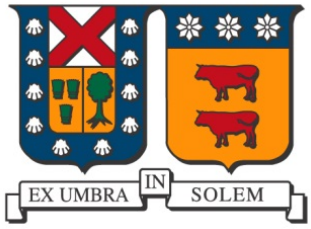
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“GAMIFICANDO EL SISTEMA DE PROGRESIÓN DE LOS JÓVENES GUÍAS Y SCOUT DE CHILE”

PATRICIO CAMPAÑA PARADA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Cecilia Reyes Covarrubias
Profesor Correferente: Andrea Vásquez Guerra

Septiembre - 2021

DEDICATORIA

Este trabajo está dedicado a mis padres: Patricio Campaña Cuello e Ingrid Parada López, quienes, aun en los tiempos más difíciles, siguieron enseñándome, guiándome y creyendo en mí.

Aun recuerdo cuando mi padre me entregó, aun estando en el colegio, aquel artículo del diario que mencionaba cierto *software* llamado *Scratch* para poder crear mis propios videojuegos sin tener que saber programar. En ese momento me entregó, quizás con cierto pre-sentimiento, quizás sin saber, quizás sólo porque quería que explorara cosas nuevas, mi primera conexión con aquello a lo que me estoy dedicando y me seguiré dedicando: el desarrollo de *software*.

Por otro lado, mi madre me ha criado con la mayoría de los valores que siempre busco integrar en mi, aunque aun sigo cometiendo errores, aprendo de ellos, y esto se lo debo, en gran medida, a la guía que mi madre me ha entregado desde que nací.

Sin duda aun me queda mucho por aprender, pero a ellos les debo gran parte de quien soy ahora, y a lo que seré en el futuro y, por eso, esta memoria se las dedico a ellos. Los quiero mucho.

AGRADECIMIENTOS

Hay tantas personas que han contribuido, tanto directa como indirectamente al desarrollo de esta memoria. Estoy infinitamente agradecido de todas las personas que he conocido, no solo en el proceso de la memoria, sino en todo mi proceso de aprendizaje.

En primer lugar, quisiera mencionar a mi gran amigo, Daniel Minte, dirigente scout quien, al escuchar mi idea de realizar un proyecto de *gamificación* para el movimiento scout, inmediatamente me apoyó contactando con José ”Pepe” Mella, director del distrito de Puerto Montt al momento de escribir este documento, y a quien también estoy enormemente agradecido por confiar en mi para llevar a cabo este proyecto. Sin ellos dos como *stakeholders*, esta memoria no habría sido posible.

También quisiera agradecer enormemente, a mi profesora guía Cecilia Reyes, quien confió en que yo podía llevar a cabo este proyecto como mi trabajo de título, quien tuvo la paciencia para acompañarme en todo este largo proceso, guiándome para escribir este documento, resolver todas mis dudas que iban surgiendo en el camino y permitirme organizar una reunión con dos profesionales, la que cambiaría positivamente el trayecto de este proyecto. Muchas gracias por todo.

Relacionado a la reunión recién mencionada, quisiera agradecer a los profesores Sven von Brand y Claudio Bustamante, quienes en una de las etapas más tempranas del proyecto, participaron en esa reunión que cambiaría la perspectiva con la que miraba este proyecto, y que permitió que llegara a donde estoy ahora.

También quisiera agradecer a otro de mis grandes amigos, Emilio Guerra, dirigente scout del Grupo Scout Altazor y al grupo, por haberme ayudado a validar una parte tan relevante de esta memoria, aun en un contexto tan complejo como la pandemia COVID19. Gracias por permitirme evaluar este proyecto.

Por supuesto, también agradecer a toda la gente de USM Games, quienes mantuvieron viva mi motivación en el desarrollo de videojuegos desde 2017 hasta ahora. Espero seguir contribuyendo a la iniciativa de una u otra forma después de mi titulación. En forma especial, quiero agradecer a aquéllos que me ayudaron directamente en la creación de modelos 3D y del ávatar para el campamento virtual de este proyecto.

Finalmente, quisiera agradecer a todos quienes me acompañaron en mi proceso universitario, con quienes formé grupos de trabajo, con quienes nos juntamos sólo para jugar un rato. Fue un período difícil, pero sin duda que todas estas personas contribuyeron a volverlo un proceso mucho más llevadero y enriquecedor.

A todas las personas que mencioné, y aquellas que no pude mencionar por el espacio disponible: **¡Muchas gracias!**

RESUMEN

Resumen— El seguimiento de la progresión personal definido por el movimiento scout es un proceso de constante observación del desarrollo personal de guías y scouts (beneficiarios), donde participan tanto ellos mismos como sus guiadoras y dirigentes, con el fin de que, estos últimos, puedan tomar decisiones estratégicas que promuevan un desarrollo integral en los jóvenes. Sin embargo, en la práctica, realizar este seguimiento no es trivial para la gran cantidad de beneficiarios que hay en un grupo, y se requieren soluciones que permitan a guiadoras y dirigentes mejorar su capacidad de seguimiento, motivando a los beneficiarios a registrar sus propios avances.

Para ello se utilizaron técnicas de *gamificación* para motivar a los beneficiarios, mediante una aplicación móvil, con un ávatar y un campamento virtual personalizable, a realizar los registros de avance y, de esta forma, generar estadísticas, para que guiadoras y dirigentes puedan analizar, desde una aplicación web, la progresión personal de los beneficiarios, y así optimizar la toma de decisiones.

Palabras Clave— Gamificación, AWS, Móvil, Unity, Scout

ABSTRACT

***Abstract*—** The monitoring of the personal progression defined by the scout movement is a process of constant observation of the personal development of guides and scouts (beneficiaries), where they, as well as their scouters, participate, so that the latter can take strategic decisions that promote an integral development in young people. However, in practice, carrying out this monitoring is not trivial because of the large number of beneficiaries in a group, and solutions are required that allow scouters to improve their monitoring capacity, motivating beneficiaries to register their own progress.

For this, gamification techniques were used to motivate the beneficiaries, through a mobile application, with an avatar and a customizable virtual campsite, to make progress records and, in this way, generate statistics, so that scouters can analyze, from a web application, beneficiary personal progression, and optimize decision making.

***Keywords*—** Gamification, AWS, Mobile, Unity, Scout

GLOSARIO

AGSCh: Asociación de Guías y Scouts de Chile
API: Application Programming Interface
AWS SAM: AWS Serverless Application Model
AWS: Amazon Web Services
B-P: Baden-Powell.
BD: Base de Datos
CDN: Content Delivery Network
CI: Continuous Integration
CLI: Command-Line Interface
DI: Departamento de Informática.
GSI: Global Secondary Index
HTTPS: HyperText Transfer Protocol Secure
JSON: JavaScript Object Notation
JWT: JSON Web Token
LSI: Local Secondary Index
RCU: Read Capacity Units
REST: Representational State Transfer
S3: Simple Storage Service
SPA: Single-Page Application
SQL: Structured Query Language
SRP: Secure Remote Password protocol
UTFSM: Universidad Técnica Federico Santa María.
WCU: Write Capacity Units
YAML: YAML Ain't Markup Language (Acrónimo recursivo)

ÍNDICE DE CONTENIDOS

RESUMEN IV

ABSTRACT IV

GLOSARIO V

ÍNDICE DE FIGURAS IX

ÍNDICE DE TABLAS XI

INTRODUCCIÓN 1

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA 3
- 1.1 CONTEXTO 3
  - 1.1.1 ORIGEN DEL MOVIMIENTO SCOUT 3
  - 1.1.2 EL ESCULTISMO Y EL GUIDISMO EN CHILE 3
  - 1.1.3 LA ASOCIACIÓN DE GUÍAS Y SCOUTS DE CHILE 4
  - 1.1.4 EL GRUPO SCOUT 4
- 1.2 EL SEGUIMIENTO DE LA PROGRESIÓN PERSONAL 5
  - 1.2.1 IDENTIFICACIÓN DEL PROBLEMA 6
  - 1.2.2 ACTORES RELEVANTES 7
- 1.3 METODOLOGÍA 8
- 1.4 OBJETIVOS PROPUESTOS 9
- 1.5 ALCANCE DE PROYECTO 9

CAPÍTULO 2: MARCO CONCEPTUAL 10
- 2.1 EL MÉTODO SCOUT 10
  - 2.1.1 LA ARQUITECTURA DEL MÉTODO SCOUT 10
  - 2.1.2 EL SISTEMA DE OBJETIVOS 12
- 2.2 GAMIFICACIÓN 13
  - 2.2.1 LA PSICOLOGÍA DE LA MOTIVACIÓN 14
- 2.3 ESTADO DEL ARTE 15
  - 2.3.1 JHATHI 16
  - 2.3.2 SENDERO SCOUT 17
  - 2.3.3 DIFERENCIACIÓN CON ESTADO DEL ARTE 18

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN 19
- 3.1 DEFINICIÓN DE LA METODOLOGÍA 19
- 3.2 PRIMER DISEÑO Y PROTOTIPO DE LA APLICACIÓN MÓVIL 21
  - 3.2.1 CÓMO SE LLEVARÍAN LOS REGISTROS 23
  - 3.2.2 PRESENTACIÓN A DIRIGENTES 23
  - 3.2.3 PROTOTIPO 24

3.2.4 REUNIÓN CON EXPERTOS Y CAMBIO DE PLANES . . . 25
3.3 DISEÑO DE LA APLICACIÓN . . . 27
3.3.1 FRAMEWORK *OCTALYSIS* . . . 27
3.3.2 PLANIFICACIÓN CON *OCTALYSIS* . . . 29
3.3.3 EL DISEÑO FINAL . . . 31
3.4 ESTRUCTURA GLOBAL DEL PROYECTO . . . 37
3.5 BACK-END: LEVANTANDO EL SERVIDOR CON AWS . . . 37
3.5.1 STACK DE TECNOLOGÍAS . . . 38
3.5.2 ENTORNO DE TRABAJO . . . 39
3.5.3 DISEÑO DE LA BASE DE DATOS . . . 39
3.5.4 ARQUITECTURA UTILIZADA . . . 42
3.5.5 REST API . . . 46
3.5.6 INTEGRACIÓN CONTINUA . . . 46
3.6 FRONT-END: PARA BENEFICIARIOS . . . 47
3.6.1 STACK DE TECNOLOGÍAS . . . 47
3.6.2 ARQUITECTURA UTILIZADA . . . 48
3.6.3 DISEÑO DE LA BASE DE DATOS . . . 51
3.6.4 VISTAS DE LA APLICACIÓN . . . 51
3.7 FRONT-END: CAMPAMENTO VIRTUAL . . . 59
3.7.1 STACK DE TECNOLOGÍAS . . . 61
3.7.2 ARQUITECTURA UTILIZADA . . . 62
3.8 FRONT-END: PARA DIRIGENTES . . . 71
3.8.1 STACK DE TECNOLOGÍAS . . . 71
3.8.2 ARQUITECTURA UTILIZADA . . . 71
3.8.3 VISTAS WEB . . . 73

CAPÍTULO 4: VALIDACIÓN . . . 83
4.1 VALIDACIÓN DE LA APLICACIÓN WEB . . . 83
4.1.1 PLAN DE VALIDACIÓN . . . 83
4.1.2 EJECUCIÓN . . . 84
4.1.3 RESULTADOS . . . 84
4.2 VALIDACIÓN DE LA APLICACIÓN MÓVIL . . . 89
4.2.1 PLAN DE VALIDACIÓN . . . 89
4.2.2 EJECUCIÓN . . . 92
4.2.3 RESULTADOS . . . 93

CAPÍTULO 5: CONCLUSIONES . . . 95
5.1 ANÁLISIS DE RESULTADOS . . . 95
5.1.1 APLICACIÓN WEB . . . 95
5.1.2 APLICACIÓN MÓVIL . . . 96
5.1.3 QUÉ PUDO HABERSE HECHO MEJOR . . . 99

ANEXO . . . 101

REFERENCIAS BIBLIOGRÁFICAS 108

ÍNDICE DE FIGURAS

1 Bitácoras y Diarios de Desafíos. 6

2 Arquitectura general de la solución propuesta. 8

3 Articulación del método scout. 11

4 Visualizador de Jhathi de los objetivos educativos. 16

5 Aplicación de Sendero Scout. 17

6 Metodología de Desarrollo 20

7 Referencia de mundo explorable con niveles 22

8 Prototipo en 2.5D y 3D 25

9 Octalysis 27

10 Segundo prototipo de videojuego Scout titulado “Espíritu Scout” 31

11 Captura de Animal Crossing: New Horizons 32

12 Entity Relationship Diagram de “Espíritu Scout” 41

13 Arquitectura de la REST API 43

14 Distribución de CloudFront 44

15 Proceso de Integración Continua 47

16 Servicios de la Aplicación Móvil 49

17 Inicio de sesión 52

18 Registro 53

19 Unirse a un grupo 54

20 Menú Principal 55

21 Edición de ávatar 56

22 Bitácora y estadísticas 57

23 Vistas de personalización de objetivos 57

24 Formulario para elegir objetivo activo 58
25 Lista de registros 59
26 Vistas de recompensas 60
27 Campamento virtual en acción 61
28 Paquetes instalados 62
29 Pantalla de Unity 63
30 Escenas 64
32 Selección de nueva zona 65
31 Ciclo de vida de zona 65
33 Componentes de interacción 66
35 Registros 67
34 Recolectable en el mundo 67
36 Buzón de correo 68
38 Componentes de Crafting 69
37 Inventario 69
39 Pista para crear una tienda de campaña de supervivencia 70
40 Sistema de construcción 70
41 Servicios de la Aplicación Web 73
42 Inicio de sesión para guiadoras y dirigentes 74
43 Creación de cuenta 74
44 Confirmar cuenta 74
45 Nuevo usuario: Instrucciones 75
46 Nuevo usuario: Formulario para inscribir grupo 75
47 Invitación para guiadoras y dirigentes 76

48 Resumen de grupo 77
49 Grupo: Resumen de grupo 77
50 Beneficiarios: Lista 78
51 Beneficiarios: Invitación 78
52 Detalle de beneficiario: Objetivos cumplidos 79
53 Detalle de beneficiario: Registros 80
54 Detalle de beneficiario: Registros de objetivo 80
55 Detalle de beneficiario: Perfil público 81
56 Guiadoras y dirigentes: Lista 81
57 Panel de administrador: Recompensas 82
58 Panel de administrador: Grupos 82
59 Captura de la validación de la aplicación web con los dirigentes 84
60 Dimensión de desafío de GAMEFULQUEST 91
61 Validación de la aplicación móvil 92
62 Resultados validación aplicación móvil 94
63 Diseño de la Gamificación del prototipo 101
64 Diseño del videojuego utilizando MDA 102

ÍNDICE DE TABLAS

1 Relación de necesidades psicológicas y elementos del diseño de juego. 15
2 Unidades centrales de Octalysis 28
3 Funcionalidades de la aplicación por Unidad Central de Octalysis 36
4 Tablas de la Base de Datos 42
5 Queries de la aplicación 104

INTRODUCCIÓN

La transformación digital ha provisto de importantes soluciones para la automatización y optimización de procesos, mejorar la productividad, promover la innovación, entre otras grandes ventajas. El diseño de *software* tradicional suele tener un fuerte enfoque en las funcionalidades, o sea, que una vez se puedan realizar todas las acciones que se necesitan en la plataforma para completar ciertos procesos, se da por finalizado el proyecto. Ésta es una forma bastante validada de desarrollar *software* y que ha tenido buenos resultados en crear la sociedad actual, que se encuentra tan ligada a la tecnología, sin embargo, ¿qué ocurre cuando el objetivo del proyecto está relacionado a aspectos más abstractos? Como podrían ser aspectos psicológicos o sociales, tal como motivar a un grupo de personas a cambiar su comportamiento en temáticas como: cuidar hábitos alimenticios, desarrollar nuevas habilidades o tener horarios de sueño estables. Si bien, como es normal, no existe una solución única para este tipo de problemáticas, la *gamificación* es una herramienta que propone abordar esto, mediante la utilización de características propias de un tipo de actividad que está en la sociedad desde hace milenios: los juegos.

El escultismo es un movimiento, fundado por Robert Baden-Powell hace más de 100 años, y que busca contribuir y promover la educación integral de jóvenes utilizando el método scout. Éste se encuentra, en la actualidad, buscando una solución a una problemática que se puede comparar a las recién mencionadas: motivar a sus beneficiarios a registrar los avances de su **progresión personal**. Si bien el concepto de **progresión personal** podría considerarse ambiguo, el marco conceptual que envuelve al movimiento define claramente cómo éllos abordan esta idea, mediante una gran lista de objetivos propuestos a los beneficiarios, clasificados en 6 áreas de desarrollo: **Corporalidad**, **creatividad**, **carácter**, **afectividad**, **sociabilidad** y **espiritualidad**, que busca promover un desarrollo personal íntegro en sus beneficiarios.

Además, para este proyecto, se buscó que los registros que realicen los beneficiarios puedan generar estadísticas que mejoren la capacidad de supervisión de sus guiadoras y dirigentes guía y scout, con el fin que puedan realizar su labor de manera más eficiente, y les permita dedicar más tiempo a las actividades con sus beneficiarios, que a los papeles.

La memoria que aquí se presenta aborda las fases de diseño, desarrollo y validación del proyecto “Espíritu Scout”, un proyecto multi-plataforma conformado de una aplicación móvil *gamificada* donde los beneficiarios del movimiento pueden realizar los registros de sus avances de progresión personal, y una aplicación web para guiadoras y dirigentes, donde éllos puedan realizar eficientemente el seguimiento de esta progresión, en base a los datos generados por los beneficiarios.

Este trabajo se conforma de 5 capítulos. En el *Capítulo 1: Definición del problema*, se presenta el contexto de la problemática y cómo ésta es percibida por los dos *stakeholders* de este proyecto: José Mella Silva y Daniel Minte Valderrama, dos dirigentes scout del distrito de Puerto Montt de la Asociación de Guías y Scout de Chile. En este capítulo también se pre-

sentan la metodología con la que se abordó la problemática, los objetivos de este proyecto, y cuál es el alcance de éste.

El *Capítulo 2: Marco Conceptual*, busca realizar una contextualización más profunda del método scout y el funcionamiento del sistema de progresión personal dentro del movimiento. Además se aborda el tema de la *gamificación*, buscando explicar por qué la *gamificación* funciona a nivel psicológico, y se realiza una revisión del estado del arte en proyectos que ya han buscado abordar esta problemática y cómo este proyecto busca diferenciarse entre estas soluciones.

En el *Capítulo 3: Propuesta de Solución*, se realiza una explicación de todo el proceso de diseño y desarrollo del proyecto, abordando cómo se realizó el diseño en base al *framework* Octalysis y la arquitectura del *back-end* y *front-end* utilizados para la aplicación web y móvil.

En el *Capítulo 4: Validación*, se revisa el plan para validar tanto la aplicación web, como la móvil con potenciales usuarios de la plataforma, y los resultados obtenidos de estas instancias.

Finalmente, el *Capítulo 5: Conclusiones*, contiene el análisis de los resultados obtenidos en la fase de validación, y cómo se planea utilizar estos resultados para el futuro del proyecto. Además, en este capítulo se plantean aquellos puntos del proceso que podrían haber sido enfocados de manera diferente, y que el autor con una visión de mejora continua y la experiencia lograda, considera relevantes concluir para el desarrollo de *software* y proyectos de *gamificación* en general.

CAPÍTULO 1
DEFINICIÓN DEL PROBLEMA

1.1. CONTEXTO

1.1.1. ORIGEN DEL MOVIMIENTO SCOUT

El origen del escultismo (*scouting*) se remonta hace más de 100 años, cuando Robert Stephenson Smyth Baden-Powell, también conocido como B-P (*Be Pe*), destacado militar británico especializado en la exploración y reconocido como héroe nacional por su labor como líder de la guarnición que participó en el sitio de Mafeking (The Editors of Encyclopaedia Britannica 2020), decide realizar en 1907 un campamento experimental donde poner a prueba sus ideas formativas con jóvenes de distintos contextos sociales, revisando temáticas como campismo, observación, salvamento, etc. Todo esto con el fin de llevar a los jóvenes a ser mejores ciudadanos. Este campamento, realizado en la Isla de Brownsea, fue todo un éxito y es considerado como el primer campamento scout del mundo (World Organization of the Scout Movement 2018).

Tras el campamento de Brownsea, B-P decidió plasmar sus diversas experiencias e ideas scout en el libro *"Escultismo para muchachos: un manual de instrucción en buena ciudadanía haciendo vida de campaña"*, texto fundamental del movimiento scout. Este manual fue publicado en el año 1908, un año después del campamento de Brownsea, y tuvo una gran recepción que llevó a que este libro, ideado originalmente como un manual para organizaciones juveniles ya existentes, diera origen a un nuevo movimiento que se extendería por el mundo entero (World Organization of the Scout Movement 2018).

1.1.2. EL ESCULTISMO Y EL GUIDISMO EN CHILE

En el libro de Rojas 2006, se explica como en Chile se experimentó una temprana expansión del escultismo gracias a la directa influencia del fundador del movimiento, Robert Baden-Powell, quien se encontraba de visita en Latinoamérica. Durante su estadía en Chile, B-P realizó una conferencia en el salón de honor de la Universidad de Chile, donde expuso el objetivo formativo detrás de la fundación de los boy scouts, invitado por dos chilenos interesados en la experiencia educativa que prometía el escultismo, Alcibíades Vicencio, connotado médico ginecólogo y fundador del Instituto de Puericultura y Joaquín Cabezas, profesor del Instituto Nacional y director fundador del Instituto de Educación Física y Manual.

Al finalizar B-P su charla, Cabezas realizó la inscripción de aquellos jóvenes interesados en formar un grupo scout, entre los que se encontraron estudiantes del Instituto Nacional y universitarios. Tras una serie de reuniones preparatorias, se realiza el 21 de Mayo de 1909,

una excursión al Río Maipo con cerca de trescientos jóvenes, y en el puente Los Morros se declara fundada, por medio de un acto solemne, la Asociación de Boy Scouts de Chile.

En 1913 surge en Rancagua la primera brigada de girl guides, punto inicial que luego llevaría a que se crease la Asociación de Girl Guides, que para 1952 ya agrupaba a seis mil jóvenes y en 1953 pasa a ser autónoma mediante un acuerdo de la Asociación de Boy Scouts.

Finalmente, entre 1970 y 1978, se realiza un proceso de unidad entre la Asociación de Boy Scouts de Chile, la Asociación de Girl Guides de Chile, la Federación de Scouts Católicos y el Movimiento de Reforma Scout, para conformar la actual Asociación de Guías y Scouts de Chile (Asociación de Guías y Scouts de Chile 2019).

1.1.3. LA ASOCIACIÓN DE GUÍAS Y SCOUTS DE CHILE

La Asociación de Guías y Scouts de Chile corresponde a una corporación de derecho privado sin fines de lucro, cuyo objeto consiste en *"promover y contribuir a la **educación integral** de niñas, niños y jóvenes en todo el territorio de la República de Chile, basada en el **método scout** creado por Robert Baden-Powell a partir de 1907"* (Asociación de Guías y Scouts de Chile 2019).

Para lograr organizar la asociación a lo largo de Chile, la corporación definió estructuras territoriales y de base. En primer lugar, una **Zona** corresponde a aquella estructura a través de la cual se aplican planes a nivel regional, y dentro de las Zonas se encuentra la estructura intermedia de la Asociación, que son los **Distritos**, una división claramente determinada del territorio nacional, y que son creados por el Consejo Nacional.

Finalmente, dentro de cada distrito existen los **Grupos**, que corresponden a la *estructura base* de la Asociación *"donde se aplica y practica el programa y el método scout"* (Asociación de Guías y Scouts de Chile 2019). Cada grupo posee una autoridad máxima, el **Consejo de Grupo**, que es presidido por el **Responsable de Grupo**. Además, dentro de un grupo participan madres, padres y/o tutores legales de las niñas, niños y jóvenes del grupo, quienes participan en el **Comité de Grupo**.

1.1.4. EL GRUPO SCOUT

En cada grupo, los niños y jóvenes que participan en él se encuentran distribuidos en ramas. Estas ramas son (*Reglamento de la Asociación de Guías y Scouts de Chile* 2019):

- **Ramas menores**: Formada por *golondrinas* y *lobatos*, que corresponden respectivamente a niñas y niños de entre 7 y 11 años. Abarcando las etapas infancia temprana e infancia tardía.

- **Ramas intermedias**: Formada por *guías* y *scouts*, que corresponden a jóvenes mujeres y hombres respectivamente, de entre los 11 y 15 años. Abarcando las etapas de prepubertad y pubertad.
- **Ramas mayores**: Formado por *pioneras* y *pioneros*, o sea, jóvenes entre 15 y 16 años, y por *Caminantes*, de entre 17 y 21 años. Abarcando las etapas de adolescencia y juventud respectivamente.

Y por cada rama, hay cierta cantidad adultos responsables conocidos como **guiadoras** y **dirigentes** cuya labor es la de acompañar y motivar a los jóvenes en su desarrollo, el número de dirigentes por rama depende principalmente de la cantidad de jóvenes en cada rama, con tal que cada guiadora y dirigente pueda estar a cargo de un máximo de 8 jóvenes (Comisión Nacional de la Rama Scouts 2016 & Comisión Nacional de la Rama Guías 2018).

1.2. EL SEGUIMIENTO DE LA PROGRESIÓN PERSONAL

Aquella parte del método scout en el que se enfocará esta memoria es el **seguimiento de la progresión personal** por parte de las guiadoras y dirigentes en las ramas intermedias. La **progresión personal** corresponde a *"el avance que un muchacho (o una niña) logra paulatinamente en la obtención de las conductas previstas en sus objetivos. Comprende todos los aspectos de su personalidad e incorpora las ideas afines de crecimiento y desarrollo. (...)"* . Mientras que el **seguimiento** *"es un proceso amable que acompaña el desarrollo, promueve la participación del joven en la Tropa (o de una niña en la Compañía), eleva su autoestima y lo ayuda a mejorar el nivel de logro de sus objetivos.* ***Es acompañamiento y no control***." (Comisión Nacional de la Rama Scouts 2016 & Comisión Nacional de la Rama Guías 2018). Los objetivos que se mencionan son personales de el o la joven que pertenece a cierto rango etario, en este caso, entre los 11 y 15 años.[1]

El seguimiento de la progresión personal es una tarea realizada tanto por el o la joven como por el dirigente o guiadora, para ello, las guías y scouts poseen el **Diario de Desafíos** y **Bitácora** respectivamente, mientras que los dirigentes utilizaban una **hoja de control de la progresión**, que se dejó de producir tras comenzar el desarrollo de una nueva versión que todavía no se ha publicado. Esto último ha causado que hayan dirigentes que tienen que llevar el registro por medio de hojas de cálculo (e.g. archivos de Excel), o que el registro no se realice formalmente.

[1] Para profundizar sobre el sistema de objetivos del método scout, revisar el capítulo 2 de esta memoria



Figura 1: Bitácoras y Diarios de Desafíos.
Fuente: Asociación de Guías y Scouts de Chile.

A lo largo del seguimiento, la guiadora o dirigente ha de realizar una evaluación de la progresión del joven a su cargo, para lo que ha de reconocer los diferentes agentes educativos que rodean al joven. Estos se pueden organizar en tres grupos (Comisión Nacional de la Rama Scouts 2016 & Comisión Nacional de la Rama Guías 2018):

- **Los padres:** Principales educadores de sus hijos y fundamentales en el proceso de evaluación de un joven.
- **Los profesores:** De la escuela o colegio al cual el joven asiste.
- **Otros agentes:** Aquí se pueden encontrar por ejemplo: autoridades de la iglesia a la que el joven pertenece, entrenadores deportivos, etc.

Y, al mismo tiempo, estos actores pueden estar interesados en saber cómo se encuentra el joven con respecto a sus objetivos personales, por lo que, finalmente, detrás de cada guía o scout, existen otros actores involucrados.

1.2.1. IDENTIFICACIÓN DEL PROBLEMA

Aun con la existencia de estas herramientas, según lo expresado por los dirigentes scouts del distrito de Puerto Montt José Mella y Daniel Minte, han percibido una dificultad en mantener la motivación de las guías y scouts (ramas intermedias) por realizar el seguimiento de su progresión personal, mostrando que la bitácora/diario de desafíos no está dando buenos resultados como herramienta de seguimiento de la progresión personal.

Este problema pone a las guiadoras y dirigentes en una situación en la cual no pueden conocer concretamente cuáles son aquellos puntos en los que la guía o el scout necesita apoyo para cumplir sus objetivos. Además, como lo expresan los manuales de rama Comisión Nacional de la Rama Scouts 2016 & Comisión Nacional de la Rama Guías 2018, es importante

que los jóvenes estén conscientes de aquello en lo que han avanzado, pues se encuentran en un período donde empiezan a desarrollar su propia personalidad, a diferenciarse, y el conocer a dónde voy y a dónde quiero ir, empieza a tener relevancia en su desarrollo, de hecho, en Comisión Nacional de la Rama Scouts 2016 & Comisión Nacional de la Rama Guías 2018 se indica que las tareas de desarrollo de la adolescencia se pueden resumir en tres ítems, de los cuales uno corresponde a *"Plantearse un proyecto de vida propio"*.

A este problema, se suma que la tarea de los dirigentes de llevar el registro se dificulta por diversos motivos, entre lo que se puede encontrar la falta de tiempo de mantener el registro, la no existencia de fichas actualizadas o, nuevamente, la falta de motivación de los scouts por ver cómo medir sus avances.[2]

1.2.2. ACTORES RELEVANTES

En el seguimiento de la progresión personal se consideran tres grupos de actores relevantes, que se encuentran asociados a cada guía o scout:

- **Guiadoras y dirigentes:** Interesados en realizar el seguimiento y evaluar al joven. Corresponden a adultos responsables voluntarios, que buscan contribuir a la educación de los jóvenes por medio de la aplicación del método scout.
- **Los agentes educativos que rodean a el o la joven:** Interesados en conocer el progreso del joven y participar en él. Corresponden a personas de diversos contextos y características, pero todos con un fin común, contribuir en al menos un aspecto, al desarrollo de un joven.
- **Guías y scouts:** Jóvenes al que se busca ayudar en su desarrollo. Niños y niñas de entre 11 y 15 años, correspondiente a las etapas de pre-pubertad y pubertad.[3]

Según una consulta realizada el día 22 de Julio de 2020 al director del distrito de Puerto Montt, José Mella, el número de jóvenes inscritos a nivel nacional en ese momento eran de 2309 guías y 1932 scouts de entre 11 y 15 años, lo que suma un total de 4241 guías o scouts. Es importante acotar que en la fecha de realizada la consulta, la Asociación de Guías y Scouts de Chile se encuentra en período de registro de nuevos integrantes y, además, en plena pandemia de la COVID19, por lo que estos números pueden variar en cantidades no despreciables.

[2]Fuente: Reunión con los dirigentes scout del distrito de Puerto Montt José Mella y Daniel Minte

[3]Para profundizar en los aspectos psicológicos que que caracterizan a los jóvenes en este período, revisar la sección 2.1.2 de esta memoria

1.3. METODOLOGÍA

En esta memoria, para afrontar la solución al problema de baja motivación de los jóvenes por realizar un seguimiento de su progresión personal, se plantea utilizar **gamificación**, que se define como el uso de mecánicas, estética y pensamiento de los juegos para involucrar a las personas, motivar acción, promover el aprendizaje y resolver problemas (Karl M. Kapp 2013). En este caso, se buscará motivar a los jóvenes a mantener su bitácora o diario de desafíos e involucrarse más con su progresión personal. La gamificación no es necesariamente desarrollar un videojuego, sin embargo, en este caso se creará uno con el fin de elaborar un contexto sobre el cual las guías y scouts se sientan motivados a registrar sus objetivos.

Este sistema se conformará de dos *puntos de entrada* para los actores, estos son (ver figura 2):

1. **Aplicación Móvil:** Videojuego donde los jugadores (guías y scouts) pueden registrar sus objetivos cumplidos, y en base a estos recibirán alguna recompensa que les motive a seguir cumpliendo y seguir con el registro.
2. **Página Web:** Sitio web donde las guiadoras y dirigentes pueden revisar la progresión personal de una guía o scout, y así realizar el seguimiento desde un lugar centralizado de fácil acceso. Además se permitirá que, con permiso del dirigente o guiadora a cargo de un joven, compartir su progresión con otros agentes educativos que participen en el círculo del joven.

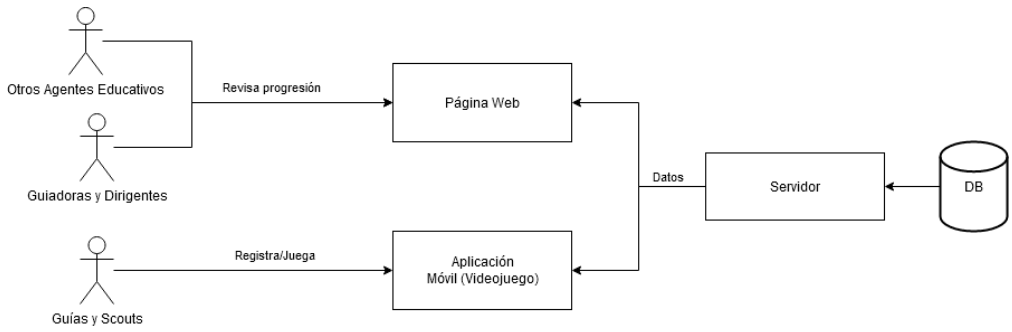


Figura 2: Arquitectura general de la solución propuesta.
Fuente: Elaboración propia.

Para el desarrollo de esta memoria se contará con el apoyo, a modo de *stakeholders*, de dos dirigentes del movimiento scout: Daniel Minte, dirigente de guías y scouts del Grupo Pumahue de Puerto Montt, y José Mella, actual director del distrito scout de Puerto Montt.

1.4. OBJETIVOS PROPUESTOS

Objetivo general: Diseñar e implementar un sistema de progresión de los objetivos de aprendizaje de cada área de desarrollo planteado en el modelo formativo de la Asociación de Guías y Scout de Chile, con el fin de aumentar la motivación de los jóvenes pre-púber y púber por alcanzar los objetivos del movimiento y, por otra parte, mejorar la gestión del progreso por parte de las guiadoras y dirigentes y otros agentes educativos.

Objetivos específicos:

- Diseñar y producir un videojuego que permita motivar a los guías y scout pre-púber y púber a llenar su bitácora personal, de tal manera de llevar un registro de los logros alcanzados en cada una de sus etapas de progresión.
- Desarrollar una página web responsiva donde los dirigentes puedan aumentar su capacidad de supervisión, y compartir con quienes consideren adecuado, las estadísticas del progreso de los guías y scout a su cargo.
- Validar que el sistema de progresión aumente la motivación de los guías y scout pre-púber y púber y mejore la gestión de su progreso, evaluando su uso con un grupo de prueba conformado por jóvenes, guiadoras y dirigentes scout del distrito de Puerto Montt.

1.5. ALCANCE DE PROYECTO

Cuando se refiere en este documento a diseñar y producir un videojuego corresponde, más específicamente, a la utilización de un entorno de videojuego, donde el objetivo es la realización de registros en la progresión personal, no el videojuego mismo.

Es importante considerar que, lo que se busca, es motivar la realización del registro de la progresión personal y guiarlos en ello, mas no el cumplir los objetivos como tal, puesto que esto último es algo que se realiza en las actividades scout. Esto es una herramienta complementaria para potenciar la motivación por realizar registros y que los dirigentes tengan esta información aprovechando las ventajas que ofrece la transformación digital, no un reemplazo del núcleo del movimiento scout.

La validación de la veracidad de los datos realizados por los beneficiarios es algo que no se tendrá en cuenta según lo conversado con los dirigentes, quienes dijeron que en caso de falta a la honestidad al utilizar la aplicación, son situaciones que ellos deben manejar personalmente con los beneficiarios.

CAPÍTULO 2
MARCO CONCEPTUAL

2.1. EL MÉTODO SCOUT

Se entiende como el **propósito del Movimiento Scout**, el "contribuir a la educación de los jóvenes para que participen en la construcción de un mundo mejor, donde las personas se desarrollen plenamente y jueguen un papel activo en la sociedad" (Asociación de Guías y Scouts de Chile 2014), y esto se logra utilizando el método creado por Robert Baden-Powell hace más de 100 años, que se conoce como el **método Scout**. Este método se define como *"un sistema de auto-educación progresiva y participativa, complementario de otros agentes educativos, que se funda en la interacción de diversos componentes articulados entre sí, varios provenientes de los valores del Movimiento"* (Asociación de Guías y Scouts de Chile 2014). De esta definición se pueden extraer las siguientes características del método:

- **Se basa en la autoeducación:** Pretende que cada joven se convierta en su principal agente de desarrollo, para que llegue a ser una persona autónoma, solidaria, responsable y comprometida.
- **Progresivo:** Desde la llegada de un joven al movimiento hasta su egreso, se espera que el método contribuya a su desarrollo gradualmente en distintos aspectos de su vida.
- **Participativo:** Como se mencionó anteriormente, se pretende que cada joven se vuelva en su principal agente de desarrollo, y esto se logra mediante la participación del joven en distintos aspectos del método, como el programa y actividades que se realicen.
- **Complementario a otros agentes educativos:** El método no busca reemplazar a la educación formal, por el contrario, funciona complementaria y articuladamente con la familia y la educación formal [4].

2.1.1. LA ARQUITECTURA DEL MÉTODO SCOUT

Como se mencionó anteriormente, el método scout se estructura en base a componentes que interactúan entre sí, los componentes son los siguientes (Asociación de Guías y Scouts de Chile 2014):

1. **Sistema de equipos:** Las guías y scouts se separan en **patrullas**, equipos de comúnmente 8 jóvenes.

[4] Art. 2 del Estatuto de las Asociación de Guías y Scouts de Chile (Asociación de Guías y Scouts de Chile 2019).

2. **Aprendizaje por la acción:** Se prepara y ejecuta un programa y realizan actividades en las cuales se busca contribuir al logro los objetivos personales de los jóvenes.
3. **Orientación de adultos:** Las guiadoras y dirigentes que acompañan al desarrollo y aprendizaje de los jóvenes.
4. **Marcos simbólicos:** Están definidos según la edad e "incentivan la imaginación, refuerzan la pertenencia y motivan el crecimiento personal".
5. **Adhesión voluntaria a la Promesa y la Ley:** La **Promesa** y la **Ley** son la expresión de un sistema de valores en el cual se consideran principios espirituales, sociales y personales, y son propuestos a los jóvenes para que las integren en su proyecto de vida.

Estos componentes por separado, sin embargo, no completan la estructura del método, aquello propio del método es la **articulación de sus componentes,** lo que puede ser descrito en forma de rombo (ver figura 3).

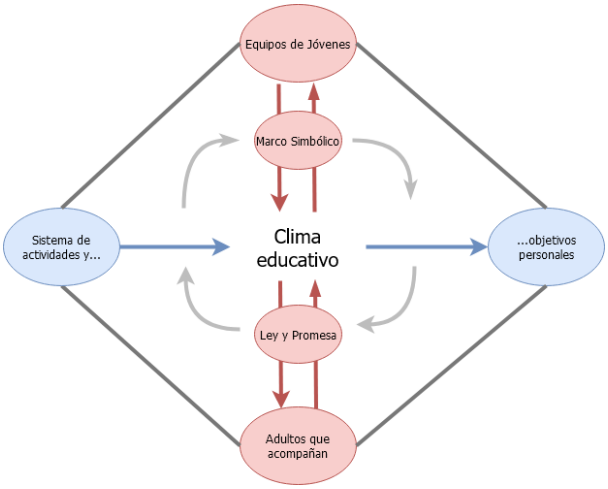


Figura 3: Articulación del método scout.
Fuente: Asociación de Guías y Scouts de Chile 2014.

Se pueden destacar tres partes de esta estructura:

1. **Línea vertical central:** Representan a las personas.
2. **Línea horizontal central:** Representa lo que las personas quieren lograr por medio del aprendizaje por la acción.
3. **Interior del rombo:** Los valores y símbolos que el método propone.

Estas son las interacciones que generan como resultado un **clima educativo,** que es aquel contexto sobre el cual se facilita la actividad educativa (Asociación de Guías y Scouts de Chile 2014).

2.1.2. EL SISTEMA DE OBJETIVOS

Dada la naturaleza integral de la educación ofrecida por el método scout, el **sistema de objetivos** del movimiento scout se crea sobre una agrupación de las dimensiones de la personalidad de un ser humano, estas agrupaciones se conocen como **áreas de desarrollo** y, desde 1989, la Asociación de Guías y Scouts de Chile ha considerado las siguientes (Asociación de Guías y Scouts de Chile 2014, Comisión Nacional de la Rama Scouts 2016 & Comisión Nacional de la Rama Guías 2018):

- **Corporalidad:** La responsabilidad de cada persona en el desarrollo de su cuerpo. En las ramas intermedias, el cambio en el cuerpo de una persona es acelerado, y el desafío de cada joven en esta área recae en asumir este nuevo cuerpo, aceptarlo y gobernarlo.
- **Creatividad:** Estimular el uso de los conocimientos adquiridos de una forma original y relevante. En las ramas intermedias los jóvenes se encuentran en un período donde están buscando obtener más conocimientos, pero hoy en día existe una masiva cantidad de información, que llega a ser confusa y hasta contradictoria entre sí, por lo que un desafío recae en promover el análisis de esta información. Otro desafío surge en algunos jóvenes que no logran obtener el rendimiento escolar esperado, lo que reprime su deseo de buscar información, entonces la tarea es que reencuentre la confianza en sus capacidades y así retomar su progresión.
- **Carácter:** Disposición de la persona para organizar sus fuerzas e impulsos de acuerdo a aquellos valores que considera relevantes. En las ramas intermedias los jóvenes están cursando una rápida transición desde la infancia, en la cual es normal estar confundido con tantos impulsos e intereses nuevos, pero es gracias a esto que pueden forjar su nueva identidad, y el desafío se encuentra en motivar el auto-conocimiento, la autocrítica y la aceptación de críticas. Y de esta forma “re-edificar” su nueva identidad.
- **Afectividad:** Integración de la vida afectiva en el comportamiento. En las ramas intermedias los sentimientos y emociones llegan a ser confusos para el mismo joven, surgiendo el desafío de lograr una mayor estabilidad que permita a los jóvenes reconocer y aceptar sentimientos, compartir sus emociones y, así, lograr progresivamente un estado de equilibrio emocional.
- **Sociabilidad:** Volver la solidaridad y el servicio a los demás como determinante en la felicidad personal. Es durante las ramas intermedias que los jóvenes empiezan a desarrollar en mayor medida la empatía y son capaces de juzgar más allá que por aspectos externos a una persona. En esta etapa es importante que el joven aprenda que todas las personas son iguales en derechos, sin importar su condición económica, social, cultural, sexual, étnica o religiosa, y que el uso de estos derechos son correlativos al cumplimiento de deberes.
- **Espiritualidad:** Búsqueda de la trascendencia de la vida humana como respuesta ante el misterio de su origen, naturaleza y destino. En las ramas intermedias es común

> que ocurran cambios en la percepción que cada joven tiene de sus creencias. Aquí es importante que las guiadoras y dirigentes no realicen un adoctrinamiento religioso, sino que fomenten en los jóvenes la expresión de actitudes como son acoger, reflexionar, escuchar, compartir, cooperar y solidarizar, con el fin de que el joven establezca vínculos con sus creencias, que la pueda re-descubrir, la acoja y se haga cargo de ella, pasando a evolucionar de una fe infantil a una fe adulta y personal, lo que converge en una fe viva.

Por cada una de estas áreas de desarrollo surgen un conjunto de objetivos que expresan aquellas conductas deseables y observables al momento que un joven egresa del movimiento, estos objetivos se conocen como **objetivos terminales** (*Objetivos educativos del Movimiento Scout* 1993-1996). Por ejemplo, el quinto objetivo terminal del área de corporalidad es:

Administra su tiempo equilibradamente entre sus diversas obligaciones, practicando formas apropiadas de descanso.

Según el rango de edad en el que cada joven se encuentra, la manera apropiada de acercarse a cada uno de estos objetivos terminales varía, de forma que se ha definido por cada objetivo terminal una serie de **objetivos intermedios** u **objetivos propios de la edad**, cada uno de ellos va dirigido a guiar a cada joven en el camino a conseguir alguno de los objetivos terminales según el rango de edad en el que se encuentra el joven. Los rangos de edad considerados actualmente son: 7-9 años (infancia media), 9-11 años (infancia tardía), 11-13 años (pre-pubertad), 13-15 (pubertad) años, 15-17 años (adolescencia) y 17-20 años (juventud) (*Objetivos educativos del Movimiento Scout* 1993-1996).

Sin embargo, hay que recordar que **cada individuo es único** y, dadas sus necesidades, intereses, aspiraciones, capacidades y potencialidades, los objetivos intermedios pueden recibir modificaciones, de hecho, los objetivos intermedios corresponden a una **propuesta de objetivos** que, en la etapa de **personalización** realizada entre una guiadora o dirigente y una guía o scout, son asumidos por este último mediante la modificación o adición de nuevos objetivos, surgiendo un nuevo conjunto de objetivos provenientes de los intermedios, conocidos como los **objetivos personales** del joven (Comisión Nacional de la Rama Scouts 2016 & Comisión Nacional de la Rama Guías 2018).

2.2. GAMIFICACIÓN

La ***gamificación*** se puede definir como *el uso de mecánicas, estética y pensamiento de los juegos para involucrar a las personas, motivar acción, promover el aprendizaje y resolver problemas* (Karl M. Kapp 2013), también se puede definir en menos palabras, y sin limitar los casos de uso, como ***el uso de elementos de los juegos en contextos no lúdicos*** (Deterding

y col. 2011), sin embargo todavía no existe una definición científica universalmente aceptada del término. Semánticamente, utilizando esta última definición se pueden distinguir cuatro componentes (Sailer y col. 2017):

- **Juego:** En este contexto, se utilizará la definición de juego de Salen y col. 2004, o sea, un sistema en el cual los jugadores participan en un conflicto artificial, y delimitado por reglas, lo que concluye en un resultado cuantitativo. Esto debido a la relación entre la gamificación y cómo los juegos poseen reglas que delimitan las acciones del usuario, a la vez que le proponen objetivos por cumplir.
- **Elementos:** Término que permite diferenciar entre la *gamificación* y *juegos serios*, que corresponden a aquellos juegos desarrollados con un propósito además de entretener, como puede ser el entrenamiento en cierta área, enseñar cierta materia, etc. En un contexto *gamificado*, los elementos del diseño de juegos corresponden a las características de estos que pueden encontrarse en varios juegos y que poseen algún significado en ellos (Sailer y col. 2017).
- **Diseño:** La *gamificación* se enfoca en el uso del diseño de los juegos en vez del uso de las tecnologías utilizadas en el desarrollo de un juego, como podría ser un motor de videojuegos.
- **Contextos no lúdicos:** La *gamificación* deja abierto el contexto en el cual se puede aplicar esta técnica como cualquier contexto fuera de los mismos juegos.

De lo anterior, podemos deducir que la gamificación no implica (necesariamente) realizar un videojuego o, si se realiza, no es de una forma tradicional, por ejemplo, algunos casos de uso de la gamificación han sido el de motivar a desarrolladores a eliminar “warnings” (Arai y col. 2014) mediante una herramienta que suma puntos según la cantidad de “warnings” removidos por cada desarrollador y equipo, con lo que se logró que la cantidad de “warnings” eliminados sean de al rededor de un 50 % más que el caso sin *gamificación*. O el de mejorar el rendimiento académico (R. N. Landers y A. K. Landers 2014) donde se realizó un estudio asignando aleatoriamente dos versiones para un mismo proyecto on-line a estudiantes, utilizando una versión *gamificada* con tabla de clasificaciones y una versión controlada sin tabla de clasificaciones[5]. En este caso, los estudiantes que utilizaron la versión gamificada interactuaron 29.61 veces más en promedio que aquellos a quiénes se les asignó la versión controlada, mostrando empíricamente que la gamificación bien implementada puede motivar a las personas a realizar ciertas acciones.

2.2.1. LA PSICOLOGÍA DE LA MOTIVACIÓN

En Sailer y col. 2017 se realizó un estudio sobre *cómo la gamificación motiva*, pues si bien en general los estudios empíricos muestran resultados mayoritariamente efectivos con respecto

[5]El proyecto no tuvo efecto en las notas finales del curso

Tabla 1: Relación de necesidades psicológicas y elementos del diseño de juego.

Fuente: Sailer y col. 2017.

Necesidad Psicológica	Mecanismo	Elemento del diseño de los juegos
Necesidad de competencia	Retroalimentación granular	Puntos
	Retroalimentación sostenida	Gráficos de rendimiento
	Retroalimentación acumulativa	Medallas
	Retroalimentación acumulativa	Tabla de clasificación
Necesidad de autonomía (libertad de decisión)	Decisiones	Avatares
Necesidad de autonomía (tarea significativa)	Compromiso voluntario	Historias significativas
Necesidad de relaciones sociales	Sentido de relevancia	Compañeros de equipo
	Meta compartida	Historias significativas

a los efectos positivos de la gamificación en la motivación, todavía no se había estudiado teóricamente cómo se podrían explicar estos resultados. Para ello se utilizó la perspectiva de *auto-determinación* de entre seis perspectivas que se tomaron en cuenta en el estudio, en esta perspectiva se consideran tres necesidades psicológicas básicas e intrínsecas que, en Rigby y Ryan 2011 defienden, son satisfechas por los videojuegos comerciales:

- **La necesidad de competencia:** Corresponde a cómo cada ser humano busca sentirse competente al interactuar con su entorno.
- **La necesidad de autonomía:** Se refiere a que el ser humano busca poder elegir entre varias vías de acción (*libertad de decisión*), a la vez que la vía de acción que escoja ha de responder al propio objetivo y actitud del individuo (*sentido de decisión*).
- **La necesidad de relaciones sociales:** Se refiere a la necesidad de pertenencia y cariño con respecto a un grupo de personas que sean importantes para el individuo mismo.

La conexión de estas necesidades con algunos elementos del diseño de los juegos se describe en tabla 1.

2.3. ESTADO DEL ARTE

A continuación se revisarán otros proyectos que se han desarrollado y que han abarcado el tema de la progresión personal.

2.3.1. JHATHI

En 2008 Juan Pablo Menichetti Wormald registra en SourceForge el software Jhathi [6] desarrollado en Java, el cual sirve como registro local de la estructura de un grupo y la progresión personal de los jóvenes en él, también conocidos como los beneficiarios. La forma en que el software está estructurado es en tres secciones que se explicarán a continuación.

En primer lugar se encuentra “Inicio”, donde se pueden crear registros de grupos scout, para luego guardarlos en formato XML, lo que permite compartir el registro de un grupo con otros dirigentes y guiadoras que utilicen el programa, o incluso subirlo a una plataforma web para compartir el registro con otros miembros del grupo.

Luego, una vez se crea un grupo, se puede acceder a la sección “Grupo”, en la cual se realizan los registros de cada unidad (manada, golondrinas, tropa, compañía, avanzada y clan), de los jóvenes que se encuentran en cada unidad, y de sus fichas médicas que incluyen datos como grupo sanguíneo, teléfono de emergencia, entre otros. Además Jhathi muestra alertas al usuario en caso de existir datos faltantes y posee un calendario en el cual se puede revisar el programa mensual del grupo, y cuyos datos provienen de los registros de la siguiente sección.

La última sección corresponde a “Programa”, donde se pueden revisar los puntajes de cada unidad, equipo o beneficiario en cada una de las áreas de desarrollo explicadas en la sección 2.1.2 de esta memoria. Estos puntajes se ganan registrando evaluaciones de progresión por medio de la ejecución de actividades que el usuario ingresa, actividades que se encuentran asociadas a dos áreas de desarrollo, una principal y una secundaria, que definen el puntaje que un beneficiario obtiene por una evaluación en cada área. Las evaluaciones se miden de 1 a 5 estrellas, y poseen el nombre del evaluador y una descripción. En esta sección también se encuentra un visualizador de los objetivos educativos en forma de malla que permite revisar y filtrar por área y edad cada parte del sistema de objetivos planteado (ver fig. 4).

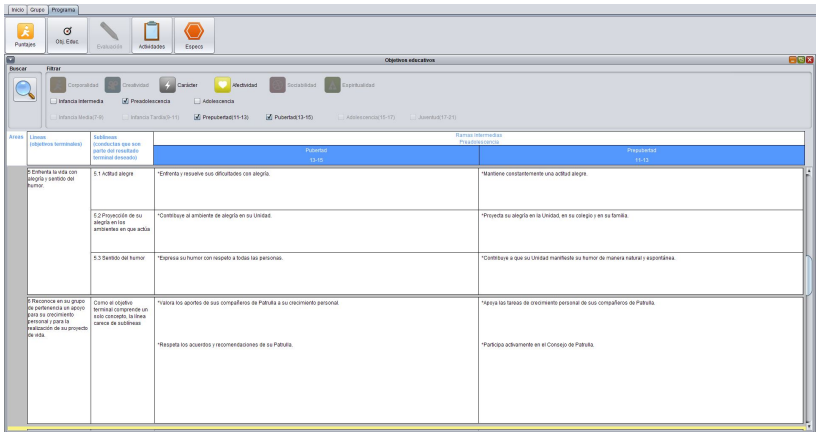


Figura 4: Visualizador de Jhathi de los objetivos educativos.
Fuente: Jhathi.

[6] https://sourceforge.net/projects/jhathi/

2.3.2. SENDERO SCOUT

El año 2019, para la XVII versión de la Feria de Software[7], el equipo de Hammersoft desarrolló Sendero Scout, un proyecto conformado por dos aplicaciones móviles, una dirigida a dirigentes y otra dirigida a los niños y cuyo fin es el de "apoyar el trabajo que realizan los dirigentes sábado a sábado" (Departamento de Informática UTFSM 2019), y donde buscan una alternativa a las bitácoras y diarios de desafíos (cartillas de progresión). Este proyecto, al momento de escribir esta memoria, se enfoca completamente en la manada, una de las unidades que conforman las ramas menores[8], sin embargo, a futuro esperan poder continuar el desarrollo en las demás ramas con el fin de contribuir con transformación digital al movimiento scout a nivel nacional.

Algunas características de Sendero Scout[9] son la de permitir a los dirigentes designar tareas por medio de misiones, con las cuales el niño pueda seguir inmerso en el mundo scout fuera de las reuniones semanales, los dirigentes también pueden observar las estadísticas del progreso de los niños, los dirigentes y los niños pueden recibir retroalimentación por las actividades de la aplicación, entre otros.

La progresión de cada niño es realizada por los dirigentes y es visualizada en forma de estrellas por cada área de desarrollo, según el progreso de los niños en estas áreas los dirigentes reciben recomendaciones de actividades que pueden realizar para reforzar aquellas en las que los niños requieren más apoyo.

Esta aplicación también utiliza técnicas de gamificación, como son la entrega de insignias para recompensar el progreso de los niños semanalmente (ver fig. 5) e incluye un minijuego de trivia que funciona como una forma de mantener a los niños más activos en la aplicación y promover su aprendizaje de aspectos del movimiento, este minijuego es el primero de entre otros que esperan desarrollar a futuro.

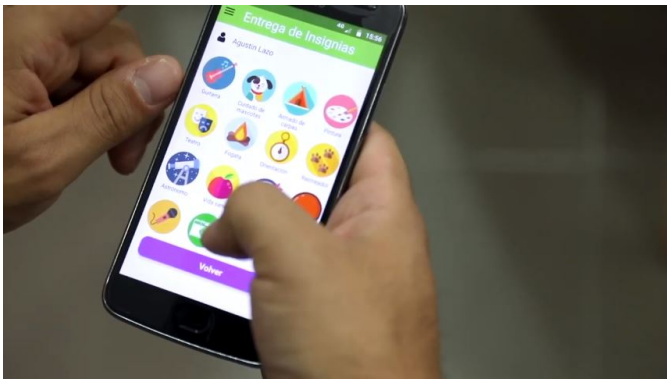


Figura 5: Aplicación de Sendero Scout.
Fuente: Departamento de Informática UTFSM 2019.

[7] http://www.feriadesoftware.cl/
[8] https://www.instagram.com/p/B5JLEp1nteZ/
[9] http://senderoscout.feriadesoftware.cl/proyectos.html

2.3.3. DIFERENCIACIÓN CON ESTADO DEL ARTE

El proyecto desarrollado para esta memoria se realizó con enfoque en las **ramas intermedias**, abriendo la posibilidad de delegar la responsabilidad de realizar el seguimiento de progresión personal a los mismos beneficiarios, y que guiadoras y dirigentes puedan dedicar mayor tiempo a gestionar actividades en base a aquellas áreas de desarrollo que requieran refuerzo.

Otro aspecto en el cual se buscó innovar con respecto a los demás proyectos, fue involucrar más el aspecto lúdico mediante la utilización de un entorno virtual y ávatar que se encuentre estrechamente relacionado con los avances del usuario en su progresión personal.

CAPÍTULO 3
PROPUESTA DE SOLUCIÓN

3.1. DEFINICIÓN DE LA METODOLOGÍA

Tal como se plantea en el artículo *"How to design gamification? A method for engineering gamified software"* de [Morschheuser y col. 2018], un parte aceptada como crucial para cualquier proyecto de *gamificación* es entender a los usuarios y el contexto en el que se encuentran, que en este caso corresponde a aquel en el que están inmersos los jóvenes guías y scout. Esto significa mantenerse dentro del marco simbólico, valores y misión del escultismo, y dentro de estos lograr conectar con guías y scouts con tal que se sientan atraídos a seguir utilizando la aplicación.

Sin embargo, en el período que se desarrolló este proyecto, la pandemia y las cuarentenas han vuelto complejo mantener actividades scout, que suelen estar muy conectadas con la naturaleza y el conocer el exterior, tanto la ciudad como el campo. Esta falta de actividades conlleva a los dirigentes a perder contacto directo con los beneficiarios, dependiendo únicamente de medios en línea.

Es importante considerar estos riesgos en el desarrollo de un proyecto de este tipo, y tener medidas de contención en caso que los planes iniciales fallen.

Para elegir la metodología se tomó en cuenta el tiempo disponible para desarrollar este proyecto y cuánto de este será dedicado a trabajar en cada parte del proyecto, que serían el servidor, la aplicación móvil y la aplicación web.

La metodología de trabajo fue elegida tomando componentes de **metodologías ágiles** que permitan flexibilidad en el desarrollo y que, en conjunto con prácticas de **DevOps**, ayuden a captar errores en fases tempranas del desarrollo mediante integración continua y el *feedback* constante de los *stakeholders*, reduciendo el riesgo por no haber logrado comprender en primera instancia, los requerimientos o el marco simbólico que orienta la *gamificación* de una metodología ya documentada en los manuales de las unidades de Tropa y Compañía, como es el seguimiento del desarrollo personal de los beneficiarios por parte de sus dirigentes y guiadoras.

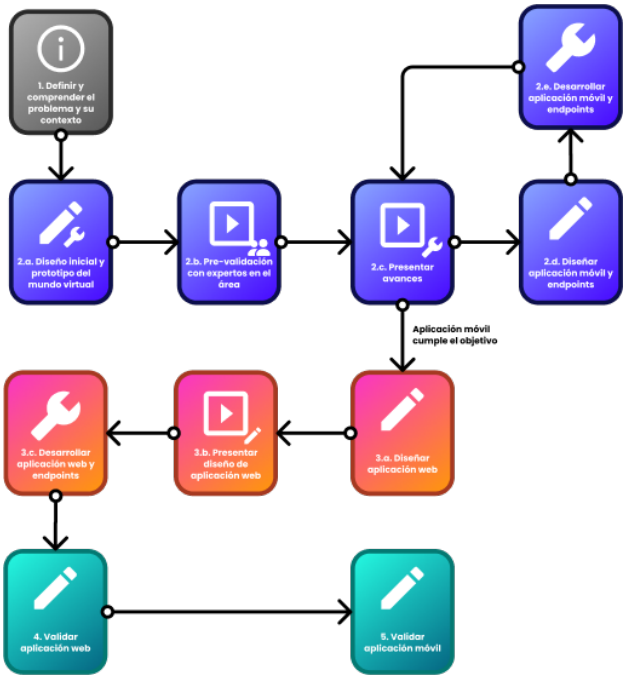


Figura 6: Metodología de Desarrollo
Fuente: Elaboración Propia

Considerando los puntos anteriores, la metodología de trabajo quedó como en la figura 6, donde el proceso fue dividido en 4 fases:

1. **Definición y comprensión del problema y su contexto:** Esto corresponde a una serie de reuniones con los dirigentes que concluyeron en los capítulos de *Definición del problema* y *Marco conceptual* de este documento. En esta fase se buscó que el desarrollador lograra obtener un conocimiento del dominio lo suficientemente amplio como para comprender qué buscan realmente los dirigentes con el *software* y bajo qué valores hay que desarrollar el mundo gamificado en el cual los beneficiarios se verán inmersos para ser motivados a realizar los registros de su bitácora o diario de desafíos.

2. **Diseño y desarrollo de la aplicación móvil:** Esta corresponde a la parte más *ágil* del desarrollo y la de mayor duración. El primer paso fue, utilizando el conocimiento obtenido en la primera fase, realizar una aproximación a un videojuego que sirva de base para construir el resto de la aplicación y consultar con conocedores del desarrollo de videojuegos sobre su opinión al respecto, el motivo de realizar este paso previo es porque, como es la primera vez del desarrollador creando un proyecto de gamificación, es mejor validar lo antes posible una idea y así, una vez recibido el *feedback* de los expertos, comenzar con la fase iterativa donde cada cierto período de tiempo de aproximadamente 3-4 semanas, se realiza una reunión con los dirigentes para recibir su *feedback* de los avances en la última iteración hasta un punto en el cual ya se de

por finalizada la aplicación para su validación con los beneficiarios. En cada paso del desarrollo se considera tanto el desarrollo del servidor con sus *endpoints* como de la aplicación móvil como tal.

3. **Diseño y desarrollo de la aplicación web:** En esta fase se utilizó una metodología más cercana a un desarrollo cascada [Balaji y Murugaiyan 2012], el motivo de esto es que la parte de mayor riesgo y tiempo de desarrollo es la aplicación móvil, por lo que era mejor utilizar la mayor parte del tiempo disponible en ello, y realizar una aplicación web en una iteración, donde se llevaron a cabo los pasos de diseño, desarrollo y validación del *software*. Al igual que en la fase 2, aquí también se considera el desarrollo de los *endpoints* necesarios en la API del servidor.

4. **Validación del proyecto:** Corresponde a la fase más importante, pues corresponde a la revisión por parte de dirigentes y beneficiarios de la aplicación web y móvil para ver que cumpla su objetivo y cuál será el trabajo futuro a realizar.

3.2. PRIMER DISEÑO Y PROTOTIPO DE LA APLICACIÓN MÓVIL

Como punto de partida para el primer diseño de la gamificación se tomó en cuenta lo revisado en la sección de Psicología de la Motivación (sección 2.2.1), donde se revisa cómo distintas necesidades psicológicas son satisfechas por ciertos elementos del diseño de los juegos. Un ejemplo de esto es la **Necesidad de autonomía (libertad de decisión)** que es satisfecha por medio de decisiones como lo puede ser elegir la ropa de un ávatar dentro de un mundo virtual.

Teniendo ya la piedra angular sobre la cual construir el resto del diseño de la *gamificación*, quedaría volverlo **scout**, para ello se analizó el **marco simbólico** de Tropa y Compañía titulado *"Explorar nuevos territorios con un grupo de amigos"* [Asociación de Guías y Scouts de Chile 2014].

De este marco simbólico se propone la idea de tener un mundo explorable, lo que también otorga **libertad de decisión** al permitir ir a donde uno desee, si bien la idea sería que entre grupos de amigos pudieran jugar entre sí, implementar un multijugador se escapa de lo factible en el tiempo de la memoria. Ahora quedaba ver cómo conectar esto con la progresión personal, para ello se tuvo la idea de que el mundo fuera compuesto por **niveles** que pudieran ser completados mediante un minijuego que sea directamente influenciado por el avance en la bitácora personal de cada beneficiario. De esta forma, se crea una sensación de progreso al ver cómo uno avanza por los niveles, al mismo tiempo que se avanza con los registros. Una referencia de esta idea sería el mundo de **Overcooked 2** de *Team17*, el cual es un conjunto de niveles que se accede explorando un mapa como el de la figura 7, incluso con niveles secretos escondidos tras montañas o activando ciertos *switches*.



Figura 7: Referencia de mundo explorable con niveles
Fuente: Overcooked 2

Además cada nivel estaría asignado a una de las seis áreas de desarrollo: **Corporalidad**, **Creatividad**, **Carácter Afectividad**, **Sociabilidad** y **Espiritualidad**, de forma que para avanzar en más niveles habría que realizar registros en distintas áreas, promoviendo un desarrollo integral.

El minijuego sería uno sólo, un juego bastante representativo de las actividades scout, que es la **guerra de pañoletas**, donde los participantes se tienen que poner la pañoleta en la parte de atrás del pantalón, y el objetivo es robar la pañoleta al contrincante. Entonces, la dificultad de los adversarios estaría directamente relacionada con el avance en la progresión personal de cada beneficiario, que serían más difíciles, o incluso imposibles de derrotar si uno no ha avanzado lo suficiente en la área de desarrollo correspondiente.

Sin embargo, en su momento esto se sentía como insuficiente para motivar a los beneficiarios a avanzar en su bitácora, pues sólo se estaba abarcando la **necesidad de competencia**, mediante puntos y un sentimiento de avance, mientras que la **necesidad de autonomía** no era suficiente y la de **relaciones sociales** se dejaba de lado completamente.

De estas dos últimas necesidades surgió una nueva idea, añadir la posibilidad de ganar ítems mediante el minijuego que pudiesen ser usados en el armado de un campamento dentro del mundo, de esta forma podrían cubrir su necesidad de autonomía y competencia al mismo tiempo, a la vez que podrían compartir este campamento con sus compañeros de grupo para satisfacer la necesidad de relaciones sociales.

En el anexo de este documento, se pueden encontrar más detalles sobre cómo se concretó este diseño utilizando el framework MDA.

3.2.1. CÓMO SE LLEVARÍAN LOS REGISTROS

Para llevar los registros se plantearon cuatro ideas no excluyentes entre sí para que los beneficiarios puedan realizar los registros de forma ordenada y que se puedan conectar con un sistema *gamificado*. Estas ideas fueron planteadas teniendo en consideración las instrucciones que provee el manual de rama de guías [Comisión Nacional de la Rama Guías 2018] y scouts [Comisión Nacional de la Rama Scouts 2016]:

1. **Un objetivo a la vez:** En la aplicación se tendría la lista de objetivos educativos propuestos que al beneficiario le quedan por completar según su edad, de los cuales tendría que elegir uno en el cual enfocarse a trabajar hasta completarlo, personalizando este objetivo a sus intereses y definiendo tareas específicas para éste, que tras completarlas considere que ha superado este objetivo.

2. **Registros granulares:** Esto corresponde a que cada cierto tiempo el beneficiario tendrá la posibilidad de ganar un nuevo ítem para su ávatar o para decorar su zona de campamento si registran algún avance en el objetivo que tienen activo, esto para motivar a realizar más registros que faciliten el seguimiento por parte de los dirigentes.

3. **Compleción de un objetivo activo:** Al terminar un objetivo activo tras completar todas las tareas que se asignó en un comienzo según lo definido en el punto 1, el beneficiario recibiría un punto de habilidad en el área de desarrollo correspondiente que le permitirá superar nuevos puntos de interés en el mundo virtual.

4. **Personalización:** Esto es análogo a un elemento planteado en los manuales del mismo nombre, corresponde a que, al entrar por primera vez a la aplicación, se le pedirá al beneficiario marcar como completados aquellos objetivos que ya se hayan considerado como ya logrados según lo conversado con su guiadora o dirigente.

3.2.2. PRESENTACIÓN A DIRIGENTES

Las ideas anteriores fueron presentadas a los dirigentes en una videoconferencia para validar si eran coherentes con lo planteado por el movimiento. Éstas fueron recibidas positivamente por los dirigentes, afirmando que la guerra de pañoletas les parecía un juego bastante representativo de los juegos de scout y que todas las ideas de cómo llevar los registros les parecen coherentes con lo que el movimiento plantea.

También se les presentaron unas primeras ideas de cómo podría ser la aplicación web en la que realizará el seguimiento de la progresión:

- El registro se podría ver en móvil y web utilizando un *framework* como Ionic[10].
- Un dirigente del grupo podría registrar este grupo desde la aplicación web y desde ahí invitar al resto de dirigentes.
- En el *dashboard* de cada grupo, guiadoras y dirigentes pueden ver gráficos representativos de la progresión por unidad, patrulla o beneficiario.

3.2.3. PROTOTIPO

Para el desarrollo del prototipo se utilizó el motor de videojuegos **Unity** en su versión 2019, dada su flexibilidad de uso para crear entornos 3D y a la experiencia que el desarrollador tiene con esta herramienta. La plataforma objetivo corresponde al sistema operativo para móviles **Android**, pues es la más común a la que pueden tener acceso los beneficiarios, aunque esto también implica que se requerirá un mayor esfuerzo en la optimización para que funcione en la mayor cantidad de dispositivos móviles. Por lo mismo, el proyecto fue creado utilizando el *Universal Render Pipeline* que está diseñado para funcionar mejor en móvil, además de entregar herramientas que facilitan la creación de visuales llamativas[11], que son importantes para un proyecto dirigido a jóvenes.

El prototipo, que se puede observar en la figura 8, se realizó con un personaje temporal (*placeholder*) con un jugador que se puede mover en un entorno 3D mediante un *joystick* en la pantalla y con dos botones, uno para saltar y otro para robar la pañoleta. Existía también una **barra de *stamina***, la cual al vaciarse limitaba las acciones que uno podía hacer por un corto período de tiempo, con tal de evitar que los jugadores puedan ganar sólo presionando el botón de robar continuamente. Además, esta barra al vaciarse te volvía vulnerable a los ataques de los contrincantes si querían robar tu pañoleta.

Los contrincantes corresponden a esferas azules que sólo se pueden mover esperando un corto tiempo en posición antes de cargar en contra del jugador para reducir su *stamina* y luego robar su pañoleta.

[10] https://ionicframework.com/

[11] https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@10.2/manual/index.html

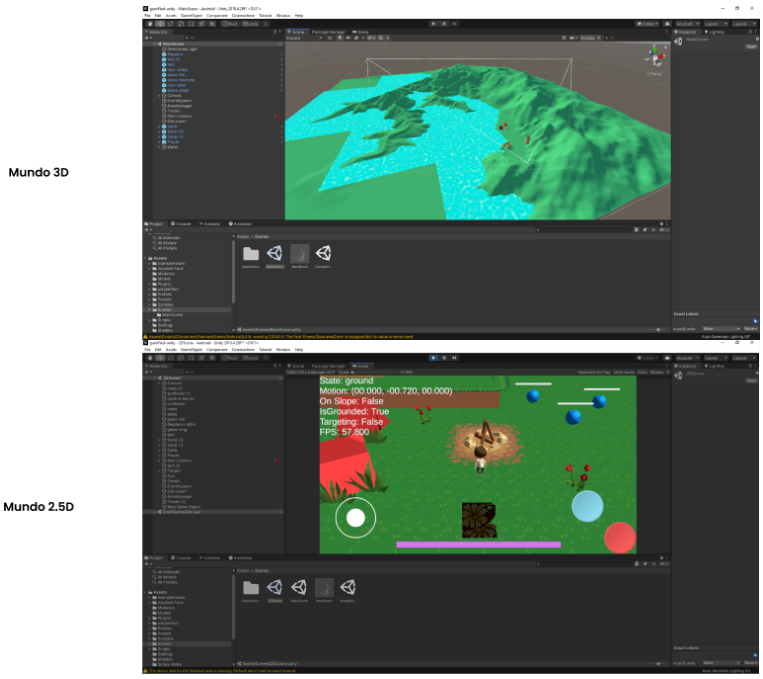


Figura 8: Prototipo en 2.5D y 3D
Fuente: Elaboración propia

3.2.4. REUNIÓN CON EXPERTOS Y CAMBIO DE PLANES

Ya teniendo ordenadas las ideas y el prototipo, se tuvo una reunión con dos profesores de la asignatura de **INF334 - TALLER DE CREACIÓN DE VIDEOJUEGOS** de la **Universidad Técnica Federico Santa María**, **Sven Von Brand** y **Claudio Bustamante**, quienes tienen amplia experiencia en el área de desarrollo de videojuegos y aceptaron amablemente a participar en esta etapa del proyecto.

La reunión fue realizada por videollamada mediante la plataforma **Google Meet**, y se les presentaron las ideas teniendo de apoyo los diagramas en **Miro** y el prototipo. Durante la reunión que duró aproximadamente 1 hora, los profesionales plantearon aspectos que sería mejor ajustar en favor de un proyecto más sólido y que pudiera ser desarrollado en menos tiempo.

Las conclusiones a las que se pudo llegar son las siguientes:

- Al menos en esta etapa del proyecto, sería mejor **dejar de lado aquellos aspectos que sean demasiado videojuego y poco *gamificación***, tal como la guerra de pañoletas.
- Se puede, en cambio, **potenciar la parte de decorar zona de campamento y su personalización**, que es la que se puede *gamificar* añadiendo más elementos comunes de

los videojuegos de exploración y que pudiesen ser conectados con los registros de la bitácora, como podrían ser:

  - *Crafting:* Crear objetos a partir de otros.
  - *Minijuegos:* Los minijuegos no tienen por qué desaparecer, pero que no sean el centro de la aplicación en esta fase.
  - *Excavar:* Encontrar ítems en el piso de las zonas de campamento.
  - *Eventos:* Que aleatoriamente ocurran cosas como que aparezcan personajes temporales que vendan o regalen algún ítem, que venga de visita un animal con el que se pueda interactuar, u otros eventos que le den vida al mundo virtual.

- Una preocupación que manifestó el desarrollador junto a los profesores fue la de **integridad de los datos**, sobre cómo mitigar que los registros realizados sean inconsistentes, para ello se plantearon algunas ideas como:

  - Integrar un sistema de moderación para revisar casos sospechosos.
  - Notificar al beneficiario que la información será revisada por los dirigentes previo a subir un registro.
  - Revisión entre pares de los registros.

- Una idea interesante planteada por **Sven Von Brand** es que las *"Relaciones humanas mejoran la duración de un juego"* , esto se puede aplicar en el proyecto de alguna forma para mantener la motivación de los beneficiarios en realizar sus registros mediante la integración de mecánicas enfocadas en la necesidad de **relaciones sociales** planteada en el marco conceptual.

También plantearon nuevas ideas para que la gamificación se integre más dentro de las actividades del grupo que, si bien no serían parte de la etapa actual del proyecto, eran interesantes para aplicar en el futuro:

- Que guiadoras y dirigentes puedan entregar puntos de desarrollo a los beneficiarios al término de cada reunión semanal.
- Poder otorgar puntaje a los beneficiarios manualmente por actividades fuera del alcance actual de la aplicación.

Con el *feedback* recibido, ya fue posible continuar desarrollando un proyecto de *gamificación* más sólido.

3.3. DISEÑO DE LA APLICACIÓN

3.3.1. FRAMEWORK *OCTALYSIS*

En un esfuerzo por potenciar más la *gamificación* se decidió utilizar un *framework* de *gamificación* que resultó ser coherente con el punto de comienzo utilizado anteriormente, que son las necesidades psicológicas planteadas por [Rigby y Ryan 2011]. Este es **Octalysis**, un *framework* de *gamificación* creado por [Chou 2019] que plantea un nuevo término para *gamificación*: *Human-Focused Design*, en oposición al desarrollo tradicional, al que se refiere como *Function-Focused Design* en [Chou 2019]. En base a esta idea, Chou plantea que los videojuegos apelan a 8 unidades centrales de motivación o *"core drives"* en los seres humanos -por ende el nombre *Octalysis*-.

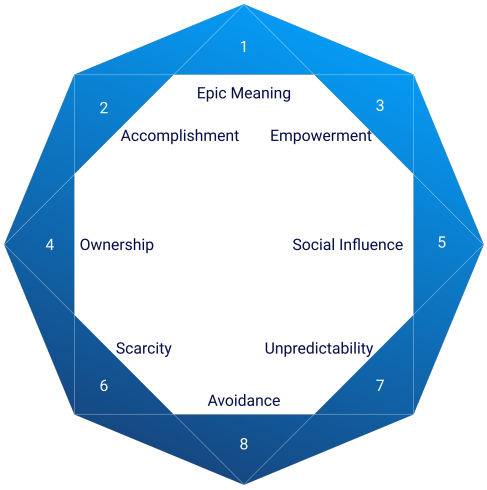


Figura 9: Octalysis
Fuente: Yu-kai Chou: Gamification & Behavioral Design

De cada uno de estos motivadores se pueden analizar distintos proyectos de gamificación o incluso videojuegos viendo a cuáles de estas unidades están apelando. En la tabla 2 se ordenaron estas unidades por descripción y ejemplos de cómo se apela a estas en proyectos de software.

Además, según la posición de cada unidad en la figura 9 a lo largo de cierto eje, adquiere ciertas características:

- **Eje vertical:** Describe si la **emoción** a la que se recurre para motivar es **positiva** o **negativa**, siendo las de más arriba emociones positivas, como alegría y satisfacción, mientras que las de abajo corresponden a emociones negativas, como el miedo o la ansiedad. Esto no significa que haya que evitar estas últimas siempre, pues si se utilizan correctamente pueden ayudar a mejorar la productividad o promover hábitos

Tabla 2: Unidades centrales de Octalysis

Fuente: Elaboración Propia

Nombre	**Traducción**	**Descripción**	**Ejemplos**
Epic Meaning & Calling	**Significado épico y el "llamado"**	El jugador siente que está haciendo algo enorme o para lo que es "el/la elegido/a"	Narrativa épica o suerte de principiante
Development & Accomplishment	**Desarrollo y realización**	El jugador siente que está realizando progreso, desarrollando habilidades o superando **desafíos**	Puntos, logros, barra de progreso
Empowerment of Creativity & Feedback	**Potenciación de la creatividad y la retroalimentación**	El jugador puede expresar su **creatividad** y ver los resultados por ello	*Feedback* inmediato, elegir qué poner en algunos lugares, desbloquear *milestones* (hitos)
Ownership & Possession	**Propiedad y Posesión**	Cuando el jugador sienten que algo les pertenece	Puntos intercambiables, bienes virtuales, ávatar
Social Influence & Relatedness	**Influencia social y familiaridad**	El jugador se siente influído por distintas **necesidades sociales** o porque algo es "familiar"	Agregar amigos, regalar ítems, botón para *presumir*
Scarcity & Impatience	**Escasez e impaciencia**	El jugador quiere algo porque no puede tenerlo	Cuenta atrás, *last mile drive* (motivación por estar a punto de terminar)
Unpredictability & Curiosity	**Impredecibilidad y curiosidad**	El jugador quiere saber qué ocurrirá después	Recompensas aleatorias, interfaz que evoluciona junto al jugador
Loss & Avoidance	**Pérdida y evitación**	El jugador tiene miedo a perder algo en lo que dedicó mucho tiempo o una oportunidad que podría no repetirse	Pérdida de progreso, oportunidad temporal

saludables, por ejemplo.

- **Eje horizontal:** Describe si la **naturaleza** de estos motivadores es **extrínseca** (más lógica) o **intrínseca** (más arraigada a aspectos creativos o sociales).

3.3.2. PLANIFICACIÓN CON *OCTALYSIS*

Ya estudiado el *framework*, se decidió que para una siguiente reunión con los dirigentes se plantearían una serie de elementos que abarquen cada una de las unidades del *Octalysis*, para luego priorizarlos en base a dos factores: **coherencia con el movimiento**, y qué tanto, en base a lo observado por los dirigentes en su experiencia con los beneficiarios, estos **contribuirían a la motivación de los beneficiarios**, utilizando como punto de partida la **naturaleza** (eje horizontal) y **emoción** (eje vertical) de cada motivador.

Esta priorización servirá para filtrar aquellos elementos que se podrían dejar fuera de la aplicación temporalmente, quedando para trabajo futuro o directamente eliminarlos si escapan de lo que se busca con el proyecto. A cada uno de los elementos se les asignó un tiempo estimado de desarrollo al que se le multiplicó una **holgura** y **factor de complejidad** que aumentan estos valores con el fin de amortiguar los errores de estimación que pueda haber cometido el desarrollador en su estimación.

La metodología utilizada durante la reunión fue la siguiente:

1. **Exponer** a los dirigentes cada uno de los motivadores planteados *Octalysis*.
2. Solicitar a los dirigentes que **ejemplifiquen** cuándo estos motivadores han estado presentes en alguna actividad scout, si es que han estado. El objetivo de esto es que se familiaricen con estos motivadores llevándolos a su área de experiencia.
3. **Enumerar** elementos de *gamificación* propuestos por el desarrollador, clasificados por unidad central de *Octalysis*.
4. **Discutir** con los dirigentes con el fin de **priorizar** y **filtrar** estos elementos, donde los dirigentes toman el rol de **expertos en el dominio** y el autor de este documento el de **experto en el desarrollo** y **conocedor del *framework***.

Mediante este sistema de discusión, se pudieron llegar a varias conclusiones que relacionan el movimiento scout con *Octalysis*:

- El movimiento scout utiliza en distintos momentos, ya sea a propósito o no, cada una de las unidades motivadoras.

- Motivadores dirigidos a **emociones positivas** como significado épico, realización o empoderamiento suelen ser apeladas por las actividades scout mismas.
- Motivadores dirigidos a **emociones negativas** como escasez e impredecibilidad son principalmente apeladas por las situaciones de campamento, donde los recursos son limitados y la naturaleza impredecible.
- El motivador de **influencia social y familiaridad** juega un rol muy importante en el movimiento, lo que se ve reflejado en el marco simbólico de Tropa y Compañía que se llama “Explorar nuevos terrenos con un grupo de amigos”

Por otro lado, se encuentran las conclusiones con respecto a cómo hay que enfocar la aplicación:

- Tal como dice el último punto de la lista anterior, hay que incluir al menos un elemento de **influencia social y familiaridad**, que incluso podría aplicarse por fuera de la aplicación como enviarse capturas de pantalla de cosas que hayan creado en la aplicación.
- Hay que incluir elementos de motivadores tanto **extrínsicos** como **intrínsicos**, pues el grupo de beneficiarios es bastante heterogéneo, por lo tanto, aquello que los motiva también.
- Si bien tanto motivadores de emociones positivas como negativas participan en algún u otro momento dentro del movimiento, es más coherente con éste los motivadores dirigidos a **emociones positivas**.

Habiendo estudiado el contexto del movimiento scout, es posible ya elegir qué elementos de *gamificación* incluir a partir de las mecánicas de un prototipo de armado de campamento realizado para optimizar el tiempo disponible de desarrollo de este proyecto. Este prototipo se puede ver en la figura 10, fue creado como videojuego para luego usar de base para el mundo virtual de este proyecto, utilizando el *feedback* entregado por los expertos en la sección anterior.

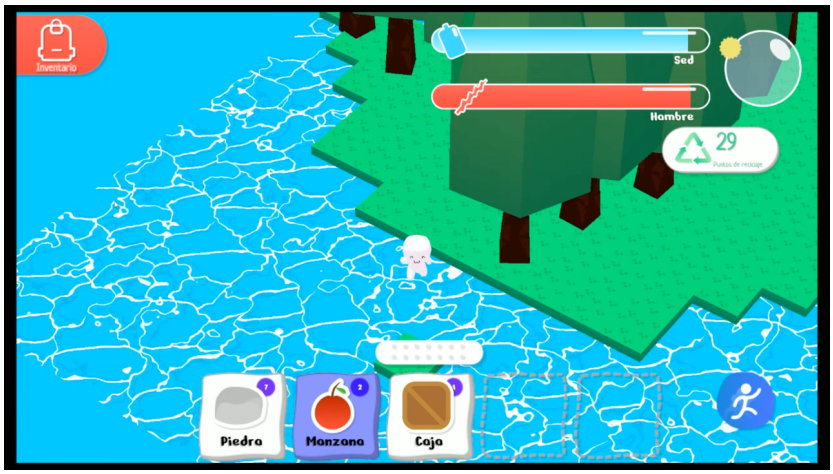


Figura 10: Segundo prototipo de videojuego Scout titulado “Espíritu Scout”

3.3.3. EL DISEÑO FINAL

Tras varias iteraciones y ajustes durante el desarrollo, se llegó al concepto final del proyecto y a un nombre, **Espíritu Scout**, bajo la idea que el ávatar y el mundo son una representación gamificada del crecimiento de un beneficiario dentro del movimiento Scout.

El concepto general del proyecto se describe en el siguiente **elevator pitch**:

> **Espíritu Scout** es una aplicación móvil para la gamificación del registro de avances de progresión personal de guías y scouts mediante un campamento virtual, donde cada guía y scout tiene un personaje y zona de campamento que pueden ir decorando con aquellos objetos que ganen con el registro de avances en su bitácora o diario de desafíos, de forma de motivarlos a llevar este registro al día y mejorar la capacidad de seguimiento de la progresión personal por parte de guiadoras y dirigentes al poder ver estadísticas de la progresión personal de su grupo desde una aplicación web.

Para explicar el diseño de la aplicación móvil se revisarán por separado cómo se creó el mundo virtual y el sistema de registros de la aplicación, para luego explicar cómo se integraron estos dos utilizando gamificación con *Octalysis*.

DEL MUNDO SCOUT AL MUNDO VIRTUAL

Para diseñar un mundo virtual para la aplicación, se tuvieron en cuenta dos aspectos:

1. Que sea interesante para los beneficiarios para que lo sigan utilizando.

2. Que se sienta Scout.

Estos dos puntos fueron discutidos con los dirigentes para conectar dos mundos, el **mundo de los campamentos Scout** y el **mundo de los videojuegos**, más específicamente, videojuegos de tipo *sandbox*, donde el jugador puede modificar el mundo, en este caso su zona de campamento, de muchas maneras.

Aquella pregunta realizada a los dirigentes, con la cual se buscó lograr esta conexión fue **"¿Qué hace que un campamento sea Scout, y no un campamento cualquiera?"**, con el fin de crear un entorno virtual que se sienta scout. Uno de los aspectos principales que acá surge es que cada participante de un campamento scout trabaja en hacer las cosas por el o ella misma, por ejemplo, armar su zona de campamento, encender un fogata con las precauciones necesarias, con tal de no ser dependiente por completo de elementos pre-hechos.

Junto a lo anterior se suma el uso del marco simbólico de scout [Asociación de Guías y Scouts de Chile 2014] que fue considerado para el primer diseño.

Dado que la innovación de este proyecto, y aquella idea que se quiere validar, es la integración de este mundo virtual con la gamificación, y no sus mecánicas, se revisaron mecánicas de videojuegos ya validadas en la industria y que sean compatibles con los puntos anteriores. Algunos referentes son videojuegos como **Animal Crossing: New Horizons** (figura 11 de *Nintendo*], que entrega la posibilidad al jugador de personalizar su ávatar y la ciudad, lo que se puede homologar en este proyecto, a personalizar la zona de campamento.



Figura 11: Captura de Animal Crossing: New Horizons

Otros videojuegos también utilizados como referentes son **Minecraft** de *Mojang* dada su alta capacidad de personalización, **Subnautica** de *Unknown Worlds Entertainment* dada su mecánica simplificada para realizar *crafting* (crear objetos a partir de otros), **Fall Guys: Ultimate Knockout**, que se usó de referente para las formas de personalización de personaje.

Aquellas ideas y mecánicas que calificaron, finalmente para el diseño del mundo virtual fueron:

- **Co-existencia con la naturaleza:** La zona de campamento comienza casi vacía, con árboles, flores y otros objetos naturales, el objetivo del beneficiario es utilizar los objetos que tenga para co-existir con la naturaleza, no destruírla. Por ejemplo, en vez de cortar un árbol, el jugador puede recoger troncos ya caídos para crear asientos o usarlos para crear otros objetos de madera.
- **Personalización del ávatar:** El ávatar se puede personalizar cambiando cada uno de los siguientes aspectos de él: Ojos, boca, prenda superior, prenda inferior y pañoleta. Este último ya que es un objeto característico de los scouts.
- **Movimiento del ávatar:** El beneficiario puede controlar este ávatar en el entorno virtual 3D, pudiendo incluso saltar.
- **Sistema de inventario:** Esto significa que el jugador puede recoger elementos del entorno y guardarlos para usar más tarde. La capacidad del inventario es infinita.
- ***Crafting*:** Los objetos del inventario pueden ser utilizados para crear nuevos objetos.
- **Interacción con el entorno:** Existen objetos con los que es posible interactuar, ya sea poniéndose en frente y presionando un botón, o arrastrando un objeto encima.
- **Ciclo día y noche:** Mientras más tiempo pase el jugador en el mundo, el día pasa, cambiando la iluminación del lugar y ocurriendo distintos eventos según la hora del día.
- **Aparición aleatoria de objetos:** Cada cierto tiempo pueden aparecer elementos en el mundo elegidos aleatoriamente, que pueden ser de utilidad al jugador para crear objetos o para interactuar.
- **Entorno modular:** El entorno virtual se divide en **Zonas**, en un comienzo el jugador parte con un espacio pequeño con el cual construir, pero al avanzar en su progresión personal, obtiene más zonas con las que puede expandir este mundo.
- **Personalización del mundo:** Algunos objetos se pueden poner desde el inventario en el mundo, con tal de que el beneficiario pueda personalizar su zona de campamento a su gusto.

La implementación de estas mecánicas en el motor de videojuegos **Unity** será explicada más adelante.

Otras ideas que fueron implementadas en el prototipo, pero que se descartaron finalmente, pues al probarlas con jugadores resultaron negativas para la experiencia usuaria (quizás con más tiempo de desarrollo podría haberse superado lo negativo) son:

- **Sistema de diálogo:** Existía la posibilidad de conversar con NPCs (Personajes no jugables) que daban instrucciones del lugar o incluso podían vender cosas al jugador. El motivo por el que fue eliminado este sistema es porque la herramienta utilizada para crear los diálogos solía entorpecer el uso de la aplicación al detener por completo la experiencia, sin la posibilidad de usar un botón para saltar el diálogo.
- **Tiendas:** Algunos personajes podían vender objetos a cambio de puntos de cada área de desarrollo. Este concepto requería más trabajo para integrarse bien con el sistema de registro.
- **Necesidades (hambre y sed):** El ávatar pasaba hambre y sed al pasar el tiempo, que podía recuperar comiendo algunos objetos específicos como manzanas. Esta mecánica sólo daba una sensación de estrés al jugador, sin otorgar un beneficio a cambio, por lo que fue removida.

DEL MUNDO SCOUT A LA BITÁCORA

El **sistema de registros de progresión** no sufrió grandes cambios durante las iteraciones de diseño y desarrollo, pues los dirigentes estaban bastante conformes con el diseño inicial y comentaron que les parecía bien adecuado para la forma en que funciona el sistema de progresión personal actualmente.

Los componentes que conforman el sistema de registro de progresión son los siguientes:

1. **Personalización:** No ha de confundirse con el sistema de personalización de ávatar, que es parte del mundo virtual. Corresponde al momento en que el beneficiario marca en la aplicación aquellas tareas que han sido completadas desde antes de usar la aplicación. Su nombre proviene del período de personalización descrito en el manual de rama guía [Comisión Nacional de la Rama Guías 2018] y scout [Comisión Nacional de la Rama Scouts 2016].
2. **Objetivos activos:** La forma de avanzar en los objetivos, una vez realizada la personalización, es completarlos uno por uno. Para ello, el beneficiario ha de elegir en qué objetivo desea enfocarse y decidir qué hacer para trabajar en este objetivo. Para esto, el proceso de selección de objetivo activo consiste de tres fases:
    a) **Enfocarse:** El beneficiario elige un objetivo de la propuesta de objetivos planteados por el movimiento.
    b) **Hacerlo propio:** El beneficiario describe en pocas palabras qué le gustaría hacer para trabajar en el objetivo propuesto. A este se le llama el **objetivo personal**.
    c) **Concretarlo:** El beneficiario define hasta 6 **tareas específicas** que correspondan a las suficientes para que el objetivo personal se considere completado.

3. **Registros granulares:** Durante el desarrollo de los objetivos, los beneficiarios pueden ir registrando los pequeños avances que realicen cada día en su objetivo en forma de **registros granulares**.

INTEGRACIÓN DE LA BITÁCORA Y EL MUNDO VIRTUAL

En la integración de la bitácora con el mundo virtual es donde entra con mayor fuerza la gamificación. Para esto se desarrolló un **sistema de recompensas**, que se encarga de convertir diferentes acciones de la aplicación, en recompensas que se vean reflejadas en el mundo virtual. Estas recompensas pueden ser las siguientes:

- **Puntos:** Puntaje que puede ser sumado al puntaje de una o más áreas de desarrollo.
- **Decoración:** Objetos decorativos para el mundo virtual.
- **Prenda de ávatar:** Prendas para personalizar el ávatar del beneficiario. Pueden ser ojos, boca, prenda superior, prenda inferior o una pañoleta.
- **Zonas:** Zonas para expandir el mundo virtual.

La implementación de estas recompensas son aleatorias según la acción realizada. Donde las **acciones recompensadas** son:

- Completar paso de personalización de objetivos.
- Completar un objetivo.
- Realizar un registro granular.

La recompensa aleatoria fue implementada permitiendo al beneficiario elegir entre un conjunto de posibles grupos de recompensas, esto permite reforzar las unidad de **impredecibilidad** del Octalysis al dar al usuario una sensación de control sobre una situación impredecible.

ELEMENTOS DE GAMIFICACIÓN POR UNIDAD CENTRAL DE *OCTALYSIS*

A continuación se ordenaron por unidad central de Octalysis los elementos de gamificación que se consideraron para este proyecto, y su estado durante la instancia de validación.

Estos elementos pueden ser o una funcionalidad, o una consecuencia de un conjunto de funcionalidades. Además existen elementos que refuerzan más de una unidad a la vez, como se observa en tabla 3.

Tabla 3: Funcionalidades de la aplicación por Unidad Central de Octalysis

Fuente: Elaboración Propia

Elemento	**Estado**
Epic Meaning & Calling	
—	
Development & Accomplishment	
Usar captura de ávatar como foto de perfil	Completado
Grandes recompensas por objetivo completado	Completado
Gráfico de radar para ver desarrollo integral	Completado
Puntaje por área de desarrollo	Completado
Estadística para medir desarrollo integral	Completado
Empowerment of Creativity & Feedback	
Personalizar ávatar	Completado
Personalizar mundo con decoración obtenida por acciones	Completado
Ownership & Possession	
Poseer un ávatar propio	Completado
Poseer un mundo propio	Completado
Social Influence & Relatedness	
Usar foto de ávatar como imagen de perfil	Completado
Mundo virtual usa elementos del movimiento Scout	Completado
Misiones de grupo	Pospuesto
Scarcity & Impatience	
Esperar un día para ganar una recompensa por registro granular	Completado
Sistema de necesidades: Hambre y sed	Cancelado
Unpredictability & Curiosity	
Recompensas aleatorias por personalización de objetivos	Completado
Recompensas aleatorias por objetivo completado	Completado
Recompensas aleatorias por registro granular	Completado
Loss & Avoidance	
—	

3.4. ESTRUCTURA GLOBAL DEL PROYECTO

En las siguientes secciones se entregan los aspectos más técnicos de cómo se implementó el diseño planteado. Pero antes, en esta sección se revisará cómo se organizó el proyecto en diferentes repositorios alojados en GitHub bajo el nombre de organización Paths Ankan[12]:

- `pps-resources`: Es un repositorio diseñado con la intención de ser incluído como un *sub-módulo*[13] en los demás repositorios. Su nombre viene de *Personal Progression System Resources*, pues contiene principalmente archivos JSON con datos estáticos utilizados en el proyecto, como valores por defecto o la organización de la propuesta de objetivos del sistema de progresión.
- `scout-progression-system-sam`: Corresponde al código del **back-end** en **Python**, más específicamente, de la REST API y el sistema de recursos de AWS que se utilizó descritos mediante **AWS SAM**. Se encuentra integrado con GitHub Actions para realizar un *deployment* automático con cada commit a la rama `master`, de forma que la rama `develop` queda únicamente para los avances que no están lo suficientemente estables para que sean subidos a producción.
- `scout-spirit`: Corresponde al código de la aplicación móvil en **Dart**, utilizando el framework **Flutter**.
- `scout-spirit-unity`: Corresponde al código del mundo virtual en 3D desarrollado en **C#** utilizando **Unity**. Está pensado para ser incluido dentro de `scout-spirit` como un *sub-módulo*. Es el único repositorio privado, dado que se están utilizando *assets* descargados por medio de la Asset Store, y que, dada su licencia, no pueden ser distribuídos públicamente.
- `scout-spirit-web-app`: Corresponde al código de la aplicación web para guiadoras, dirigentes y otros agentes educativos, desarrollado en **TypeScript** utilizando **Angular 11**.

3.5. BACK-END: LEVANTANDO EL SERVIDOR CON AWS

El servidor se levantó en el servicio de computación en la nube **Amazon Web Services** (AWS) que provee varias herramientas que facilitan crear un *back-end* robusto en poco tiempo.

En las siguientes sub-secciones se revisará el ***stack* de tecnologías** utilizado y cómo fue organizado en el **entorno de trabajo**. Luego se revisará cómo se realizó el **diseño de la base de datos** y cómo se encuentra estructurado el código y la plataforma en la **arquitectura utilizada**.

[12] https://github.com/pankandev
[13] https://git-scm.com/book/en/v2/Git-Tools-Submodules

3.5.1. STACK DE TECNOLOGÍAS

La herramienta que se utilizó para levantar el servidor en AWS es **AWS Serverless Application Model**[14] (AWS SAM) que corresponde a un framework para estructurar aplicaciones *serverless* mediante un archivo YAML[15] que define los recursos AWS a utilizar y cómo interactúan entre ellos. Esto incluye REST APIs con sus endpoints, diseño de la base de datos, *buckets* de archivos, recursos de autentificación como AWS Cognito, CDNs, etc. A continuación se enlistan los principales recursos AWS utilizados en este proyecto:

- **Lambda Functions**[16]**:** Este recurso es el que permite ejecutar código en varios lenguajes de programación sin necesidad de un servidor completo, pues son funciones ejecutadas sólo cuando se utilizan. Para este proyecto las funciones *Lambda* fueron utilizadas para describir el comportamiento de los endpoints de la REST API, principalmente para comunicar con la base de datos DynamoDB utilizando Python.
- **API Gateway**[17]**:** Este recurso permite diseñar una API ***RESTful*** definiendo los endpoints bajo los que se ejecutará cierta función *Lambda*. Además por cada endpoint se puede definir un autorizador (*authorizer*) que se encargue de asegurar que quien está llamando a un endpoint sea quien dice ser, para que luego los permisos para ese usuario específico sean verificados por la función *Lambda*.
- **DynamoDB Tables**[18]**:** DynamoDB es una base de datos NoSQL (Not-Only SQL) distribuída de tipo documental llave-valor, esto quiere decir que permite guardar datos con mayor flexibilidad que en SQL permitiendo trabajar sin un esquema definido mas que para la llave primaria y además una gran velocidad de acceso por su sistema distribuído. Más adelante se revisará el diseño de esta base de datos.
- **Cognito User Pools**[19]**:** Cognito corresponde a un servicio de Amazon para gestionar usuarios de forma segura almacenándolos en *User Pools*. Además Cognito permite dar a usuarios permisos a otros recursos de la plataforma mediante un *Identity Provider* para permitir, por ejemplo, que usuarios puedan subir archivos a un Bucket S3.
- **Bucket S3**[20]**:** Abreviación para *Simple Storage Service* (S3), corresponde al servicio en el cual se suben todos los archivos de la plataforma, como son los archivos estáticos de la aplicación web, el código de las funciones *Lambda* y los archivos subidos por los usuarios.

[14] https://aws.amazon.com/serverless/sam/
[15] https://yaml.org/
[16] https://aws.amazon.com/lambda/
[17] https://aws.amazon.com/api-gateway/
[18] https://aws.amazon.com/dynamodb/
[19] https://aws.amazon.com/cognito/
[20] https://aws.amazon.com/s3/

- **CloudFront**[21]**:** Corresponde a un *Content Delivery Network* (CDN), esto significa que permite tener en *caché* los responses para ciertas rutas de un dominio en AWS, reduciendo considerablemente la latencia: Además, CloudFront permite tener el acceso a la REST API, aplicación web y archivos subidos por los usuarios bajo un mismo dominio.

El código del back-end fue subido a un repositorio público de GitHub[22], siendo considerado entonces como una plataforma *Open Source*.

Para realizar una integración continua de la aplicación también se utilizó **Travis CI**, que realizó *testing* y *deployment* por cada *commit* al repositorio por gran parte del desarrollo. Sin embargo, este fue reemplazado cerca del final del desarrollo por **GitHub Actions** al terminarse los créditos del plan gratuito de Travis CI.

3.5.2. ENTORNO DE TRABAJO

El entorno de trabajo fue dispuesto utilizando **AWS SAM CLI** dada sus herramientas de *testing* local de la API y **Docker Compose** para levantar la base de datos localmente. Este conjunto de herramientas permite tener un servidor local, que tiene más latencia que subirlo a AWS, sin embargo esto se ve compensado al no tener que esperar que se realice el *deployment* de la aplicación por cada cambio en el código, agilizando el desarrollo además de reducir costos en subida de archivos.

3.5.3. DISEÑO DE LA BASE DE DATOS

Qué es DynamoDB

La base de datos en DynamoDB se conforma, al igual que una BD con SQL, de **tablas**, donde cada tabla se conforma de **filas**, que vendrían a ser los documentos. Por cada tabla existen uno o dos atributos que conforman la llave primaria, y en las cuales se tuvieron las siguientes consideraciones:

- **HASH (PARTITION) KEY** (Obligatoria): Se encarga que las filas con una misma *HASH KEY* pertenezcan siempre a una misma partición. Si no se utiliza una *SORT KEY*, esta debiese ser única, de lo contrario, se debe buscar que esta llave cumpla lo siguiente:
    1. En combinación con la *SORT KEY*, se debiese formar un valor único por cada fila.
    2. Debe ser lo suficientemente única como para no sobrecargar una partición.

[21] https://aws.amazon.com/cloudfront/
[22] https://github.com/pankandev/scout-progression-system-sam/

3. Debe ser lo suficientemente amplia como para que, las operaciones que necesita la aplicación, sean realizables mediante una *query*, que sólo se pueden ejecutar en una partición, de lo contrario hay que utilizar una operación *scan* que requiere mayor nivel de procesamiento.

- **RANGE (SORT) KEY** (Opcional): Define el orden en que se almacenan las filas en una partición. Es con esta llave en la que se pueden realizar filtros en una *query*, los que pueden ser, por ejemplo:

  - **Si la *RANGE KEY* es un número:** Obtener filas donde su *RANGE KEY* esté entre dos números, sea mayor a algún valor, sea igual a algún valor, etc.
  - **Si la *RANGE KEY* es un *string*:** Obtener filas donde su *RANGE KEY* comience con algún *sub-string* (esto se utilizó bastante en el diseño de la base de datos), que sea exactamente igual a algún *string*.

Al utilizar el *Free Tier*[23] que ofrece AWS para esta etapa del desarrollo, existe un límite de unidades de capacidad de escritura y lectura en la base de datos, los llamados **RCU** (*Read Capacity Units*) y **WCU** (*Write Capacity Units*) que definen cuánto nivel de procesamiento (*throughput*) tendrá la base de datos. Por ejemplo 1 RCU permite una lectura consistente o dos eventualmente consistentes para ítems de hasta 4KB, mientras que 1 WCU representa una escritura para un ítem de hasta 1KB.

El *Free Tier* permite tener hasta 25 WCU y 25 RCU, por lo que el nivel de procesamiento de la BD se encuentra restringido por estos valores.

Diseño

Para explicar el diseño de la base de datos se comenzará con el ER Diagram (*Entity Relationship Diagram*) que explica las entidades que componen el proyecto y sus relaciones en la figura 12.

[23] https://aws.amazon.com/free/

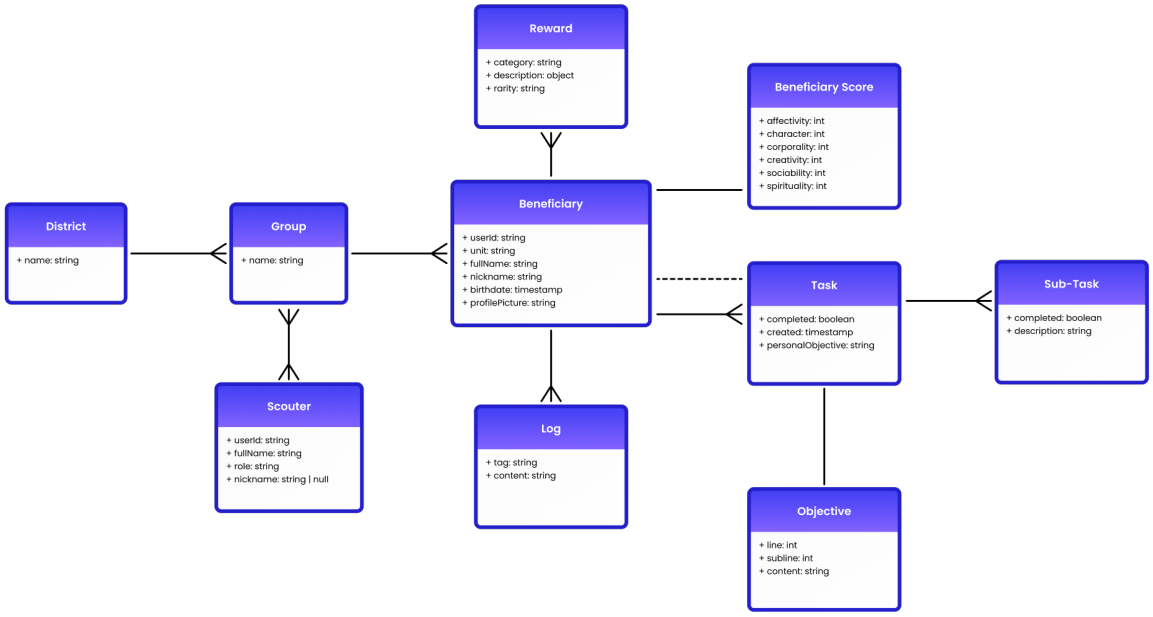


Figura 12: Entity Relationship Diagram de “Espíritu Scout”
Fuente: Elaboración Propia

Las entidades más relevantes son las siguientes:

- **Distrito (*District*):** Es el nivel de organización territorial más grande, un distrito se conforma de varios grupos guía y scout.
- **Grupo (*Group*):** Corresponde a la estructura base de la AGSCh. En un grupo participan **guiadoras y dirigentes** (*scouters*) y los **beneficiarios** (*beneficiaries*).
- **Guiadoras y dirigentes (*Scouter*):** Son los interesados en realizar seguimiento de la progresión personal de los beneficiarios. Una guiadora o dirigente puede participar en más de un grupo a la vez.
- **Beneficiarios (*Beneficiary*):** Son la entidad central de este proyecto, pues son quienes realizan los registros de progresión, marcan objetivos como cumplidos y generan las estadísticas que luego guiadoras y dirigentes usarán para realizar el seguimiento de progresión. Además cada beneficiario tiene un puntaje asociado por cada área de desarrollo (*Beneficiary Score*)
- **Registros (*Log*):** Son diversos registros generados mediante la interacción de los beneficiarios con la aplicación, como pueden ser los registros granulares, cuando el beneficiario gana una recompensa, completa un objetivo, etc.
- **Recompensas (*Reward*):** Son las recompensas que un beneficiario puede ganar, pueden ser de muchos tipos y con descripciones diferentes entre sí que definen aspectos de una recompensa como su color.

Tabla 4: Tablas de la Base de Datos

Fuente: Elaboración Propia

Tabla	Índice	*Partition Key*	*Sort Key*	*Queries*	*RCU/WCU*
districts	*Default*	`code`	–	D1	1/1
groups	*Default*	`district`	`code`	G1	2/2
beneficiaries	*Default*	`userId`	–	–	3/4
beneficiaries	ByGroup	`groupCode`	`join( unitCode, userId )`	B1, B2	4/3
logs	*Default*	`userId`	`join( ...tags )`	L1	5/5
rewards	*Default*	`category`	`releaseId`	R1	5/5
tasks	*Default*	`userId`	`objectiveId`	T1	5/5

- **Tareas (*Task*):** Corresponden a los objetivos personales de un beneficiario, ya sea que ya ha cumplido, o que está progresando en él. Cada uno de estos posee sub-tares (*Sub-Task*), que son actividades más concretas de qué hacer para completar un objetivo personal.
- **Objective (*Objective*):** Corresponde a los objetivos propuestos por el sistema de objetivos, de aquí surgen los objetivos personales (*Task*).

Las tablas finales de la base de datos se encuentran descritas en la tabla 4. Y su proceso de diseño a partir de las *queries* utilizadas en la aplicación, se puede encontrar en el anexo de este documento.

3.5.4. ARQUITECTURA UTILIZADA

El código del *back-end* utiliza una arquitectura de **micro-servicios**, utilizando 8 **funciones Lambda** encargadas cada una de un recurso específico de la **REST API** que se utilizará para que la aplicación web y móvil interactúen con la base de datos.

Como se puede observar en la figura 13, la API se encuentra definida por medio del servicio **Amazon API Gateway** que, a partir de una *request* HTTPS enviada por el cliente, llama a la función Lambda correspondiente para interactuar con la base de datos, ya sea para escribir o leer datos.

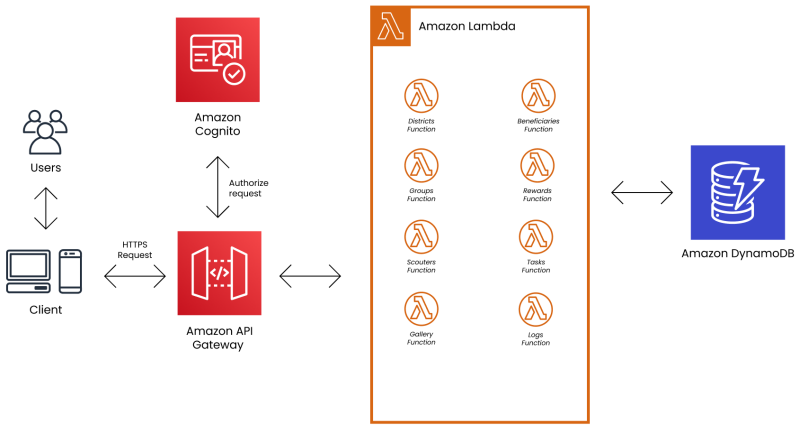


Figura 13: Arquitectura de la REST API
Fuente: Elaboración Propia

Funciones Lambda

Aun siendo micro-servicios, existe código y dependencias comunes para cada función Lambda, lo que ocuparía un espacio innecesario en el bucket S3 que almacena el código de estas funciones. Es por esto que AWS provee los llamados **Layers**, que son un archivo comprimido que guarda todos estos archivos comunes y que son cargados en cada función Lambda al llamarse, reduciendo el espacio ocupado por estas funciones.

Autentificación

Cada *request* es autorizada por medio del servicio **Amazon Cognito**, que recibe un *JWT* enviado por medio del *header "Authorization"* al que los clientes tienen acceso por medio de un **Amazon Cognito User Pool Client** configurado para realizar inicio de sesión mediante el protocolo *SRP* (*Secure Remote Password*), que habilita la autentificación sin enviar la contraseña directamente al servidor, previniendo el robo de contraseñas de alguien que esté observando la conexión.

Además, este cliente se encuentra conectado con un **Amazon Cognito Identity Pool**, que se encarga de permitir el acceso a los usuarios directamente a ciertos recursos, como lo es el *Bucket S3* dedicado a almacenar los archivos subidos por usuarios, más específicamente imágenes. Este *Bucket* fue utilizado para que los beneficiarios puedan cambiar su foto de perfil en la aplicación.

Organización de Usuarios

Amazon Cognito permite organizar los usuarios en los llamados **User Pools**. En este proyecto existen tres tipos de usuarios: *Scouters* (guiadoras y dirigentes), *Beneficiaries* (beneficiarios) y *Admins* (administradores), sin embargo sólo se utilizó una *User Pool* llamada `users`, donde para diferenciar entre tipos de usuarios se utilizan **Grupos de usuarios** que permiten dar

permisos diferentes a ciertos conjuntos de usuarios.

Al grupo *Admins* sólo es posible unirse siendo un desarrollador actualizando al usuario desde la consola de AWS. Mientras que a los grupos *Scouters* y *Beneficiaries* los usuarios son agregados automáticamente al unirse a un grupo scout desde la aplicación web o móvil respectivamente.

CDN

La aplicación web fue construida como un SPA (*Single-Page Application*) utilizando Angular, donde toda la interacción con el *back-end* es realizada por medio de JavaScript asíncronamente. Esto significa que la parte web de esta aplicación se conforma sólo de archivos estáticos (`.js`, `.html` y `.css`) que no cambian, por lo que pueden ser guardados en *caché* para que sean accesibles más rápidamente que si se obtuvieran directamente desde el *Bucket S3* donde están guardados y, usualmente, a menor costo.

Este trabajo de guardarlos en *caché* es realizado por el *CDN* de AWS llamado **Amazon CloudFront**, que permite dejar diferentes orígenes, como pueden ser los *Bucket S3* bajo un CDN con reglas de *caching* específicas para distintos orígenes.

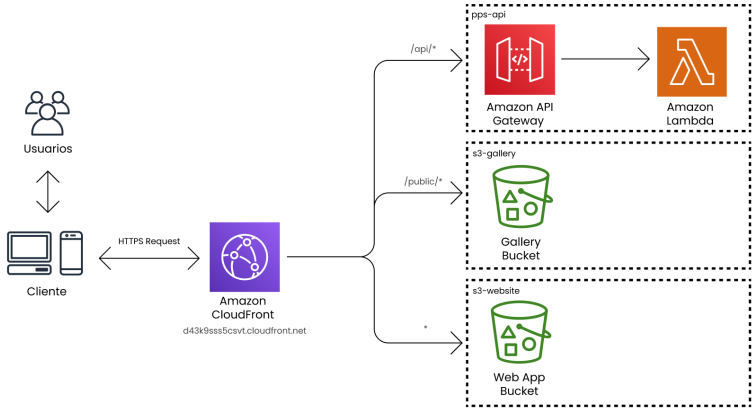


Figura 14: Distribución de CloudFront
Fuente: Elaboración Propia

Los orígenes utilizados se encuentran representados por la figura 14, en él se describen los siguientes orígenes:

- `pps-api`: Es el acceso a la REST API, este origen se encuentra detrás de CloudFront únicamente para compartir dominio con el resto de los orígenes, pues su *caché* se encuentra deshabilitado. Se puede acceder tras las rutas "`/api/*`".
- `s3-gallery`: Es el acceso a las imágenes subidas por los beneficiarios a S3, que en

este punto del desarrollo corresponde únicamente a sus imágenes de perfil. Se puede acceder tras las rutas "`/public/*`".

- `s3-website`: Es el acceso a la aplicación web para dirigentes y guiadoras. Esta aplicación se encuentra almacenada en su propio *Bucket S3*. Se puede acceder tras las rutas "`*`".

Sistema de recompensas

El sistema de recompensas utilizado para la *gamificación* fue implementado a nivel de *backend* en la REST API, cuyo *endpoint* será definido en la siguiente sección.

Este sistema fue implementado como el medio con el cual recibir recompensas al ejecutar ciertos *endpoints*, y poder reclamarlas más adelante al servidor, verificando que esta recompensa sea válida.

Para esto se consideró la utilización de *tokens* firmados por el servidor para validar su legitimidad. La tecnología utilizada para esto fue *JWT* (JSON Web Token) firmados con una JWK (JSON Web Key) por el servidor.

De esta forma, el servidor puede generar los que se llamaron ***tokens* de recompensa** al ejecutar ciertos *endpoints* específicos bajo ciertas condiciones. Estos *tokens* luego se pueden enviar por medio de otro *endpoint* para reclamar las recompensas que éste guarda.

El motivo por el cual no se entregan las recompensas inmediatamente al ejecutar un *endpoint* es para que los beneficiarios puedan elegir entre un conjunto de recompensas que están indicados en el *token* y, además, poder separar la responsabilidad de estos *endpoint*.

El contenido de un *token* de recompensa es el siguiente:

```
{
  "sub": "<user-id>",
  "iat": 12345,
  "exp": 12345,
  "static": [...],
  "boxes": [[...], [...], ...],
  "index": 42,
  "area": "<development-area>",
  "reason": "<reward-reason>"
}
```

Donde:

- `sub`: ID del usuario al que le pertenece este *token*.
- `iat`: Timestamp de emisión de este *token*.
- `exp`: Timestamp de expiración de este *token*.
- `static`: Son las recompensas que el beneficiario va a obtener definitivamente al reclamar este *token*.
- `boxes`: Son una lista de listas de recompensas. El beneficiario puede ganar una de estas recompensas al reclamar este *token*, que se suman a las recompensas definidas en `static`.
- `index`: Es el índice de este *token*. Permite evitar que este *token* se reclame más de una vez por el beneficiario.
- `area`: Define si este *token* fue generado para recompensas de un área de desarrollo en específico, por ejemplo, otorgar puntos a sólo un área de desarrollo.
- `reason`: Especifica el motivo por el cual se obtuvo esta recompensa.

3.5.5. REST API

La forma que utiliza tanto la aplicación móvil como la aplicación web para interactuar con el *back-end* de la aplicación, es mediante la REST API. El diseño de esta API se puede encontrar en el anexo de este documento, donde se detalla la lista completa de los endpoints disponibles en la plataforma clasificados por micro-servicio (o sea, función Lambda).

3.5.6. INTEGRACIÓN CONTINUA

Con el fin de agilizar el proceso de *deployment*, se decidió utilizar un sistema de CI (*Continuous Integration*) que instala dependencias, ejecuta los *testings* unitarios por medio de *pytest* y, en caso de tener resultados satisfactorios, realiza el *build* y *deploy* de la plataforma en AWS.

La figura 15 describe el proceso de integración continua realizado por Travis CI y, más adelante en el proyecto, por GitHub Actions para la integración continua.

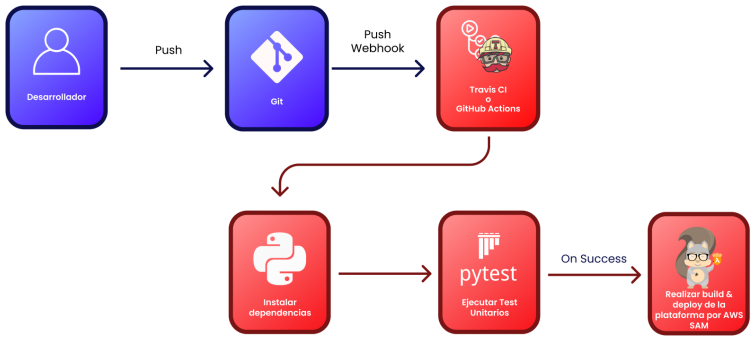


Figura 15: Proceso de Integración Continua
Fuente: Elaboración Propia

3.6. FRONT-END: PARA BENEFICIARIOS

La aplicación móvil corresponde a aquella parte del proyecto dirigida a los beneficiarios y a la implementación de la gamificación.

A continuación se explicarán las tecnologías utilizadas para el desarrollo de esta parte, para luego revisar cómo se implementó la comunicación con el mundo virtual en Unity y finalmente mostrar las vistas principales de la aplicación.

3.6.1. STACK DE TECNOLOGÍAS

La principal tecnología utilizada fue **Flutter**, un *kit* de herramientas de UI para el desarrollo móvil creado por Google para el lenguaje de programación **Dart**. *Flutter* es una herramienta de desarrollo muy ágil y con una comunidad muy activa al momento de escribir este documento, existiendo una gran variedad de paquetes que facilitan diferentes partes del desarrollo.

Uno de estos paquetes que es parte crítica del proyecto, es `flutter_unity_widget`[24] desarrollado por *xraph.com*. Pues lo que provee es un *widget* de **Flutter** que permite integrar una aplicación desarrollada en el motor de videojuegos Unity compilada para Android o IOs, dentro de esta aplicación, además de habilitar la comunicación bi-direccional entre Unity y Flutter. De esta forma se posee tanto la agilidad de desarrollo que ofrece **Flutter** para interfaces, como la capacidad de crear entornos 3D interactivos de Unity.

[24]https://pub.dev/packages/flutter_unity_widget

Otro de los paquetes relevantes para el desarrollo de esta aplicación fue `amplify_flutter`[25]. AWS Amplify es un conjunto de herramientas que permite levantar un *back-end* simple en poco tiempo, a la vez que permite conectar sin problemas *front-end* desarrollados en diversas herramientas con AWS. Para este proyecto sólo se utilizó el paquete para agilizar la conexión desde **Flutter** al servidor con los recursos ya creados por AWS SAM. Si bien hubo que realizar algunas configuraciones extras, fue más rápido que hacer las llamadas HTTP desde cero. Además, Amplify ofrece herramientas muy útiles para la autentificación, almacenando automáticamente el estado actual de éste, y realizando la obtención y refresco del *JWT* utilizado para las llamadas a la API por medio del Amazon Cognito User Pool Client correspondiente.

Finalmente, para almacenar datos del usuario como el estado actual de su mundo y su inventario, se utilizó una base de datos NoSQL interna llamada *Hive*.

3.6.2. ARQUITECTURA UTILIZADA

A continuación se revisarán las partes del código más críticas del proyecto, como son: la estructura MVC utilizada para comunicar con la REST API, cambiar el estado de la aplicación y actualizar la vista acorde, la comunicación entre Flutter y Unity, y el almacenamiento de los datos del usuario utilizando Hive.

Patrón MVC

El patrón MVC en Flutter es bastante útil para separar la lógica de negocio (modelos), el cambio de estados (controladores) y la lógica de presentación (vistas). En este código se aplicó por medio de **Modelos**, **Servicios** y **Widgets** respectivamente:

- **Modelos (*Model*):** Son clases que guardan la estructura de los datos recibidos por la API por medio de los servicios.
- **Servicios (*Controller*):** Corresponden a clases, usualmente *Singleton* (sólo permiten utilizar una instancia en toda la aplicación), encargadas de cambiar el estado de la aplicación en base a lo solicitado por los Widgets y los datos que solicita desde la REST API.
- **Widgets (*View*):** El concepto *Widget* es específico de **Flutter**, refiriéndose a elementos de la interfaz que pueden ser reutilizados en distintas partes de la aplicación.

El elemento central para la comunicación con la REST API son entonces los servicios. Si bien AWS Amplify ofrece herramientas para comunicarse con la REST API, éstas no lograron interactuar correctamente con su propio sistema de autentificación, por ello se implementó la

[25] https://pub.dev/packages/amplify_flutter

interacción por medio del paquete `http`[26], un paquete provisto por *Dart Team* para realizar este tipo tareas, y se integró con el sistema de autentificación de AWS Amplify para obtener el *JWT* válido que actualmente permite comunicarse con la API.

La comunicación de los servicios se encuentra descrita por la figura 16.

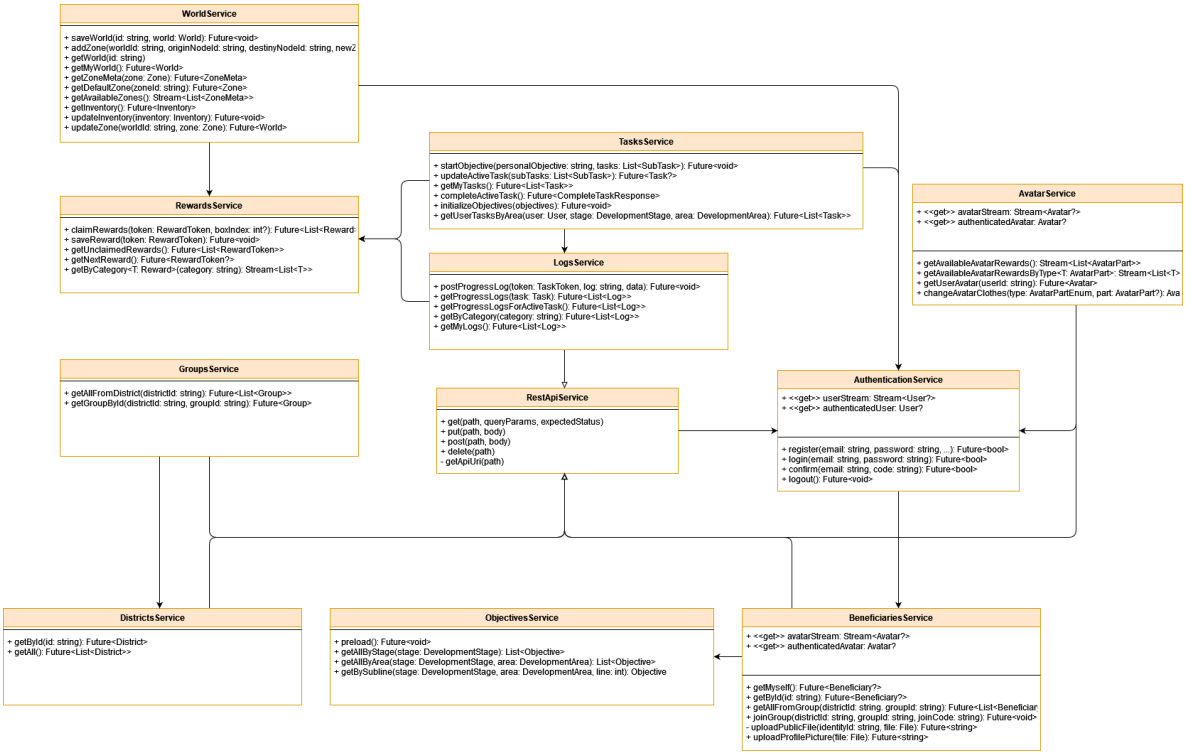


Figura 16: Servicios de la Aplicación Móvil

Comunicación Flutter-Unity

Como se comentó anteriormente, la comunicación bi-direccional entre Flutter y Unity fue provista por el paquete `flutter_unity_widget`. Sin embargo, si no se mantenía organizada esta comunicación, considerando la gran cantidad de mensajes que se tienen que enviar ambas partes, era fácil perder el orden y causar inconsistencias.

Las herramientas disponibles para la comunicación son dos: enviar un mensaje desde Flutter a Unity y enviar un mensaje desde Unity a Flutter. Se consideraron dos cosas para implementar el sistema de comunicación:

1. Para reducir el acoplamiento, la mayor parte de la comunicación es de Unity pidiendo datos a Flutter asíncronamente. Unity no se comunica directamente con la REST API.

[26]https://pub.dev/packages/http

2. Flutter necesita enviar mensajes a Unity, pero no esperar una respuesta. Por ejemplo, al presionar un botón en la interfaz de Flutter, se espera que Unity reaccione de alguna manera.

El segundo punto ya es cubierto con las herramientas que provee `flutter_unity_widget`, mientras que el primer punto requiere utilizar un patrón de mensajería. Por ello, se implementó un sistema de mensajería que utiliza el patrón *request-response*, donde una entidad, en este caso Unity, envía un mensaje a otra, que sería Flutter, esperando una respuesta. Se eligió este patrón pues permite a Unity solicitar datos asíncronamente a Flutter.

La implementación, en grandes rasgos, se realizó asignando a cada *request* de Unity un ID incremental, partiendo de 1, y este ID es enviado junto a los datos de la solicitud -que se revisarán en los siguientes parrafos- para que Flutter, al completar la solicitud, envíe su *response* a Unity con este número, tal que Unity esté informado de a qué *request* corresponde esta respuesta.

El contenido de los mensajes enviados entre ambas partes sólo puede ser un *string*, es por esto que se decidió utilizar JSON en toda la comunicación.

Los *requests* realizados por Unity poseen la siguiente estructura:

```
{
    "index": 42,
    "method": "<method-name>",
    "arguments": {...}
}
```

Donde:

- `index`: ID del *request*.
- `method`: Nombre del método al que se quiere llamar en Flutter. Por ejemplo `GetAvatar`, `SaveWorld`, etc.
- `arguments`: Es un objeto en formato JSON entregado como argumento al método informado en la llave `method`.

Ahora quedaría informar a Flutter de qué método usar cuando reciba un *request* desde Unity. Para esto se utilizó el patrón *Publish-Subscribe* tal que, al inicializar la aplicación de Unity dentro de Flutter, se suscribe un método de Flutter a algún *string* correspondiente a alguno de los nombres enviados en la llave `method` desde Unity. De esta forma, al recibir un *request*, se ejecutará el método correspondiente en Flutter, y su respuesta será enviada a modo de respuesta a Unity.

3.6.3. DISEÑO DE LA BASE DE DATOS

La base de datos interna de la aplicación móvil utilizada fue *Hive*, una base de datos NoSQL de tipo llave-valor.

Hive funciona ordenando los datos en *Boxes* (cajas), donde una caja puede almacenar uno o varios tipos de dato. En esta etapa del desarrollo sólo se utilizaron para almacenar los datos del mundo del jugador, su inventario y sus *tokens* de recompensa disponibles. El motivo de guardar estos *tokens* es por si el usuario cierra la aplicación antes de reclamar este *token*, este puede volver a intentar reclamarlo al volver a abrir la aplicación.

Entonces existen tres cajas:

- `world`: Almacena el mundo de un beneficiario. Usa como llave el id del usuario.
- `inventory`: Almacena el inventario de un beneficiario. Usa como llave el id del usuario.
- `rewards`: Almacena los *tokens* de recompensa disponibles para el beneficiario. Usa como llave una combinación del ID del usuario y el `index` del *token*.

3.6.4. VISTAS DE LA APLICACIÓN

A continuación, con el fin de guiar al lector por la implementación final de la aplicación, se revisarán las vistas que conforman la aplicación móvil clasificadas por función y ordenadas por orden aproximado de aparición para un nuevo beneficiario.

Autentificación

La figura 17 corresponde a la pantalla inicial de la aplicación, desde la cual un beneficiario puede acceder inmediatamente a su cuenta, o crear una nueva cuenta que lo lleve a las vistas de registro presentadas en la figura 18.



Figura 17: Inicio de sesión

Las vistas de registro de la figura 18, más específicamente las de las imágenes *a* y *b*, permiten crear una cuenta en la aplicación por medio de Amazon Cognito en dos pasos: **Credenciales** y **Datos personales**. Para finalmente, en la imagen *c*, confirmar la cuenta mediante un código enviado al correo electrónico del beneficiario (este es un paso requerido por Amazon Cognito):

Unirse a un grupo

Al iniciar sesión por primera vez, se solicita al beneficiario unirse a un grupo, y de eso consisten las vistas de la figura 19, donde los beneficiarios indican el distrito y grupo al que pertenecen, para que luego tengan que ingresar el código de grupo que les proveerá su guiadora o dirigente.

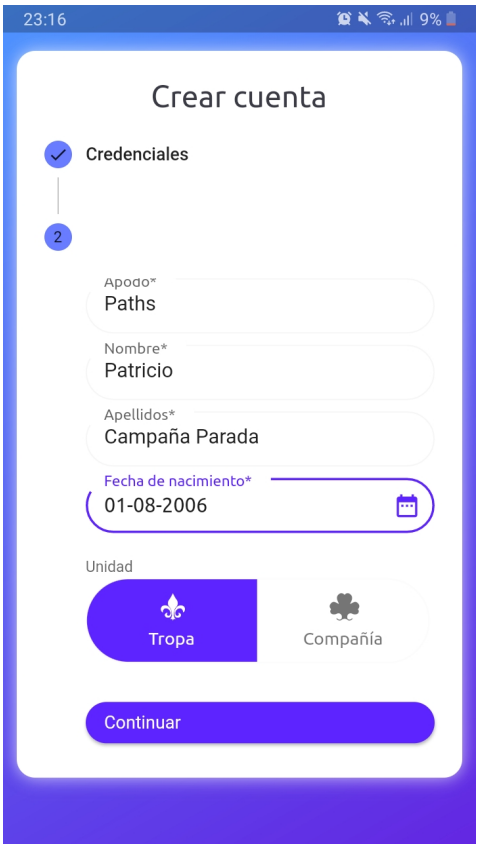
Menú Principal

La vista de la figura 20 corresponden al menú principal, desde aquí se puede acceder a todo el resto de la aplicación. De arriba hacia abajo se pueden ver:

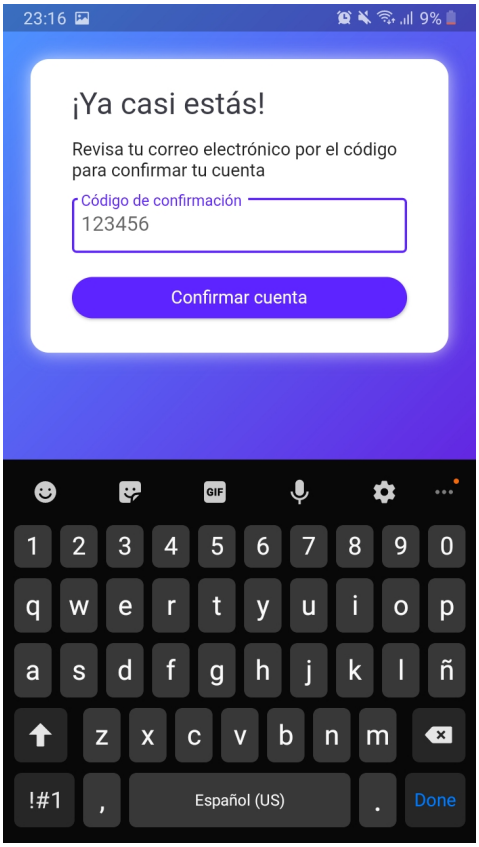
1. Un cuadro con la información básica del perfil del beneficiario y un botón para editar el ávatar.
2. En el caso de no haber realizado el paso de personalización de objetivos (donde se indican los objetivos previamente completados), una alerta indicando que aun debe realizar ese paso.
3. El objetivo activo del beneficiario o un botón para comenzar a seleccionar el objetivo.
4. Botones para acceder a las vistas de Bitácora, Estadísticas y Registros.
5. Los créditos.



(a) Registro: Credenciales



(b) Registro: Datos personales



(c) Confirmación de cuenta

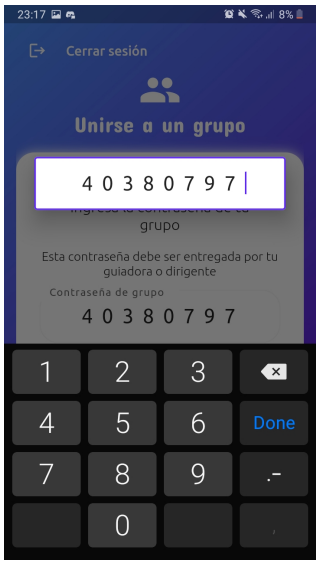
Figura 18: Registro



(a) Seleccionar distrito



(b) Seleccionar grupo



(c) Ingresar código de grupo

Figura 19: Unirse a un grupo

6. Abajo a la izquierda hay un *Floating Action Button* para acceder al campamento virtual del beneficiario.

Edición de Ávatar

En la figura 21 se observa lo que es la edición del ávatar del beneficiario. Esto se implementó mediante un *widget* de Unity que recibe los cambios que uno realiza en Flutter, y al presionar guardar, envía estos cambios a la REST API para actualizar el Ávatar en la base de datos.

Las partes del ávatar que se pueden cambiar son: cada ojo por separado, la boca, la prenda superior, la prenda inferior y la pañoleta.

En la esquina superior derecha se pueden observar los botones para guardar los cambios y tomar una captura del ávatar. Al sacar una captura del ávatar, Flutter envía un mensaje a Unity solicitando la imagen, luego Unity realiza la captura y retorna a Flutter la ruta de la imagen generada, para finalmente mostrar al usuario la vista *c*, que permite cortar la imagen como desee para usar como imagen de perfil.

Bitácora y estadísticas

Las vistas de la figura 22 son el histórico de cómo va el beneficiario con su progresión personal. Se divide en dos pantallas:

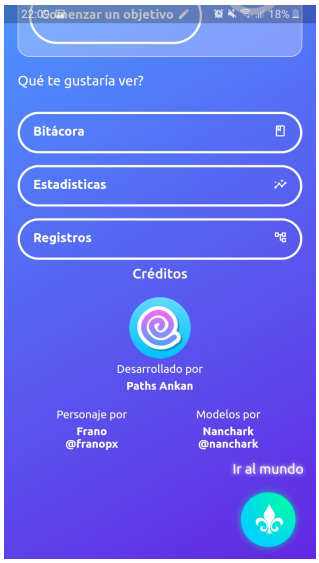
1. **Bitácora:** Aquí se muestra cómo va el beneficiario a nivel de cantidad objetivos cumplidos, teniendo acceso a su lista de objetivos cumplidos y a un gráfico de radar mostrando la cantidad de objetivos por área relativa en una sóla imagen.



(a) Menú principal de nuevo usuario



(b) Menú principal con objetivo activo

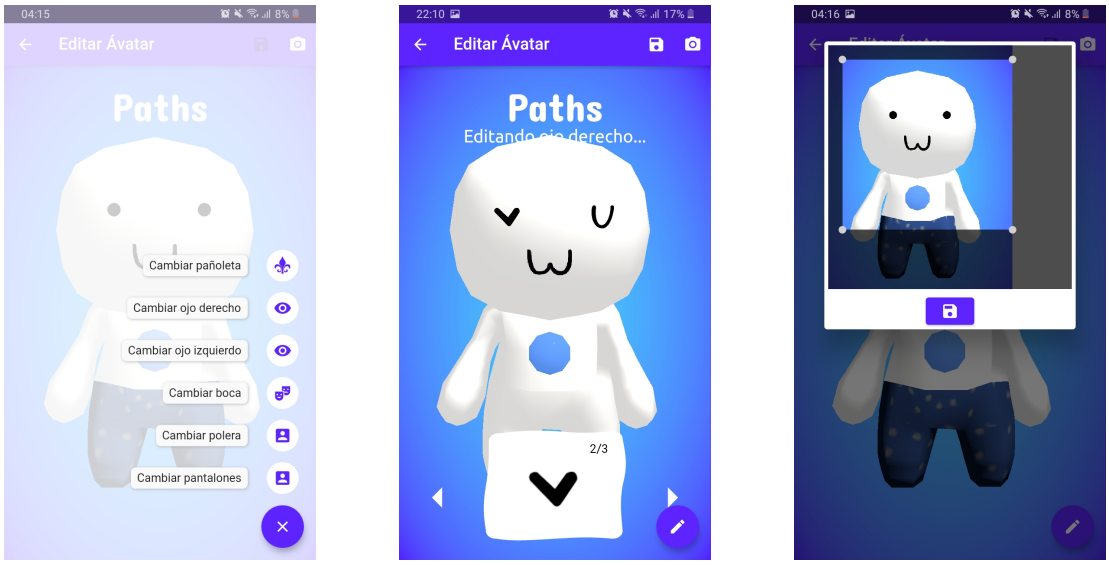


(c) Menú principal: Créditos



(d) Hint: ¿Qué es un objetivo activo?

Figura 20: Menú Principal



(a) Selección de parte de ávatar (b) Edición de ojo derecho (c) Toma de captura de ávatar

Figura 21: Edición de ávatar

2. **Estadísticas:** Aquí se muestra cómo va el beneficiario con respecto a su puntaje, el puntaje corresponde a una medida de cuánto ha avanzado el beneficiario en cada área de desarrollo considerando, no sólo objetivos cumplidos, sino también registros granulares o personalización de objetivos. En esta vista también se puede ver el **factor de desarrollo integral**, que corresponde a un parámetro calculado a partir del porcentaje de área cubierta en el gráfico de radar de puntajes, sirviendo como una guía de qué tan integral ha sido el desarrollo del beneficiario en su uso de la aplicación.

Personalización (de objetivos)

La personalización de objetivos de la figura 23 se realiza área por área, seleccionando por cada una los objetivos que ya considera completados desde antes de usar la aplicación. Al completar este procedimiento desaparece el mensaje de alerta del menú principal.

Formulario para elegir objetivo activo

Al presionar el botón de “Comenzar un objetivo” del menú principal, como se muestra en la figura 20a, se despliega el formulario para comenzar un objetivo activo, como se muestra en la figura 24. Este se compone de tres partes, como se menciona en la sección 3.3.3:

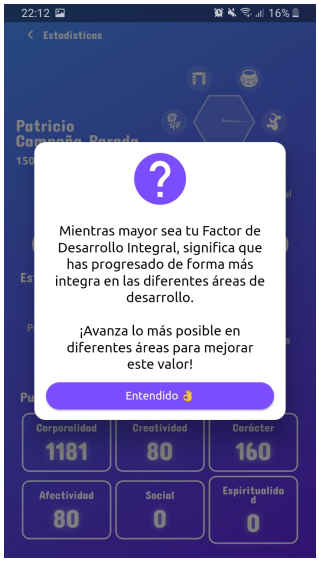
- **Enfócate:** El beneficiario elige el objetivo propuesto en el que quiere trabajar.
- **Hazlo tuyo:** El beneficiario adapta este objetivo a sus propios intereses de desarrollo.
- **Concrétalo:** El beneficiario elige tareas específicas para cumplir este objetivo.



(a) Vista de bitácora

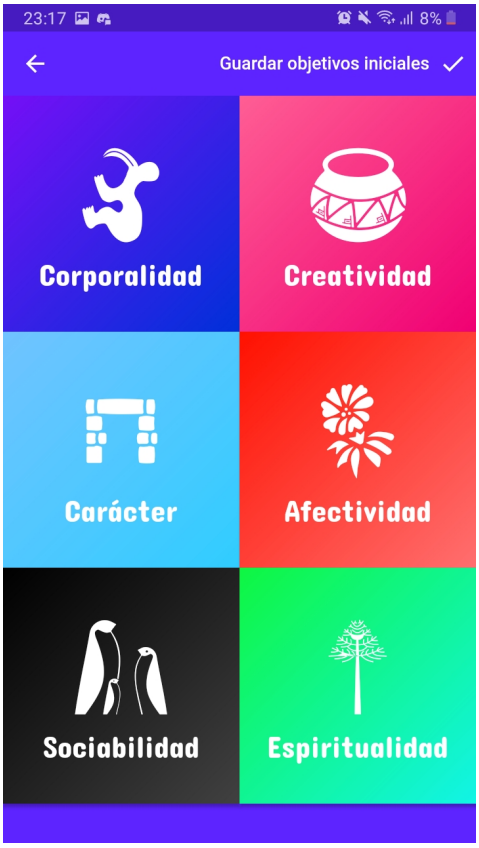


(b) Vista de estadísticas

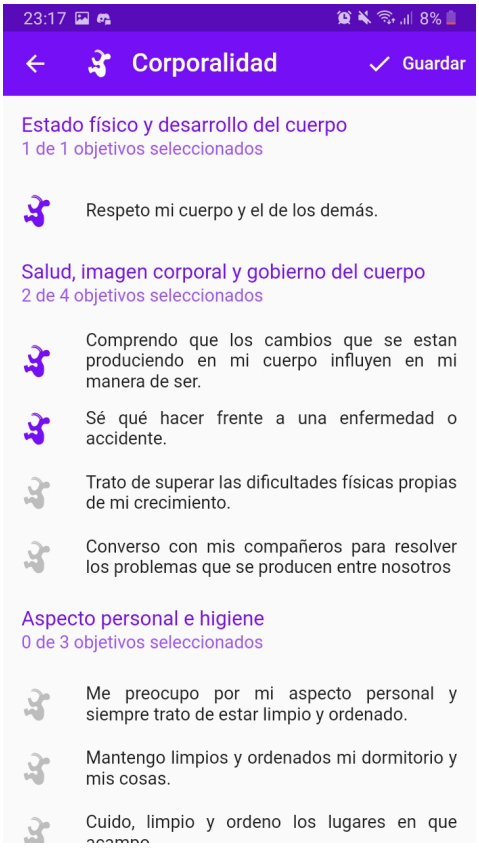


(c) Hint: Explicación del Factor de desarrollo integral

Figura 22: Bitácora y estadísticas

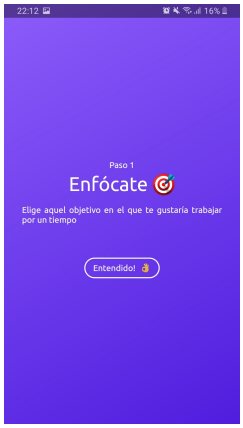


(a) Lista de áreas de desarrollo

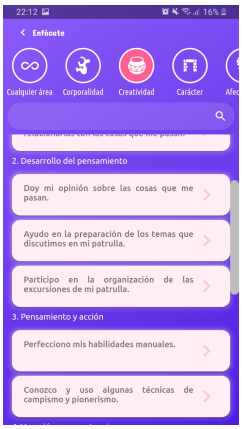


(b) Lista de objetivos de un área de desarrollo

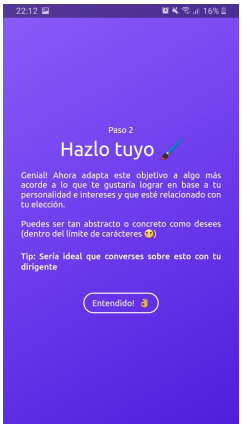
Figura 23: Vistas de personalización de objetivos



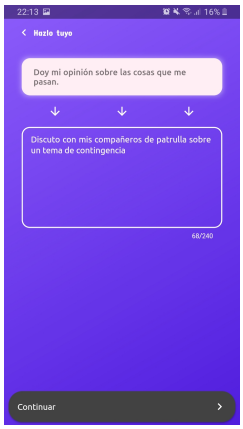
(a) Enfócate: Instrucciones



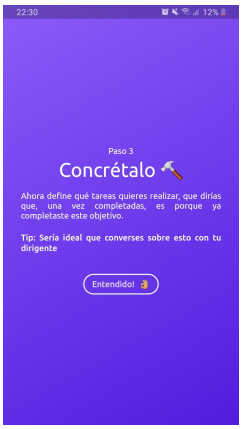
(b) Enfócate: Formulario



(c) Hazlo tuyo: Instrucciones



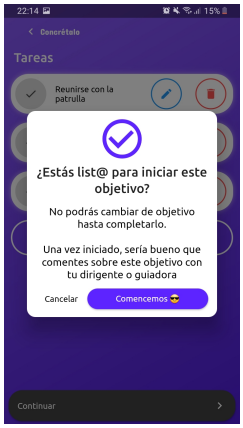
(d) Hazlo tuyo: Formulario



(e) Concrétalo: Instrucciones



(f) Concrétalo: Formulario



(g) Finalizar

Figura 24: Formulario para elegir objetivo activo

Registros

La vista de la figura 25 corresponde al registro de toda la actividad que ha realizado el beneficiario en la aplicación, para que pueda ver sus avances a un nivel más detallado.

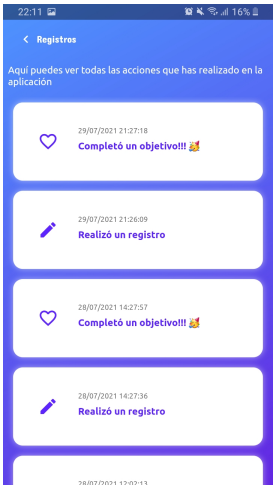


Figura 25: Lista de registros

Reclamo de recompensas

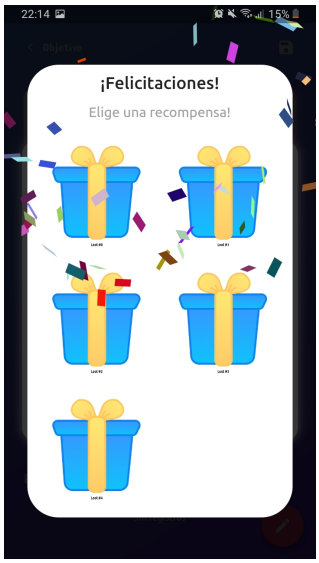
En la figura 26 se pueden observar las vistas que aparecen al realizar alguna acción que ofrezca una recompensa, tales como:

1. Realizar personalización de objetivos.
2. Realizar registro granular cada 24 horas.
3. Completar un objetivo.

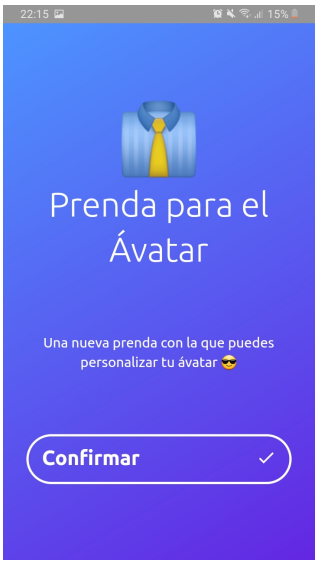
En la vista 26a los beneficiarios pueden elegir entre las cajas que vienen en el *token* de recompensas provisto por el servidor, cada una de estas posee un conjunto de recompensas diferente, luego, según las recompensas que haya ganado, se le muestran al usuario las recompensas que ganó con su progreso.

3.7. FRONT-END: CAMPAMENTO VIRTUAL

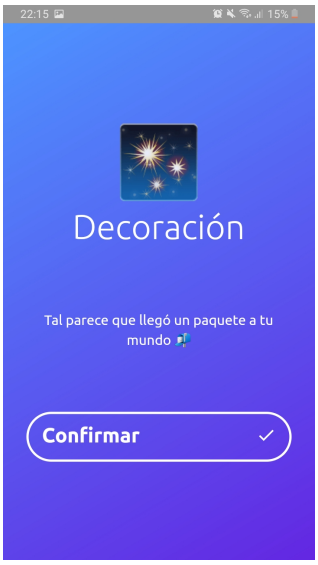
El campamento virtual corresponde al entorno 3D personalizable y perteneciente a cada beneficiario. A continuación se revisará el stack de tecnologías utilizado para esta parte, la arquitectura del código y las vistas de la interfaz.



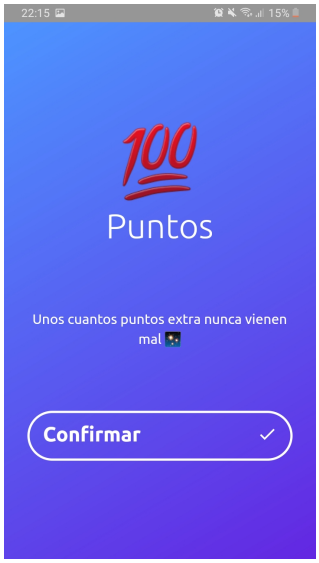
(a) Elección de Loot Boxes



(b) Vista de obtención de prenda de ávatar



(c) Vista de obtención de decoración



(d) Vista de obtención de puntos



(e) Vista de obtención de zona de campamento

Figura 26: Vistas de recompensas



Figura 27: Campamento virtual en acción

3.7.1. STACK DE TECNOLOGÍAS

La principal tecnología utilizada fue, como ya se ha mencionado en secciones anteriores, **Unity**, un motor de videojuegos 2D y 3D desarrollado por *Unity Technologies*. Existen varios motivos por el cual se decidió utilizar este motor, entre ellas:

- Es un motor en el que el desarrollador ya tiene experiencia.
- Permite compilar un programa en múltiples plataformas, entre ellas: Android y IOs.

Para este proyecto se utilizaron múltiples recursos de terceros para agilizar el desarrollo, estos se obtuvieron por tres medios:

1. ***Package manager***: Un gestor de paquetes, un paquete en Unity se le llama a un contenedor de varios tipos de características o assets[27], algunos paquetes utilizados en el proyecto se encuentran listados en la figura 28, siendo los más importantes los siguientes:
   - *Cinemachine*: Provee utilidades para trabajar con la cámara en un entorno 3D.
   - *Post Processing*: Entrega herramientas para el post-procesado de la imagen para realizar tareas como corrección de color.
   - *TextMeshPro*: Ofrece una forma de mostrar textos en alta calidad en la interfaz.

[27] https://docs.unity3d.com/Manual/Packages.html

- *Universal RP*: Corresponde a un *Render Pipeline* optimizado para móvil y tener acceso a características gráficas que permitan generar entornos 3D estéticamente atractivos rápidamente, tales como el *Shader Graph*, que permite generar shaders en una interfaz visual.

2. ***Asset Store***: Es una tienda provista por Unity para que la comunidad comparta los Assets que desee regalar o vender. Se utilizó principalmente para acceder a Assets 3D, de audio o utilidades programación.

3. **USM Games**: Una iniciativa estudiantil de la UTFSM comprometida con incentivar y apoyar la ideas de videojuegos, principalmente de estudiantes, en Chile. El autor de este documento es, al momento de la redacción de esta memoria, parte de esta iniciativa así que, por medio de la comunidad de *USM Games* se realizó el contacto con dos modeladores 3D, cuyos créditos se pueden ver en la figura 20c, quienes realizaron el diseño del personaje y algunos modelos 3D para este proyecto.

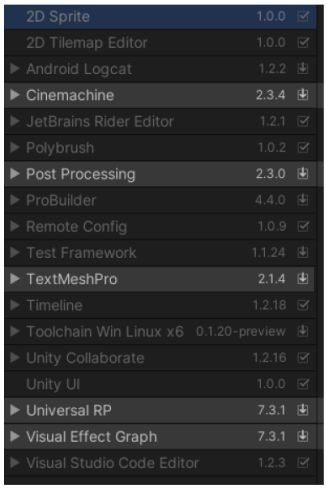


Figura 28: Paquetes instalados

3.7.2. ARQUITECTURA UTILIZADA

Primero, se explicarán algunos conceptos de Unity que es importante definir para comprender cómo se diseñó el código del campamento virtual:

- **Escena:** Es un entorno 3D o 2D conformado de varios *GameObject*.
- ***Game Object*:** Un objeto dentro de una Escena, compuesto de varios *Componentes*.
- **Componente:** Son aquellos elementos que, asignados a un *GameObject*, le entregan un comportamiento.

- ***Scriptable Object*:** Es una forma de guardar datos como un Asset re-utilizable. Puede servir para que varios componentes compartan un mismo grupo de datos, y que al modificar este dato se replique en todos los objetos al mismo tiempo.

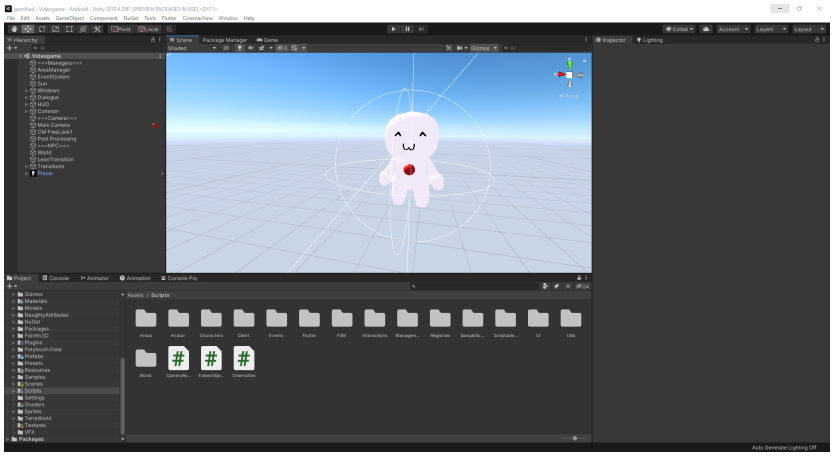


Figura 29: Pantalla de Unity

Estructura de las escenas

La organización de las escenas es importante para una comunicación consistente con la aplicación de Flutter que contiene a Unity. Por eso se dividió la aplicación en 3 escenas principales, como se ve en la figura 30a:

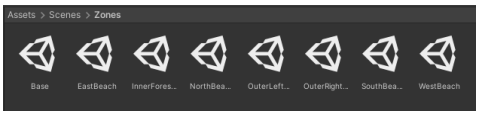
1. **Hub:** Es la escena inicial, corresponde a una escena casi vacía, sólo con los *GameObjects* necesarios para la comunicación con Flutter. Estando aquí, Flutter envía un mensaje a Unity solicitando cambiar a alguna de las otras dos escenas.
2. **Avatar:** Escena donde se realiza la personalización del Ávatar.
3. **Videogame:** Escena encargada de cargar las zonas del campamento virtual.

Además de estas escenas, existen las escenas de zona, como se ven en la figura 30b. Estas corresponden a las planillas que se usan de base para una zona del campamento virtual, cada una de esas escenas corresponden a una recompensa en la base de datos, y se componen de:

1. **Spawns:** Son los puntos en los cuales puede aparecer el jugador al abrir el campamento virtual o entrar a esta zona.
2. **Loaders:** Son *colliders* (espacios 3D que detectan colisiones), que detectan cuando el jugador pasa por ahí para cargar la zona que se encuentra en cierta salida.



(a) Escenas Principales



(b) Escenas de zonas

Figura 30: Escenas

Usualmente cada salida (Norte, Sur, Este, Oeste, etc.) de una zona posee un **Spawn** y un **Loader** para poder entrar y salir respectivamente. Y estos son marcados con un *Game Object* para posicionarlos cómodamente en el mundo.

Ciclo de vida de zona

El **ciclo de vida de zona** se define en este proyecto como el proceso por el cual pasa el campamento virtual para cargar una zona, con todos sus objetos, sólo cuando es necesario. Este proceso es descrito en el diagrama de la figura 31, y se compone de las siguientes etapas:

- Acciones de carga:
  - **Cargar datos del mundo**: Se solicita a Flutter la información del mundo, el inventario del jugador y su ávatar.
  - **Cargar escena de nueva zona:** Se carga la escena que el mundo tiene marcada como la zona actual a partir del nombre de la escena.
  - **Mover personaje a Spawn correspondiente:** Se utiliza la información del mundo para saber en qué Spawn de la escena hacer aparecer el personaje.
  - **Llenar objetos de nueva zona:** Se utiliza la información del mundo para saber qué objetos van en esta zona. Este es el último paso antes de que el jugador tenga el control sobre su personaje en el mundo.
  - **Jugador selecciona una nueva zona o cancela acción:** Esto ocurre cuando un jugador sale de una zona, y llega a un camino que todavía no tiene una zona asignada. En este caso se entrega la elección al jugador de qué zona añadir en este camino (figura 32b). Si cancela la acción, o no hay zonas disponibles en sus recompensas (figura 32a), entonces vuelve a la escena en la que se encontraba antes.
- Acciones del jugador:
  - **Jugador personaliza mundo:** El jugador utiliza los recursos que obtuvo mediante sus registros de progresión para personalizar la zona actual durante un tiempo.
  - **Jugador sale de zona por un Loader:** El jugador sale de la zona por un Loader, lo que activa las acciones de salida.
- Acciones de salida:



(a) Mensaje de camino bloqueado



(b) Ventana para abrir nuevo camino

Figura 32: Selección de nueva zona

- **Guardar zona actual:** Se buscan todos los objetos del mundo que tienen el componente `SaveableObject` o un hijo de éste. Esto corresponde al patrón de diseño Memento[28], lo que implica que el responsable de recolectar los datos de un objeto en el mundo, en formato JSON, son los mismos objetos, no un único gran recolector de información.
- **Limpiar y descargar escena de la zona actual:** La escena de la zona actual se descarga de la memoria y se limpian los objetos restantes para dar paso al cargado de la siguiente escena en las acciones de carga.

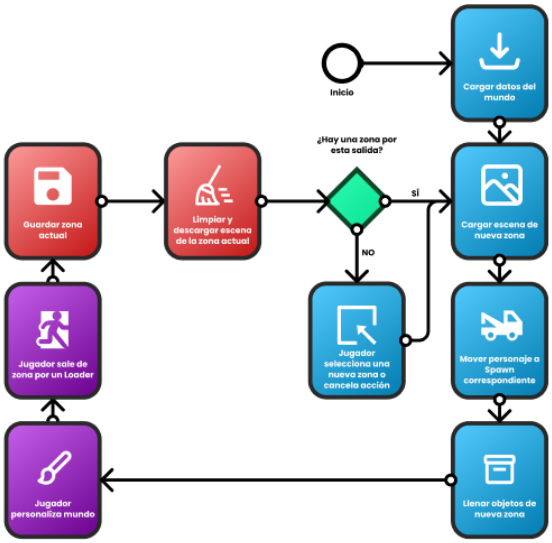
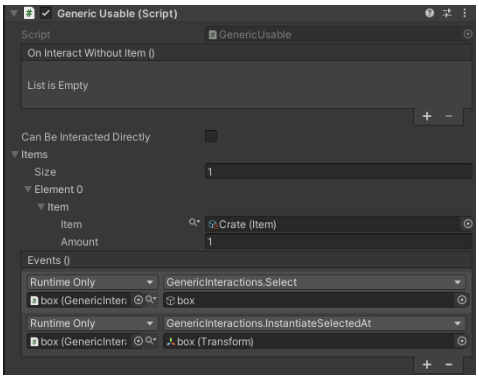
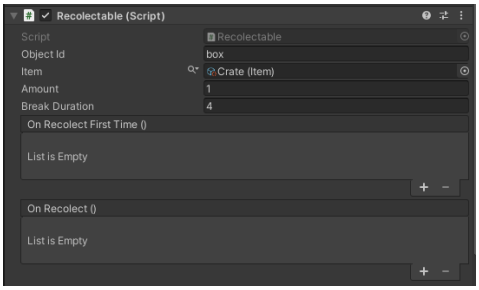


Figura 31: Ciclo de vida de zona

[28]https://refactoring.guru/design-patterns/memento



(a) Componente de Interacciones



(b) Componente de Recolectable

Figura 33: Componentes de interacción

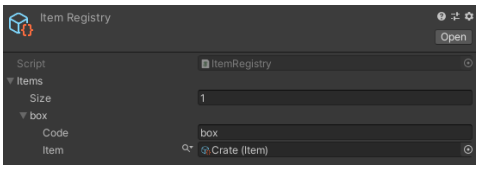
Interacciones del jugador

Para permitir las interacciones del jugador con el mundo se utilizaron principalmente 2 componentes, como se pueden ver en la figura 33. Estos corresponden a los siguientes:

- `GenericUsable`: Permite configurar qué ocurre cuando el jugador interactúa con un objeto, ya sea presionando el botón de interactuar, o arrastrando algún objeto compatible encima por medio de *Unity Events*, que permiten llamar a métodos de otros componentes de la escena definidos mediante la interfaz de Unity. En la figura se observa cómo al arrastrar una caja (definido por al atributo `Item`) desde el inventario a encima del objeto, se crea una instancia de una caja en un punto dado, gastando 1 caja (definida en `Amount`) en el proceso.
- `Recolectable`: Son los objetos que pueden ser recolectados del mundo, los atributos más importantes aquí son:
  - *Item*: Define qué ítem entrega el recolectar este objeto.
  - *Amount*: Define cuántas unidades del ítem elegido entrega el recolectar a este objeto.
  - *Break Duration*: Define cuánto tiempo tarda el jugador en recolectar este objeto (cuánto tiempo tiene que mantener presionado el botón de recolectar).



(a) Registro de prendas de Ávatar



(b) Registro de Ítems

Figura 35: Registros



Figura 34: Recolectable en el mundo

Registros

Los registros son una clase de este proyecto que hereda de Scriptable Objects, y se encarga de definir los códigos (*strings*) que representan diferentes elementos del mundo 3D. El motivo para usar estos, es porque la comunicación entre Flutter y Unity es a través de JSON, por lo que la única forma de identificar cada recompensa, es por medio de códigos, como se puede ver en la figura 35, donde se utilizan códigos para representar prendas de ávatar e ítems respectivamente.

En el caso del **registro de prendas de Ávatar** se define un código para distintos **materiales**, que indican los colores, texturas y otros aspectos gráficos de las prendas del personaje. Mientras tanto, el **registro de ítems** define un *ScriptableObject* de tipo `Item` por código, que corresponde a una descripción de un objeto que puede tener el jugador en su inventario. También existe un **registro de objetos** que define un código por *Game Object*, para permitir cargar objetos 3D en la escena a partir de un *string*, que sería el código.

Recolección de ítems de recompensas

El jugador puede recolectar los ítems que ha ganado por medio del sistema de recompensas

interactuando con un buzón como se ve en la figura 36.

Primero, al aparecer el objeto en el mundo, el buzón solicita a Flutter una lista de *Ítems* que quedan por reclamar. Durante esta solicitud Flutter revisa las recompensas que ha ganado el beneficiario y las resta a las que ya han sido reclamadas, y en base a esto entrega estos ítems al beneficiario.

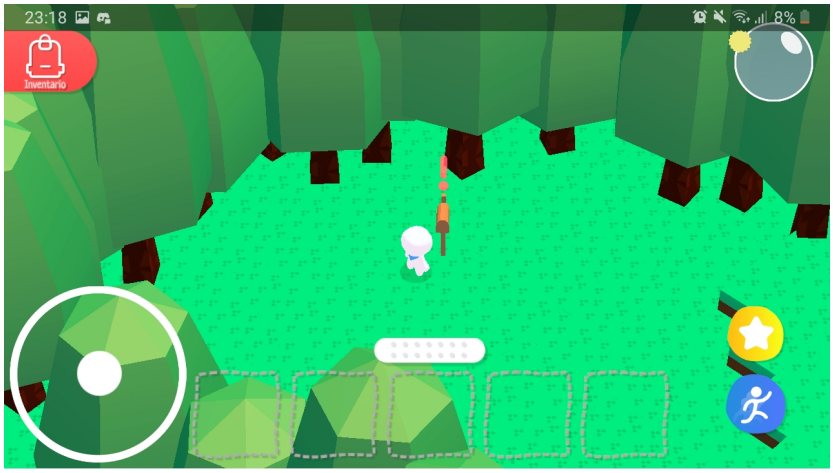


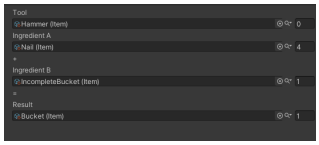
Figura 36: Buzón de correo

Otras implementaciones

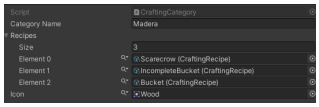
En la extensión disponible para escribir este documento es imposible abarcar todos los aspectos del mundo virtual por lo que, a continuación, se describirán, a grandes rasgos, cómo se realizó la implementación de otros aspectos del proyecto:

- **UI y HUD:** La interfaz se implementó utilizando assets de Lean GUI[29] para facilitar las interacciones *Touch* en móvil, como arrastrar ítems en el inventario y rotar la cámara con gestos táctiles en la pantalla.
- **Gestión del inventario:** El **inventario** es gestionado por un componente asignado al jugador. Éste posee métodos para agregar o quitar **Ítems** de su inventario y del *hotbar*, que corresponde a la barra de acceso rápido para utilizar ítems sin necesidad de abrir el inventario completo. El **Ítem** se implementó como un *Scriptable Object* que describe aspectos del ítem como su *Sprite* (su imagen) y su nombre en el inventario.

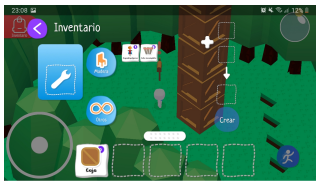
[29] http://carloswilkes.com/Documentation/LeanGUI



(a) Receta de *Crafting*



(b) Categoría de *Crafting*



(c) Menú de *Crafting*

Figura 38: Componentes de Crafting



Figura 37: Inventario

- **Movimiento del jugador:** Se utilizó una **máquina de estados finita (FSM)** que, a partir del *input* del jugador entregado por su interacción con la interfaz, cambiaba de estados entre: `Grounded` (en el piso) y `OnAir` (en el aire, por ejemplo al saltar).
- **Crafting:** El *crafting* se implementó por medio de **recetas** como la de la figura 38a, que son *Scriptable Objects* que almacenan dos Ítem (**ingredientes**) que, juntos, forman otro Ítem (**resultado**). Se puede, además, definir cuánto de cada ingrediente es necesario, y cuánto del resultado genera. Otro ítem que se puede indicar en el editor es una **herramienta**, que define que es necesario un tercer ítem para generar el resultado. Para ordenar mejor las recetas de *Crafting*, se utilizaron **Categorías**, que corresponden a *Scriptable Objects* con una lista de recetas y un ícono representativo de la categoría, como se ve en la figura 38b.
- **Movimiento de la cámara:** Para el control de la cámara se utiliza una combinación del paquete *Cinemachine*, que entrega herramientas para el control de la cámara y el seguimiento del ávatar en el mundo, y el asset *Lean GUI*, que se encarga, por medio de un *script* intermedio, de controlar la rotación de la cámara al rededor del jugador.
- **Pistas:** Las Pistas son objetos que, al ser utilizados por el jugador, muestran un mensaje. Por ejemplo, en la figura 39 se muestran carteles con comentarios que entregan una

pista en la interfaz para realizar acciones en el mundo, como el de la parte superior de la captura.



Figura 39: Pista para crear una tienda de campaña de supervivencia

- **Sistema de construcción:** Para posicionar objetos en el mundo, el jugador puede arrastrar ítems de su inventario y arrastrarlo al escenario, si selecciona un espacio válido, se cambiará la interfaz a la de construcción (figura 40), donde el jugador puede rotar el objeto y cancelar o confirmar su posición.

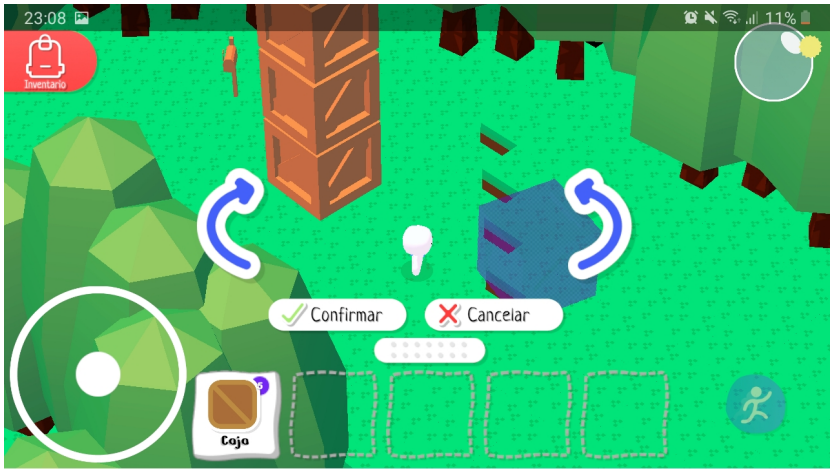


Figura 40: Sistema de construcción

3.8. FRONT-END: PARA DIRIGENTES

La aplicación web para guiadoras y dirigentes corresponde a la parte del proyecto donde ellos pueden realizar el seguimiento de la progresión personal de los beneficiarios a su cargo y del grupo en general.

3.8.1. STACK DE TECNOLOGÍAS

El *framework* principal que se utilizó para el desarrollo de esta aplicación móvil, fue **Angular 11.2.8**[30], una herramienta para el desarrollo de **Single-Page Applications** (*SPA*). Se eligió por la experiencia del desarrollador en esta tecnología, y al existir herramientas que permiten integrar la aplicación con AWS rápidamente.

La integración con AWS se realizó entonces con los paquetes de NPM `aws-amplify`[31] y `@aws-amplify/ui-angular`[32], que proveen herramientas para interactuar con los recursos de AWS rápidamente desde JavaScript y TypeScript.

Esta aplicación fue desarrollada teniendo en cuenta que se espera que sea utilizado tanto en computadoras como en móvil, por lo que se utilizó **Bootstrap 4.5.0**[33], un *framework* creado con la motivación de ayudar a construir interfaces responsivas.

Además, para acelerar el desarrollo del proyecto se utilizó Angular Material[34], que provee varios componentes de Angular ya implementados, utilizando *Material Design*[35].

3.8.2. ARQUITECTURA UTILIZADA

A continuación se enumeran y explican los elementos que componen la aplicación web en Angular.

PROGRAMACIÓN ASÍNCRONA

Con el fin de realizar las comunicaciones con el servidor sin congelar la aplicación, el desarrollo *front-end* suele proveer herramientas para realizar tareas asíncronas fácilmente. En el caso de Dart (el lenguaje de programación de Flutter), existen los `Future`s, que corresponden a tareas que tardan un tiempo en retornar una respuesta. Por otro lado, en *JavaScript*, se encuentran las `Promise`s, que tiene el mismo propósito.

[30] https://angular.io/
[31] https://www.npmjs.com/package/aws-amplify
[32] https://www.npmjs.com/package/@aws-amplify/ui-angular
[33] https://getbootstrap.com/docs/4.5/getting-started/introduction/
[34] https://material.angular.io/
[35] https://material.io/design

Sin embargo, también existen un conjunto de librerías llamadas *ReactiveX*[36] que proveen herramientas para la programación asíncrona y basada en eventos utilizando secuencias observables. Para este proyecto, la librería utilizada fue *RxJS*.

Esta librería fue de gran ayuda para generar observables que actualizan los datos que se muestran en pantalla a partir de los cambios en la URL actual. Por ejemplo, es posible utilizar *ReactiveX* para observar cuándo cambian los parámetros de *query* de la ruta, o al cambiar el ID del beneficiario o grupo cuya vista está cargada actualmente, y solicitar al servidor obtener los datos necesarios para la vista actual. Pues, hay que recordar que esta aplicación es un SPA, lo que causa que no se vuelva a cargar la página, necesariamente al moverse entre rutas.

El uso de observables queda graficado en el diagrama de servicios de la figura 41, donde cada observable corresponde a un atributo con un signo `$` al final de su nombre.

SERVICIOS

Los servicios[37] que se implementaron a lo largo de toda la aplicación, como unidades de código modulares y re-utilizables, se encuentran descritos por el diagrama 41. A continuación se revisarán los más importantes:

- **Servicio de autentificación (`AuthenticationService`):** Se encarga de controlar el estado de la autentificación utilizando el módulo `Auth` de AWS Amplify. Esto permite obtener los datos del usuario usando Amazon Cognito como el proveedor.
- **Servicio de API (`ApiService`):** Provee utilidades para realizar la comunicación con la REST API utilizando el módulo `API` de AWS Amplify.
- **Servicios de recursos (`GroupsService`, `DistrictsService`, `BeneficiariesService`, `RewardsService`):** Utilizan `ApiService` para obtener los datos desde el servidor, o actualizarlos si es necesario.
- **Servicio de alerta (`AlertService`):** Se encarga de mostrar alertas en la vista, por ejemplo un *Snackbar*[38] de Angular Material, o un Sweet Alert[39] para pedir confirmaciones al usuario, por ejemplo, al cerrar sesión.
- **Servicio de objetivos (`ObjectivesService`):** Se encarga de obtener una lista de objetivos a partir de un archivo JSON.

[36] http://reactivex.io/intro.html
[37] https://angular.io/guide/architecture-services
[38] https://material.angular.io/components/snack-bar/overview
[39] https://sweetalert2.github.io/

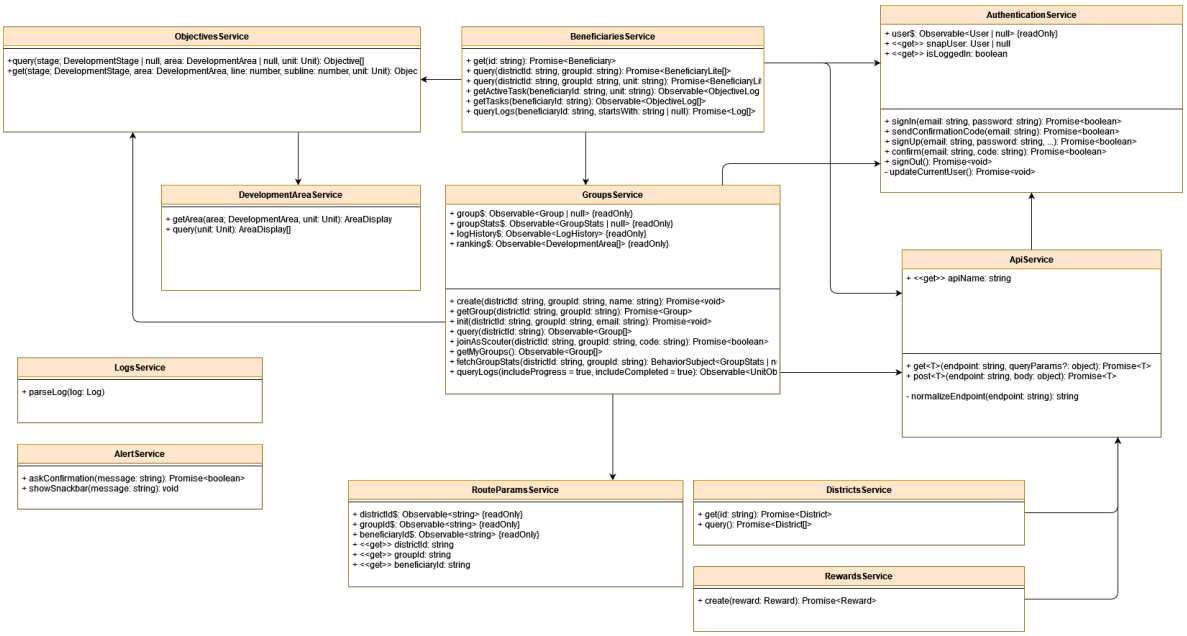


Figura 41: Servicios de la Aplicación Web

3.8.3. VISTAS WEB

A continuación se revisarán las vistas que componen la aplicación web.

Autentificación

El proceso de autentificación se compone de tres pasos:

1. **Crear cuenta (figura 43):** En esta vista los dirigentes pueden crear su cuenta, aquí pueden ingresar sus datos básicos, credenciales y un apodo con el cual podrán ser identificados más fácilmente.

2. **Confirmar cuenta (figura 44):** Este es un paso requerido por AWS donde, al crear su cuenta, tienen que ingresar el código que les llegue a su correo electrónico para confirmar su cuenta.

3. **Iniciar sesión (figura 42):** Corresponde también a la vista inicial de la aplicación, desde aquí se puede ingresar rellenando el correo electrónico y contraseña con el cual la guiadora y dirigente se haya registrado.

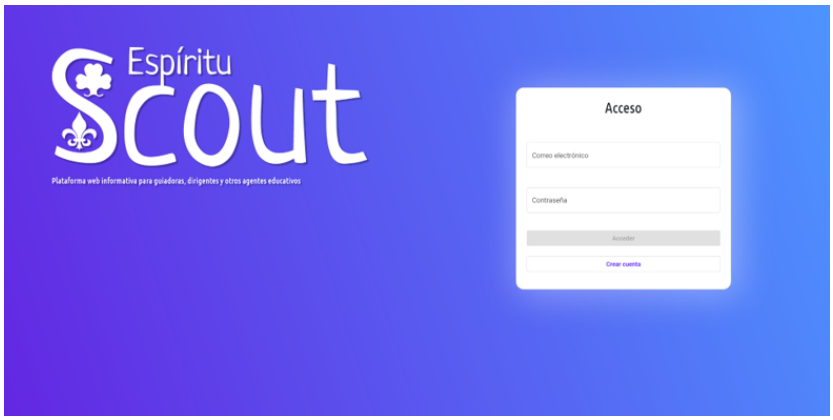


Figura 42: Inicio de sesión para guiadoras y dirigentes

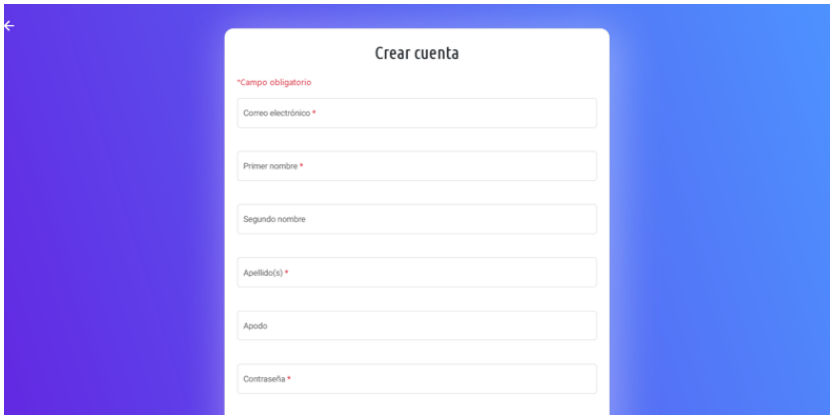


Figura 43: Creación de cuenta

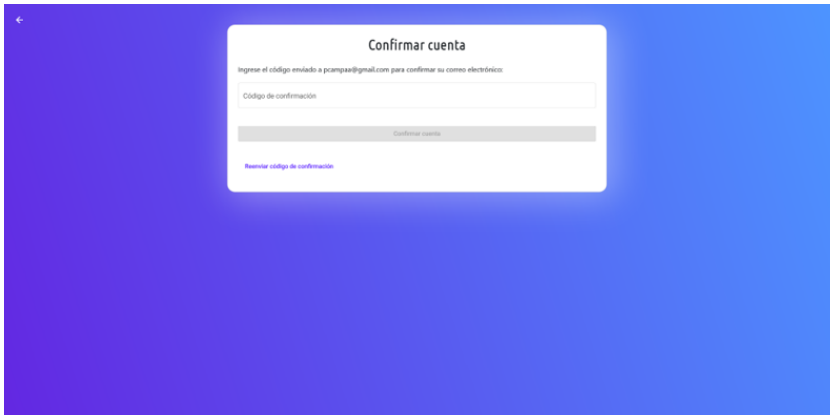


Figura 44: Confirmar cuenta

Nuevo usuario

Al ingresar una guiadora o dirigente a la aplicación sin pertenecer a un grupo, el usuario verá las instrucciones de la figura 45, indicando que hay dos formas de unirse a un grupo: creando uno desde el formulario indicado (figura 46) o unirse a un grupo ya existente mediante el enlace de invitación que le haya compartido otro usuario.

El motivo por el cual los grupos no pueden registrarse por su cuenta todavía, es para mantener los registros moderados en la fase de pruebas, sin embargo se espera a futuro abrir este registro para que se pueda realizar desde la misma aplicación. Por el momento, un administrador de la aplicación debe agregar al primer dirigente a un grupo nuevo para empezar a utilizar la aplicación.



Figura 45: Nuevo usuario: Instrucciones

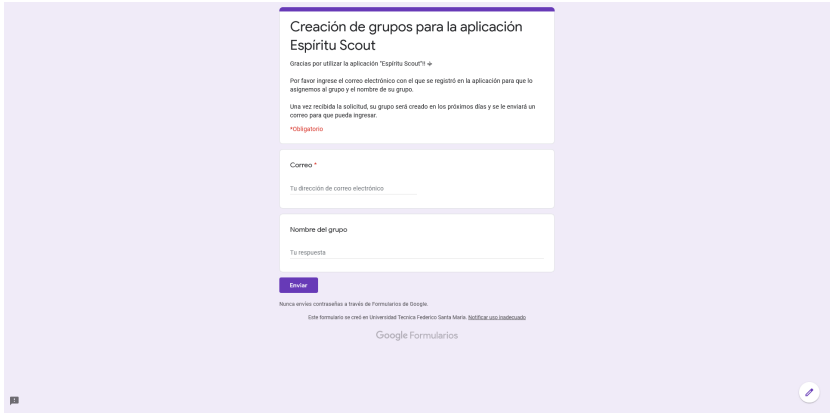


Figura 46: Nuevo usuario: Formulario para inscribir grupo

Unirse a un grupo

Para unirse a un grupo, cualquier usuario de la aplicación puede ingresar por medio de un enlace de algún usuario que ya se encuentre dentro de un grupo. Al recibir este enlace, que almacena la contraseña de dirigente/guiadora como parámetro

de *query*, por ejemplo: `https://example.com/districts/pankan/groups/grupo-de-prueba/invite?code=91252efd7d4c2c7762507221c6585d07cce49f42`.

Al ingresar al enlace, el usuario verá la vista de la figura 47, donde al presionar el botón "Aceptar invitación", realizará la solicitud a la REST API para unirse al grupo utilizando el código y, si el código es válido, el grupo añadirá a este usuario como un nuevo dirigente o guiadora.



Figura 47: Invitación para guiadoras y dirigentes

Dashboard: Grupo

La primera vista a la que acceden los usuarios al entrar a un grupo son las de la figura 48, donde se verá en modo retrato o *landscape* dependiendo de la orientación que tenga el dispositivo.

Luego, una vez los beneficiarios hayan ingresado registros de progresión personal, esta vista se actualizará para verse similar a como en la figura 49. Los registros que se pueden encontrar en esa vista son:

1. **Número de beneficiarios** registrados en el grupo.
2. **Objetivos completados** en total por todos los beneficiarios del grupo.
3. **Uso de la aplicación** en los últimos 4 meses usando como medidas acumulativas el número de objetivos completados, el número de avances registrados y la suma de estos dos.
4. ***Ranking* de las áreas de desarrollo** con más avances registrados.
5. **Gráfico de radar** con las áreas de desarrollo con más avance por los beneficiarios del grupo.



(a) Grupo: Resumen en modo retrato



(b) Grupo: Resumen en modo *landscape*

Figura 48: Resumen de grupo

Estas estadísticas buscan ayudar a los dirigentes a tener una visión general de cómo está el grupo en las diferentes áreas de desarrollo, tanto históricamente, como en el último tiempo.

En la parte superior del cuadro blanco, se pueden ver las pestañas que permiten navegar por las tres secciones de las vistas de grupo: resumen, beneficiarios y guiadoras y dirigentes.



Figura 49: Grupo: Resumen de grupo

Dashboard: Beneficiarios

Desde la vista de beneficiarios, guiadoras y dirigentes pueden ver la lista de beneficiarios registrados en la plataforma. Desde aquí el usuario puede seleccionar un beneficiario para ver un resumen de sus detalles, como se ve en la figura 50, o compartir el código de invitación de beneficiarios como en la figura 51, donde pueden copiar el código o compartir por WhatsApp.

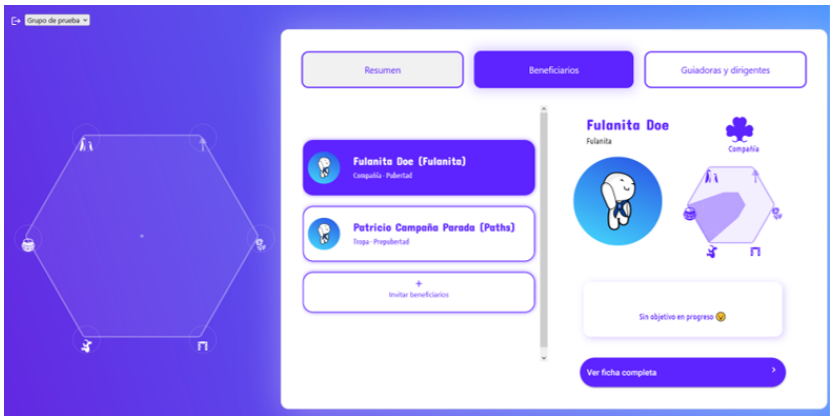


Figura 50: Beneficiarios: Lista

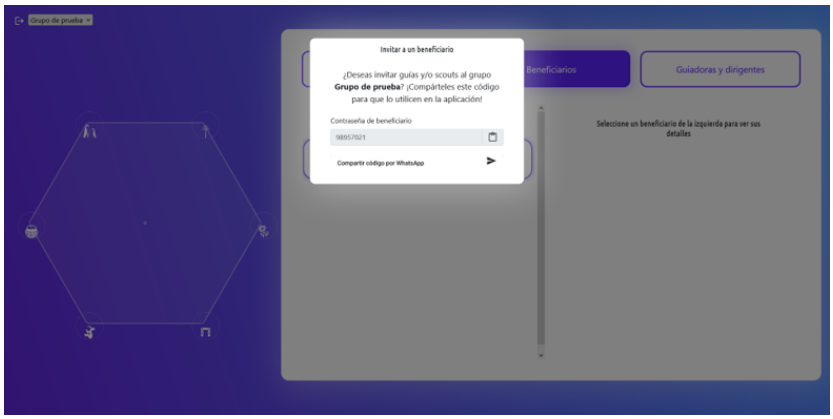


Figura 51: Beneficiarios: Invitación

Detalle de beneficiario

Desde la vista de la figura 50, al presionar el botón de “Ver ficha completa”, el usuario puede acceder a un mayor detalle de los avances de este usuario, teniendo acceso a:

- **Resumen:** A la izquierda se puede observar el perfil público de un usuario, indicando:
  - Etapa (pre-pubertad o pubertad).

  - Unidad (tropa o compañía), que se indica mediante un ícono (flor de lis o trebol respectivamente).
  - Gráfico de radar personal indicando cómo va en su avance por área de desarrollo.
  - Objetivo en progreso.

- **Objetivos cumplidos (figura 52):** La lista de objetivos cumplidos por el beneficiario, indicando su objetivo original (u objetivo propuesto) y el objetivo personal que se propuso.
- **Registros (figura 53):** Los registros de uso del beneficiario de la aplicación, para que se pueda realizar un seguimiento más detallado.
- **Registros de objetivo (figura 54):** Al presionar en la lista de objetivos cumplidos, el botón de “Ver registros” de uno de los objetivos, se puede acceder la lista de registros, filtradas por los objetivos relacionados a un objetivo en particular.



Figura 52: Detalle de beneficiario: Objetivos cumplidos

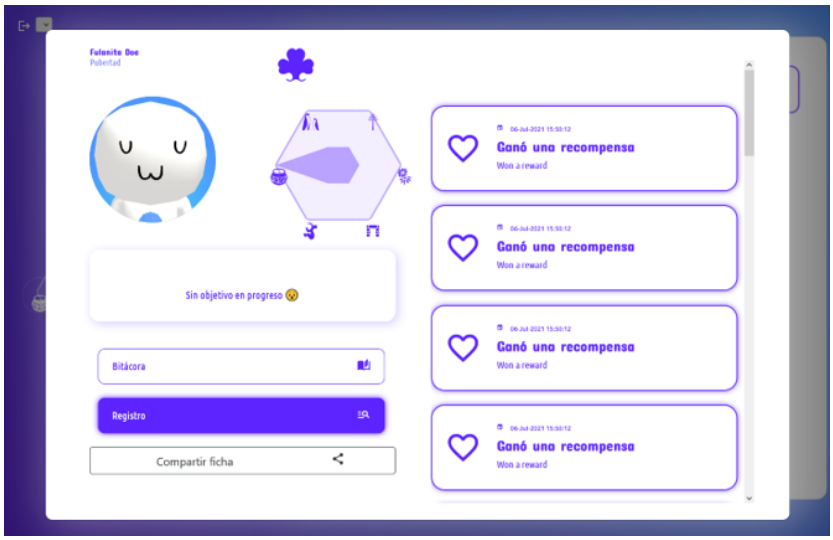


Figura 53: Detalle de beneficiario: Registros



Figura 54: Detalle de beneficiario: Registros de objetivo

Además, al presionar el botón “Compartir ficha”, el usuario podrá compartir un enlace para acceder a esta ficha sin necesidad de haber iniciado sesión. El motivo de esta opción es para que otros agentes educativos, como los padres de un beneficiario, puedan también participar del seguimiento de una progresión personal. Un ejemplo de este perfil público se puede observar en la figura 55.

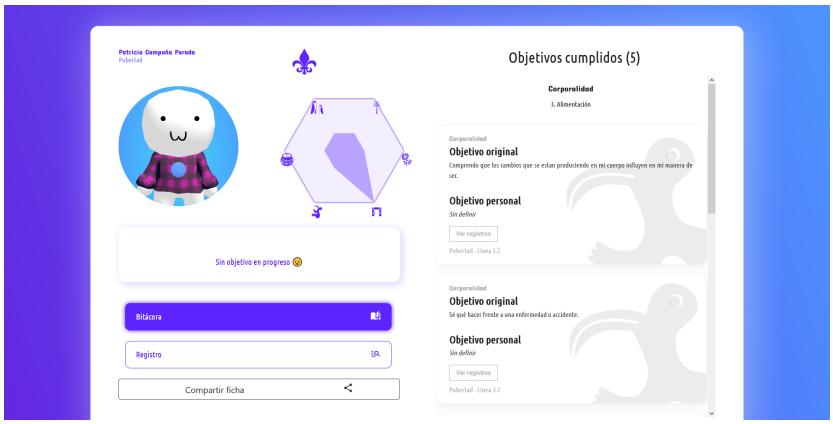


Figura 55: Detalle de beneficiario: Perfil público

Dashboard: Guiadoras y dirigentes

En la última pestaña de la vista de grupo, el usuario puede ver la lista de guiadoras y dirigentes registrados en el grupo, además del enlace para unirse a éste, tal como se ve en la figura 56.

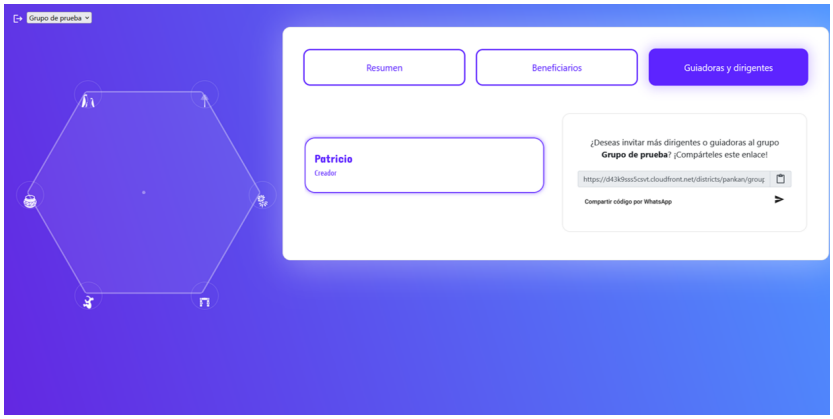


Figura 56: Guiadoras y dirigentes: Lista

Panel de administrador

El panel de administrador corresponde a una vista a la cual sólo pueden acceder usuarios que pertenezcan al grupo `admin` en Amazon Cognito. Desde aquí es posible ver la lista y crear nuevas recompensas que estarán disponibles en la aplicación (figura 57), además de gestionar los grupos de la plataforma, permitiendo crear nuevos grupos en un distrito, ver la lista de grupos y asignar un usuario como guiadora y dirigente creador de un grupo (figura 58).

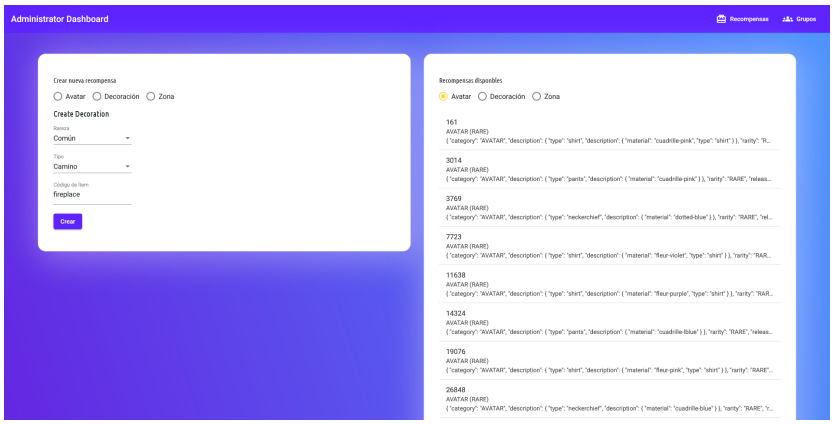


Figura 57: Panel de administrador: Recompensas

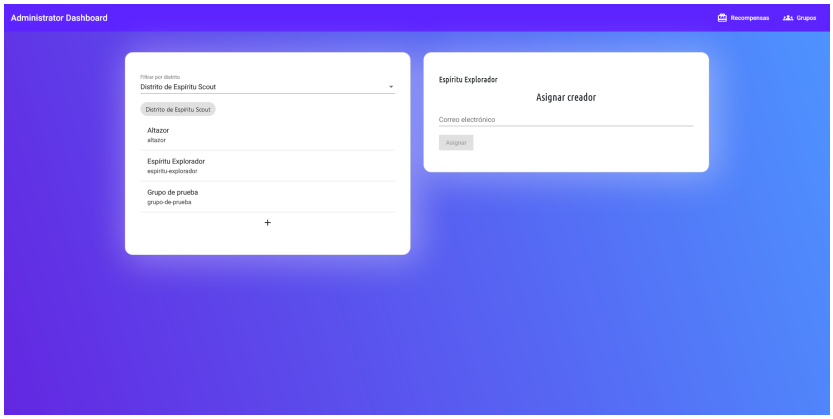


Figura 58: Panel de administrador: Grupos

CAPÍTULO 4
VALIDACIÓN

En esta sección se busca validar la solución propuesta para resolver el problema planteado. Tanto la aplicación web, como la aplicación móvil pertenecen a un mismo proyecto de una misma temática, pero que van dirigidos a usuarios distintos y, además, a objetivos específicos diferentes. Por eso, se realizaron dos instancias de validación con grupos de potenciales usuarios específicos para cada aplicación. En las siguientes secciones se explicará, por cada instancia, el plan de validación, cómo se ejecutó y sus resultados. Las conclusiones a partir de estos resultados serán comentadas en el siguiente capítulo.

4.1. VALIDACIÓN DE LA APLICACIÓN WEB

4.1.1. PLAN DE VALIDACIÓN

El objetivo específico de la aplicación web es el siguiente:

> Desarrollar una página web responsiva donde los dirigentes puedan aumentar su capacidad de supervisión, y compartir con quienes consideren adecuado, las estadísticas del progreso de los guías y scouts a su cargo.

Se decidió que la mejor forma de validar esta parte del proyecto, sería realizando una **capacitación** a los dirigentes interesados en el proyecto, para que puedan probar cada parte de la aplicación ellos mismos y comentar al respecto desde su experiencia.

Para ello se realizó una planificación de cómo mostrar las partes de la aplicación, siguiendo los pasos desde crear una cuenta en la plataforma, hasta revisar las estadísticas de grupo. En orden, la instancia se dividió en 6 partes:

1. Creación de cuenta e inicio de sesión
2. Crear y unirse a un grupo
3. Invitar a otros dirigentes y guiadoras al grupo
4. Invitar beneficiarios
5. Revisar estadísticas de beneficiarios
6. Revisar estadísticas de grupo

Durante cada parte se permite a los dirigentes realizar los comentarios que consideren pertinentes.

4.1.2. EJECUCIÓN

La validación de la aplicación web se realizó el día 2 de Julio de 2021 a las 22:30 en conjunto a los dirigentes interesados en el proyecto: **José Mella Silva** y **Daniel Minte Valderrama** del distrito de Puerto Montt.

La instancia se realizó por medio de Google Meet, como se puede observar en la figura 59, y tuvo una duración aproximada de 1 hora 40 minutos. Ésta se grabó con el consentimiento de los participantes, con el fin de servir de referencia para escribir este documento.

La capacitación se realizó en base a una presentación donde se detalla cada sección y un ejemplo de las vistas principales y cómo éstas aparecen en web y móvil.

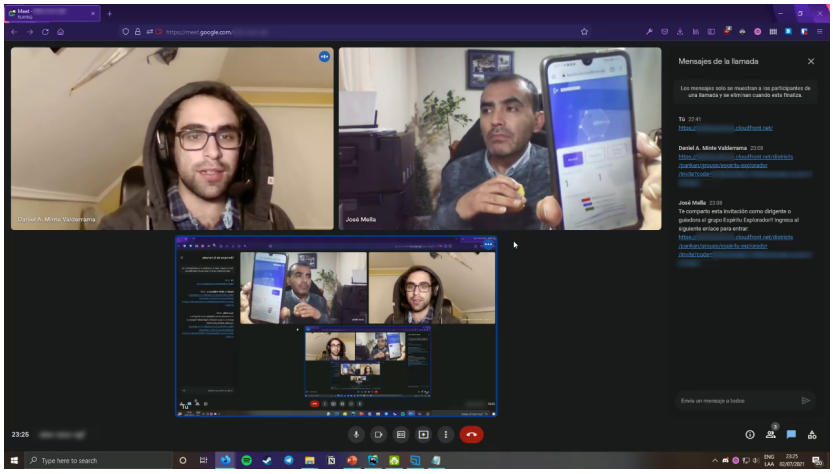


Figura 59: Captura de la validación de la aplicación web con los dirigentes

4.1.3. RESULTADOS

En esta sección se presentan los comentarios más relevantes, realizados por los dirigentes, durante la instancia de validación.

Problemas con la contraseña

Al comenzar a usar la aplicación ocurrió un problema a partir de una situación que no había sido probada a nivel de experiencia de usuario lo suficiente por el desarrollador, que es, al crear una cuenta con una contraseña que no cumple los parámetros mínimos de seguridad (mínimo de carácteres, que no tenga mayúsculas, etc.), aparece un error indicando

que vuelva a intentar más tarde, por lo que recomendaron añadir unas indicaciones de las características mínimas de la contraseña.

Sobre el traspaso de información de beneficiarios

El siguiente comentario fue expresando cuando, durante la presentación, se estaba realizando el movimiento de un beneficiario del grupo de pruebas internas, al grupo "Espíritu Explorador" que fue creado durante la presentación para que los dirigentes hicieran sus propias pruebas. Durante esta acción, el desarrollador, comentó sobre cómo los datos de un beneficiario estaban asociados al beneficiario, no a un grupo, por lo que al mover este beneficiario de un grupo a otro - acción que se realiza, por el momento, cambiando un atributo del modelo del beneficiario directamente en la base de datos - no se requieren realizar más acciones para que sus estadísticas asociadas, sean integradas dentro del nuevo grupo. Esto llamó la atención, positivamente, de los dirigentes, y Daniel Minte comentó lo siguiente:

> **Daniel:** De hecho, esa es una de las cosas que más complica, que no falta el chico que se cambia de grupo o, incluso, y esto es errores, muchas veces, dentro del grupo, pero hay muchas veces que **el traspaso de información entre un grupo a otro grupo, cuando un chico se cambia, es poca**.

Sobre la interpretación del lenguaje de dominio a *software*

Mientras se les explicó a los dirigentes sobre cómo invitar beneficiarios a un grupo, los dirigentes comentaron, a nivel global, sobre cómo se utilizó el lenguaje del movimiento scout dentro del *software*, y se llegó a mencionar, incluso al software Jhathi que se describió en la sección del estado del arte en esta memoria. Los principales comentarios fueron sobre cómo se interpretó el lenguaje de dominio dentro del proyecto completo, refiriéndose tanto a la aplicación móvil y web, y cómo esto les facilita el trabajo como dirigentes, lo que responde al objetivo específico relacionado a esta parte del proyecto, de aumentar su capacidad de supervisión:

> **José:** Sabes que una de las cosas que más valoro de tu trabajo, aparte obviamente que el programa funcione, es que ha interpretado tan bien el lenguaje nuestro. Por ejemplo, lo que acabas de decir es, como, genial, es como, para nosotros es tan natural, pero quizás para cualquier otro no lo es. Y de hecho, lo que más ha costado en las plataformas digitales que usa la asociación -no tenemos para progresión-... alguna vez tuvimos una, que se llamaba Jhathi.
>
> **Daniel:** Oh sí, yo lo busqué, busqué imágenes y, efectivamente, era como muy de adulto, como que era una plataforma que, la verdad, la comparaba con la que se usa en el hospital[40], por ejemplo, era más o menos igual.

[40] Acotación: Daniel Minte es estudiante de enfermería, por eso realizó esta analogía

José: Era como llenar formularios *nomás*. Entonces eso nos facilita el flujo que uno tiene con los *chiquillos* porque uno tiene que dedicarle tiempo a observarlos, en cambio **cuando tú veas esta gráfica, y se arma el gráfico ahí, y se entienda va a ser súper fácil para uno: "Oh, este tiene una tendencia a tal cosa"**

Daniel: Sí

José: Entonces ahí uno puede, efectivamente, luego en la unidad, sin tener que analizar tanto, empezar a preguntarse **cómo los invito a que desarrollen la otra para que no sea tan deforme su crecimiento, para que sea más integral**. Entonces, primero es súper fácil, pero a lo que iba a decir que, usaba la palabra Jhathi, exclusivamente, porque en el fondo, Jhathi es un elefante, en "El libro de las tierras vírgenes", entonces claro, un elefante tiene mucha memoria, por lo tanto, guarda mucha información, y eso, quizás, lo hacía bien, pero eso nomás. En cambio aquí hay todo como una... Hasta yo lo encuentro entretenido, y a mi no me gustan los juegos, de hecho no juego. No tengo ningún videojuego, no tengo un videojuego con el que juego en mi computador, pero esto me entusiasma.

Sobre el uso del marco simbólico

También se refirieron, poco después, al uso del marco simbólico del movimiento scout dentro del proyecto:

José: Es que, me llama la atención que hayas contado **no sólamente el lenguaje, las palabras, sino como el espíritu que uno mira este mundo**, y los *cabros* en el fondo tienen un marco simbólico que lo van aprendiendo de a poco, pero uno que ya ha pasado por el movimiento scout es como que todo le calza. Y en el lenguaje que tú nos hablas, a través de este programa, como que todo fluye para mi. Es como: se nota que estudió.

Daniel: Aparte también, siempre se habla de que, **el marco simbólico se vive, no se habla**, entonces nos faltaba, por ejemplo, que uno decía, por ejemplo: "No, la idea de ser más creativos", de esto, de mostrarle la típica pelea de que el calendario sea, que no sea simplemente una carta Gantt poco menos, sino de maneras más creativas, pero uno de repente ve lo suyo propio, y la verdad es la misma carta Gantt que te presentaron los chicos. Entonces, claro, estas cosas ayudan bastante a que efectivamente se mantenga esa idea del marco simbólico.

Sobre el gráfico de radar de los beneficiarios

Cuándo se revisó la vista de estadísticas de los beneficiarios, más específicamente, sobre el gráfico de radar utilizado para representar la progresión personal de un beneficiario en las diferentes áreas de desarrollo:

José: Patricio, mira, de verdad te lo digo, voy a cumplir 44 años de escultismo ininterrumpido el próximo año, entré a los 6 años, el próximo año cumplo 50, y cumplo 44 años en el escultismo. Nunca he tenido receso, y créeme, que éste gráfico es la **mejor representación rápida** que uno necesita tener para poder colaborar de mejor forma con los chiquillos. Porque uno tiene 3 horas, ya, ponle 4 horas, una vez a la semana con ellos. Y tú no tienes tiempo para llenar libros, ni administración, ni burocracia para entender si el chico está progresando o no. Y **este índice te va a mostrar *al tiro* si hay equilibrio o no en ese desarrollo que buscamos que sea en forma integral**, es sumamente fácil. E incluso **si le incorporas más datos más adelante va a ser espectacular**, pero esto para un dirigente es lo que necesitamos. Porque nosotros todo el otro tiempo tenemos que ocuparlo exclusivamente a observarlos, entonces si nos dedicamos a llenar papeles, perdemos tiempo. (…) además que nuestro sistema es un sistema de educación no formal, entonces no podemos tener herramientas que nos inviten a llenar más formularios.

(…)

Daniel: De hecho, la gran mayoría de los dirigentes **cuando uno les pregunta por algún formulario que está, la gran mayoría te dice que no tiene tiempo, obvio si ya el poco tiempo que tienen por el trabajo, lo están ocupando en los scouts.** Entonces, la idea es que el voluntariado… tratar de que le quite el menor tiempo posible a la persona para que pueda hacer su vida, si al final es su tiempo libre.

Sobre el gráfico de radar de grupo

Con respecto al gráfico de radar utilizado para ver la progresión personal por área del grupo completo, los dirigentes dijeron que preferían tener la posibilidad de usar distintos filtros para ver este gráfico, pues les permite tomar decisiones más estratégicas al momento de planificar actividades. Y especialmente les gustaría tener la posibilidad de ver los **equipos de los beneficiarios**, que corresponden a sub-conjuntos de beneficiarios dentro de una unidad:

Daniel: Entonces esta gráfica sería de el…

Patricio: De todo el grupo, el de la izquierda

Daniel: ¿De todo el grupo?

Patricio: Sí, de todo el grupo, no de una unidad, sino de todo el grupo. ¿Quizás prefieren que sea sólamente de su unidad? ¿O quizás prefieren tener distintos filtros mejor?

Daniel: Yo lo pondría con distintos filtros. El poder ver, de partida, de los equipos, porque **es común que cuando los chicos se juntan por grupos de amigos, la afinidad tiende a que los chicos vayan a por objetivos más o menos similares**. Entonces de repente tú ves que hay un equipo completo como tal que son más

orientados hacia un lado que otro. Entonces ahí, a veces, tratando de invitar a una persona a hacer otras *cuestiones* no funciona tan bien si es que no ves que es el equipo completo el que tienes que tratar de moverlo.

José: De hecho, la diversidad de un equipo, como son amigos, eso es lo que se espera, que sean un grupo de amigos, tienen ideas comunes y acciones comunes, son *chiquillos* que nosotros esperamos que se junten en la semana sin tener a un dirigente con ellos, y van al cine a ver una película que les gusta, o se juntan en la casa de alguno de los compañeros por alguna razón determinada. Entonces igual andan moviéndose con cosas que lo unieron, cosas afines que tenían, por algo hicieron equipo.

Daniel: Y si, por ejemplo, es un equipo que se unieron en sí por jugar al fútbol, es un equipo que tú, probablemente los veas, y van a tener un lado marcado como equipo. Ya ahí, uno puede moverlos un poco, buscar como ese fútbol puedes llevarlo a los diferentes áreas de desarrollo.

Patricio: Permite decisiones bastante estratégicas el trabajar en equipo.

Daniel: Sí

Comentarios finales

Al finalizar la instancia de validación, los dirigentes realizaron comentarios globales de la aplicación web y el proyecto en general, incluso teniendo altas expectativas sobre lo que podría llegar a ser:

Daniel: (...) No, *viejo*, está súper. De hecho, ojalá que a los *cabros* les interese harto esto, porque así como está... yo, hasta buscaría la opción de que la asociación o algo se ponga para poder tener una base de datos grande y tratar de dejar totalmente nacional este tema, obviamente si es que se puede, porque en verdad sirve *caleta*, o sea, la pura parte del resumen donde aparece el mapa hexagonal, donde te aparece si hay algún objetivo en progreso o no. **Ya solamente eso es un ahorro de tiempo gigantesco para cualquier persona que asuma un rol dentro de la unidad**: algún dirigente nuevo... porque cuando llega un dirigente nuevo no falta que te pregunte "Oye, ¿y hay ficha de los chicos o algo como para conocer a los chicos?" y tú llegas y le presentas ahí los papeles (realiza gesto de poner una gran cantidad de papeles sobre la mesa), ahí varios que llegan y "*Eheeh*... no".

Y lo mismo que te decía, o sea, los chicos que se cambian muchas veces: alguien de fuerzas armadas, chicos que se cambian de colegio, son chicos que se pierde el progreso de ellos porque se cambian de ciudad. Piensa que hubo un compañero que estuvo, literalmente, en Rapa Nui, la Antartida y Puerto Montt. O sea, si hubiese estado con algún dirigente en ese momento, y hubiera entrado al grupo scout, te puedo asegurar que quien sea, no tiene ni idea si es que estuvo en algún grupo en Rapa Nui, y menos el contacto.

Dado que ya se habían recibido varios comentarios positivos del uso de la aplicación, el desarrollador decidió preguntar por aquellos aspectos que mejorarían en la aplicación web, con el fin de recibir críticas constructivas de cómo llevar el proyecto de aquí en adelante, a lo cual respondieron dando dos recomendaciones y comentando sobre el potencial que tiene este proyecto para seguir creciendo a futuro:

> **Daniel:** Hasta ahora, **el tema del equipo**, y poco más. **Los detalles que habíamos hablado de la contraseña**... ah, y **si es que estaba la opción de dejar a alguien como encargado de distrito**, como para no sobrecargar a, ti en este caso, el desarrollador, para armar los grupos, porque al final, el distrito tiene que saber qué grupos están conformados, y una vez que terminaron su validación: "Perfecto, aquí esta, de paso, su validación..."
>
> **José:** O sea, (...), es un hecho que **quien valida la existencia del grupo, es la dirección del distrito**. Lo otro, que decía Minte, con respecto a usarlo a nivel nacional, es súper posible. Esto no es como un, ojalá que ocurra, va a ocurrir, primero, **por la naturaleza que tiene el programa**, o lo que yo por lo menos veo, **la potencialidad que tiene**; es una necesidad, **creo que responde a lo que nosotros necesitamos tener para que dediquemos más tiempo a la educación, y menos tiempo a los papeles**, y los muchachos, como te decía yo: a mi no me gusta jugar siquiera, y de verdad que es interesante, es como: me dieron ganas de jugar. Y yo hace rato que no juego, me refiero que lo último que jugué fue como *Space Invaders* o algo así, pero este ávatar es interesante. A mi me encantaría crearme un ávatar y jugar, para ver qué tanto provecho se le puede sacar al asunto.

4.2. VALIDACIÓN DE LA APLICACIÓN MÓVIL

4.2.1. PLAN DE VALIDACIÓN

Los objetivos específicos relacionados a la aplicación móvil, corresponden a los siguientes:

> Diseñar y producir un videojuego que permita motivar a los guías y scout pre-púber y púber a llenar su bitácora personal, de tal manera de llevar un registro de los logros alcanzados en cada una de sus etapas de progresión.

> Validar que el sistema de progresión aumente la motivación de los guías y scout pre-púber y púber y mejore la gestión de su progreso, evaluando su uso con un grupo de prueba conformado por jóvenes, guiadoras y dirigentes scout del distrito de Puerto Montt.

Planes

Teniendo en consideración la situación de pandemia COVID-19 que afecta a la sociedad durante el desarrollo de esta memoria, fue importante contar con más de un plan de validación para que, en caso que no resultara el primero, contar con medidas de contingencia. Esto debido a que, en situación de pandemia, ha sido complicado mantener actividades scout activas para varios grupos, y esto incluye a aquellos en el distrito de Puerto Montt.

En un comienzo se plantearon tres ideas, en orden de prioridad:

1. **Validación por videollamada con beneficiarios y dirigentes de grupos scout del distrito de Puerto Montt:** La primera idea fue reunir a beneficiarios que pertenezcan al público objetivo (guías y scouts entre 11 y 15 años), junto a sus guiadoras y/o dirigentes, por medio de un formulario de inscripción a compartir a través de José Mella y Daniel Minte a los grupos del distrito de Puerto Montt. La instancia consistiría de una videollamada en la cual se realizaría una capacitación del uso de la aplicación a los beneficiarios para que conozcan cada aspecto de ésta, seguido de la resolución de una encuesta para conocer su percepción de la aplicación.
2. **Validación por videollamada con beneficiarios y dirigentes flexibilizando requisitos de participación:** Este plan es similar al anterior, pero buscando beneficiarios en un rango más cercano, flexibilizando un poco los requisitos de participación, como puede ser la edad, con el fin de asegurar un contacto más rápido. Si bien, no resultaría en una validación tan completa como en el plan número 1, permitiría obtener resultados para saber qué pasos seguir a continuación, para luego realizar pruebas más completas al terminar la pandemia (en un período posterior al término de esta memoria), ejecutando el primer plan.
3. **Validación por redes sociales:** Este plan es el menos ideal, pero que era mejor tenerlo en caso que los anteriores fallen, y consistía en abrir al uso público la aplicación móvil para que diferentes personas puedan probar el *software* y responder la encuesta.

Encuesta

El instrumento a utilizar para validar si el proyecto logra motivar a los beneficiarios a realizar los registros de progresión personal, fue elegido teniendo en consideración que la herramienta para la motivación es la *gamificación*, por lo que se busca tener una medida de la sensación de juego que entrega la aplicación. Esto es a lo que los creadores de la encuesta **GAMEFULQUEST** (Högberg y col. 2019) se refieren como *gamefulness*. GAMEFULQUEST es una herramienta para modelar la experiencia individual de *gamefulness* al utilizar un *software* alrededor 7 dimensiones: Logros (*accomplishment*), desafío (*challenge*), competición (*competition*), guiado (*guided*), inmersión (*immersion*), lúdico (*playfulness*) y social (*social experience*).

Por cada dimensión existen una serie de afirmaciones a las que los encuestados asignan un valor según qué tan de acuerdo están con cada ítem utilizando una escala Likert de siete puntos con las siguientes alternativas: (1) Completamente en desacuerdo, (2) En desacuerdo, (3) Ligeramente en desacuerdo, (4) Ni de acuerdo ni en desacuerdo, (5) Ligeramente de acuerdo, (6) De acuerdo, (7) Completamente de acuerdo.

Estas alternativas fueron ordenadas en una matriz de alternativas como en la figura 60 y, se siguieron las indicaciones de la encuesta, que sugerían que, para obtener mejores resultados, definir que las filas de las matrices, y el orden de las dimensiones sea aleatoria.



Figura 60: Dimensión de desafío de GAMEFULQUEST

4.2.2. EJECUCIÓN

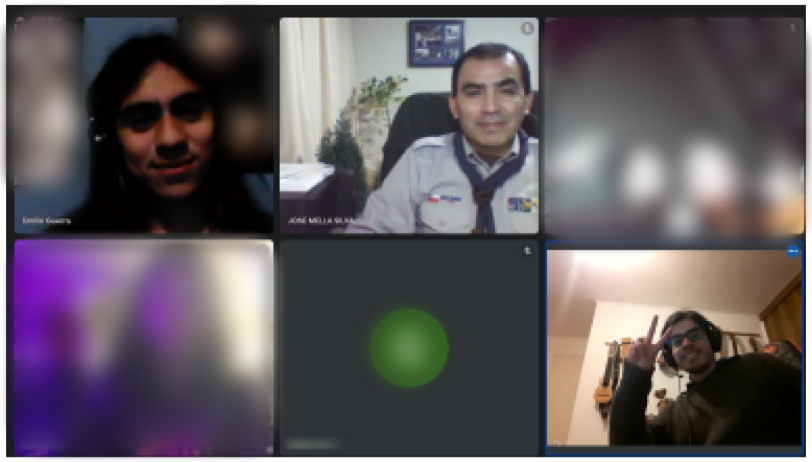


Figura 61: Validación de la aplicación móvil

Para iniciar el proceso de búsqueda de participantes, se comenzó con el primer plan, enviando a través de los dirigentes una encuesta para que beneficiarios y guiadoras y dirigentes del distrito de Puerto Montt puedan inscribirse en la validación de la aplicación móvil, explicando mediante un *elevator pitch*, en qué consiste la aplicación, y el origen de su desarrollo. Esta encuesta sólo logró obtener una respuesta tras ser compartida dos veces, por lo que se pasó a ejecutar al plan número 2.

El desarrollador intentó contactar con un grupo scout que ya conocía, el Grupo Altazor de Concón que, si bien estaba en proceso de inscripción en la Asociación de Guías y Scouts de Chile hasta el comienzo de la pandemia, aun así sus comentarios serían de mucho valor al ya conocer el movimiento scout y su funcionamiento.

Mediante contacto directo con el dirigente de este grupo, se logró organizar la instancia de validación para el día 7 de Agosto de 2021 a las 17:30 por medio de Google Meet como se ve en la figura 61 (por confidencialidad, se censuraron nombres y rostros de los beneficiarios). Esta validación contó con la participación de 4 beneficiarios (2 hombres y 2 mujeres) con edades de 13, 16, 17 y 19 años, junto a su dirigente, Emilio Guerra Astudillo y el director del distrito de Puerto Montt, José Mella.

De los beneficiarios, 3 llevan más de 4 años en el movimiento scout, y el beneficiario restante lleva entre 1 y 2 años, y corresponde a la joven de compañía de 13 años.

En la instancia los beneficiarios descargaron el Android Package (`.apk`) de la aplicación para probarlo en sus propios celulares o *tabletas* con sistema operativo Android.

Si bien se detectaron errores, ninguno de los errores fue crítico ni afectó en lo que se busca validar en la memoria. Algunos errores fueron que, al cerrar forzosamente la aplicación, es-

tando en el campamento virtual, se perdía la información del mundo, esto debido a que el guardado ocurre sólamente durante el menú de escape y al moverse entre zonas. Otro error menor, fue cómo se manejan las recompensas aleatorias, pues el servidor, para un mismo tipo recompensa, tenía una tendencia a entregar siempre el mismo ítem a todos cuando se realizó la transacción en una misma ventana temporal.

A pesar de estos errores, los beneficiarios se vieron muy interesados en por lo innovador que era el proyecto, y cómo se estaba llevando a cabo el registro de la progresión personal de una forma “más allá” que sólo completar formularios.

4.2.3. RESULTADOS

Los resultados de la encuesta fueron ordenados por dimensión de GAMEFULQUEST y se asignaron valores de entre -3 y 3 a las alternativas de la escala Likert, donde números negativos corresponden a respuestas “negativas” (desacuerdo), mientras que los valores positivos corresponden a respuestas “positivas” (de acuerdo).

Para visualizar mejor los datos, se graficaron los resultados en 7 *box plots*, con el fin de representar las respuestas de los beneficiarios teniendo una visión general de la distribución de los datos. Este gráfico se presenta en la figura 62, donde el eje horizontal corresponde al código del ítem, y el eje vertical al valor de la respuesta entregada por los beneficiarios.

El análisis y las conclusiones a partir de estos resultados será descrito en el siguiente capítulo de este documento.



Figura 62: Resultados validación aplicación móvil
Fuente: Elaboración Propia

CAPÍTULO 5
CONCLUSIONES

A continuación se presentan las conclusiones del autor con respecto a distintos aspectos del proyecto. Este capítulo se compone de cuatro secciones: Análisis de resultados de la validación, los alcances y limitaciones del proyecto desarrollado, el trabajo futuro y sugerencias para nuevos proyectos de este tipo.

5.1. ANÁLISIS DE RESULTADOS

5.1.1. APLICACIÓN WEB

La aplicación web fue recibida muy positivamente por los *stakeholders* de este proyecto, que son los dirigentes Daniel Minte y José Mella.

Uno de los aspectos más valorados fue cómo se realizó la traducción del lenguaje de dominio (del movimiento scout) dentro del software, y cómo esto les facilita su tarea de realizar el seguimiento, reduciendo el tiempo que dedican a llenar papeles, pudiendo dedicar un mayor tiempo a trabajar con los beneficiarios, respondiendo directamente con el objetivo de aumentar su capacidad de supervisión. El principal impulsor de esta visión, pareciera ser el gráfico de radar que representa el avance en cada una de las áreas de desarrollo, que permite indicar también cuando hay un desarrollo integral.

Sin embargo, hay que tener en cuenta que ésta fue una primera impresión de parte de los dirigentes al ver el gráfico, en una validación de 1 hora y 40 minutos, todavía quedaría validar este gráfico en una situación real, durante la ejecución de un ciclo de programa de un grupo, para ajustar qué parámetros son los realmente relevantes para un correcto análisis de la progresión personal de un beneficiario. Como dijeron los dirigentes, es posible que integrar más datos a esta métrica ayude a una mejor medida, pero se ve necesario realizar una validación en terreno para elegir estos valores.

Por otro lado, se pudo comprobar con los dirigentes, que el botón “Compartir ficha” en la ficha de beneficiarios, permite incorporar a otros agentes educativos, mediante la invitación de las guiadoras y dirigentes, abarcando el objetivo completo de la aplicación web para esta fase del proyecto.

5.1.2. APLICACIÓN MÓVIL

Todas las conclusiones para esta sección serán a partir de los resultados plasmados en la figura 61. Se analizarán las respuestas entregadas por los beneficiarios ordenadas por dimensión planteada por GAMEFULQUEST (Högberg y col. 2019), tomando en cuenta cómo se utilizó esta dimensión dentro de este proyecto, con el fin de, en la última parte de esta sección, llegar a una conclusión de qué dimensiones requieren más trabajo, y funcionalidades *gamificadas* que se podrían agregar.

Logros (*Accomplishment*)

La dimensión de *accomplishment* se relaciona a la sensación de logro al completar una tarea, lo que en este caso sería completar un objetivo personal. Esta dimensión se puede relacionar directamente con el "core drive" de Octalysis del mismo nombre.

En general hubo reacciones positivas ante esta dimensión, donde el único ítem en el que hubo una fuerte discordancia en las respuestas fue el ítem A05 que dice "Me hace luchar para llevarme al siguiente nivel". Esto puede deberse a que el tiempo de validación fue muy corto como para observar realmente una *lucha* utilizando la aplicación, sin embargo, también podría deberse a que el beneficiario que votó negativamente a este ítem no sintió que hayan muchas limitantes para finalizar un objetivo, e iniciar uno nuevo, como si no hubiese realmente una sensación de "cierre" al completar un objetivo.

Mientras tanto, aquellos que sí sintieron una "*lucha*", quizás la sintieron en la definición de tareas específicas al momento de comenzar un objetivo, donde han de definir tareas, que luego deben completar con tal de obtener las recompensas para su campamento virtual, pero esto pudo no haber tenido el mismo efecto en la sensación de todos los beneficiarios.

Desafío (*Challenge*)

La dimensión de *challenge* tiene que ver con la motivación que sienten algunas personas al encontrar un obstáculo, lo que, en el contexto de este proyecto, se podría relacionar al desafío de tener un objetivo que completar en la vida real para personalizar más su campamento virtual, o tener que realizar registros diarios para el mismo objetivo.

Igual que en el caso anterior, en general hubo una respuesta positiva en esta dimensión, sin embargo hubo dos ítems con diferencias de opinión: Ch01 y Ch04 que corresponden a "Me incentiva a superar mis límites" y "Me desafía" respectivamente. Esto podría estar fuertemente relacionado con el motivo por el que, en la dimensión anterior, el ítem relacionado a la *lucha* se encuentra en la misma situación, considerando que la respuesta negativa corresponde también a quién respondió negativamente al ítem A05. Por lo que, es probable, que sea la misma sensación la que llevó a estas respuestas en estos ítems.

Competición (*Competition*)

La dimensión de *competition* se relaciona a la sensación de competitividad con respecto a ser un ganador entre varios usuarios, o en aspectos del servicio mismo.

Junto a inmersión, corresponde a la dimensión con mayores diferencias de opinión entre los encuestados, donde los ítems más bajos en puntaje son Co01, Co07 y Co08, correspondientes a "Se siente como que estoy participando en una competición", "Se siente como estar en una carrera" y "Me hace sentir que necesito ganar para triunfar" respectivamente. En cambio, el ítem Co06, correspondiente a "Hace que la victoria se sienta importante", o sea, una afirmación más ligada a la sensación de triunfo que a la de competir, tiene una evaluación positiva.

Guiado (*Guided*)

La dimensión *guided* responde a cómo el usuario se siente guiado por el servicio, como puede ser: ser fiel al plan original, estructurar el trabajo y obtener *feedback* de su rendimiento.

Esta es una dimensión con una evaluación generalmente positiva, donde los ítems G01 ("Me hace sentir guiad@"), G05 ("Me da la sensación que tengo un instructor") y G08 ("Me da retroalimentación útil para poder adaptarme") corresponden a los de opiniones más divididas.

Estos puntos se pueden relacionar directamente con la experiencia de usuario (UX) que, si bien parece estar bien encaminada, podría mejorar en los puntos mencionados, mediante una mejor retroalimentación.

Inmersión (*Immersion*)

La dimensión de *inmersion* se relaciona a cómo el servicio entrega una experiencia en la cual se sienta inmerso, como puede ser tener reacciones emocionales a eventos que ocurren dentro de la aplicación.

Esta es una de las dimensiones que requiere más trabajo, donde el ítem más bajo corresponde al I04, llamado "Me hace perderme en lo que estoy haciendo".

Este aspecto, entonces, sería uno de los que GAMEFULQUEST recomendaría mejorar más urgentemente en el corto plazo. Es probable que estas opiniones sean debido a la gran separación, a nivel de interfaz, de lo que es el mundo de juego, de completar formularios. Ya que, esto puede causar que la *gamificación* no se sienta completamente integrada dentro de la aplicación y el usuario sienta como si está utilizando la aplicación, y que esto afecta dentro del campamento virtual, pero no llegar a estar inmerso en este mundo.

Lúdico (*Playfulness*)

La dimensión de *playfulness* se describe por cómo el servicio deja espacio para la imaginación y la creatividad, por la exploración, o inclusive *misterio*.

Corresponde a la dimensión con la mayor cantidad de evaluaciones positivas, probablemente al ser una aplicación que utiliza un entorno virtual directamente dentro de la aplicación en la cual puede aprovechar su imaginación (P03), sentir que puede ser creativo (P04), tener la sensación de poder explorar cosas (P05) y, con respecto al ítem P07 que tuvo una evaluación máxima unánime, tener una sensación de querer saber qué viene después.

Estos ítems parecieran estar relacionados a dos *core drives* de Octalysis: impredecibilidad (*unpredictability*) y pertenencia (*ownership*), pues se encuentran ligados a la sensación de exploración y creación a las que apela el campamento virtual.

Social (*Social Experience*).

La dimensión de *social experience* se relaciona a cómo la presencia de otras personas producen experiencias sociales, como ser motivados por otras usuarios, o que sean *notados* por ellos al cumplir un objetivo. En el estudio de GAMEFULQUEST se menciona que hay servicios que parecieran ser capaces de crear experiencias sociales sin la presencia de personas reales.

Este es uno de los aspectos cuya evaluación fuertemente positiva fue la que más sorprendió al desarrollador, al ser una dimensión en la que no hubo realmente muchas funcionalidades relacionadas en su *core drive* análogo de Octalysis: *influencia social*.

Analizando esta dimensión, se llegó a la conclusión que esto se podría deber principalmente a dos cosas: la naturaleza del movimiento scout y los aspectos creativos de la aplicación. Pues, el campamento virtual y la edición del ávatar corresponden a dos elementos fuertemente visuales y que se pueden compartir fácilmente con compañeros de equipo, o al grupo en general, mediante capturas de pantalla, permitiendo que los beneficiarios compartan entre sí sus mundos y avatares, generando, indirectamente, una experiencia social.

Conclusiones de la aplicación móvil

Si bien, todavía queda por hacer para tener un proyecto completamente sólido en su gamificación. En su estado actual, tras realizar las correcciones de errores necesarias, ya se encontraría en posición de utilizarse a nivel de producción, aunque sea en un grupo reducido de usuarios. Pues, en general, la *gamificación* pareciera haber tenido resultados positivos en la evaluación realizada con los beneficiarios.

Aun así, existen dimensiones que requieren más trabajo, especialmente las de inmersión y competición. En el caso de inmersión, esto se podría corregir integrando el mundo virtual, o más elementos de éste, como capturas del campamento o el ávatar, en más partes de la aplicación, como podría ser en tutoriales, o el proceso de selección del objetivo activo, lo que apoyaría igualmente a la dimensión de guiado, al tener, casi literalmente, un instructor, que sería el mismo ávatar.

Las, relativamente, bajas evaluaciones en la dimensión de inmersión, pueden servir como punto de partida para integrar el aspecto lúdico de la aplicación como apoyo al resto de las

dimensiones, con el fin de abarcar de mejor manera cada aspecto de *gamefulness* que indica GAMEFULQUEST.

Si bien Octalysis no comparte directamente las mismas dimensiones que GAMEFULQUEST, si fue un *framework* útil para el diseño de la *gamificación* al permitir priorizar ciertos aspectos de la *gamificación* en esta primera fase de desarrollo, que luego tendrían efecto en cada dimensión de GAMEFULQUEST, como se observó durante la validación.

El próximo paso, tras corregir los errores encontrados, sería evaluar la aplicación en un caso real, donde, a partir del próximo año, se pueda utilizar como mínimo en un grupo scout a lo largo de un ciclo de programa (de una duración de entre 3 y 4 meses), con el fin de validar esta aplicación en la práctica, y revisar si los resultados de GAMEFULQUEST son los mismos en esa situación. Una vez teniendo estos resultados, sería posible definir qué funcionalidades desarrollar a continuación con el fin de potenciar las dimensiones más débiles.

5.1.3. QUÉ PUDO HABERSE HECHO MEJOR

Sin duda, la pandemia COVID19 fue una situación que influyó de forma directa al proyecto. Una parte muy importante de los proyectos de *gamificación* es comprender a los usuarios, que en este caso son los beneficiarios, pues la *gamificación* no es una herramienta que motive a las personas por sí sola, es necesario estudiar el contexto, y esto se realizó en base a la experiencia de los *stakeholders* en su labor como dirigentes y, levemente, de la experiencia del desarrollador como ex-beneficiario del movimiento scout. Sin embargo, lo ideal hubiese sido trabajar directamente con los beneficiarios; durante el proceso de diseño se intentó contactar con ellos para crear un *focus-group*, pero la situación nacional dificultó bastante el proceso de inscripción.

Ahora, no sería justo establecer como único motivo a la pandemia en esta dificultad del proceso de diseño, la inexperiencia del desarrollador al momento de comenzar este proyecto llevó a varios errores, especialmente en las primeras etapas del desarrollo, que luego fueron corregidas por medio del *framework* Octalysis.

Uno de estos errores fue intentar diseñar el proyecto directamente desde la psicología de la *gamificación* combinado con un *framework* para videojuegos (MDA), pues el objetivo final es utilizar la *gamificación* para motivar, pero no crear un videojuego únicamente. Estudiar el aspecto psicológico fue de bastante ayuda, sin embargo, no fue hasta la aplicación de *Octalysis*, que el proyecto pudo ser diseñado más eficaz y eficientemente.

Otro error, fue la sobre-estimación de las habilidades del desarrollador en crear un proyecto conformado por tantas partes en el tiempo disponible. Todo esto fue corregido cuando se organizó la planificación aumentando “artificialmente” los tiempos que el desarrollador estimó para cada funcionalidad, en base a su conocimiento de las herramientas y el nivel de detalle al que había que llegar.

Con respecto a las herramientas de desarrollo utilizadas, AWS es una excelente herramienta para crear un *back-end* robusto, sin embargo, para nuevos usuarios y, especialmente para crear una primera versión de un *software* no tradicional, ésta no es fácil de aprender ni utilizar correctamente. El autor se encuentra agradecido de haber aprendido esta herramienta, sin embargo opina que, de comenzar de nuevo, hubiera optado por otro servicio en la nube más apto para primeras versiones de *software*, como puede ser Google Cloud mediante Firebase.

También la elección de la base de datos hubiése sido distinta, en la aplicación se utilizan una gran cantidad de *queries* que requieren la creación de índices extra, o un diseño de base de datos más complejo de lo que debería ser para un proyecto más enfocado en la innovación. Esta complejidad podría ser reducida mediante la utilización de otro motor de base de datos con más herramientas para *queries* como podría ser MongoDB o una base de datos SQL, para tener acceso a `JOIN`s, que permitirían generar estadísticas más eficientemente.

ANEXO

CONCRETANDO EL PRIMER DISEÑO

CONCRETANDO LA *GAMIFICACIÓN*

Para concretar más las ideas de la primer diseño, se realizó un diagrama como el de la figura 63 en la plataforma Miro[41]. En esta tabla se organizaron todas las ideas destinadas a motivar a los beneficiarios en las tres necesidades ya mencionadas: competencia, autonomía y relaciones sociales, tal como se ve en la figura 63.

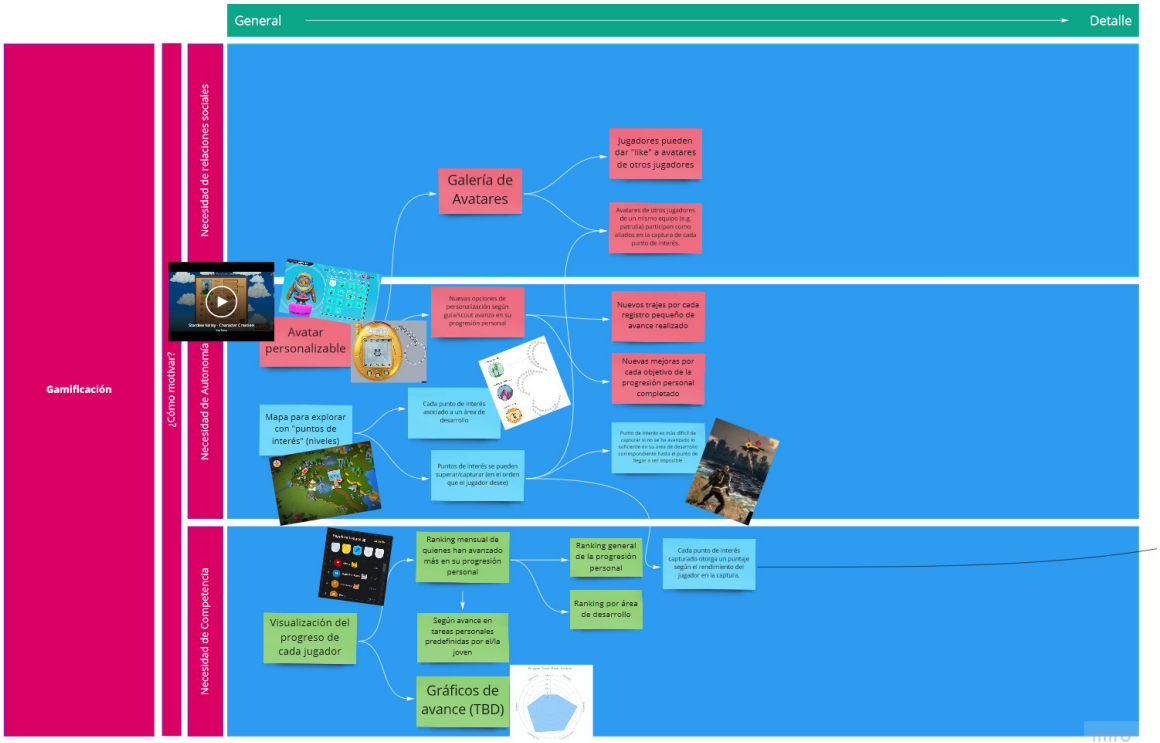


Figura 63: Diseño de la Gamificación del prototipo
Fuente: Elaboración propia

[41]https://miro.com/welcomeonboard/bHBMZDZWU1Y1NTdWUDQ3dGRoc0J1MWxRRUtKQl-haaTZlTnp1WlBJSFluV1ptTnlvOGU2aFJ3RnJ3M2ZHaXl6WHwzMDc0NDU3MzQ5NjM0MjQ4NTQ1

FRAMEWORK *MDA*

Además se creó en la misma plataforma un diagrama utilizando el *MDA framework* (*Mechanics, Dynamics and Aesthetics framework*) planteado en Hunicke y col. 2004, donde se describe todo el diseño de un videojuego utilizando los siguientes conceptos:

- **Mecánicas:** Nivel más bajo de donde surgen los demás, corresponde a las reglas, representación de datos y algoritmos utilizados.
- **Dinámicas:** Surgen a partir de las mecánicas, corresponden a cómo se comportan las mecánicas mediante la interacción del usuario y las otras mecánicas.
- **Estética:** Es la parte más abstracta del *framework*, corresponde a la respuesta emocional esperada del jugador.

Este *framework* fue utilizado para describir la parte del minijuego, construcción y otras interacciones del mundo como lo son tiendas para comprar ítems con los puntos obtenidos del minijuego y los avances de los registros.

También se clasificaron las ideas según su prioridad, con tal de poder ir retirando ideas si el tiempo disponible no era el suficiente.

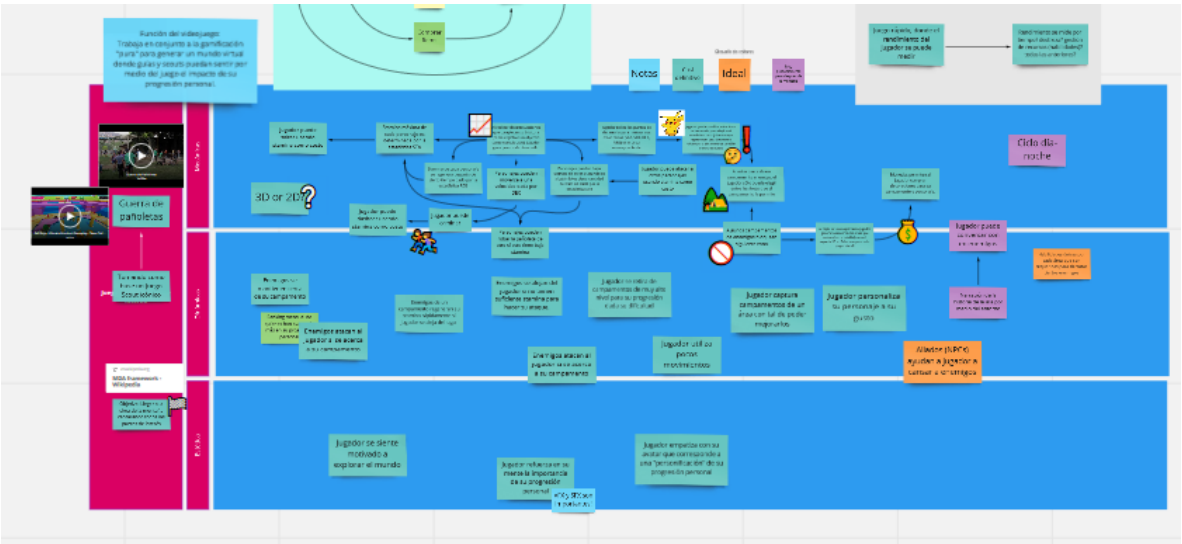


Figura 64: Diseño del videojuego utilizando MDA
Fuente: Elaboración propia

DISEÑO DE LA BASE DE DATOS

Dada la estructura de los datos de la base de datos, es necesario tener definidas las *queries* a realizar en la aplicación, con tal que el modelado de datos funcione en pos de las transacciones a realizar. Estas se encuentran descritas en tabla 5.

Es a partir de estas *queries*, y tras un proceso de normalización y des-normalización de las entidades según los requerimientos del proyecto, se tomaron las siguientes decisiones para las tablas:

- *Beneficiary Score* es una entidad que puede formar parte de *Beneficiary* en forma de `map` donde la llave es un área de desarrollo y el valor es el puntaje en esa área, pues es una relación 1:1.
- Los *Scouter* de un grupo pueden ser almacenados en la fila de un *Group*, pues los datos de un *Scouter* son guardados por Amazon Cognito, y lo único que se ha de tener en cuenta es el rol de un dirigente en un grupo y usar redundancia para guardar su nombre completo.
- Para saber qué recompensas ha obtenido un *Beneficiary* se pueden utilizar los datos almacenados como *Log*.
- Las *Sub-task* de una *Task* pueden ser guardados en el mismo documento como un `Array`.
- Los *Objective* no son necesarios en una tabla extra, pues son estáticos y no han cambiado ni, según lo conversado con los dirigentes, van a cambiar en un tiempo pronto, por lo que pueden ser almacenados como archivos estáticos en cada parte de la aplicación en vez de en la base de datos.

Considerando los puntos anteriores, las tablas finales y sus llaves primarias se encuentran descritas por la tabla 4. En esa tabla se utilizan algunos términos que se encuentran detallados a continuación:

- El método `join(...)` se usa como representación de una concatenación de *strings* por medio de otro *string*, que internamente es `::`. Esto permite utilizar *queries* con `BEGINS_WITH`, para realizar filtros más complejos.
- `releaseId`: Corresponde a un ID diseñado para obtener una recompensa aleatoria y que sea compatible con cierta versión, esto se realiza generando para una versión n, si la recompensa es:
  - COMÚN: Un `releaseId` aleatorio entre $1000 \cdot n + 1$ y $1000 \cdot (n + 1)$
  - RARA: Un `releaseId` aleatorio entre $-1000 \cdot n - 1$ y $-1000 \cdot (n + 1)$

Tabla 5: Queries de la aplicación

Fuente: Elaboración Propia

ID	Entidad	*Query*
D1	*District*	Obtener todos los distritos registrados
G1	*Group*	Obtener todos los grupos de un distrito
B1	*Beneficiary*	Obtener todos los beneficiarios de un grupo
B2	*Beneficiary*	Obtener todos los beneficiarios de un grupo por unidad
L1	*Log*	Obtener todos los registros ordenados por tiempo de creación de cierto tipo: *recompensa*, *recompensa de cierto tipo*, *usados para calcular estadísticas*, *registro granular de cierto objetivo*, *todos los anteriores*.
R1	*Reward*	Obtener una recompensa aleatoria que puede ser “RARA” o “COMÚN” compatible con la versión actual de la aplicación.
T1	*Task*	Obtener todos los objetivos cumplidos por un beneficiario por área de desarrollo.
O1	*Objective*	Obtener todos los objetivos propuestos por: *etapa*, *área de desarrollo* y *línea*
S1	*Scouter*	Obtener todos los dirigentes y guiadoras de un grupo

MICRO-SERVICIOS Y *ENDPOINTS*

Los valores rodeados de llaves ({ y }) en las rutas corresponden a parámetros de la ruta como IDs.

Para gestionar el *routing* en una función *Lambda* se implementó un método que permite asignar a una ruta de la API, un esquema[42] para validar la estructura del cuerpo de un *request* y definir si requiere estar autentificado o no.

Beneficiarios

- `GET /districts/{district}/groups/{group}/beneficiaries/{unit}/`: Obtiene todos los beneficiarios de una unidad en un grupo dado.
- `GET /districts/{district}/groups/{group}/beneficiaries/`: Obtiene todos los beneficiarios de un grupo.
- `GET /beneficiaries/{userId}`: Obtener datos de un beneficiario. Si el que ejecuta este endpoint es el mismo beneficiario o pertenece al grupo *Scouter* tiene acceso a más datos del beneficiario relacionados a su progresión, como su objetivo activo. No requiere estar autentificado para ser accesible, esto para permitir a agentes educativos a realizar el seguimiento de un beneficiario sin necesidad de crear una cuenta.
- `PUT /beneficiaries/{userId}`: Actualiza los datos de un beneficiario, estos pueden ser su Apodo (`nickname`) y/o su imagen de perfil (`profile_picture`).

Distritos

- `GET /districts/`: Obtiene todos los distritos disponibles.
- `GET /districts/{districtCode}`: Obtiene los datos de un distrito.

Galería

- `GET /beneficiaries/{userId}/avatar/`: Obtiene el ávatar de un beneficiario.
- `PUT /beneficiaries/{userId}/avatar/`: Actualiza el ávatar de un beneficiario.

Grupos

[42] https://pypi.org/project/schema/

- `GET /districts/{districtCode}/groups/`: Obtiene todos los grupos de un distrito.
- `GET /districts/{districtCode}/groups/{groupCode}`: Obtiene los datos de un grupo. Si el usuario que ejecuta este endpoint es un dirigente o guiadora encargado de este grupo, entonces también recibe las contraseñas para invitar a otros dirigentes o beneficiarios.
- `GET /districts/{districtCode}/groups/{groupCode}/stats`: Obtiene las estadísticas de progresión del grupo, tales como registros granulares realizados y objetivos completados.
- `POST /districts/{districtCode}/groups`: Crea un grupo en el distrito dado. Este *endpoint* es sólo accesible por un administrador.
- `POST /districts/{district}/groups/{group}/beneficiaries/join`: Recibe la contraseña de beneficiario (8 dígitos) y crea un beneficiario en la base de datos asignado al grupo dado.
- `POST /districts/{district}/groups/{group}/scouters/join`: Recibe la contraseña de dirigente (contraseña generada a partir del código de distrito y grupo mediante SHA1) y crea un beneficiario en la base de datos asignado al grupo dado.
- `POST /districts/{district}/groups/{group}/init`: Define el estado inicial de un grupo, por el momento sólo agrega a un usuario dado como el dirigente y guiadora gestor de un grupo, para que pueda invitar al resto de los dirigentes y guiadoras. Este *endpoint* es sólo accesible por un administrador.

Tareas (*Tasks*)

- `GET /users/{userId}/tasks/`: Obtiene todos los objetivos personales cumplidos por el usuario.
- `GET /users/{userId}/tasks/{stage}/`: Obtiene todos los objetivos personales cumplidos por el usuario de cierta etapa de desarrollo (pre-pubertad o pubertad).
- `GET /users/{userId}/tasks/{stage}/{area}`: Obtiene todos los objetivos personales cumplidos por el usuario de cierta etapa y área de desarrollo.
- `GET /users/{userId}/tasks/{stage}/{area}/{subline}/`: Obtiene un objetivo personal dado. Retorna un código HTTP 404 si no se ha cumplido.
- `GET /users/{userId}/tasks/active/`: Obtiene el objetivo en progreso del usuario dado.
- `POST /users/{userId}/tasks/{stage}/{area}/{subline}/`: Da inicio a un objetivo en progreso.

- `POST /users/{userId}/tasks/active/complete/`: Marca el objetivo en progreso como completado.
- `POST /users/{userId}/tasks/initialize/`: Define los objetivos personales cumplidos previos al uso de la aplicación.
- `PUT /users/{userId}/tasks/active/`: Actualiza el objetivo en progreso.

Registros (*Logs*)

- `GET /users/{userId}/logs/`: Obtiene todos los registros asociados a un usuario.
- `GET /users/{userId}/logs/{tag}`: Obtiene todos los registros asociados a un usuario cuyo `tag` comience con el *sub-string* dado.
- `POST /users/{userId}/logs/{tag}`: Permite crear un registro personalizado como un registro granular, recibe el **tag** con el que se quiere guardar el registro el cual, por el momento, sólo puede ser *"PROGRESS"*, que son los registros granulares de avance.

Recompensas (*Rewards*)

- `GET /rewards/{categoryCode}/{release}/`: Obtiene todas las recompensas de cierta categoría disponibles para la versión dada en `release`.
- `GET /rewards/{categoryCode}/{release}/{id}/`: Obtiene los datos de una recompensa.
- `GET /rewards/mine/{categoryCode}`: Obtiene las recompensas que el usuario autentificado ha obtenido de cierta categoría.
- `POST /rewards/{categoryCode}/{release}/`: Crea una recompensa para una versión dada. Este *endpoint* es sólo accesible por un administrador.
- `POST /rewards/claim/`: Reclama un *token* de recompensa.

Dirigentes y guiadoras (*Scouters*)

- `GET /districts/{districtCode}/groups/{groupCode}/scouters/`: Obtiene todas las guiadoras y dirigentes de un grupo.
- `GET /districts/{districtCode}/groups/{groupCode}/scouters/{userId}`: Obtiene la información y el rol de una guiadora o dirigente en el grupo dado.

REFERENCIAS BIBLIOGRÁFICAS

Arai, S., Sakamoto, K., Washizaki, H. & Fukazawa, Y. (2014). A Gamified Tool for Motivating Developers to Remove Warnings of Bug Pattern Tools, En *2014 6th International Workshop on Empirical Software Engineering in Practice*.

Asociación de Guías y Scouts de Chile. (2014). *Marco conceptual del método y modalidad de animación del programa de jóvenes*.

Asociación de Guías y Scouts de Chile. (2019). Estatuto de la Asociación de Guías y Scouts de Chile.

Balaji, S. & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), 26-30.

Chou, Y.-k. (2019). *Actionable gamification: Beyond points, badges, and leaderboards*. Packt Publishing Ltd.

Comisión Nacional de la Rama Guías. (2018). *Manual de la Rama Guías*. Comisión de Educación de Guías y Scouts de Chile.

Comisión Nacional de la Rama Scouts. (2016). *Manual de la Rama Scouts*. Comisión de Educación de Guías y Scouts de Chile.

Departamento de Informática UTFSM. (2019). Sendero Scout. https://www.youtube.com/watch?v=F5V1NeDtyqI

Deterding, S., Dixon, D., Khaled, R. & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining "Gamification", En *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, Tampere, Finland, Association for Computing Machinery. https://doi.org/10.1145/2181037.2181040

Högberg, J., Hamari, J. & Wästlund, E. (2019). Gameful Experience Questionnaire (GAMEFULQUEST): an instrument for measuring the perceived gamefulness of system use. *User Modeling and User-Adapted Interaction*, *29*(3), 619-660.

Hunicke, R., Leblanc, M. & Zubek, R. (2004). MDA: A formal approach to game design and game research, En *In Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*, Press.

Karl M. Kapp, R. M., Lucas Blair. (2013). *The Gamification of Learning and Instruction Fieldbook: Ideas into Practice*.

Landers, R. N. & Landers, A. K. (2014). An Empirical Test of the Theory of Gamified Learning: The Effect of Leaderboards on Time-on-Task and Academic Performance. *Simulation & Gaming*, *45*(6), https://doi.org/10.1177/1046878114563662, 769-785. https://doi.org/10.1177/1046878114563662

Morschheuser, B., Hassan, L., Werder, K. & Hamari, J. (2018). How to design gamification? A method for engineering gamified software. *Information and Software Technology*, *95*, 219-237. https://doi.org/https://doi.org/10.1016/j.infsof.2017.10.015

Objetivos educativos del Movimiento Scout. (1993-1996). Organización Mundial del Movimiento Scout.

Reglamento de la Asociación de Guías y Scouts de Chile. (2019).

Rigby, S. & Ryan, R. M. (2011). *Glued to games: How video games draw us in and hold us spellbound: How video games draw us in and hold us spellbound*. AbC-CLIo.

Rojas, J. (2006). *Los boy scouts en Chile, 1909-1953* (1.ª ed.). Dirección de Bibliotecas, Archivos y Museos.

Sailer, M., Hense, J. U., Mayr, S. K. & Mandl, H. (2017). How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in Human Behavior*, *69*, 371-380. https://doi.org/https://doi.org/10.1016/j.chb.2016.12.033

Salen, K., Tekinbaş, K. S. & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. MIT press.

The Editors of Encyclopaedia Britannica. (2020). *Robert Baden-Powell, 1st Baron Baden-Powell*. www.britannica.com/biography/Robert-Stephenson-Smyth-Baden-Powell-1st-Baron-Baden-Powell

World Organization of the Scout Movement. (2018). *Baden-Powell*. https://www.scout.org/node/52292/introduction