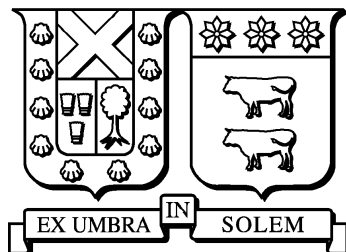


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SANTIAGO – CHILE



“PROPUESTA DE UN FRAMEWORK PARA EL
DESARROLLO DEL PENSAMIENTO
COMPUTACIONAL EN COLEGIOS CHILENOS
UTILIZANDO GAMIFICATION”

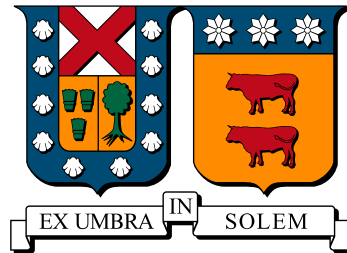
JOSÉ TOMÁS DOCMAC LORENZINI

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: PEDRO FRANCISCO GODOY BARRERA

DECIEMBRE 2022

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



**“PROPUESTA DE UN FRAMEWORK PARA EL
DESARROLLO DEL PENSAMIENTO
COMPUTACIONAL EN COLEGIOS CHILENOS
UTILIZANDO GAMIFICATION”**

JOSÉ TOMÁS DOCMAC LORENZINI

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO**

PROFESOR GUÍA: PEDRO FRANCISCO GODOY BARRERA
PROFESOR CORREFERENTE: ANDREA FERNANDA VÁSQUEZ GUERRA
FEDERICO JESÚS MEZA MONTOYA

DECIEMBRE 2022

MATERIAL DE REFERENCIA, SU USO NO INVOLUCRA RESPONSABILIDAD DEL AUTOR O DE LA INSTITUCIÓN

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres. Sin su apoyo, esta etapa de mi vida hubiese sido mucho más difícil. Después, me gustaría agradecer a mis amigos, tanto a los de Curicó como a los nuevos amigos que hice en la universidad. Sin ustedes, no estaría donde estoy ahora, por lo que estaré por siempre agradecido. Y finalmente, me gustaría agradecer también a mi profesor guía Pedro Godoy. Él fue quién me enseñó a programar cuando entré a la universidad en primer año, y fue quién me ayudó y guió en el desarrollo de este trabajo en último año. Por eso, siempre estaré agradecido también.

Resumen

El presente trabajo muestra una propuesta de un Framework para trabajar y desarrollar el Pensamiento Computacional a través de actividades que utilicen la Gamification como herramienta principal, con foco en el aula de clases de la enseñanza básica. La metodología para construir el Framework requirió de un exhaustivo estudio de la bibliografía tanto en términos conceptuales como la experimentación que se ha realizado en la educación escolar y universitaria en distintos países del mundo. Con esto, se aíslan los elementos clave de Pensamiento Computacional y de Gamification que son necesarios para trabajar, y se unen en base a un criterio lógico que permite potenciar el Pensamiento Computacional, siempre cuidando que los elementos de Gamification coexistan en armonía en la actividad. Así, el Framework permite a los educadores de distintas asignaturas crear actividades gamificadas que les permiten trabajar el Pensamiento Computacional, sin tener conocimientos profundos en diseño de juegos, lo cual parece ser uno de los principales problemas e impedimentos para los educadores, pues es algo que está fuera de su área.

Abstract

This work presents an approach of a Framework that has the purpose of develop and improve the skills associated with Computational Thinking, using Gamification as the main tool. The objective of this Framework is to gamify activities in school level education, helping teachers with the activity design. The Framework was built after a comprehensive revision of several studies associated with this subject, as different authors had already studied this subject and others had tested teaching Computational Thinking using a Gamification approach. Having said that, the key components of both Computational Thinking and Gamification were isolated and then mixed based on a logical criteria, with the main focus being always to maximize the Computational Thinking skills improvement, and being careful that Gamification components can coexist together in spite of the activity nature.

Finally, the Framework fulfills the main objective and helps teachers of different subjects to create gamified activities that accomplish to improve Computational Thinking skills of the students without having a very extense knowledge of game design.

Índice de Contenidos

Agradecimientos	III
Resumen	IV
Abstract	V
Índice de Contenidos	VI
Lista de Tablas	XI
Lista de Figuras	XII
Glosario	XV
Introducción	1
1. Definición del Problema	3
1.1. Causas del problema	4
1.1.1. Brecha Digital	4
1.1.2. Investigación a nivel nacional	4
1.1.3. Currículo escolar desactualizado	5
1.1.4. Técnicas educativas tradicionales	5

1.2.	Consecuencias	7
1.2.1.	Reprobación de cursos universitarios	7
1.2.2.	Resolución de problemas de alta complejidad	8
1.2.3.	Problemas en el mundo laboral	9
1.3.	Objetivos	9
1.3.1.	Objetivo principal	9
1.3.2.	Objetivos específicos	9
2.	Marco Conceptual	11
2.1.	Pensamiento Computacional	12
2.1.1.	Componentes del Pensamiento Computacional	14
2.2.	Gamification	15
2.2.1.	Gamification y Educación	17
2.2.2.	Gamification, Educación y Pensamiento Computacional	22
2.3.	Game Design	26
2.3.1.	El juego	26
2.3.2.	Las Reglas	28
2.3.3.	Los Elementos	29
2.3.4.	Las Mecánicas	30
2.3.5.	Game Design, Gamification y Educación	31
2.4.	Framework	32
2.4.1.	Gamification Frameworks	33
2.5.	Trabajo Colaborativo	37
2.5.1.	Elementos de la cooperación	37
3.	Propuesta	39

3.1.	Idea General	41
3.2.	Antecedentes del Framework	41
3.2.1.	Pensamiento Computacional	42
3.2.2.	Gamification	44
3.3.	Diseño del Framework	45
3.4.	Framework de Pensamiento Computacional	51
3.4.1.	Framework de Pensamiento Computacional de Primer Nivel	51
3.4.2.	Framework de Pensamiento Computacional de Segundo Nivel	52
3.4.3.	Reglas de puntuación del Framework	54
3.4.4.	Consideraciones del Framework	58
3.4.5.	Secciones de una Actividad	60
3.4.6.	Enfoques de las actividades de acuerdo al vértice	62
3.5.	Secuencia de uso: cómo utilizar el Framework	64
3.5.1.	Cuestionario inicial	65
3.5.2.	Diseño de la actividad I	67
3.5.3.	Diseño de la actividad II	70
3.5.4.	Diseño de la actividad III	72
3.5.5.	Puntuación y figura del Framework	72
3.6.	Diagrama de flujo	73
3.6.1.	Cuestionario Inicial	74
3.6.2.	Diseño I	74
3.6.3.	Diseño II, Diseño III y Final	74
3.7.	Encuesta de percepción y feedback	74
3.7.1.	Sobre el Pensamiento Computacional y el Trabajo Colaborativo	76
3.7.2.	Sobre Gamification	77

3.7.3.	Sobre la actividad	78
3.7.4.	Posibles Respuestas	79
3.8.	El Framework en funcionamiento: un ejemplo demostrativo	81
3.8.1.	Diseño de la actividad	83
3.8.2.	Figura y Puntaje	87
3.8.3.	Encuesta de percepción	88
4.	Implementación	90
4.1.	Experimentos	91
4.1.1.	Primer experimento: Python 3	91
4.1.2.	Segundo experimento: Microsoft Excel y Visual Basic	92
4.2.	Herramienta basada en el Framework	92
4.2.1.	Primera parte: Bienvenida, Teoría Previa e Instrucciones de Uso	92
4.2.2.	Segunda parte: Diseño y Hoja de Puntaje	94
4.2.3.	Tercera parte: Encuesta y Comparativa	95
4.3.	Aprendizajes y modificaciones	96
4.4.	Validación del Trabajo	97
4.4.1.	Primera instancia: Usuario utiliza la herramienta con estrategia Think Aloud	97
4.4.2.	Segunda instancia: Usuarios ven la herramienta en uso a través de visita guiada y juicio experto	99
4.4.3.	Otros	100
5.	Conclusiones	102
	Conclusiones	102
5.1.	Aprendizajes y limitantes	102
5.2.	Trabajo Futuro	104

6. Anexos	105
Anexos	105
6.1. ANEXO A: Actividad-tipo de acuerdo al vértice	105
6.2. ANEXO B: Imagenes de la herramienta basada en el Framework	111
Bibliografía	122

Índice de tablas

1.1. Porcentaje de reprobados por año en Casa Central en Programación.	7
1.2. Porcentaje de reprobados por año en Santiago en Programación.	8
1.3. Porcentaje de reprobados total por año en Programación.	8
2.1. Resumen de definiciones y conceptos.	15
2.2. Resumen de definiciones y componentes.	15

Índice de figuras

1.1. Árbol del problema.	10
2.1. Octalysis Framework.	35
3.1. Diagrama del flujo de la propuesta.	40
3.2. Núcleo del Framework basado en Pensamiento Computacional.	45
3.3. Subgrupos con los elementos de PC y Gamification seleccionados.	49
3.4. Subgrupo del Trabajo Colaborativo y Gamification.	50
3.5. Framework de Pensamiento Computacional de Primer Nivel.	51
3.6. Framework de Pensamiento Computacional de Primer Nivel graduado.	52
3.7. Framework de Pensamiento Computacional de Segundo Nivel.	53
3.8. Framework de Pensamiento Computacional de Segundo Nivel graduado.	53
3.9. Seccionado de una actividad.	61
3.10. Resumen de la secuencia de diseño.	73
3.11. Ejemplo de figura en base a actividad.	74
3.12. Flujo del Cuestionario Inicial.	75

3.13. Flujo del Diseño I.	76
3.14. Flujo del Diseño II, III y Final.	77
3.15. Encuesta: Preguntas relacionadas a PC.	78
3.16. Encuesta: Ejemplo de pregunta para información adicional.	78
3.17. Encuesta: Preguntas relacionadas al Trabajo Colaborativo.	79
3.18. Encuesta: Preguntas relacionadas a Gamification.	80
3.19. Encuesta: Preguntas relacionadas a Gamification.	81
3.20. Encuesta: Preguntas relacionadas a la actividad en general.	82
3.21. División del espacio de trabajo (Espacio Real).	84
3.22. Figura de la actividad de para Educación Física basada en el Framework.	89
4.1. Pantalla Teoría Previa V1.0.	96
6.1. Pantalla de Bienvenida.	111
6.2. Pantalla de Teoría Previa.	112
6.3. Pantalla de Teoría del Framework (1/3).	113
6.4. Pantalla de Teoría del Framework (2/3).	113
6.5. Pantalla de Teoría del Framework (3/3).	114
6.6. Pantalla de Instrucciones de uso (1/2).	114
6.7. Pantalla de Instrucciones de uso (2/2).	115
6.8. Pantalla de Cuestionario Inicial.	116
6.9. Pantalla de PC y TC.	117

6.10. Pantalla de Diseño I (1/2).	117
6.11. Pantalla de Diseño I (2/2).	118
6.12. Pantalla de Diseño II.	118
6.13. Pantalla Hoja de Puntaje.	119
6.14. Pantalla Instrucciones Encuesta.	119
6.15. Encuesta en Google Forms.	120
6.16. Pantalla Planilla Respuestas.	121
6.17. Pantalla Comparativa.	121

Glosario

UTFSM: Universidad Técnica Federico Santa María.

PC: Pensamiento Computacional.

TC: Trabajo Colaborativo.

On the go: Para este caso, es sinónimo de runtime o tiempo de ejecución.

Wrapper: Para este caso, es un elemento que engloba a otro más pequeño.

Dummies: Casos de prueba.

Snowball: Bola de nieve. En este caso, se utiliza para definir el efecto "snowball", que corresponde a algo pequeño que al ponerse en movimiento, se hace cada vez más grande.

Introducción

Con el avance de la tecnología, el mundo cambia constantemente. La aparición de los computadores significó un cambio radical en la forma de hacer muchas cosas, y en todos los ámbitos de la vida. La ciencia, el arte, la robótica, la cocina y hasta las manualidades se han visto influenciadas por la computación, pasando parte de su desarrollo hacia la digitalización. Este avance de la tecnología también ha permitido que las personas empiecen a cambiar la forma de pensar. Problemas que antiguamente se veían imposibles de resolver, ahora se pueden calcular tras un par de *clicks*, e incluso en aparatos electrónicos que caben en el bolsillo. Así, el uso del pensamiento lógico se ha vuelto casi que un estándar para todas las personas, pues es la única forma de adaptarse a los cambios que se generan en el día a día.

Esto ha llevado a que, con el tiempo, distintos expertos comenzaran a estudiar la relación entre la resolución de problemas a través de la computación y la forma de pensar de las personas. Después de muchos años de experimentación dentro de la computación y la informática, la conclusión revela que no sólo se necesita tener claro **qué** conceptos son los que hay que saber, sino que, además, hay que saber **cuándo** hay que aprenderlos.

Aquí es donde aparece el concepto de **Pensamiento Computacional**, el cual fue reconocido por la UNESCO como una de las Competencias del siglo XXI, pues es esencial para el mundo moderno en el que vivimos. Además, la educación y el desarrollo del PC como habilidad debe comenzar, por lo menos, en la etapa de la educación escolar básica de las personas.

Actualmente, son varios los países que han incursionado en agregar al currículo escolar cursos asociados al Pensamiento Computacional, pero la mayoría se encuentra muy por

detrás. Chile no es la excepción, pues -aunque es uno de los países que más estudios al respecto tiene en sudamérica, junto con Argentina y Colombia- las iniciativas para enseñar y desarrollar el PC son escasas, y generalmente, son en el ámbito extra programático. El gran problema con que cuenta el desarrollo del PC actualmente - y esto es independiente del lugar donde se quiera trabajar - es que los mecanismos tradicionales de la educación son poco eficientes para su enseñanza (por ejemplo, las clases lectivas). Así, generar instancias para trabajar el PC resulta difícil para las personas, sobre todo para el cuerpo docente del país. Para eso aparece la Gamification, que resulta ser una herramienta muy poderosa para generar estas instancias, pero con la complicación que hay que aprender a usarla, pues su génesis ocurre al implementar elementos que son del mundo lúdico en contextos más serios.

Es así, finalmente, que este trabajo de memoria intenta dar una solución al problema de crear e implementar instancias que sirvan para desarrollar el Pensamiento Computacional, y lo hace a través de una propuesta de un Framework de desarrollo de actividades para colegios. Para llevar a cabo el diseño del Framework, se requiere de un completo estudio sobre Pensamiento Computacional y Gamification, además de un barniz de lo que es el Game Design y cómo sus elementos conviven dentro del mundo de los juegos y videojuegos. Con esto, es posible agrupar los elementos del Pensamiento Computacional que se quieren trabajar con los elementos de Gamification que sirven para potenciar a los de PC, y así, entregarle una serie de directrices de diseño a los educadores donde, además, podrán analizar de forma cuantitativa su diseño, sumado a una comparación directa con la percepción que tuvieron los estudiantes acerca de la actividad.

Capítulo 1

Definición del Problema

El mundo es cada vez más competitivo y, hoy en día, los desafíos que se proponen para los nuevos profesionales son distintos a los de hace unos años atrás. Con esto, cambian los objetivos, las especialidades, los trabajos y las competencias que requieren los problemas modernos.

Con respecto a estas últimas, la UNESCO ha catalogado a las más importantes como las Competencias del siglo XXI, entre las cuales aparecen el pensamiento crítico, el pensamiento lógico, la resolución de problemas y el trabajo colaborativo, entre otras [25].

Aquí es donde se vuelve necesario hablar sobre el Pensamiento Computacional, pues esta habilidad engloba diversas competencias, como lo son la abstracción, el pensamiento algorítmico (o lógico), el reconocimiento de patrones y la subdivisión de problemas, convirtiéndose así en una habilidad fundamental para los profesionales del futuro.

El Pensamiento Computacional se ha transformado en un eje a desarrollar en la educación escolar, y así lo demuestran los países que se han aventurado a incluirlo, paulatinamente, en la formación de sus estudiantes. Por ejemplo, dentro de la misma región, Argentina es el país que tiene más investigación por parte de la academia, sumado al proyecto “Program.ar”[4], el cual está encargado de complementar la formación académica tradicional en los colegios con cursos enfocados en desarrollar el Pensamiento Computacional. Otro ejemplo, y siendo el más avanzado de todos, corresponde a lo que ha hecho el Reino Unido y España (específicamente en Madrid). Ellos ya cuentan con un plan académico elaborado para ser

incluido dentro de los colegios, el cual trabaja el Pensamiento Computacional enfocándose en medidas innovativas y dándole opciones metodológicas a los educadores [26]. En Chile, la enseñanza del Pensamiento Computacional (y otras competencias) se ha visto retrasada en comparación con otros países, lo que significa que tanto las actuales generaciones de estudiantes como las que están por venir, que van desde el colegio hacia la universidad, están quedando sin la formación necesaria para los desafíos actuales del mercado y del país. A continuación, se revisan las causas del problema planteado.

1.1. Causas del problema

1.1.1. Brecha Digital

La Brecha Digital se entiende como “las diferencias en la capacidad de acceso a las tecnologías de información y comunicaciones de distintos grupos de población en un mismo país o a nivel internacional”[5], y según el último estudio de Con@cción, al año 2020 hay aún 3,4 millones de personas sin acceso a internet en nuestro país [1].

El Pensamiento Computacional es una habilidad que se encuentra fuertemente potenciada por la tecnología. Es posible enseñar y practicar esta competencia sin el uso de ella (a través de actividades “unplugged”), pero limita las posibilidades en una gran medida, pues la tecnología permite crear actividades que se diseñan al detalle y que generan un vínculo de interactividad con el estudiante. Además, se ve más limitada aún si la técnica educativa se cierra a los métodos tradicionales de clases lectivas sin ningún tipo de apoyo moderno.

1.1.2. Investigación a nivel nacional

Según el estudio de Roig-Vila et al. al año 2018 Chile no tiene ninguna publicación asociada al Pensamiento Computacional en la educación, siendo superados en la región por Perú, Colombia, Argentina y Brasil [21].

Una forma de potenciar el trabajo y la inclusión de estas competencias en el currículo escolar chileno es lograr que la academia y los profesionales del área de la educación investiguen

para sentar las bases de la educación del futuro.

1.1.3. Currículo escolar desactualizado

Pese a que el currículo escolar chileno ha tenido cambios, estos no han estado enfocados en incluir competencias como lo es el Pensamiento Computacional. Los cambios están enfocados mayoritariamente en incluir o quitar horas de las asignaturas históricas del currículo, pero ningún cambio ha sido planeado para incluir actividades que fomenten las nuevas competencias y que se salgan de las técnicas educativas tradicionales.

1.1.4. Técnicas educativas tradicionales

Las competencias del siglo XXI, en su mayoría, son consideradas como “soft skills”, por lo que el modelo de educación tradicional - entendiéndose como clases 100 % lectivas, sumado a interrogaciones de pregunta-respuesta - no es el óptimo para enseñarlas y para ponerlas en práctica.

Para trabajar estas competencias, se vuelve entonces necesario actualizar y/o complementar las técnicas tradicionales con algunas más modernas, como lo son las actividades lúdicas, clases invertidas, actividades kinestésicas, trabajo en equipo y actividades “unplugged”.

Con todo esto dicho, resulta imperativo encontrar una forma de normalizar la enseñanza de estas competencias en los colegios chilenos. Pese a la poca investigación por parte de la academia, según M. Sarmiento, Chile es uno de los países que más experiencias educativas para el desarrollo del Pensamiento Computacional realiza en Iberoamérica, en conjunto con Argentina y España [7]. Estas experiencias son realizadas, en su mayoría, por universidades y por académicos que realizan actividades extracurriculares para colegios (entre ellos, académicos de esta misma casa de estudios). Aún así, es posible inferir que esto no es suficiente, puesto que la mayor parte -sino todos- de los colegios en Chile no han integrado un mecanismo de enseñanza de estas competencias, sumado a los colegios vulnerables que se encuentran completamente aislados de la tecnología.

Los actores de esta problemática son tanto los alumnos como los profesores de la educación escolar. En el caso de los estudiantes, ellos no están recibiendo la formación integral que la educación superior (y posteriormente la vida profesional) requiere, y en el caso de los educadores, la falta de investigación y de un plan de trabajo serio e integrador por parte del Estado los ha dejado a su suerte, teniendo que implementar actividades extracurriculares o, en su defecto, no generar instancias para el desarrollo de estas competencias.

Actualmente, Chile cuenta con el Plan Nacional de Lenguajes Digitales [3], el cual ha progresado en términos de diseñar un curso de acción con respecto al desarrollo del Pensamiento Computacional, construyendo un currículo en forma de trayectoria formativa, separando distintos cursos de acuerdo a la edad de los estudiantes. Esta alternativa resulta ser la iniciativa más potente que existe en el país.

El problema que tiene este plan, fuera de si su implementación ha funcionado o no, es que depende exclusivamente de la tecnología, pues se basa exhaustivamente en programación, robótica, inteligencia artificial y desarrollo de aplicaciones. Esto deja fuera automáticamente a todos los colegios vulnerables donde la tecnología no llega, sumado a los lugares más recónditos del territorio nacional donde la conexión a internet es escasa. Además, y siendo otro aspecto importante, discrimina cursos que no están asociados fuertemente a la tecnología o que no siguen un lineamiento tan lógico dentro de sus contenidos, como lo son Lenguaje, Ciencias Sociales o Educación Física. Naturalmente, nace la necesidad de generar instancias donde, dentro de la mayoría -o de ser posible, todos- de los cursos del currículo escolar chileno, se puedan implementar actividades enfocadas en desarrollar el Pensamiento Computacional.

Dado que este problema es algo que viene ocurriendo desde hace años, hay diversos elementos que podrían atribuirse como consecuencias.

1.2. Consecuencias

1.2.1. Reprobación de cursos universitarios

En la universidad, sobre todo en carreras técnicas y de ingeniería, son diversos los cursos donde se utiliza el Pensamiento Computacional. Si desde el colegio existe una falencia en enseñar y poner en práctica las competencias del siglo XXI (entre ellas, el pensamiento lógico), es normal ver que exista parte de la población de estudiantes que falla en los cursos donde estas competencias son útiles, como lo son los cursos de programación, matemáticas, física, etc.

Es importante reconocer este fenómeno para explicar parte de la reprobación de los cursos, pero también es crucial mencionar que esta, bajo ningún caso, es la única de las razones. Para ilustrar el caso, se muestran, a modo de ejemplo, las Tablas 1.1 y 1.2 con datos del 2016 al 2018 del curso de Programación de esta casa de estudios, tanto en Casa Central como en Santiago. En la Tabla 1.3 se ilustran los promedios de reprobación total por año.

Casa Central				
1° Semestre			2° Semestre	
Año	Total alumnos	% Reprobados	Total alumnos	% Reprobados
2016	669	36,7 %	552	44,0 %
2017	732	42,8 %	578	28,6 %
2018	734	36,9 %	490	26,1 %

Tabla 1.1: Porcentaje de reprobados por año en Casa Central en Programación.

Fuente datos: Departamento de Informática - UTFSM.

Finalmente, con los datos de las tablas anteriores se ilustran los promedios anuales de ambos campus en la Tabla 1.3

Santiago				
1° Semestre			2° Semestre	
Año	Total alumnos	% Reprobados	Total alumnos	% Reprobados
2016	515	25,6 %	470	37,4 %
2017	590	37,5 %	520	33,6 %
2018	753	35,0 %	438	29,5 %

Tabla 1.2: Porcentaje de reprobados por año en Santiago en Programación.

Fuente datos: Departamento de Informática - UTFSM.

Promedio total de reprobación por año	
2016	35,9 %
2017	35,6 %
2018	31,9 %

Tabla 1.3: Porcentaje de reprobados total por año en Programación.

Fuente datos: Departamento de Informática - UTFSM.

Los fenómenos que llevan a un 30-35 % de los estudiantes a reprobado son diversos, y no es posible establecer porcentajes para cada uno de ellos. Lo que sí se puede considerar es que, fortaleciendo la enseñanza y el trabajo de Pensamiento Computacional desde el colegio, ese 30 % se puede ver reducido en futuras generaciones. Y así también los respectivos porcentajes de reprobación en otros cursos y en otras universidades e institutos técnicos del país.

1.2.2. Resolución de problemas de alta complejidad

Como consecuencia directa de la ausencia del desarrollo del pensamiento lógico (más allá de lo que se ve en el curso escolar de Matemáticas), la resolución de problemas de alta complejidad se ve mermada negativamente dado que los estudiantes de educación superior y profesionales de hoy en día carecen de las habilidades necesarias para obtener resultados óptimos.

1.2.3. Problemas en el mundo laboral

Al igual que la consecuencia anterior, las empresas y el mercado actual son cada vez más exigentes dados los problemas que el mundo moderno propone, haciendo que los desafíos requieran de profesionales cada vez mejor formados. El Pensamiento Computacional es una de las competencias más usadas en el mundo laboral de hoy, por lo que formar profesionales que carecen de estas habilidades es una desventaja en la competencia y un problema de crecimiento a nivel país.

Entonces, en palabras simples, la propuesta que busca exponer esta memoria es la creación de un Framework que desarrolle el Pensamiento Computacional a través del uso de la Gamification. Para poder acercar el Framework a los profesores, quienes son los principales usuarios, se diseña también un programa que permita la abstracción de la parte más teórica, permitiendo así que los profesores no deban aprenderse todas las reglas de memoria, pues es el programa quién los va guiando.

Los objetivos que se buscan cumplir al desarrollar esta memoria son los siguientes.

1.3. Objetivos

1.3.1. Objetivo principal

Diseñar un Framework que permita trabajar el Pensamiento Computacional en colegios chilenos utilizando Gamification

1.3.2. Objetivos específicos

- Generar una herramienta que facilite a los profesores diseñar actividades que utilicen el Trabajo Colaborativo
- Buscar la sinergia entre los conceptos del Pensamiento Computacional y los elementos

de Gamification que sirven para potenciar y trabajar adecuadamente esos conceptos

- Habilitar un instrumento que permita al profesor contrastar su diseño con la percepción de los estudiantes de este mismo

Finalmente, se resume el problema, sus causas, consecuencias y objetivos de la memoria en la Figura 1.1.

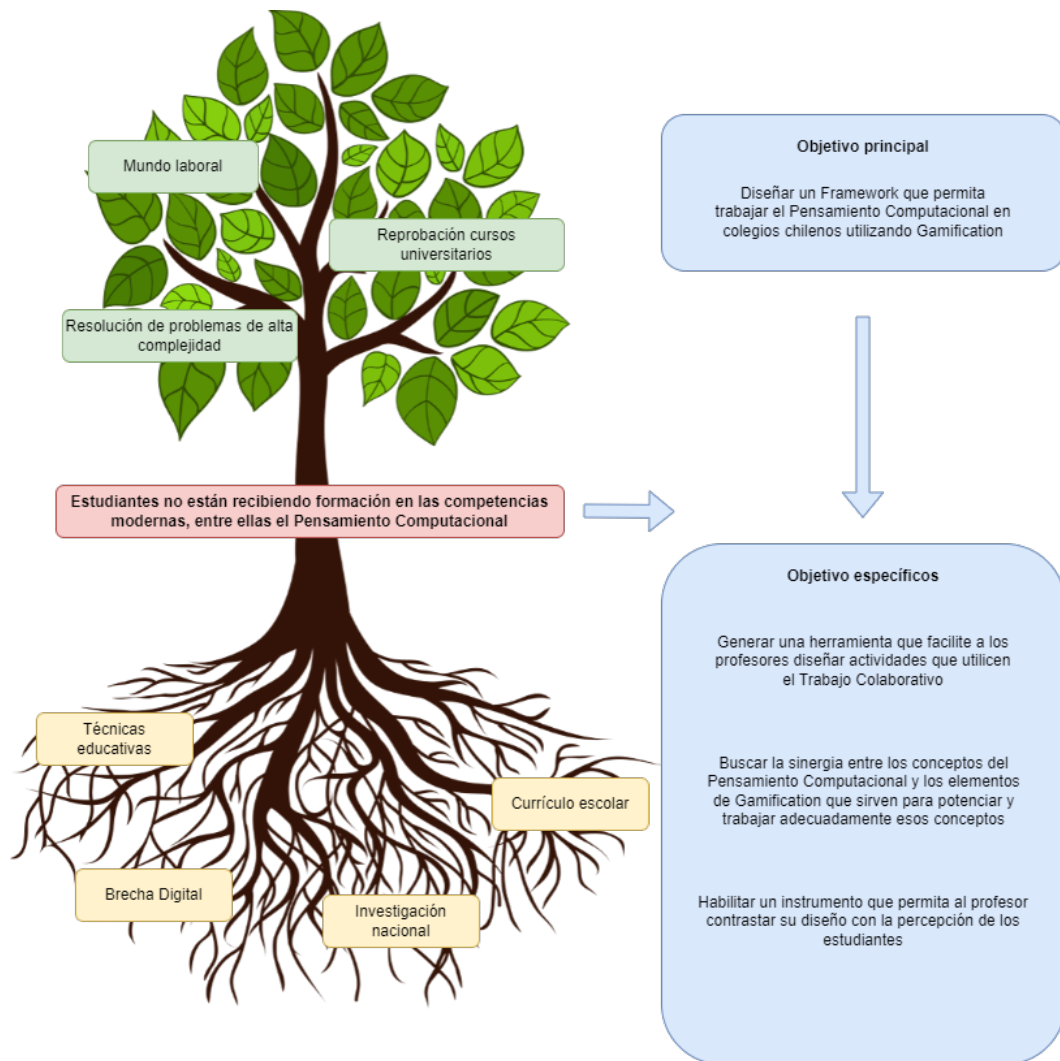


Figura 1.1: Árbol del problema.

Fuente dibujo:

<https://view.genial.ly/60cd23463adac80cfe6f5cba/interactive-content-arbol-de-problemas>

Fuente representación: Elaboración propia.

Capítulo 2

Marco Conceptual

Antes de definir los conceptos importantes que tratan este trabajo de memoria, se vuelve fundamental señalar el orden de jerarquía en que un componente engloba a otros más pequeños. En primer lugar, es importante entender qué es el Pensamiento Computacional y cuáles son sus atributos más importantes, además de sus respectivas definiciones.

Luego, corresponde identificar el concepto de Gamification. Con el tiempo, este ha recibido distintas definiciones, algunas muy amplias y otras muy específicas, pero existe una en un punto medio que abarca de sobremanera los alcances de este trabajo de memoria. Bajo el alero de Gamification, también resulta necesario agregar lo que es el concepto de Game Design y los componentes que se utilizan para diseñar juegos y videojuegos, pues son esenciales para las actividades lúdicas que se diseñan en este trabajo.

Finalmente, se debe hacer una revisión sobre lo que es y significa un Framework, esto con el objetivo de comprender cómo es que el set de actividades lúdicas que se diseñan en este trabajo servirán para que el educador tenga una base para construir otras actividades, siempre manteniendo como núcleo central las componentes del Pensamiento Computacional.

2.1. Pensamiento Computacional

La primera definición que se debe revisar es la que dio la propia autora del concepto, Jeannette Wing, el año 2006. Según ella, el Pensamiento Computacional se define como “la habilidad de dar respuesta a problemas haciendo uso de herramientas computacionales” [30]. El problema de esta definición es que queda algo ambigua. ¿Qué se considera como herramienta computacional? ó ¿Qué clase de problemas se buscan resolver? Por esto mismo, utilizando como base aquella definición, la autora vuelve a proponer una contextualización más específica, diciendo que el Pensamiento Computacional corresponde a “procesos del pensamiento humano implicados en la formulación de problemas y su resolución, aplicando un agente de procesamiento de información, como un ser humano y/o una máquina”[30]. Esta definición es fundamental, pues reconoce que el Pensamiento Computacional abarca toda clase de problemas, desde su formulación hasta la resolución. Además, y siendo lo más importante, es que desliga al Pensamiento Computacional de la tecnología, asumiendo que el ser humano es también un agente de procesamiento de información. En Chile, el Plan de Lenguajes Digitales utiliza esta definición como base para su programa [3].

El año 2010, Jeannette Wing vuelve a entregar una definición, esta vez intentando complementar a qué se refiere con la solución de un problema. Así, el Pensamiento Computacional corresponde al “proceso de pensamiento donde están involucradas la formulación de problemas y sus soluciones, donde las soluciones están representadas en una forma que se pueden llevar a cabo con eficacia por un agente de procesamiento de información” [31]. Mantiene lo importante de su definición previa (agente de procesamiento de información) pero agrega que el enfoque que se tiene con respecto a la búsqueda de soluciones es a encontrar representaciones óptimas, recalcando de forma implícita que no todas las soluciones son útiles al resolver un problema.

Con el tiempo, el Pensamiento Computacional es un concepto que se ha trabajado en distintas áreas de investigación, por lo que resulta natural que otros autores lo hayan redefinido según lo que ellos estimaron correcto para su trabajo.

Por ejemplo, Robert Panoff define Pensamiento Computacional como “enfoque que sirve para desarrollar habilidades metacognitivas en los estudiantes quienes adquieren la capacidad de descomponer los problemas en pequeñas partes para así darle solución por medio de

una computadora” [19]. Panoff da una definición bastante parecida a la segunda que entrega Wing el 2006, pero hace referencia en un computador como medio clave. Quizás para su investigación el término funciona, pero para este trabajo y, en general, para la forma en que se están integrando actividades que desarrollan Pensamiento Computacional alrededor del mundo, no es apta.

Peter Dening entrega una definición amplia, pero que encaja en lo que se busca cuando se habla de desarrollar Pensamiento Computacional. Según el autor, el Pensamiento Computacional corresponde a “un método de aproximación a la solución de problemas representando el problema como un proceso de información con solución algorítmica”[8].

Con esto, la definición de Wing del 2010 sigue siendo la más exacta cuando se quiere entender el Pensamiento Computacional como una habilidad que trasciende la tecnología (y por sobre todo el uso de computadores), al reconocer que el ser humano es también un agente de información y, como tal, es capaz de razonar y resolver problemas lógicos por sí mismo.

La última definición a revisar corresponde a la que dieron desde la Sociedad Internacional para la Tecnología en Educación (ISTE) y la Asociación de Docentes en Ciencias de la Computación (CSTA), quienes formularon una propuesta enfocada en la educación escolar. Así, el Pensamiento Computacional corresponde a “un método para resolver problemas de manera que puedan ser implementados con una computadora. Los estudiantes no son solo usuarios de la herramienta sino constructores de ellas; utilizando un conjunto de conceptos, tales como la abstracción, la recursividad y la iteración, para procesar y analizar los datos y crear artefactos reales y virtuales”[6].

Esta definición es, sin lugar a dudas, la que más se acerca a lo que se busca desarrollar en este trabajo. Al igual que Wing, reconoce que las personas (en este caso, los estudiantes) son parte del proceso de información, y, además, reconoce elementos clave que son parte del Pensamiento Computacional de forma explícita, como lo son la abstracción y la recursividad.

Como se menciona al inicio de este capítulo, existen jerarquías donde un concepto engloba diversos componentes. Siendo este el caso, resulta necesario señalar y definir estas piezas que componen al Pensamiento Computacional, donde ya se mencionaron algunas en la definición de Barr y Stephenson [6].

2.1.1. Componentes del Pensamiento Computacional

Según la fundación Google for Education [2], algunos de los conceptos relacionados al Pensamiento Computacional son:

- **Abstracción:** La abstracción es la identificación y extracción de información relevante para definir la idea principal.
- **Diseño de Algoritmos:** Es la creación de una serie ordenada de instrucciones para la solución de problemas similares, o para hacer una tarea.
- **Descomposición:** Se descomponen datos, procesos o problemas en partes más pequeñas y manejables.
- **Reconocimiento de patrones:** Es la observación de patrones, tendencias y regularidades en los datos.
- **Recolección de datos:** Recopilación de información.
- **Representación de datos:** Es representar y organizar datos en gráficos, tablas apropiadas, palabras o imágenes.
- **Automatización:** Utilizar ordenadores o máquinas para hacer las tareas repetitivas.

De este listado, para trabajar correctamente esta memoria de título, se consideran los cuatro primeros componentes: **abstracción, diseño de algoritmos, descomposición y reconocimiento de patrones.**

Para recapitular todas las definiciones conceptuales que se han mencionado hasta ahora, se presentan la Tablas 2.1 y 2.2 que muestra el acercamiento que tienen los autores con los conceptos y componentes más importantes (para este trabajo) con respecto al Pensamiento Computacional.

	Resolución de Problemas	Proceso de Pensamiento	Computación/Tecnología	Aplicación en diversas áreas
J. Wing 1 (2006)	X		X	
J. Wing 2 (2006)	X	X		
J. Wing (2010)	X	X	X	X
R. Panoff	X	X	X	
P. Dening	X	X		
ISTE/CSTA	X	X	X	

Tabla 2.1: Resumen de definiciones y conceptos.

Fuente Tabla: Elaboración propia.

	Abstracción	Diseño de Algoritmos	Descomposición	Reconocimiento de Patrones
J. Wing 1 (2006)				
J. Wing 2 (2006)				
J. Wing (2010)				
R. Panoff			X	
P. Dening		X		
ISTE/CSTA	X	X	X	X

Tabla 2.2: Resumen de definiciones y componentes.

Fuente Tabla: Elaboración propia.

Con este resumen, se puede concluir que la definición de la ISTE/CSTA es la que más interesa para este trabajo, pues contiene casi todos los elementos más importantes: los componentes del Pensamiento Computacional, el PC como proceso de pensamiento y resolución de problemas, el uso de tecnología y el reconocimiento del ser humano como agente de procesamiento de información. Lo único que queda sin mención es la aplicación del PC para distintas áreas, pero dado que esta definición no tiene ninguna incongruencia con la definición de Wing del 2010, entonces se asume que este concepto es plausible dentro de la definición y, con esto, decanta en una definición completa para guiar este trabajo.

2.2. Gamification

La *gamification* -o gamificación en español- es un concepto que se ha puesto de moda en los últimos años, esto debido a su utilidad en distintas áreas tanto del quehacer diario

como en otras más específicas. El comercio, la educación, los bancos y hasta las actividades deportivas están adoptando este concepto para mejorar la experiencia de sus usuarios. Pero, dada su versatilidad, esto lleva a pensar que debe ser un concepto complejo y/o diverso, así que se hace necesario hacer una revisión de los distintos autores que han entregado una definición desde su punto de vista para poder analizar las similitudes y diferencias, y así establecer una definición que se acerque a lo que se busca para guiar este trabajo.

La primera definición a mencionar corresponde a la de Nick Pelling quién, por el año 2002, definió el concepto para referirse exclusivamente a un giro que debía tener el diseño de interfaces de usuario. Así, Pelling dice que el concepto de Gamification corresponde a “aplicar diseño de interfaces de usuario acelerado, como las que se usan en juegos, para hacer las transacciones electrónicas más rápidas y disfrutables”[20]. Obviamente, esta definición está absolutamente alejada del uso actual que se le da al término, aunque no por eso carece de importancia. Según el mismo autor, esta definición sirvió para que empresas como Apple sean el gigante electrónico que son hoy en día, pues la implementación de esta definición cambió el paradigma de cómo los desarrolladores diseñaban sus productos.

Actualmente, diversos autores han intentado pulir una definición que logre abarcar diversas áreas, esto sumado a la integración de otros elementos asociados a los juegos y videojuegos.

Por ejemplo, Cunningham y Zichermann definieron Gamification como el “uso del pensamiento lúdico y las mecánicas de juego para fomentar la participación de los usuarios y resolver problemas”[33]. Esta definición es interesante, pues propone que la idea de integrar elementos lúdicos tienen un impacto positivo en la resolución de problemas por parte de los usuarios. Pero cuenta con una falencia, y es que deja abierta la puerta al uso indiscriminado de elementos lúdicos. El problema asociado a esto es que es de suprema importancia utilizar elementos lúdicos **manteniendo la formalidad de la actividad**, y no ludificar la actividad hasta que se transforma mera y llanamente en un juego. Por otro lado, Deterding et al. buscaron una definición para el uso de gamification en actividades formales (como la educación, por ejemplo), definiendo como “uso de elementos de diseño de juegos en contextos no lúdicos”[9]. Esta definición no es tan completa como la anterior, pero sirve para ilustrar que es plausible la integración de elementos lúdicos en actividades que son, ciertamente, de

carácter no lúdico.

Por sí solas, ambas definiciones fallan en entregar una visión que pueda aportar al enfoque de este trabajo, pero juntas hacen una combinación perfecta. Cunningham y Zichermann aportan con una pieza clave: **resolver problemas**. A través de esto es cómo se puede conectar la Gamification con el Pensamiento Computacional. Por otro lado, Deterding et al. dieron con el significado que hoy en día más se utiliza, que hace referencia a las actividades no lúdicas, como la educación o el comercio.

Tal como señala Diego Vergara y Ana Isabel Gómez, la gamification como estrategia se insertó en el comercio desde hace ya varios años, aunque el concepto “gamification” como tal sea desconocido por la mayoría. Por ejemplo, hoy en día las casas comerciales y supermercados ofrecen acumular puntos al comprar productos, pudiendo estos ser intercambiados por otro producto al juntar una cierta cantidad [32]. Lo mismo ofrecen las aerolíneas al viajar y pagar con determinadas tarjetas de crédito, donde se acumulan puntos en forma de millas y luego se intercambian por pasajes hacia algún destino que el usuario seleccione. Lo que se utiliza aquí es el concepto de “puntaje”, un elemento que aparece al diseñar casi cualquier juego o videojuego. Otros elementos se verán en un subcapítulo más adelante, pues resulta interesante revisar cuáles elementos sirven y cuáles no a la hora de utilizar gamification evitando ludificar completamente una actividad.

Dado que este trabajo se enfoca en la gamification como método para introducir la enseñanza y trabajo del Pensamiento Computacional en colegios, lo primero que hay que revisar, luego de haber definido el concepto, es la relación entre la gamification y la educación.

2.2.1. Gamification y Educación

Con todo lo que se ha escrito hasta ahora, y sabiendo que por definición la Gamification corresponde a la inclusión de elementos lúdicos en actividades no lúdicas, es que aparece la siguiente pregunta: ¿Es posible integrar la Gamification en la educación?

Según Joey Lee y Jessica Hammer, la estructura básica que se utiliza en la educación ya hace uso de elementos de juegos [17]. Por ejemplo, en el colegio un estudiante (jugador) debe asistir durante el transcurso de un año a un curso (nivel) y para completarlo debe obtener, al menos, el mínimo de notas (puntaje) para pasar al siguiente curso (siguiente nivel), hasta

terminar su estadía en el colegio (terminar el juego).

Aunque el ejemplo resulta ser una abstracción que no es tan notoria, sirve para demostrar que existe, de forma implícita, una conexión más que estable entre la Gamification y la educación. Y sobre esta conexión, resulta interesante realizar dos interrogantes: ¿Cómo implementarlo? y ¿Para qué hacerlo?

La primera pregunta, según Lee y Hammer, se responde revisando las técnicas que existen para gamificar actividades. Entre ellas, se deben considerar las que responden a tres estímulos esenciales para potenciar la educación:

- **Cognitivo:** Sirve para transformar la **perspectiva** del estudiante. A este último, generalmente, se le dice qué hacer mientras que no se le explica (y tampoco lo entiende por sí mismo) los beneficios de lo que está haciendo en el largo plazo. Así, la gamification busca explotar la idea de que el beneficio a corto plazo motiva al estudiante. En un videojuego, un jugador se hace la pregunta: ¿Si ya dominé este nivel, qué debo hacer después para dominar el juego? Extrapolando la idea, lo cognitivo busca que el estudiante se haga la misma pregunta al dominar una materia, pues esto le sirve para “dominar” el colegio.
- **Emocional:** Los juegos evocan emociones, y lo logran igual que otros medios audiovisuales, como por ejemplo el cine. Según Nicole Lazzaro, las emociones que evocan los juegos son poderosas, y van desde la curiosidad hasta la frustración y la alegría [16]. Dicho esto, en la educación escolar ocurre un fenómeno bien particular asociado al fracaso. El estudiante, en general, tiene pocas chances de practicar y, además, arriesga mucho cuando le toca rendir alguna evaluación. Esto hace que el estudiante, en vez de sentir alguna clase de ansiedad por el desafío que tiene que enfrentar, sienta miedo, frustración e incluso agobio.

Entonces, lo que busca el enfoque emocional en la técnica de gamification no es en crear instancias donde el estudiante deje de fracasar, sino instancias donde el estudiante aprenda que el **fracaso es parte del aprendizaje**, y que al igual que en un videojuego, donde se pueden volver a reintentar los niveles, en el colegio también se puede fallar (de forma positiva) y eso no significa el fin del mundo. El estudiante comienza a ver el fracaso como una oportunidad, empujándolo a mejorar en lo que sea

que esté fallando.

Para lograrlo, la estrategia debe converger en instancias donde el riesgo sea bajo, las fallas sean controladas y cíclicas (al igual que un proceso iterable), y dónde se premie el esfuerzo por sobre el dominio de la actividad.

- **Social:** En los juegos y videojuegos, los jugadores pueden tomar el rol de una infinidad de papeles disponibles. Puede ser un fontanero vestido de rojo y azul, una copia exacta del agente 007 ó un mago que mata dragones con hechizos de fuego. En el colegio, ocurre un contexto similar, pero con roles reales y más humanos (aunque, obviamente, a todos nos gustaría ser James Bond), y esto trae problemas para aquellos estudiantes que no les resulta fácil sociabilizar. Esto último es muy importante, dado que el aprendizaje está directamente relacionado con el ambiente donde se estudia.

Así es como la técnica de gamification se debe enfocar en crear roles que, aunque no suenen tan convencionales, ayuden a integrar a estos estudiantes y hacer que se sientan parte teniendo una identidad. Una técnica con un diseño robusto permitiría a estos estudiantes no solo obtener reconocimiento explícito de parte de los profesores, sino también de parte de sus compañeros, creando así un ambiente donde todos los estudiantes logren **encontrar su espacio a través de una identidad**.

Luego queda por responder a la siguiente pregunta y, según los autores, el para qué tiene tanto riesgos como beneficios que deben ser considerados. Dentro de los beneficios, se dice que mientras el diseño de las actividades donde se utiliza gamification sea robusto, este puede ayudar a los estudiantes a entusiasmarse con la enseñanza de una forma que las actividades “no lúdicas” no pueden. Además, impulsa a los estudiantes a dar el 100 %, dado que ven que sus esfuerzos traen recompensas en el corto plazo. También ayuda a los profesores, pues les abre un camino de experimentación dentro de la enseñanza que los métodos tradicionales no consideran.

Por otro lado, esto también tiene riesgos y desafíos que podrían terminar en una actividad que no sólo no aporta a la educación, sino que resta. Una actividad mal diseñada tendría efectos negativos tanto en profesores como estudiantes. En el caso de los profesores, estos podrían verse desgastados, tanto energéticamente (en caso que la actividad sea demandante de tiempo o esfuerzos físicos) como financieramente (si es que la actividad requiere de elementos

que el profesor debe costear por su cuenta). En el caso de los estudiantes, una actividad mal diseñada podría terminar en disminuir el entusiasmo, y mal acostumbrar al estudiante a que toda actividad exitosa tiene una recompensa en el corto plazo.

Es imperativo comprender que la Gamification en educación no es una panacea, y no va a solucionar todos los problemas que se tienen en la educación actual. Pero es una buena técnica que, cuando está bien implementada, produce resultados muy positivos en el aprendizaje de los estudiantes.

Antes de revisar cómo se conecta este concepto con el Pensamiento Computacional, hace falta definir los elementos que componen a la Gamification. Al ser un concepto general que se puede adaptar a diversas situaciones, considerarlo en esa forma para este trabajo podría ser contraproducente, pues, aquí la Gamification es una herramienta de apoyo y no el objetivo principal de la propuesta.

Werbach [29] definió tres conceptos clave para la Gamification, y son diversos los estudios existentes que se basan en su trabajo para integrar Gamification al **aula de clases**.

Los conceptos son:

- **Dinámicas:** Estructura implícita del juego, como lo son la narración, las emociones o la progresión.
- **Mecánicas:** Procesos que provocan el desarrollo del juego. Por ejemplo, el concepto de competición, retos, y recompensa.
- **Componentes:** Implementación de las Dinámicas y Mecánicas. Avatares, puntaje, ranking.

Otros autores coinciden de una u otra forma con Werbach, pero dan distintas definiciones a los elementos o discrepan en la pertenencia de los componentes en una u otra categoría.

Por ejemplo, para Rosana Rosas [22] la Gamification en el aula se compone de la siguiente manera:

1. Mecánicas

Corresponden a las reglas del juego, o en otras palabras, a lo que define la experiencia.
Algunas mecánicas son:

- Mundo
- Avatar
- Reglas
- Niveles
- Recompensas
- Progreso
- Ranking
- Equipo

2. Dinámicas

Las dinámicas corresponden a las acciones que surgen cuando los jugadores utilizan las mecánicas.

Entre ellas:

- Aprendizaje
- Retos
- Socialización
- Emociones
- Narrativa

3. Estética

La estética hace referencia al diseño que hace emerger emociones, y es la puerta de entrada al mundo gamificado. Está relacionado con el diseño gráfico y la experiencia de usuario.

La mayoría de estos componentes se definen y explican en el apartado 2.3., pues los conceptos vienen desde el Game Design y los juegos en sí.

2.2.2. Gamification, Educación y Pensamiento Computacional

Una forma de verificar que la Gamification y la Educación efectivamente tienen una conexión que puede funcionar -y que se puede trabajar el Pensamiento Computacional- es realizar experimentos en situaciones reales con personales reales. Esto ya se ha hecho, y para mostrar los resultados, resulta necesario revisar la bibliografía.

Diego Vergara y José María Mezquita diseñaron un estudio donde sometieron a cuatro generaciones de estudiantes de secundaria (equivalente a la educación media en Chile) a utilizar un videojuego llamado TRIVIREC, el cuál se basa en el popular juego Trivia [27]. El juego consiste en que los estudiantes deben responder preguntas de distintas categorías, siendo ganador quién responda correctamente a una pregunta de cada categoría. Para ajustarlo al experimento, los autores consideraron que las preguntas debían ser de una materia en específico, por lo que decidieron usar el curso de “Economía de la Empresa”, haciendo que las categorías del juego fuesen los contenidos macro del curso. Además, para reforzar el trabajo colaborativo entre los estudiantes, los autores decidieron que el juego se jugaría en equipos de tres integrantes, y estos integrantes serían seleccionados por los mismos profesores. La razón de esto fue para poder conformar equipos con un estudiante con calificaciones altas, uno con calificaciones medias y uno que estuviese directamente reprobando la asignatura. También incluyeron como regla que, cuando un alumno responde una pregunta, debe esperar a que sus dos compañeros participen para poder volver a intentar responder otra pregunta. En caso que el estudiante tenga dudas al momento de responder una pregunta, puede pedir 1 minuto de tiempo cronometrado para conversar con sus dos compañeros de equipo, y así poder ajustar la respuesta.

Al finalizar el juego, los profesores obtienen métricas a través del software, lo que les permite realizar una clase magistral sobre las categorías donde existen más fallos en cada sesión de juego. Al respecto de esta actividad, las cuatro generaciones de estudiantes respondieron una encuesta al finalizar la última sesión, obteniéndose los siguientes resultados:

- 86 % de los alumnos NO habían tenido una actividad así
- 100 % de los alumnos encontraron que la experiencia fue MOTIVADORA
- Un 57 % de los alumnos consideró que fue una buena herramienta para repasar contenidos para la prueba de selección
- 100 % de los alumnos indicaron que les gustó la actividad
- Un 71 % indicó que le gustó que la actividad fuese una mezcla de juego-clase teórica
- Un 0 % indicó que no le gustaría que la actividad fuese un juego en su totalidad
- Del 71 % que indicó que le gustó que fuese mezcla, el 67 % prefiere que en esta actividad tenga más peso la clase teórica (tradicional) por sobre el juego.

De este estudio se desprenden varias cosas importantes. En primer lugar, muestra que, efectivamente, se puede integrar Gamification en la educación (al menos en lo que corresponde a educación media). Además, la respuesta de los alumnos fue satisfactoria en cuanto a lo que se quería testear en este estudio. En general, a la amplia mayoría le gustó la actividad y una mayoría más pequeña encontró que la actividad sirvió para repasar contenidos. También mostró que ningún estudiante prefiere que la actividad sea sólo lúdica, reafirmando la tesis de que la Gamification funciona cuando mantiene el carácter serio de las actividades.

En el caso del Pensamiento Computacional, esta habilidad no fue trabajada -al menos de forma explícita- en la actividad, pero sí el trabajo colaborativo, el cual es importante para este trabajo, pues está considerado de manera opcional para el diseño de actividades.

Otro estudio realizado por Vergara, Mezquita y Gómez consistió en someter a dos cursos, uno de preparatoria y otro de bachillerato, a una actividad donde los estudiantes debían responder preguntas utilizando el software Quizizz [28]. Con este software, los profesores

tenían control total de la actividad a través de feedback instantáneo, además de tener la posibilidad de mostrar distintas preguntas a los estudiantes para evitar posible copia entre ellos.

Dividieron la actividad en 3 fases, las cuales quedaron de la siguiente forma:

- **Fase 1 - Individual:** El estudiante debe contestar 20 preguntas
- **Fase 2 - Grupal:** Grupos de 3 estudiantes deben formular preguntas y sus respectivas respuestas. Estas preguntas y respuestas son enviadas a los profesores
- **Fase 3 - Grupal:** Los grupos de estudiantes deben competir entre ellos, y gana quién obtiene la mayor cantidad de aciertos en el menor tiempo posible al intentar responder 20 preguntas. Estas 20 preguntas salen a elección de los profesores, y son seleccionadas del grupo de preguntas-respuestas que los mismos estudiantes formularon en la Fase 2.

Los estudiantes respondieron una encuesta antes de realizar la actividad, y luego al finalizar la actividad.

Los resultados de este estudio son más discretos con respecto al anterior, ya que los profesores contaron con una menor cantidad de métricas. Lo importante a señalar es que los estudiantes encontraron que la actividad era amena y entretenida, y sintieron que efectivamente se reforzó el trabajo colaborativo, sobre todo en la fase 3. Con respecto a esto mismo, también se notó una diferencia entre los dos cursos, pues en el caso de los estudiantes de bachillerato, el trabajo colaborativo no era tan bueno y además fue decayendo a medida que pasaba el tiempo en la actividad. Otro tema interesante es que durante la fase 2, los grupos de estudiantes de preparatoria se sentían muy preparados para formular las preguntas, mientras que los de bachillerato eran más precavidos.

Para este trabajo en particular, resulta importante destacar que de estos resultados se puede hacer una observación de algo que hasta ahora no se había considerado: al parecer, la edad de los participantes influye en el desempeño en la actividad, y por ende, es un elemento importante a tener en cuenta al momento de realizar el diseño de la actividad.

También es importante destacar que, al igual que el estudio anterior, se desarrolló el trabajo

colaborativo en dos fases del experimento, impulsando el apoyo mutuo para dos tareas distintas. Con respecto al Pensamiento Computacional, la fase 2 es la más importante de todas, puesto que para la formulación de preguntas (con su respectiva respuesta) resulta necesario utilizar tanto de forma implícita como explícita los componentes del PC, como lo son la abstracción y la resolución de problemas.

Finalmente, otro estudio realizado por Ros et al. consistió en realizar una actividad donde se pone en práctica el Pensamiento Computacional a través de la resolución de problemas [11]. En este caso, dividieron a un curso de estudiantes en dos grupos, los cuales tenían que realizar la misma tarea pero con distintas especificaciones. El primer grupo debía usar únicamente la plataforma Scratch para la resolución de los problemas, mientras que el otro grupo, además de Scratch, tenían una conexión a un Arduino a través del escudo Echidna. Como herramienta de Gamification, utilizaron la plataforma Kahoot! para hacer la evaluación del desarrollo de la actividad con respecto al Pensamiento Computacional. Para esto, se realizó un pre-test (antes de comenzar a resolver problemas) y un post-test una vez las sesiones de resolución de problemas habían concluido.

Los resultados obtenidos en Kahoot! indicaron que casi la totalidad del grupo que trabajó sólo con Scratch bajó el puntaje obtenido en el post-test con respecto al obtenido en el pre-test. En el caso del grupo con el Arduino, la mayoría de los integrantes lograron mejorar su puntaje obtenido en el post-test con respecto al pre-test.

De este estudio se desprenden dos ideas interesantes para tener en cuenta en el desarrollo de este trabajo. Primero, y lo más evidente, es la perspectiva de utilizar la Gamification como un elemento secundario. En este caso, se utiliza como instrumento de medición. La segunda, y relacionada con el experimento mismo, es que al parecer el desarrollo de este tipo de actividades tienen un impacto mayor en los estudiantes que cuentan con algún elemento que conecte la tarea a realizar con la realidad. En este caso, los estudiantes que tenían el Arduino (mundo real) tuvieron mejores resultados que los estudiantes que sólo trabajaron con Scratch (mundo virtual).

2.3. Game Design

A lo largo de este marco conceptual, se ha definido lo que es la Gamification y se ha mostrado cómo funciona y para qué sirve. Aunque, tal como se mencionó en la sección 2.2., resulta importante resolver el vacío que deja la definición sobre los llamados “elementos de juegos y videojuegos”, dado que existen en una gran cantidad y muchos de ellos sirven para ludificar las actividades de sobremano, complicando así el desarrollo de las actividades donde se quiere implementar la Gamification. Es por esto que esta sección, titulada *Game Design* -o diseño de juegos en español-, se encarga de englobar y explicar los conceptos fundamentales que se extraen del diseño de juegos y videojuegos, a través de la amplia bibliografía que existe tanto de diseñadores que trabajan en la industria de los juegos como de quienes se dedican al estudio de los juegos desde la academia.

Antes de comenzar con los elementos, se debe explicar y definir lo que es el Game Design. Tal como el nombre lo dice, se trata de diseñar juegos utilizando elementos lúdicos de forma coherente, buscando encontrar un equilibrio en la obra que se quiere crear.

Una definición directamente desde la literatura, y probablemente la más utilizada, corresponde a la de Salen y Zimmerman, quienes definen el Game Design como “proceso en el cual un diseñador de juegos crea un juego para ser confrontado por un jugador, desde donde emerge el juego (verbo) significativo”[23].

Aunque el objetivo de este trabajo es utilizar Gamification y no diseñar juegos, es importante tener en cuenta cómo es que el diseño influye directamente en el uso de los elementos lúdicos en otras actividades.

2.3.1. El juego

Un juego es, en su definición más simple, una actividad lúdica compuesta por diversos elementos. Pero esta definición queda corta a la hora de revisar los juegos y videojuegos de la actualidad, pues un juego es una cosa mucho más compleja. Para comprender mejor el concepto de juego como tal, resulta esencial revisar las definiciones de la literatura.

En primer lugar, Johan Huizinga entrega una definición de “jugar” en su trabajo *Homo*

Ludens. Este propone como tesis que el ser humano (*homo sapiens sapiens*), tal cuál es ahora, evolucionó hacia el *homo ludens* a través del juego. Básicamente, que la actividad lúdica fue parte de la evolución del ser humano en el transcurso de su vida. Según Huizinga, **jugar** es “una actividad libre que conscientemente se expone fuera de la *vida ordinaria*, dado su contexto de no-seriedad, pero que al mismo tiempo absorbe al jugador intensamente. Es una actividad conectada con un interés no material, y dónde no se puede obtener ganancia. Funciona en base a sus propios límites, su propio tiempo y espacio y sus reglas definidas.” [12]. Esta definición es una buena aproximación para empezar con las definiciones de lo que significa “juego”, ya que aunque no es explícita para el concepto, sí determina diversos elementos que pertenecen a lo que es un juego.

La definición que se utiliza para este trabajo también corresponde a la de Salen y Zimmerman, la cual corresponde a “un juego es un sistema en el cual los jugadores se enfrentan en un conflicto artificial, definido por reglas, que resulta en un desenlace cuantificable.” [23]. Esta definición es extremadamente completa, y es posible separar los conceptos que utiliza y definirlos por sí mismos. En términos de la Gamification, esta definición contiene lo esencial para el desarrollo de las técnicas de creación de actividades con componentes lúdicos. Las palabras clave de esta definición son sistema, jugador, artificial, conflicto, reglas y desenlace cuantificable.

- **Sistema:** Un grupo de elementos interactivos, interrelaciones o interdependientes que forman un todo completo. Según Salen y Zimmerman, un juego es de forma intrínseca y esencial, un sistema.
- **Jugador:** Participantes que interactúan con el sistema (juego). En el contexto de este trabajo, se refiere a los estudiantes.
- **Artificial:** Hace referencia a los límites que separan el tiempo y el espacio entre la vida real y el contexto donde ocurre el juego.
- **Conflicto:** El conflicto es un elemento central de un juego. Se refiere a una competencia de poder, aunque esta competencia no necesariamente hace referencia a una rivalidad. También puede tratar de cooperación.

- **Reglas:** Elemento crucial de un juego. Es la estructura que da pie al nacimiento del “jugar”, y delimita lo que los jugadores pueden o no pueden hacer. Los tipos de reglas se definen en la sección 2.3.2.
- **Desenlace Cuantificable:** Este es un elemento diferenciador de las actividades no lúdicas. Es un indicador (como el puntaje, por ejemplo) de un objetivo o un fin. Generalmente, cuando el juego termina, el jugador gana o pierde.

2.3.2. Las Reglas

Tal como se menciona en la sección 2.3.1., las reglas son un elemento fundamental (o de base) en un juego. Es lo que delimita qué es lo que es el juego, y también separa una actividad sin un fin específico de una actividad lúdica. El ejemplo más clásico es el de los hombres boteando una pelota. En un caso, un hombre botea una pelota sin intención alguna. Si la pelota se le escapa, va a recogerla y la sigue boteando, hasta que considere que ya no necesita hacerlo más. En el otro caso, un hombre botea una pelota, pero teniendo una serie de consideraciones. Cada vez que la pelota pegue en el suelo y vuelva a su mano de forma satisfactoria, subirá su contador en 1 punto. Si la pelota da dos rebotes antes de tomarla con la mano, entonces reduce su contador en 1 punto. Si la pelota se le escapa, entonces su contador vuelve a 0, y debe comenzar desde ahí. El puntaje mínimo es 0, y el máximo es infinito.

Esta serie de consideraciones no son nada más que las reglas del juego, y son las que hacen que la actividad que está realizando el hombre del caso dos sea un juego, mientras que la del caso uno no lo sea.

Según Salen y Zimmerman, las reglas se dividen en tres subgrupos importantes [23]:

- **Reglas Operacionales:** Estas son las reglas explícitas del juego. Es la guía de reglas que viene en un manual dentro de la caja de un juego de mesa. Por ejemplo, las reglas que se mencionaron en el juego de botear la pelota son, precisamente, reglas operacionales -o explícitas-.

- **Reglas Constitutivas:** Las reglas constitutivas son las que van por debajo de los lineamientos, pero que no son implícitas. Las más clásicas son reglas lógicas y matemáticas, y un buen ejemplo de dónde existen estas reglas es en el juego del gato (tic-tac-toe en inglés).
- **Reglas Implícitas:** Tal como el nombre lo indica, son las reglas que no están escritas en el manual del juego. Un ejemplo, en el caso del juego de botear la pelota, sería que está prohibido hacer trampa sumando puntos extra al contador.

Además, las reglas tienen ciertas características que son, en general, compartidas en cualquier diseño de juegos. Nuevamente citando a Salen y Zimmerman, las características más importantes son [23]:

- Las reglas limitan las acciones de los jugadores
- Las reglas no pueden ser ambiguas
- Las reglas son compartidas por todos los jugadores por igual
- Las reglas son fijas, no cambian durante el transcurso de la actividad
- Las reglas están hechas para ser obedecidas
- Las reglas son repetibles

2.3.3. Los Elementos

Esta sección se encarga de mostrar los grupos grandes que componen a un juego. En otras palabras, son las partes que, de forma independiente, componen el todo completo (sistema).

Para Jesse Schell, los elementos que hacen a un juego ser lo que es son cuatro [24], y estos son:

- **Mecánicas:** Corresponden a los procedimientos y reglas del juego. Describen el objetivo del juego y cómo los jugadores pueden y no pueden intentar alcanzar ese objetivo, y lo que ocurre si lo intentan.

- **Historia:** No se refiere necesariamente a una historia con personajes o diálogos, sino a la secuencia de eventos que desarrollan el juego.
- **Estética:** La estética se refiere a cómo el juego se ve, cómo suena, cómo se siente. Este elemento influye directamente con la experiencia del jugador, pues es lo que le entrega feedback asociado a sus sentidos.
- **Tecnología:** Se refiere a cualquier material necesario para la existencia del juego. Desde un computador de alta gama, hasta un papel y lápiz. En otras palabras, es el medio para que el juego exista.

2.3.4. Las Mecánicas

Las mecánicas son el núcleo del juego. Tal como están definidas en la sección 2.3.3., son las encargadas que el juego sea de una forma determinada, y que se juegue de esa forma y no de otra. Las mecánicas engloban a las reglas como un concepto, junto a otros más. Nuevamente, según Jesse Schell [24], estas son las mecánicas que componen un juego:

- **Espacio:** Hace referencia al espacio físico o imaginario-virtual donde ocurre el juego. Por ejemplo, en el caso del tic-tac-toe, sería el tablero de 3x3.
- **Tiempo:** De forma literal, se refiere al flujo del tiempo. En el mundo real no se puede controlar el tiempo y, dada nuestra percepción, sólo se viaja hacia adelante, tal como el flujo de un río. En un juego, el tiempo es relativo y ajustable a un contexto, y se puede manifestar de diversas formas. Por ejemplo, el tiempo se puede dejar de forma continua, o discretizar si se vuelve necesario. O algo más simple y menos teórico, se puede revisar el tiempo en base a un cronómetro para medir la demora en dar una vuelta a la pista en un juego de carreras de autos.
- **Objetos, Atributos y Estados:** Los objetos son los elementos que llenan el mundo compuesto por espacio y tiempo. Además, estos objetos tienen atributos, y están en un estado específico que, según las reglas del mundo, puede cambiar o no. Por ejemplo, el

objeto puede ser un lápiz, su atributo es ser largo y de plástico, y su estado es estático acostado sobre una hoja de papel.

- **Acciones:** Se refiere al verbo “hacer” dentro del juego. Una acción puede ser un movimiento de una pieza de ajedrez, o marcar una casilla en el tic-tac-toe. Las acciones se pueden complejizar a medida que las reglas del juego se van volviendo más complejas.
- **Reglas:** Ya se encuentran definidas y descritas en la sección 2.3.2.
- **Habilidad:** La habilidad es importante porque cambia el foco. La habilidad no se refiere a algo intrínseco del juego, sino del jugador. Todo juego requiere de cierta habilidad para ser jugado, y esta habilidad puede ser física, mental o social.
- **Suerte:** La suerte -o chance, en inglés- hace referencia a que la diversión del juego tiene asociado, generalmente, un poco de azar. La suerte genera imprevistos y sorpresas en el jugador, lo que le agrega al juego incertidumbre.

2.3.5. Game Design, Gamification y Educación

Esta sección está para capitalizar todo lo expuesto en el apartado de Game Design, y poder hacer la relación directa con el uso de Gamification en la Educación. Es fácil darse cuenta que, en general, la definición del juego y la identificación de sus componentes no resultan del todo trivial, y que por algo existe una diversa bibliografía al respecto (aunque en este trabajo se utilizó mayoritariamente el libro de Salen y Zimmerman, que es algo así como la biblia del Game Design).

Como ya se estableció en las secciones anteriores, la Gamification sí funciona al ser integrada en una actividad no lúdica como lo es la Educación, pero hay que resguardar que la Gamification no ludifique la actividad, sino que la apoye al integrar los elementos lúdicos para darle una perspectiva más fresca y entusiasme a los estudiantes a aprender y trabajar. Para esto, entonces se vuelve necesario revisar Las Mecánicas pero desde el punto de vista de la Gamification, para así tener un set de componentes enfocados en diseñar bien las actividades.

El **Espacio** es una mecánica que da mucha libertad para trabajar con Gamification, y su definición no debería ser alterada. Al igual que los ejemplos, a través de la Gamification el espacio puede seguir siendo un lugar físico (una hoja de papel, la sala de clases, el pizarrón, etc.) ó un lugar virtual (que se accede a través de un computador, tablet o smartphone).

El **Tiempo** carece de importancia dentro de una sala de clases, por lo que al menos a simple vista, sería un componente con el cuál no conviene trabajar al utilizar Gamification.

Los **Objetos, sus Atributos y Estados** sirven al diseñar actividades que ocurren en mundos alternativos, o en su defecto, en actividades que realizan esfuerzo físico. Esto sí es una mecánica importante para utilizar con Gamification, sobre todo en actividades diseñadas para el curso de Educación Física.

Las **Acciones** son esenciales dentro del juego. Desde el punto de vista del jugador, es el estado de interactividad que el juego le proporciona, por lo que es de suma importancia dentro de la Gamification.

Las **Reglas** definen cualquier juego o actividad (incluso no-lúdica) por lo que no tiene sentido discutir algo.

La **Habilidad**, al estar enfocada en el jugador más que en el juego, es un elemento que se considera pero del que no se saca provecho. La Gamification en Educación no busca encontrar fórmulas para dificultar el juego y filtrar a sus jugadores (estudiantes), sino que en este caso, necesita neutralizar la dificultad para tratar de llegar a todos los que están en el aula.

Finalmente, la **Suerte** es un elemento que genera incertidumbre tanto en el juego mismo como en el diseño del juego, por lo que se tiene que tener mucho cuidado al utilizar en actividades no-lúdicas. El elemento Suerte podría transformarse en la clave para ludificar completamente una actividad, perdiendo así el sentido de la Gamification.

2.4. Framework

Tal como está titulado este trabajo de memoria, la solución que se plantea más adelante es el diseño de un framework como una propuesta para enseñar Pensamiento Computacional, pero eso lleva a una pregunta bien simple: ¿Qué es un *framework*?

Una definición de diccionario (como el diccionario de Oxford) sería “la estructura de un

sistema particular”[10], pero esto es demasiado general como para definir un framework en la actualidad.

Para el alcance de este trabajo, Ralph Johnson entrega una definición acertada y seccionada en 3 partes [14].

Así, un framework es:

1. Suma de componentes y patrones.
2. Diseño reutilizable del todo o partes de un sistema que está representado como un set de clases abstractas y la interacción entre aquellas clases.
3. Esqueleto de una aplicación que es modificable por su desarrollador.

Pasando a palabras más simples esta enumeración, un framework es una plantilla de diseño que permite optimizar el trabajo de lo que se está desarrollando al enfocar los esfuerzos de los desarrolladores en elementos específicos, y este diseño se puede utilizar para distintas aplicaciones que se basan en el mismo tema.

2.4.1. Gamification Frameworks

En la actualidad, son diversos los frameworks que distintos autores han creado para integrar elementos lúdicos en actividades a través de la gamification. Como ya se mencionó en los apartados anteriores, aplicar gamification no es algo trivial, pero el diseño de las actividades tienen elementos en común, independiente del foco que se le quiera dar.

Por ejemplo, al diseñar una actividad integrando gamification para dos fines distintos, muy probablemente difieran entre ellas en el tema, en los objetivos o incluso en la escala, pero también muy probablemente tengan varios elementos en común, sobretodo los que vienen desde la gamification. Es aquí donde se vuelve necesario crear frameworks de gamification, pues el diseño que proponen estos frameworks permiten agilizar el diseño de las actividades a través de la reutilización de partes y, además, permiten tener cuidado con lo más importante: no ludificar una actividad hasta transformarla en un juego.

Para mostrar y entender cómo funcionan los frameworks de gamification, a continuación se presentan algunos ejemplos.

GAME Framework

Andrzej Marczewski, hace ya algunos años, creó un framework de gamification titulado Marczewski's Gamification Framework. Es un framework bastante completo y con muchas componentes, por lo que a través del trabajo continuo, testeo y aplicación, llegó a una simplificación de su framework original a la que llamó GAME Framework.

El GAME Framework busca establecer directrices de cómo trabajar con Gamification, y cada letra representa una acción de diseño, las cuales se definen, según el autor [18], así:

- **Gather:** Hace referencia a la palabra reunir en español, y se refiere a la obtención de información necesaria a través de preguntas, tales como ¿Qué estás gamificando? y ¿Para qué estás gamificando?.
- **Act:** Actuar en español. Se refiere a diseñar la actividad una vez la información está lista.
- **Measure:** Medir en español. Se refiere a realizar una medición continua de los resultados que obtienen los usuarios al participar de la actividad gamificada.
- **Enrich:** Enriquecer en español. Hace referencia a mejorar el trabajo realizado iterando cambios en las componentes de la actividad una vez se tienen a mano resultados de los participantes.

Octalysis Framework

Octalysis es un gamification framework creado por Yu-kai Chou, un autor que lleva ya varias décadas testeando y mejorando este framework. A diferencia del GAME Framework que concentra su estudio en el diseño, Octalysis es un framework que concentra su estudio en las componentes de Gamification. Para Chou, son 8 los componentes claves que se repiten y se utilizan comúnmente, por eso el nombre de su framework [15]. Así, el framework

está representado por un octágono donde cada vértice corresponde a un elemento clave para utilizar gamification. La Figura 2.1 muestra gráficamente la representación.

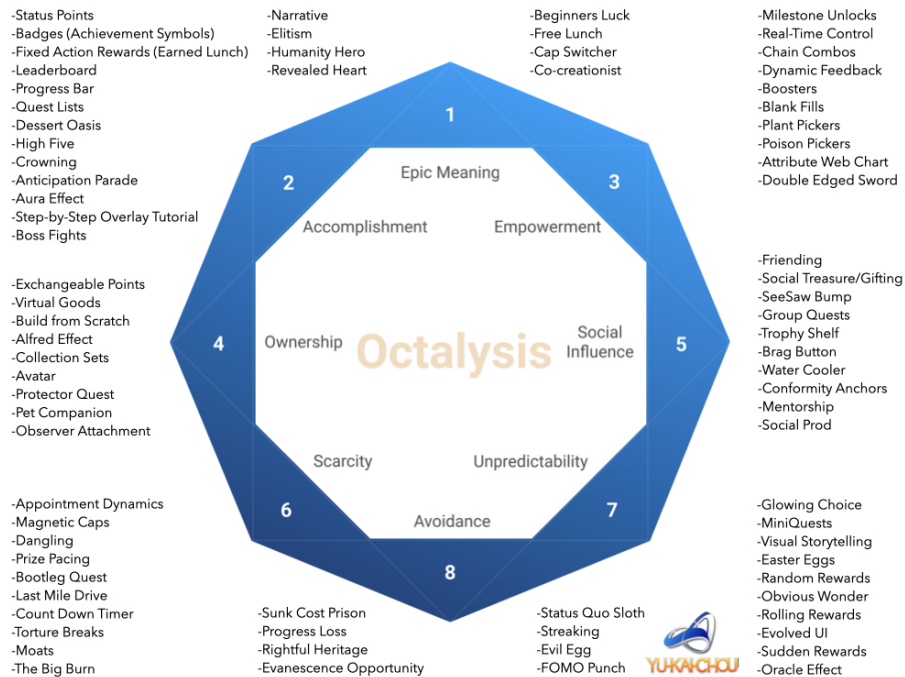


Figura 2.1: Octalysis Framework.

Fuente: <https://i0.wp.com/yukaichou.com/wp-content/uploads/2020/12/Gamification-Framework.jpeg?ssl=1>

Esto representa el nivel 1 del framework, pues la iteración durante décadas le ha permitido al autor hacer un framework multinivel que le da distintas perspectivas y asociaciones a los elementos. Aún sabiendo esto, el primer nivel es suficiente para explicar la forma en que opera este framework.

El autor posiciona los elementos de una forma estratégica, y no de forma aleatoria. Esto dado que el octágono está segmentado en izquierda versus derecha, y arriba versus abajo. La representación izquierda versus derecha nace de los lados del cerebro humano, y tal como menciona explícitamente el autor, esto es solo una representación y no ciencia real del cerebro.

Los elementos del octágono que se encuentran al lado izquierdo son **motivadores extrínsecos**, y nacen de la motivación natural de obtener algo o lograr algo (por ejemplo: obtener un tesoro). Al contrario, los elementos del lado derecho son **motivadores intrínsecos**, y hacen

referencia a elementos que son una motivación por sí mismos. Por ejemplo, no se necesita una motivación específica para juntarse con amigos. El hecho de juntarse con amigos es una motivación por sí sola.

Por otro lado, la segmentación arriba versus abajo utiliza el concepto blanco/negro como algo positivo y negativo (en cierto sentido). Los elementos que se encuentran en la parte superior del octágono se conocen como **Gamification de sombrero blanco**, y son elementos que motivan de forma positiva. Por ejemplo, el poder ejercer una acción donde puedes demostrar que se tiene gran conocimiento o habilidad. El segmento inferior se conoce como **Gamification de sombrero negro**, y son elementos que motivan de forma negativa. Por ejemplo, el poder ejercer una acción de la cual no se puede saber qué es lo que se va a obtener, y donde el riesgo puede ser muy alto.

La forma en que el framework funciona para evaluar una actividad o producto al cual se le añadieron elementos de gamification es a través de sus 8 vértices. Se realiza una encuesta al usuario donde, a su juicio, evalúa cada uno de los elementos con nota de 0 a 10. La existencia de notas 0 no necesariamente significan algo malo, como la existencia de notas 10 algo bueno. Por ejemplo, si la actividad no tiene un carácter social (o ese no es su objetivo), pero dada la encuesta los usuarios creen que la actividad tiene un impacto social muy fuerte como componente, evaluando entre 8-10, entonces hay un problema serio en el diseño.

Para el diseñador, lo importante es que los elementos que sí están presentes y los cuales son objetivos dentro del diseño de la aplicación tengan un puntaje alto, y que el puntaje no sea homogéneo en caso que los elementos están en una mayor o menor medida dentro de la misma actividad. Además, una sobrecarga en los motivadores (izquierda versus derecha) o en los sombreros (arriba versus abajo) tampoco significan algo malo, dado que puede que la actividad diseñada busque específicamente desarrollar y explorar esos elementos del octágono en específico. Lo importante a considerar, eso sí, es que tras años de ensayo y error son las actividades **balanceadas** las que tienen un mejor impacto y cumplen mejor su función. Si por ejemplo el diseño de la actividad tiene sólo motivadores negativos y casi nulos motivadores positivos, hay una alta probabilidad que los usuarios se aburran o se nieguen a continuar una vez iniciada la actividad.

2.5. Trabajo Colaborativo

El trabajo colaborativo cumple un rol secundario en este trabajo, pero corresponde, de igual forma, hacer una revisión de lo que lo define y compone. Para esto, se utiliza el trabajo de Johnson y Johnson [13] sobre el trabajo colaborativo y sus reglas. Como bien exponen los autores, al formar grupos de trabajo de varios integrantes, no siempre ocurre lo que se denomina un grupo de aprendizaje colaborativo, sino más bien un grupo de pseudoaprendizaje o un grupo de aprendizaje tradicional. Estos dos últimos ocurren cuando se producen grupos de integrantes que trabajan de forma individual y que en vez de generar un ambiente colaborativo, crean un ambiente competitivo, pues no entienden que el éxito del grupo depende de todos. Estos grupos se caracterizan por actitudes individualistas de los integrantes, donde trabajan sólo los que más saben y tienen la disposición, mientras que los otros deciden no aportar ya que no existe interés en el trabajo ni cohesión con el resto del grupo.

Por el otro lado, un grupo de aprendizaje colaborativo es el que sus integrantes entienden que deben trabajar juntos para lograr el objetivo, el cuál es común para todos. Si triunfa uno, triunfan todos. Si uno falla, fallan todos. Los que más saben ayudan a los que se van quedando atrás, y todos tienen la disposición de avanzar hacia el objetivo. Para que la cooperación en los grupos funcione, hay cinco elementos que deben ser considerados.

2.5.1. Elementos de la cooperación

1. **Interdependencia positiva:** Hace referencia a la existencia de un objetivo en común, y dónde los integrantes tienen claro que sus esfuerzos personales no sirven sólo a ellos mismos, sino a todo el grupo. O ganan todos, o pierden todos.
2. **Responsabilidad individual:** Cada miembro debe asumir la responsabilidad de cumplir con su parte del trabajo en conjunto. Nadie se puede aprovechar del trabajo del otro, y además los integrantes deben ser capaces de identificar su progreso hacia el objetivo y su progreso individual en el trabajo.
3. **Interacción estimuladora (o cara a cara):** Este elemento está asociado a una característica cognitiva de las personas. Dado que en el trabajo colaborativo existe esta

cooperación, esta se logra de mejor manera cuando las personas están en contacto directo (cara a cara), dónde se pueden apoyar y promover el aprendizaje.

4. **Técnicas interpersonales y de equipo:** Consiste en generar un ambiente de cooperación sano en el grupo de trabajo. Trabajar en grupo es más complejo que trabajar individualmente, pues las personas no sólo deben estar pendientes del trabajo y lo que se les asigna para desarrollar, sino también de llevarse bien y cohesionar con los demás integrantes del grupo. Resulta natural que en grupos de trabajo aparezca el conflicto, por lo que las personas deben estar preparadas para saber cómo resolverlo.
5. **Evaluación grupal:** Los integrantes realizan una evaluación de los otros integrantes del grupo y de ellos como grupo en general. La idea es que sepan identificar los aspectos tanto positivos como negativos, y entiendan cómo transmitir la información y proponer soluciones para ir corrigiendo lo negativo.

Capítulo 3

Propuesta

Con lo planteado en los dos capítulos anteriores, la propuesta de este trabajo de memoria puede comenzar a formularse. La secuencia de la propuesta de solución opera de la siguiente forma:

1. **Diseño de un Framework para trabajar Pensamiento Computacional:** a través del estudio desarrollado en el Marco Conceptual, se tiene suficiente información para diseñar y desarrollar un framework que busque colaborar en trabajar el Pensamiento Computacional en colegios, utilizando gamification.
2. **Desarrollo de un programa:** Una vez diseñado el Framework, se desarrolla un programa (software) utilizando Excel, la herramienta de Office desarrollada por Microsoft. Dado que esta es la génesis del Framework, el programa busca ser un primer acercamiento del Framework a los usuarios. La elección de esta tecnología se debe a la facilidad de implementar una UI básica y, además, la posibilidad de usar Visual Basic como lenguaje de programación, ya que cumple a cabalidad con lo que se necesita implementar.
3. **Diseño de la Encuesta de Percepción para los alumnos:** Se definen las preguntas que debe tener la encuesta, pues resulta ser una parte importante del Framework para comparar resultados.

4. **Testeo del programa:** Se pone a prueba el programa de forma real con profesores que enseñan y trabajan el Pensamiento Computacional en colegios.

La Figura 3.1 muestra una representación del flujo de trabajo para lograr la propuesta.

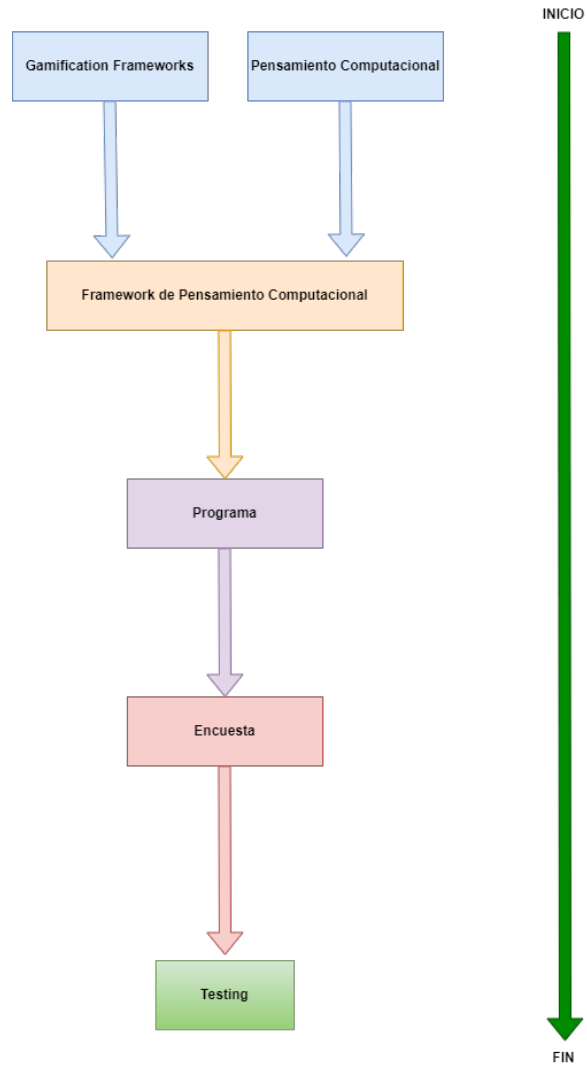


Figura 3.1: Diagrama del flujo de la propuesta.

Fuente: Elaboración propia.

3.1. Idea General

El problema práctico que intenta ayudar a resolver esta propuesta radica en la disonancia que existe actualmente al diseñar actividades gamificadas enfocadas en Pensamiento Computacional. Así, el Framework busca ser una herramienta de colaboración para docentes, profesores, educadores y cualquier persona que sepa y entienda los conceptos principales del Pensamiento Computacional, pero no tenga conocimientos específicos de Gamification, ya sea teóricos o prácticos.

El Framework sirve para desarrollar tanto ideas de actividades que aún están en proceso de *fragar*, cómo para evaluar actividades ya diseñadas, las cuales necesitan de una suerte de filtro que entregue información cuantitativa para tomar decisiones. Sea cual sea el caso, el Framework debería responder adecuadamente a la actividad, mientras cumpla con los requisitos mínimos: los 4 conceptos de Pensamiento Computacional desarrollados, sumado a elementos de Gamification (y, de forma opcional, el Trabajo Colaborativo).

Además, otra parte importante al desarrollar este tipo de actividades resulta ser el feedback. Si no hay una retroalimentación directa, el diseñador de la actividad no tiene cómo saber si lo realizado estuvo bien implementado, independiente de los resultados finales a largo plazo (por ejemplo, los resultados de una prueba donde se evalúan los contenidos presentes en la actividad). Para esto, el Framework busca también levantar una instancia de recopilación de datos, de manera directa con los usuarios, de forma que el diseñador pueda evaluar cuantitativamente los elementos, comparando su percepción del diseño contra la percepción de los usuarios al realizar la actividad.

Los detalles del Framework, desde su génesis hacia el detalle del flujo de desarrollo y utilización, se pueden encontrar en las secciones siguientes de este capítulo.

3.2. Antecedentes del Framework

Para poder comenzar estrictamente con el diseño, primero se deben separar los componentes más importantes y de mayor utilidad de cada uno de los elementos principales que darán forma al Framework: Pensamiento Computacional y Gamification.

3.2.1. Pensamiento Computacional

En el caso del Pensamiento Computacional, los componentes mencionados en el capítulo anterior son los necesarios y a tener en cuenta: **abstracción, diseño de algoritmos, descomposición y reconocimiento de patrones.**

Estos componentes serán los ejes fundamentales en el diseño del Framework, permitiendo que las actividades que se diseñen basándose en él puedan cuantificar cuan presente tienen estos componentes.

Dado que se busca que este Framework se utilice en colegios chilenos, se debe identificar de qué forma, actualmente, aparecen los elementos del PC en la sala de clases. Un buen ejemplo (y el más común) sería la resolución de problemas en la asignatura de Matemáticas. Todo quien haya pasado por la educación escolar chilena debería recordar haber revisado este tipo de ejercicio: Un párrafo con mucho texto (contexto) y números (valores), finalizado por una o varias preguntas (lo que se busca encontrar). El objetivo de este tipo de problemas es lograr que el estudiante aprenda a separar información y a resolver preguntas específicas en contextos donde se satura con datos, algunos necesarios y otros no tanto.

Para explicarlo, se presenta un ejemplo con su respectivo uso del PC:

“Juan va camino a la tienda que queda cerca de su casa. Su mamá lo envió a comprar una serie de cosas, ya que su despensa se estaba quedando vacía. Juan, que tiene muy buena memoria, recuerda al detalle lo que su mamá le pidió: 7 manzanas, 5 zanahorias, 2 lechugas, 4 litros de leche y 2 kilos de pan. Además, su mamá, que también tiene muy buena memoria, recuerda los precios de la tienda y sabe que el total de los productos cuestan menos de \$10.000, por lo que para realizar la compra, le entregó a Juan un billete de \$10.000 y espera que este le de el vuelto al volver a casa. Cuando Juan llega a la tienda, se da cuenta que al fondo del mesón hay una pizarra colgada en la pared, la cual tiene escrito con tiza de distintos colores: Tomates: \$200 cada uno, Lechugas: \$600 cada una, Lentejas: \$800 el kilo, Leche: \$1.000 el litro, Zanahorias: \$400 cada una, Arroz: \$800 el kilo, Manzanas: \$300 cada una, Pan: \$500 el kilo.”

Preguntas a responder:

- a) ¿Cuánto es el vuelto que Juan le dará a su mamá cuando vuelva a casa después de realizar la compra?
- b) ¿Cuánto sería el total si Juan, a sabiendas que su mamá sabe aproximadamente cuánto vuelto es lo que le debe devolver y que lo puede retar si le entrega menos, decide agregar a la compra 5 dulces de \$50 cada uno y 2 chicles de \$100 cada uno?

Este tipo de ejercicio es un problema clásico dentro de la enseñanza básica escolar, y resulta que, de forma intrínseca, utiliza casi todos los componentes del Pensamiento Computacional. Aunque claro, un estudiante de ese nivel escolar, a esa edad, no lo sabe.

La explicación, componente por componente, es la siguiente:

- **Abstracción:** El estudiante debe identificar, dentro de todo ese texto, cuál es la información que necesita para responder a la primera pregunta. Ante esto, el estudiante se debe dar cuenta que lo importante es lo siguiente: la cantidad de dinero con la que cuenta para hacer la compra, la lista de objetos y la cantidad solicitada de cada uno, y finalmente, el precio de cada uno en la tienda. Todo el resto de la información es absolutamente innecesaria, y solo está ahí para confundir al estudiante.
- **Diseño de Algoritmos:** El estudiante debe resolver el problema, y para eso, sin siquiera saber lo que es un algoritmo, diseña uno para encontrar la solución. En este caso, el algoritmo es bastante sencillo: por cada elemento, multiplicar la cantidad solicitada por su precio, y luego sumar todos los resultados. Finalmente, restar ese resultado al dinero con el que cuenta inicialmente y listo, tiene el vuelto que la mamá de Juan está esperando.
- **Reconocimiento de Patrones:** El estudiante se enfrenta a una nueva pregunta, pues ahora se suman dos elementos nuevos a la lista, y cada uno con una cantidad y un precio diferente. Además, la pregunta es por el total de la compra, no por la diferencia (el vuelto). El estudiante debe percatarse que el algoritmo que le permitió solucionar la primera pregunta, en parte, le sirve para la segunda. El estudiante, en la primera pregunta, aplicó conocimientos de suma y multiplicación con diversos productos, como la leche y el pan, y en la segunda pregunta, el estudiante debe ser capaz de entender que la pregunta requiere, exactamente, los mismos conocimientos (y la misma aplicación)

para encontrar su solución.

- **Descomposición:** Este componente no está presente en el ejemplo tal cual está redactado, pero podría aparecer si, por ejemplo, consideramos que la tienda donde va Juan tiene alguno de sus productos agotados, por lo que Juan debe acercarse a otras tiendas (una o más) para encontrar todos los productos, los cuales se venden a otros precios distintos a los de la primera tienda. Con eso, el estudiante se enfrenta a un problema más grande y complejo, con aún más datos. Una forma de resolver esto, evidentemente, es separando el problema en partes, considerando que cada parte es una tienda donde Juan realiza una compra. Así, divide el problema principal en pequeños subproblemas y, para finalizar la solución, solo necesita sumar todos los subresultados obtenidos.

La aplicación intrínseca de estos componentes aparece en otras unidades dentro de Matemáticas -como la resolución de sistemas de ecuaciones- y también en otras asignaturas, como por ejemplo Lenguaje, pero se deben diseñar actividades para trabajarlas. En el apartado 3.2. se muestra de qué forma estos componentes conforman el núcleo del Framework y cómo estos definen si una actividad está bien diseñada o no para trabajar Pensamiento Computacional.

3.2.2. Gamification

En el desarrollo del Marco Conceptual, se expusieron una gran cantidad de componentes para Gamification. Además de su definición, se incluyeron elementos del juego y Game Design -importantísimos para el diseño de actividades- y algunos ejemplos de Frameworks de Gamification que existen y se utilizan en la actualidad.

La perspectiva que utilizan los autores Lee y Hammer para integrar Gamification en educación es lo suficientemente buena para ser considerada en este trabajo. Así, este Framework debería contener los tres enfoques definidos en el capítulo anterior: **Cognitivo, Emocional y Social**.

Además, la estructura que entrega Rosana Rosas también es útil, sobretodo porque la forma en que agrupa y distribuye los elementos está enfocada en aplicar Gamification en el aula de clase.

Dado que el objetivo del Framework a diseñar está, principalmente, enfocado en Pensamiento Computacional, utilizar un Framework de Gamification en su totalidad sería un despropósito que podría terminar anulando al objetivo principal, pero hacer una selección de componentes eliminaría esa preocupación. Para esto, en el siguiente apartado se procede a reagrupar las **Mecánicas** y **Dinámicas** (la estética no es importante para el Framework, sí para las actividades), además de componentes de Frameworks como Octalysis, junto con cada uno de los componentes del PC, de forma que cada subgrupo (elemento PC + elementos Gamification) quede condensado en un concepto, el cual puede ser trabajable y cuantificable.

3.3. Diseño del Framework

Como se menciona en el apartado anterior, son cuatro los elementos principales del Pensamiento Computacional que se esperan trabajar en esta propuesta de Framework. Una posible representación podría ser la de la Figura 3.2. Este rombo corresponde al núcleo del Framework, y ahora hay que combinar cada elemento con componentes de Gamification, creando distintos conceptos que representan subgrupos de cosas.

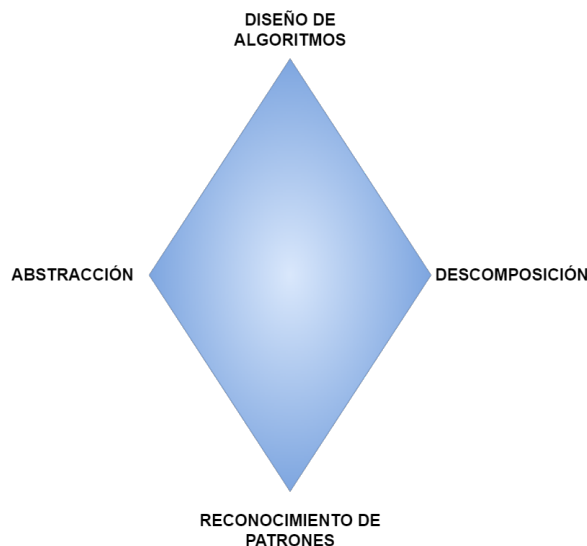


Figura 3.2: Núcleo del Framework basado en Pensamiento Computacional.

Fuente: Elaboración propia.

De la misma forma que estos elementos del Pensamiento Computacional, también se mencionaron distintos elementos de Gamification y de Game Design, pero se hace necesaria una separación de acuerdo a si es útil en este contexto o no, y además, de acuerdo a qué elemento del PC serviría para trabajar.

De todo el conjunto, y de acuerdo a la definición de cada componente del PC, se seleccionan los siguientes elementos: **Narrativa, Reglas, Dificultad, y Niveles.**

Abstracción y Narrativa

La forma de trabajar la abstracción se basa, mayoritariamente, en el análisis de la información. En el problema de ejemplo en la sección 3.1.1., el proceso de abstracción aparece cuando, en vez de entregarle al alumno una ecuación a resolver, se le entrega un **contexto** con el cuál trabajar, y desde ahí, el alumno debe realizar un proceso de selección de información para resolver el problema (o construir la ecuación, al menos de forma implícita). En otras áreas, como la programación por ejemplo, la abstracción ocurre cuando el programador busca representaciones de estructuras de datos, siendo que, dentro del computador, todo se almacena en base a direcciones de memoria. Y también está el caso del arte, donde el artista realiza un proceso de abstracción al dibujar o pintar ciertas figuras, para que luego algún espectador de su obra deba, también en un proceso de abstracción, entender lo que el artista quiere expresar con su trabajo.

Básicamente, la Abstracción necesita de un contexto desde el cual permite realizar el proceso de análisis, y de ese contexto se encarga la Narrativa. La Narrativa no hace referencia necesariamente a que la actividad debe contar una historia con un contenido fantástico y personajes con súper poderes, pero sí debe proveer de la materia prima para la Abstracción: datos e información puesta en desarrollo.

La Narrativa por sí sola es un concepto que engloba a otros elementos, como lo son Espacio, Tiempo, Avatar, Historia y Emociones. En general, cualquier elemento que sirva para nutrir a la Narrativa de la actividad.

Diseño de Algoritmos y Reglas

El Diseño de Algoritmos se trabaja cuando el diseño de la solución a un problema requiere de cierta **secuencia**. Si el alumno se enfrenta a un problema donde la solución no es trivial, entonces debe enfocarse en identificar, paso por paso, los hitos que lo llevan a encontrar esa solución.

Mientras más reglas tenga una actividad, menos libertad existe para que los estudiantes hagan lo que quieran. En este sentido, se vuelve necesario crear un sistema de reglas que guíe a los alumnos hacia el diseño del mejor algoritmo que puedan crear para una posible solución. Mientras más relajado esté el sistema de reglas de la actividad, mayor será el espacio para que los estudiantes se desvíen hacia algoritmos que, incluso, puede que no estén ni siquiera apuntando hacia una verdadera resolución del problema dado. Las reglas, en este sentido, existen de dos formas distintas: implícitas y explícitas. Tal como se mostró en el capítulo anterior, los autores diversifican las reglas, generalmente, en tres o más grupos. Pero dado que la Gamification es sólo la herramienta utilizada y no el foco del Framework, resulta más conveniente separarla en esos dos grupos. Tal como dice su nombre, las Reglas Implícitas son las que no están especificadas pero se sabe de antemano que existe. Por ejemplo, que está prohibido hacer trampa. Por otro lado, las Reglas Explícitas son las que el diseñador debe tener mayor cuidado al diseñar la actividad, pues demuestran qué tan bien pensada está, da los espacios para que esta se desarrolle de buena forma y, lo más importante, guía a los estudiantes a desarrollar algoritmos enfocados en resolver el problema.

Reconocimiento de Patrones y Niveles

Los estudiantes se ven sometidos, normalmente, a enunciados de problemas con distintas preguntas que decantan del mismo problema. Aunque algunas veces las preguntas requieren distintos conocimientos para ser respondidas, otras veces se requiere el mismo conocimiento, pero aplicado desde otra perspectiva. Como ejemplo clásico, el estudiante aprende a contar manzanas, pero se le pregunta por la cuenta de naranjas, y luego la de naranjas y sandías juntas. El Reconocimiento de Patrones aparece cuando el alumno logra **identificar** el concepto

de la suma (o cuenta), y lo aplica para distintas preguntas que requieren el mismo conocimiento, o en algunos casos, requieren una solución -o parte de ella- anterior que ya encontró. En un juego, la forma de generar dinamismo y la sensación de progreso es a través de distintos niveles (o *biomas*, como les llaman más actualmente), independiente de la dificultad. Así, la forma de darle la oportunidad al estudiante de identificar patrones requiere que la actividad a desarrollar cuente con distintas instancias para aplicar los conocimientos, y no sólo un ambiente monótono. Además, la sensación de progreso siempre debe estar presente al aplicar distintos niveles, pues el entusiasmo se pierde al desarrollar actividades -o jugar juegos- que no le demuestran al usuario que está avanzando en el contenido. Dicho esto, es que los Niveles se descomponen en dos elementos esenciales: Diversificación y Sensación de Progreso.

Descomposición y Dificultad

Ante un problema simple, generalmente viene una solución simple. Pero cuando el problema resulta ser más **complejo**, las personas tienden a desarrollar una sensación de pesadumbre, pues no saben cómo abordar el problema. La Descomposición es un elemento que se hace presente cuando se presenta un problema dónde la mejor forma de enfrentarlo es dividiéndolo en problemas más pequeños, por lo que la composición de todos ellos juntos recrean el problema original. Para esto, los estudiantes deben reconocer que el problema tiene distintos matices y preguntas que se relacionan pero que requieren distintas soluciones, por lo que particionan el problema original.

La dificultad debe estar enfocada en complejizar los problemas o ejercicios buscando que los estudiantes separen el problema original en subproblemas y, además, que los más adeptos a la materia puedan demostrar su habilidad al resolver problemas difíciles. Pero, el aumentar la dificultad, se corre el riesgo de generar esa sensación de pesadumbre, haciendo que los estudiantes prefieran rendirse y no intentar resolver el problema. La forma de solucionar esto es a través de un sistema de recompensas, pues esto motiva a los estudiantes a intentar resolver el problema y puede que sea el estímulo necesario para recorrer la última milla y llegar a una solución. Entonces, la Dificultad tiene dos componentes que la ayudan a existir: el Sistema de Recompensas y la Habilidad.

Estos subgrupos, para ser representados en un sólo concepto, se nombran de acuerdo a una palabra representativa de su contenido. Así, al subgrupo de **Diseño de Algoritmos** se le llama **Secuencia**, al de **Abstracción** se le llama **Contexto**, al de **Descomposición** se le llama **Complejidad** y, finalmente, al de **Reconocimiento de Patrones** se le llama **Identificación**. Una representación de esto se puede ver en la Figura 3.3.

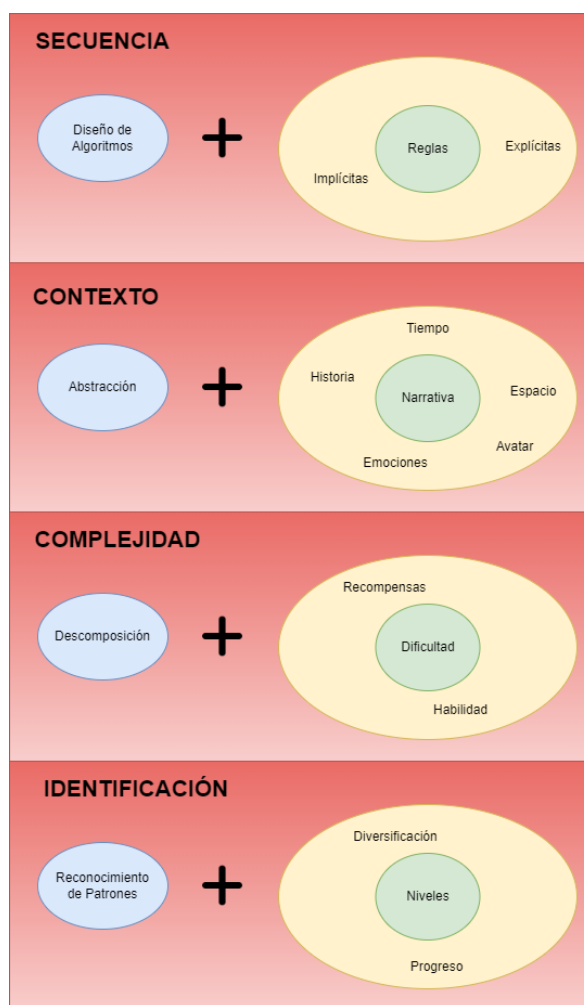


Figura 3.3: Subgrupos con los elementos de PC y Gamification seleccionados.

Fuente: Elaboración propia.

Además, dado que el ambiente donde se desarrollarán las actividades basadas en este Framework es una sala de clases, con diversos estudiantes que componen un curso en un entorno **social**, es que sería un error no considerar, al menos como componente opcional, al **Trabajo Colaborativo**.

Trabajo Colaborativo

El **Trabajo Colaborativo**, al igual que el Pensamiento Computacional, es una de las habilidades que se consideran como esenciales para el siglo XXI. Esta habilidad se pone en marcha cuando se considera el trabajo en equipo dentro del desarrollo de una actividad, según los elementos de la cooperación expresados en el Marco Conceptual.

Un buen ejemplo para esto último sería el de contestar preguntas en grupo, en voz alta. Cada pregunta correcta suma puntos, mientras que las incorrectas no suman ni restan. Los estudiantes no pueden pasar, y una vez que a un integrante del grupo le toca responder, debe esperar que los otros miembros del grupo respondan en sus respectivos turnos para que pueda responder otra vez. Además, se pueden ayudar entre ellos, pero sin *soplar* la respuesta correcta.

Aquí se utilizan elementos sociales que existen en Gamification, como lo son las actividades grupales, la competitividad entre equipos utilizando puntaje, la asignación de roles dentro de los equipos y la posibilidad de generar instancias donde los estudiantes practiquen sus habilidades de liderazgo y enseñanza a los que no están al día con las materias a través de pequeñas tutorías. Estos elementos de Gamification ayudan a que los elementos de cooperación del Trabajo Colaborativo se integren de manera correcta en una actividad y que, efectivamente, la actividad tenga Trabajo Colaborativo y no sólo trabajo en grupos.

El nombre que se le da a este nuevo subgrupo es el de **Conjunto**, y contiene al Trabajo Colaborativo junto con los elementos sociales de Gamification. La Figura 3.4 muestra una representación del subgrupo.

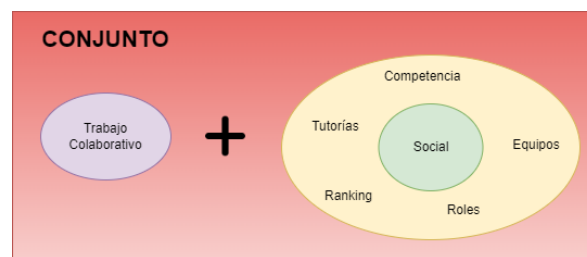


Figura 3.4: Subgrupo del Trabajo Colaborativo y Gamification.

Fuente: Elaboración propia.

3.4. Framework de Pensamiento Computacional

Tomando en consideración todo lo expresado y lo representado en las Figuras 3.3 y 3.4, el Framework de Pensamiento Computacional queda definido en dos niveles, separados básicamente por la existencia (o no existencia) del componente social y la habilidad del Trabajo Colaborativo.

Eso sí, la forma cuantitativa que tiene el Framework para evaluar las actividades diseñadas es igual en ambos niveles. Una explicación y ejemplos al respecto se pueden revisar en los siguientes apartados.

3.4.1. Framework de Pensamiento Computacional de Primer Nivel

El Framework de Primer Nivel se considera al que está enfocado en actividades **individuales**, por lo que el diseño de actividades basadas en este nivel del Framework solamente toman en cuenta los subgrupos de PC y Gamification condensados en los conceptos de Secuencia, Contexto, Identificación y Complejidad.

Una representación del Framework del Primer Nivel se puede apreciar en la Figura 3.5

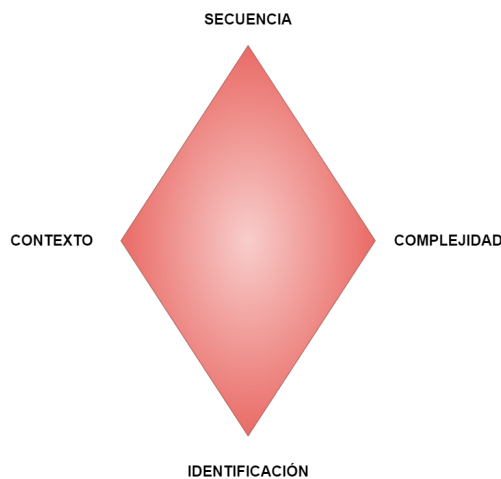


Figura 3.5: Framework de Pensamiento Computacional de Primer Nivel.

Fuente: Elaboración propia.

El Framework escala el valor **subjetivo** de cada uno de los componentes, desde 0 hasta

10. De forma automática, al trabajar en este nivel el valor de Conjunto está ajustado en 0, y no se puede variar. La subjetividad se debe a que no existe una forma directa de medir el impacto con exactitud que tiene cierto componente en una actividad. Para esto, el Framework se encarga de guiar al diseñador a tener un estimado pensando en qué componente del PC quiere trabajar (y en qué medida), y con eso, aplicar los elementos de Gamification asociados al subgrupo.

Una representación del Framework, graduado, se puede ver en la Figura 3.6.

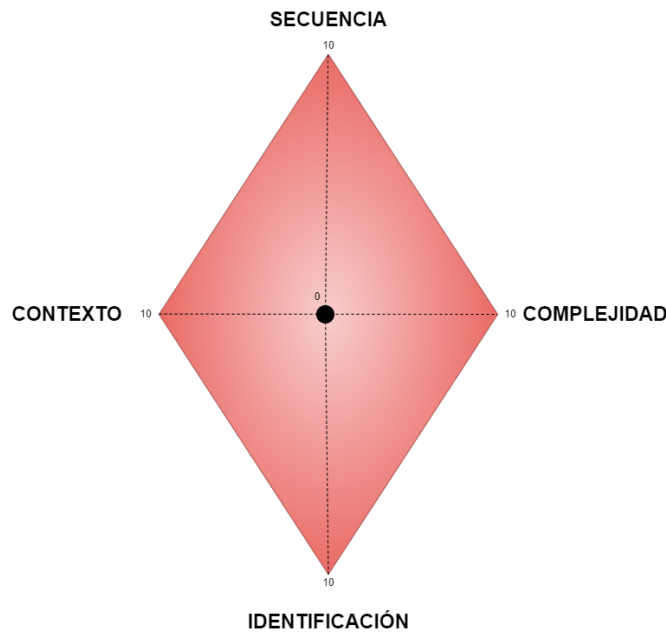


Figura 3.6: Framework de Pensamiento Computacional de Primer Nivel graduado.

Fuente: Elaboración propia.

3.4.2. Framework de Pensamiento Computacional de Segundo Nivel

El Framework de Segundo Nivel se considera al que está enfocado en actividades **grupales**, por lo que el diseño de actividades basadas en este nivel consideran todos los elementos del Framework, incluyendo Conjunto. Además, los demás elementos se ven influenciados por las mecánicas sociales incluidas en el concepto Conjunto. La representación del Framework de Segundo Nivel se puede ver en la Figura 3.7.

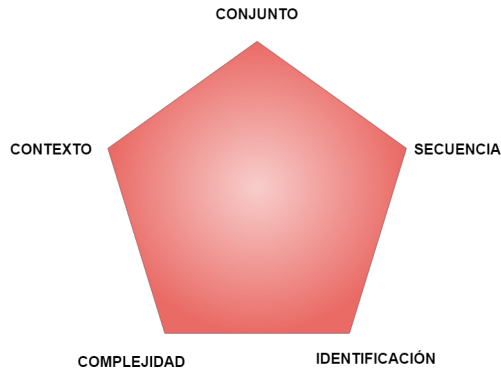


Figura 3.7: Framework de Pensamiento Computacional de Segundo Nivel.

Fuente: Elaboración propia.

La forma de graduar el Framework de Segundo Nivel sigue la misma lógica subjetiva que el del primer nivel. Una representación del Framework graduado se puede ver en la Figura 3.8.

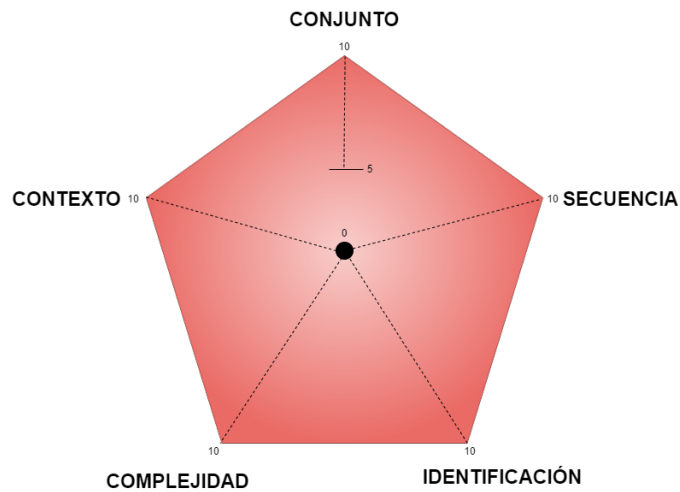


Figura 3.8: Framework de Pensamiento Computacional de Segundo Nivel graduado.

Fuente: Elaboración propia.

3.4.3. Reglas de puntuación del Framework

El sistema para puntuar cuenta con Reglas Generales y con Reglas Específicas. Las Reglas Generales corresponden a aquellas que son objetivas para sumar puntos, como la presencia/ausencia de un elemento en particular. Se aplican en los elementos de Pensamiento Computacional a trabajar en la actividad. Las Reglas Específicas corresponden a las que necesitan de cierta subjetividad para asignar un valor, sobre todo a cosas asociadas a Gamification.

La razón de haber escalado la puntuación de 0 a 10 se debe a que es una buena medida para evitar dispersión de los resultados, al estar esto fuertemente influenciado por la percepción. En una escala más pequeña (por ejemplo, de 0 a 5) la holgura es menor y los participantes, al momento de evaluar, podrían verse confundidos con qué valor tomar. Por otra parte, una escala holgada (por ejemplo, de 0 a 100) permitiría tener mayor precisión para los participantes, pero dificultaría el análisis de los resultados.

Reglas Generales

- Cada elemento del Framework asociado a Pensamiento Computacional puede tener un puntaje mínimo de 0 puntos, y uno máximo de 10 puntos.
- El elemento Conjunto está forzado en este nivel, por lo que el puntaje mínimo es de 5 puntos, y su máximo son 10 puntos (Segundo Nivel).
- Agregar un elemento de Pensamiento Computacional suma 5 puntos en el vértice correspondiente.
- Agregar un segundo elemento de Pensamiento Computacional suma 3 puntos en el vértice correspondiente. Agregar un tercer elemento suma 2 puntos, y un cuarto elemento suma 1 punto.
- Añadir un elemento de Gamification en el vértice de Conjunto suma 2 puntos (Segundo Nivel).
- Si un elemento del Pensamiento Computacional no está presente, entonces ese vértice sólo puede sumar un máximo de 5 puntos.

Reglas Específicas: Puntuación de las Reglas

El puntaje de las Reglas es subjetivo dado que depende de algo conceptual, y no de si son implícitas o explícitas, o de la cantidad. Cuando se habla de Reglas, se entiende como el sistema que guía al curso de acción de la actividad, pero también que restringe lo que se puede hacer. El concepto que se opone a las Reglas corresponde al de **Libertad**. Mientras mayor sea la Libertad, menor será el espacio restringido por las Reglas.

En una actividad de juego libre, los participantes tienen total *libertad* de acción para hacer lo que quieran, mientras que en una actividad definida y guiada por reglas, los usuarios realizan lo que el diseñador de la actividad quiere que hagan.

Así, la puntuación de las Reglas, aunque subjetiva, es bien sencilla:

- Si el sistema de reglas no permite libertad a los usuarios, suma 5 puntos al vértice Secuencia
- Si el sistema de reglas permite total libertad a los usuarios, no suma puntos
- Si el sistema de reglas es mixto (y permite a los usuarios salir -en alguna medida- de las restricciones), suma entre 1 y 4 puntos al vértice Secuencia, esto definido por el diseñador **según su criterio**

Máximo puntaje a obtener: 5 puntos.

Reglas Específicas: Puntuación de la Dificultad

La dificultad se puede revisar desde distintas perspectivas, pero la que más interesa es que sea una mezcla entre algo escalonable y algo agrandable. Cuando la dificultad es escalonable, los problemas aumentan su dificultad de forma gradual. En otras palabras, se comienza con algo sencillo a forma de introducción, para luego ir aumentando la dificultad, costando un poco más de esfuerzo dar con la respuesta. Por otro lado, el término agrandable hace referencia a que lo que se busca es que el problema sea de un tamaño mayor, lo que permita al usuario partirlo en trozos más pequeños y abordables.

Entonces:

- Si la dificultad es escalonada y agrandable, suma 3 puntos al vértice de Complejidad
- Si la dificultad es escalonada y no agrandable, suma 2 puntos al vértice de Complejidad
- Si la dificultad es estándar y agrandable, suma 2 puntos al vértice de Complejidad
- Si la dificultad es estándar y no agrandable, suma sólo 1 punto al vértice de Complejidad

Además, como ya se ha mencionado, la dificultad resulta lograr, de forma natural, disuadir a los usuarios de participar de cualquier cosa. Independiente que existan personas que sienten cierta atracción por la dificultad, la mayoría no es así. Por eso, la forma de paliar este efecto negativo es el Sistema de Recompensas. Las recompensas que se pueden entregar son de dos tipos: reales y virtuales.

Un premio real hace referencia a una recompensa fuera de la actividad (por ejemplo, una nota, décimas, dulces, etc). Premio virtual hace referencia a una recompensa dentro de la actividad (por ejemplo, al derrotar a un enemigo en un juego de rol, puedes recibir una espada milenaria muy poderosa).

- Si hay una recompensa real, suma 1 punto al vértice Complejidad
- Si hay una recompensa virtual, suma 1 punto al vértice Complejidad

Máximo puntaje a obtener: 5 puntos.

Reglas Específicas: Puntuación de los Niveles

Los Niveles sirven para diversificar y marcar progreso en el desarrollo de la actividad. Para este Framework, lo que interesa es que los niveles diversifiquen buscando la identificación de patrones.

- Si los niveles diversifican permitiendo identificar patrones, se suman 3 puntos al vértice Identificación
- Si existen niveles pero no diversifican de ninguna forma más que para cambiar el ambiente, entonces suma 1 punto al vértice Identificación
- Si no hay niveles presentes en la actividad, entonces no hay suma de puntos

Al agregar niveles, hay que cuidar que un mal diseño genere una sensación de estancamiento (que todos los niveles sean iguales, aunque parecieran ser distintos). Una forma de romper esa sensación es a través de un Sistema de Progreso, el cual se descompone en progresión real y progresión virtual. Progresión real hace referencia al avance del usuario en la vida real con respecto a lo que se está trabajando (por ejemplo, gracias a la realización de la actividad, ahora el usuario puede resolver operaciones de matemáticas más complejas que antes). Progresión virtual hace referencia al avance del usuario dentro de la actividad (por ejemplo, al derrotar a un enemigo en un juego de rol, el jugador recibe puntos de experiencia, los cuales le permiten subir de nivel y aprender nuevas habilidades).

- Si hay progresión real, se suma 1 punto al vértice Identificación
- Si hay progresión virtual, se suma 1 punto al vértice Identificación

Máximo puntaje a obtener: 5 puntos.

Reglas Específicas: Puntuación de la Narrativa

La Narrativa tiene diversos elementos de Gamification, pero lo que se busca es que utilicen los que permiten que el usuario trabaje con información y datos, y también los que permiten que el usuario busque otras formas de representación de cosas. Es por esto que los elementos que agregan valor al Framework son pocos, y se insta al diseñador a que los use.

La Narrativa se define en un Espacio real (por ejemplo, la sala de clases, o la sala de computación, o el patio) y en un Espacio Imaginario (por ejemplo, una escape room, una selva, una

cafetería, etc). Además, la forma de introducir a los usuarios a este mundo imaginario es a través de una historia (o un enunciado).

- Si se organiza el espacio real para la actividad, se suman 1 puntos al vértice Contexto
- Si se crea un espacio imaginario para representar el mundo donde ocurre la actividad, se suman 2 puntos al vértice Contexto
- Si existe una historia o enunciado preparado, se suma 2 punto al vértice Contexto

Máximo puntaje a obtener: 5 puntos.

3.4.4. Consideraciones del Framework

La figura perfecta existe, pero sólo en el papel

La figura perfecta se considera al rombo o pentágono con todas sus aristas puntuadas en 10, tal como se ven en la Figura 3.5 y Figura 3.7. En cierto sentido, y en base a la lógica, es posible crear una actividad basada en el Framework que, por las reglas de puntuación, cumpla con los parámetros para ser una figura perfecta. El problema se hace ver a través de la puesta en práctica, pues una actividad que se represente por la figura perfecta, probablemente, sea un rotundo fracaso. La percepción de los participantes distará mucho de la figura perfecta, y el feedback general de la actividad no será de utilidad dada la dispersión de conceptos. Esto último hace referencia a que la existencia de tantos conceptos (tanto de PC como de Gamification) hará que los usuarios no logren identificar ninguno, dado que los usuarios generalmente logran notar focos.

Menos es más

El diseñador, al crear actividades basadas en el Framework, debe buscar formar focos que se puedan identificar. Para crear focos, se utiliza la regla que aparece en el título: Menos es más. Siempre será mejor una actividad que contemple sólo un vértice del Framework, muy

bien diseñada y que utilice bien los elementos de Gamification, a una actividad con tres o cuatro vértices cargados hacia los 10 puntos. Con esto, los participantes podrán identificar claramente qué es lo que están trabajando y cómo lo están desarrollando, de forma que el feedback será más específico y su percepción de la actividad concluirá con una figura parecida a la del diseñador.

Además, considerando que el objetivo principal del Framework es enseñar Pensamiento Computacional, la evidencia dice que los estudiantes retienen mejor los conocimientos al trabajar los conceptos por partes, en vez de todos juntos en una misma instancia.

Elementos focalizados

Cuando se trabaja con más de un componente del Pensamiento Computacional en una misma actividad (por ejemplo, dos), es importante seleccionar una cómo la principal y la otra como secundaria, para así poder escalar. Las reglas de puntuación del Framework ya colaboran al dar 5 puntos a la principal y 3 a la secundaria, pero aquí es dónde el diseñador debe utilizar los elementos de Gamification de forma eficiente, pues en base a estos puede hacer que la componente secundaria obtenga un puntaje mayor que la principal. Si se quiere lograr construir focos, se debe priorizar el puntaje para la actividad principal, cuidando que la secundaria quede con un puntaje menor pero siempre resguardando un buen diseño (aunque puedan existir casos borde donde esto no se cumpla y la actividad funcione de todas formas).

La Gamification es flexible

Dependiendo de la actividad, los elementos de Gamification pueden tomar distintos sentidos. En una, puede que un sistema de roles para los usuarios sea la piedra angular para el buen desarrollo, mientras que para otro, el sistema de roles tiene una utilidad en cuanto a generar interés a los participantes. En ambos casos, el uso del sistema de roles es correcto y vale la pena utilizarlo. Una gran ventaja de la Gamification y sus elementos es la versatilidad que tienen, y para encontrar el mejor uso dependiendo de la actividad, es deber del diseñador el explorar y descubrir el universo de posibilidades.

La práctica hace al maestro

La forma de mejorar en el diseño de actividades utilizando este Framework es a través del ensayo y error. Cada vez que se diseña una actividad, se debe tener en cuenta **la evaluación**. La evaluación se debe realizar desde dos puntos de vista: del diseñador y del usuario. En el caso del aula de clases, el profesor (diseñador) debe realizar su evaluación al momento de diseñar la actividad (puntuar con el Framework), y luego debe encuestar a los estudiantes (usuarios) sobre su **percepción** de la actividad, una vez finalizada esta. Este feedback es esencial para contrastar la opinión personal de la actividad con el desarrollo real, y el resultado de este proceso lleva a ajustar mejor esta subjetividad.

3.4.5. Secciones de una Actividad

Antes de poder diseñar una actividad, primero hay que entender cómo es que las actividades se desenvuelven y en qué consisten.

La mayoría de las actividades podrían ser divididas en tres partes, con un separador claro entre las dos últimas: el objetivo o condición de victoria. Una representación de esto se puede ver en la Figura 3.9.

Inicio

La etapa de inicio de una actividad es el momento que tienen los usuarios para asimilar a lo que se estarán enfrentando más adelante. El participante comienza a comprender el mundo ficticio en el que va a trabajar, el sistema de reglas que tendrá que seguir, la lista de mecánicas que le permitirá actuar, el objetivo a cumplir (y la condición de victoria, de ser necesaria) y el punto de término de la actividad. Además, en caso de ser una actividad social y/o *unplugged*, es el momento donde se transforma el espacio real (por ejemplo, sala de clases) en el espacio de *juego*, y el momento donde se conforman los equipos y se reparten los roles, en caso de existir.

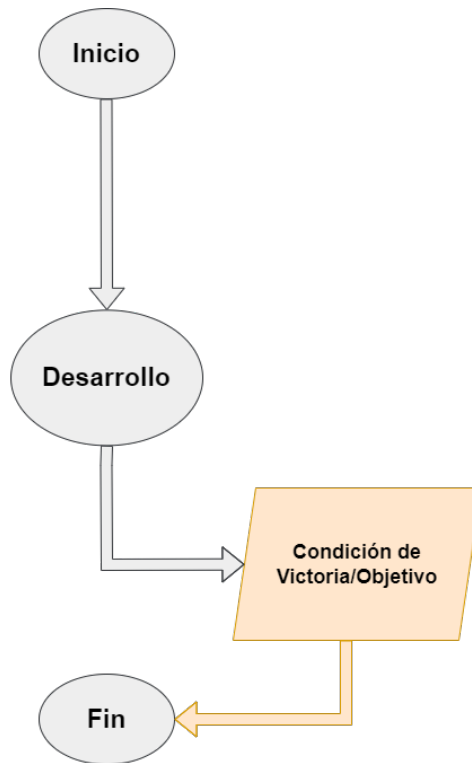


Figura 3.9: Seccionado de una actividad.

Fuente: Elaboración propia.

Desarrollo

El desarrollo es la etapa que comienza cuando se supera el inicio. Los participantes, a través de las mecánicas, sortean las dificultades para progresar hacia el objetivo. Desde un punto de vista técnico, el desarrollo se podría representar como un *loop*: la actividad no termina hasta que se haya cumplido la condición de victoria o se haya alcanzado el objetivo.

Condición de victoria/Objetivo

La condición de victoria es el *breakpoint* de la actividad. El *loop* del desarrollo termina una vez se alcanza este punto. Hay actividades que no cuentan con una condición de victoria (por ejemplo, las que no son competencia), pero sí cuentan con un objetivo. En esos casos, el alcanzar el objetivo marca el final del desarrollo.

Fin

El fin de la actividad es el momento donde los usuarios ya no están sujetos al sistema de reglas y ya no pueden seguir utilizando las mecánicas que les permiten progresar, pues el desarrollo terminó. En esta sección es cuando se hace conteo de puntaje, se definen ganadores, se hace revisión del desarrollo y/o se repasa contenido. Una vez eso termina, se puede dar por finalizada la actividad en su totalidad.

3.4.6. Enfoques de las actividades de acuerdo al vértice

Contexto

En el caso de contexto (que contiene la abstracción), los conceptos que se deben utilizar son la **representación** y a la **imaginación**. El usuario de la actividad debe pensar en formas de trabajar con información y ejecutar una acción que lleve a resultados concretos en base a su entendimiento. Por ejemplo, tal como se menciona anteriormente, un estudiante trabaja la abstracción al resolver problemas de enunciado en Matemáticas, o al dibujar de acuerdo a ciertos parámetros en Arte. Además, también existe un proceso de abstracción en el curso de Lenguaje cuando se trabaja con poesía, pues el proceso para entenderla requiere que el lector deje de pensar de forma lógica/literal, y se centre en el sentido figurado de las cosas, para así dar con la intención del autor. Sin lugar a dudas, contexto es el componente más difícil de trabajar y explotar al diseñar actividades, por lo que se recomienda explorar de distintas formas el trabajo en esta área.

Secuencia

En el caso de secuencia (que contiene el diseño de algoritmos), los conceptos principales deben ser los de **ordenamiento** y **secuencia de pasos**. A diferencia de contexto, secuencia es más fácil de entender y también de explicar. Cualquier actividad que invite al usuario a ordenar elementos y crear listas de instrucciones (y mientras más detalladas, mejor) trabaja el componente secuencia.

Complejidad

Los conceptos asociados a complejidad (que contiene la descomposición de problemas) son la **dificultad escalonada** (o progresiva) y los **problemas divisibles**. Es importante que el problema divisible NO se entregue ya dividido, sino que debe ser parte de la dificultad el que el usuario de la actividad identifique los subproblemas y la forma de resolverlos. Además, la dificultad del problema (o problemas) debe ser variante y ordenada. Esto quiere decir que se debe comenzar con algo fácil, a modo de tutorial, para luego ir incrementando la complejidad. Es importante también tener siempre claro que la dificultad es un arma de doble filo. Cuando las cosas son demasiado complejas, el usuario pierde interés, independiente de la recompensa que pueda obtener, debido a la frustración que produce la sensación de estancamiento.

Identificación

Finalmente, la identificación (que contiene la identificación de patrones) utiliza el concepto de los **problemas similares** y el **contenido equivalente**. Básicamente, para trabajar identificación se quiere que el usuario logre detectar conceptos equivalentes dentro de la materia que se está estudiando, y pueda aplicarla en distintos contextos. Tal como se dijo anteriormente como ejemplo: si un estudiante sabe sumar y se le pregunta por una suma de manzanas y luego por una suma de naranjas, el estudiante debe identificar que la operación aritmética de la suma es el contenido común y que le sirve para responder la segunda pregunta. Obviamente, esto también sirve para otras materias: en Arte, el estudiante debe identificar patrones del estilo que se utilizaba en una época determinada, o en Educación Física, al realizar trabajos aeróbicos que requieren de ciertos movimientos similares, como en la invertida y en la rueda.

Conjunto

El componente conjunto se trabaja añadiendo cualquier concepto **social** a la actividad. El uso de este componente cambia el Nivel del Framework, y contempla cualquier elemento

que colabore a que trabajar en equipo sea mejor que trabajar de forma individual. Se puede utilizar en casi cualquier curso del colegio, siempre dependiendo de la materia que se esté revisando. Para que el componente conjunto quede bien implementado dentro de la actividad, es esencial lograr un equilibrio con las componentes sociales de Gamification a utilizar. Si, por ejemplo, se utiliza un sistema de roles dentro de los equipos, se debe asegurar que cada uno de ellos tenga una tarea/objetivo que cumplir, en toda parte del trayecto. Por el contrario, si lo que se quiere generar es una competencia (individual o grupal, ya que es un elemento que está presente en ambos Niveles del Framework) se necesita tener en consideración que los que se quedan atrás no pierdan el entusiasmo por participar, y además, en pos de la competencia, evitar el efecto *snowball*. Este efecto hace referencia a un fenómeno donde quién lleva la ventaja se vuelve inalcanzable, y un ejemplo de esto podría ser un sistema de puntos con multiplicadores, donde mientras mejor sea el rendimiento en la actividad, mayor es el multiplicador. Con esto, los puestos abajo del primero estarán en una desventaja que se vuelve constante, hagan lo que hagan.

3.5. Secuencia de uso: cómo utilizar el Framework

Antes de exponer el trabajo necesario para diseñar una actividad, es importante recalcar que existen diversas formas de crear instancias gamificadas, y lo que aquí se representa es sólo una de ellas, pero con un foco hacia lo que busca el Framework. En otras palabras, el diseñador podría formular una actividad utilizando otro Framework (como, por ejemplo, uno de los Gamification Frameworks mencionados en el Marco Conceptual) pero de todas formas deberá pasar su actividad por los esquemas que propone este Framework, para así puntuar sus elementos y conocer los focos de trabajo y la figura.

También se pueden utilizar actividades-tipo, las que se pueden encontrar en el Anexo A del Capítulo 6.

3.5.1. Cuestionario inicial

Lo primero que hay que hacer, antes de comenzar a diseñar la actividad, es contestar las siguientes preguntas:

¿Qué quiero trabajar?

Esta pregunta apunta al elemento del Pensamiento Computacional a reforzar con la actividad. Para responder adecuadamente esta pregunta, el mejor caso sería no sólo seleccionar el o los elementos, sino que además tener claro cuál elemento será el principal y cual/es los secundarios. Aún así, solo teniendo seleccionados los elementos ya es una respuesta válida a la pregunta. Por recomendación, lo mejor es que no sean mas de **dos** elementos los seleccionadas.

Respuestas posibles: Abstracción, Diseño de Algoritmos, Identificación de Patrones, Subdivisión de Problemas (más de una)

¿Es una actividad individual o en grupo?

Responder esta pregunta antes de comenzar a diseñar la actividad permite saber de antemano cuál nivel del Framework utilizar. Si la actividad será individual, entonces se utiliza el Framework de Primer Nivel y la referencia pasa a ser el rombo. En caso que la actividad busque reforzar también el Trabajo Colaborativo, entonces se utiliza el Framework de Segundo Nivel y la referencia es el pentágono. Esta decisión es importante saberla antes, dado que los elementos de Gamification a seleccionar en el diseño de la actividad se pueden ver afectados por la presencia (o no) del componente Conjunto del Framework.

Respuestas posibles: Individual ó Grupal

¿Qué tipo de actividad es?

Dado que existen distintos tipos de actividades, se sugiere tener claro antes de comenzar a diseñar los mecanismos a trabajar. La actividad puede ser teórica o práctica (o un poco de las dos). También puede ser utilizando un medio tecnológico, como un computador o un smartphone, o puede ser una actividad *unplugged*.

Respuestas posibles: Teórica, Práctica, Unplugged (o una mezcla)

¿Cuál es el nivel de Gamification que quiero incluir?

El nivel de Gamification hace referencia a qué tan gamificada está la actividad diseñada. Los niveles se dividen en tres calificaciones: **baja**, **media** y **alta**. Una actividad con un nivel de gamificación bajo es aquella donde la actividad no dista mucho de ser una clase normal, una cátedra o una resolución de una guía, pero con algún elemento gamificador como lo es el puntaje o el ranking. Por otro lado, una actividad con un nivel de gamification alto corresponde a un amplio uso de la gama de elementos de gamification existente, al punto que lo que resultaría ser una clase normal o una resolución de una guía de ejercicios, se transforma en una actividad donde los usuarios perciben que ya no están en el mismo ambiente que una clase normal, y por sobre todo, se están divirtiendo (algo que, por lo general, no ocurre normalmente en el aula de clases).

Respuestas posibles: Alta, Media, Baja.

¿Cuanto es el tiempo estimado que tengo/quiero/necesito para desarrollar la actividad?

Esto es, sencillamente, para tener en cuenta la cantidad de tiempo disponible y así pensar mejor lo que se puede hacer y no hacer. Si, por ejemplo, la clase cuenta con 1 hora y 30 minutos en total, pensar en una actividad de 3 horas significaría un problema, como también si la actividad dura tan sólo 20 minutos (a no ser que se quiera realizar una clase mixta).

Respuestas posibles: Cantidad de tiempo.

3.5.2. Diseño de la actividad I

En esta sección se definen los elementos que pertenecen al núcleo de la actividad, y son esenciales para el INICIO. Primero, se homologan los componentes del Pensamiento Computacional con los componentes correspondientes del Framework. Luego, cada elemento de Gamification se define en base a la siguiente plantilla: se hacen preguntas con respuesta sí/no (algunas tienen otra respuesta que no afecta a la definición sino a la puntuación), y luego se definen los elementos en base a esas respuestas.

La Gamification es una herramienta flexible que utiliza diversos elementos del diseño de juegos, independiente del tipo de actividad a realizar y de los focos de Pensamiento Computacional a trabajar seleccionados. Si bien hay elementos de Gamification que están asociados a un determinado elemento del PC, es importante recalcar, una vez más, que una actividad gamificada puede -o tiene que- contar con elementos asociados a componentes del PC que NO se están trabajando, todo en pos de lograr una actividad balanceada, entretenida y bien lograda.

1. Reconocimiento de el/los vértice(s), Focos y Nivel del Framework

Teniendo definido el/los componentes del PC que se quieren trabajar, se requiere hacer el *match* correspondiente con el Framework de Pensamiento Computacional, y reconocer los Focos de trabajo (primario, secundario, etc).

- Abstracción → Contexto
- Desarrollo de Algoritmos → Secuencia
- Descomposición → Complejidad
- Reconocimiento de Patrones → Identificación
- Trabajo Colaborativo → Conjunto

Si no se utiliza el Trabajo Colaborativo, entonces el diseño se enfrenta al Primer Nivel del Framework. En caso contrario, hay que utilizar el pentágono del Segundo Nivel.

2. Definición de la Narrativa

La Narrativa viene a ser, en resumidas cuentas, de lo que se va a tratar la actividad. Puede estar muy definida y utilizar muchos componentes de Gamification, como también puede estar solo para dar un poco de contexto.

- ¿Es Contexto un vértice seleccionado?

Respuesta posible: sí/no

Luego:

- Definir Espacio Real
- Definir Espacio Imaginario
- Definir Historia (o enunciado)
- Definir otros elementos de la Narrativa (Tiempo, Personajes, etc)

3. Definición de elementos Sociales (SOLO EN EL SEGUNDO NIVEL)

Algunas veces los elementos de Gamification pertenecientes a Conjunto alteran la forma normal de crear reglas y mecánicas, por lo que deben ser definidas antes de estas últimas.

1. ¿Hay un sistema de roles?

Respuesta: sí/no

El sistema de roles permite generar instancias de colaboración donde todos los usuarios participan en la actividad con un objetivo concreto.

2. ¿Hay un sistema de puntaje?

Respuesta: sí/no

El sistema de puntaje sirve para marcar progreso y también para cumplir objetivos (como la condición de victoria).

Con esto:

- Definir Equipos (opcional) (en cantidad, cómo se forman, etc).
- Definir Sistema de Roles
- Definir Sistema de Puntaje
- Definir otros elementos necesarios

4. Definición de las Reglas y Mecánicas

Tal como ya se ha mencionado, las reglas son el elemento más importante dentro de cualquier actividad Gamificada. Fuera de una actividad de *juego libre*, todas tienen reglas, y las reglas son las que definen y diferencian a la actividad.

Además de las reglas, se deben definir las mecánicas. Las mecánicas son las acciones que pueden realizar los usuarios para participar en la actividad (por ejemplo, la mecánica del tic-tac-toe le permite al usuario marcar con una X o una O, según corresponda, en una casilla vacía).

1. ¿Es Secuencia un vértice seleccionado?

Respuesta: sí/no

2. ¿Qué tanta libertad tendrá la actividad?

Respuesta: mucha/intermedia/poca/nula

Para puntuar en el Framework.

Luego:

- Definir Reglas explícitas
- Definir Reglas implícitas
- Definir Mecánicas

5. Definición del Objetivo y/o Condición de Victoria

El objetivo siempre debe estar presente, mientras que la condición de victoria depende del tipo de actividad que se esté diseñando, por lo que podría no estar presente. Por ejemplo, en una actividad del tipo *escape room*, el objetivo de la actividad sería escapar de la sala, mientras que la condición de victoria corresponde a lograr abrir el candado antes del tiempo límite.

Así:

- Definir Objetivo
- Definir Condición de Victoria (de ser necesario)

3.5.3. Diseño de la actividad II

En esta sección se definen los elementos que son esenciales para el DESARROLLO de la actividad.

6. Definición de la Dificultad

La Dificultad es una herramienta que permite aumentar la cantidad de esfuerzo necesario para resolver un determinado tipo de problema, volviéndolo más complejo. También se puede utilizar desde otras perspectivas, como por ejemplo, en vez de aumentar la dificultad directamente a un problema, se puede utilizar alterando una mecánica que le permite al usuario resolver rápidamente el problema, obligándolo a pensar de otra forma. Dependiendo de si Complejidad es foco de trabajo en la actividad, el área de interés de la Dificultad es una dificultad agrandable, la que permite al usuario subdividir un problema.

1. ¿Es Complejidad un vértice seleccionado?

Respuesta: sí/no

2. ¿Es la dificultad un elemento diferenciador en la actividad?

Respuesta: sí/no

Cuando ambas respuestas anteriores son **no**, es mejor dejar una dificultad estándar para toda la actividad. En el otro caso, es importante escalar la dificultad en la actividad: fácil → difícil.

3. ¿Hay un sistema de recompensas?

Respuesta: sí/no

Recomendado cuando la actividad no posee dificultad estándar o al detectar que los usuarios pueden perder el interés.

Luego:

- Definir Dificultad (estándar/escalada + agrandable)
- Definir Sistema de Recompensas (premio real y/o premio virtual)

7. Definición de los Niveles

Los Niveles sirven para diversificar el DESARROLLO de la actividad, evitando la monotonía y la pérdida del interés por parte de los usuarios. En caso que Identificación sea foco de trabajo, los Niveles permiten repetir elementos del contenido que se está trabajando en distintas instancias, permitiendo que los usuarios identifiquen patrones.

1. ¿Es Identificación un vértice seleccionado?

Respuesta: sí/no

2. ¿La actividad necesita diversificar el ambiente?

Respuesta: sí/no

Los niveles sirven para diversificar y poder moverse entre distintos elementos de trabajo (por ejemplo, distintas materias o mezclarlo con una dificultad escalada).

3. ¿Hay un sistema de progresión en la actividad?

Respuesta: sí/no

Un sistema de progresión es necesario cuando se implementan distintos niveles, sino la sensación que queda es la de estancamiento, independiente de la diversificación.

Finalmente:

- Definir Niveles (cantidad, materia a trabajar, preguntas/ejercicios)
- Definir Sistema de Progresión (progresión real y/o progresión virtual)

3.5.4. Diseño de la actividad III

Esta sección está para agregar cualquier elemento de Gamification del cuál no se haya hecho mención en las secciones I y II, y que pudiese ser necesario para lograr un buen diseño de la actividad gamificada. En caso que ese elemento influya directamente en alguno de los componentes del Framework, entonces se podrá utilizar en la puntuación. En caso contrario, no agregará valor a la puntuación.

Finalmente:

- Definir otros elementos de Gamification

Un resumen de la secuencia se puede ver en la Figura 3.10

3.5.5. Puntuación y figura del Framework

Una vez está diseñada la actividad, hay que asignar los puntos a los vértices de acuerdo a las reglas de puntuación del Framework, las cuales fueron mencionadas anteriormente. Por ejemplo, luego de realizar el proceso de diseño y asignar los puntos correspondientes a una actividad que busca trabajar el Diseño de Algoritmos (Secuencia en el Framework), la lista de vértices queda de la siguiente manera:

- Contexto = 2 puntos

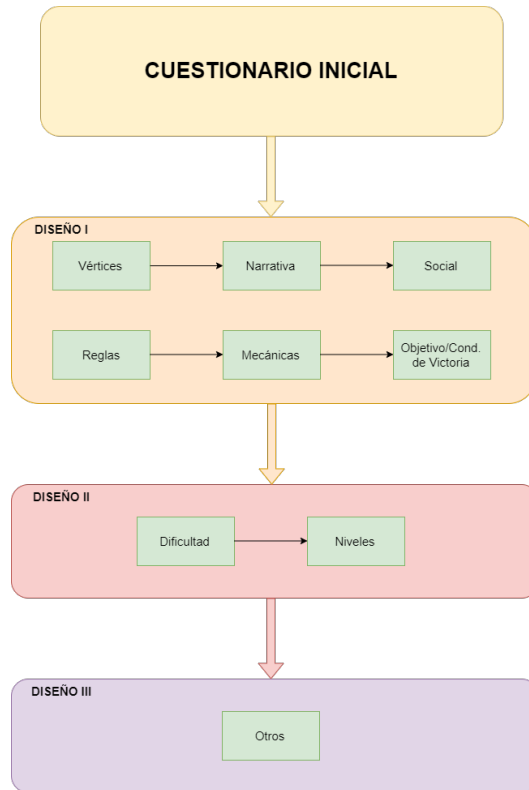


Figura 3.10: Resumen de la secuencia de diseño.

Fuente: Elaboración propia.

- Secuencia = 10 puntos
- Complejidad = 3 puntos
- Identificación = 2 puntos

En base a este puntaje, una representación de acuerdo al Framework se puede ver en la Figura 3.11.

3.6. Diagrama de flujo

A modo de poder sintetizar toda la secuencia de instrucciones y las reglas de puntaje, se presenta un diagrama de flujo estructurado en secciones para tener un acceso facilitado a la información.

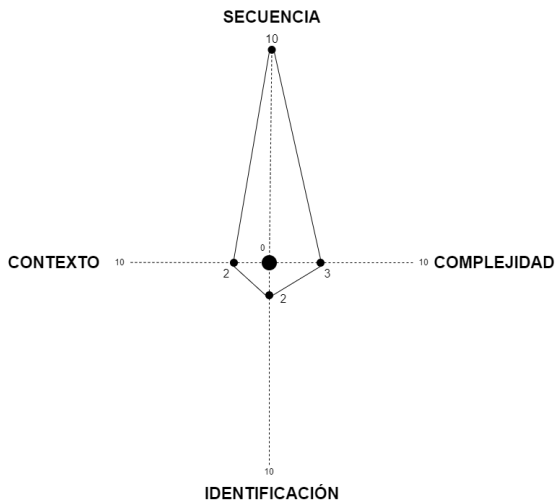


Figura 3.11: Ejemplo de figura en base a actividad.

Fuente: Elaboración propia.

3.6.1. Cuestionario Inicial

Ver Figura 3.12

3.6.2. Diseño I

Ver Figura 3.13

3.6.3. Diseño II, Diseño III y Final

Ver Figura 3.14

3.7. Encuesta de percepción y feedback

La encuesta de percepción y feedback se le entrega a los usuarios una vez finalizada la actividad (después de la etapa FIN). Esta encuesta busca recolectar información valiosa para

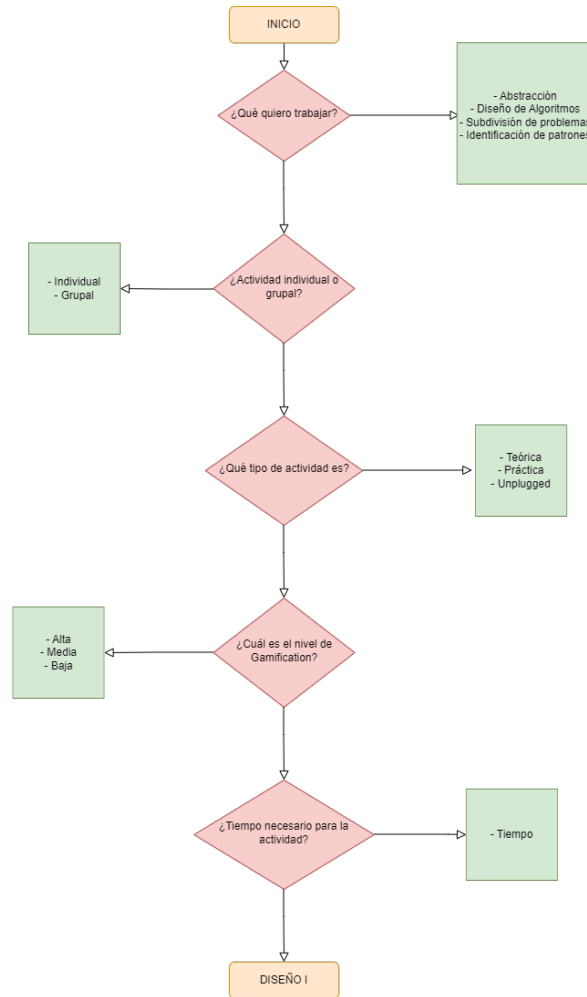


Figura 3.12: Flujo del Cuestionario Inicial.

Fuente: Elaboración propia.

el diseñador, puesto que a través del procesamiento de esta información, el diseñador puede contrastar lo que los usuarios sintieron durante el desarrollo de la actividad.

La encuesta está dividida en tres partes: la primera parte es para consultar directamente por los componentes del Pensamiento Computacional y el Trabajo Colaborativo, la segunda parte es para consultar sobre los elementos de Gamification y la tercera parte es para preguntar por el desarrollo de la actividad en general.



Figura 3.13: Flujo del Diseño I.

Fuente: Elaboración propia.

3.7.1. Sobre el Pensamiento Computacional y el Trabajo Colaborativo

Esta parte de la encuesta es simple y directa. Los participantes seleccionan, según su perspectiva, cada uno de los componentes principales del PC y Trabajo Colaborativo en cuanto a la presencia en la actividad, marcando las opciones “Mucho”, “Regular”, “Parcial”, “Casi nulo”. Además existe el espacio para que expresen, según sus palabras, la forma en que se trabajó dicho componente en la actividad.

Sobre Pensamiento Computacional

Según tu criterio, evalúa las componentes del Pensamiento Computacional de acuerdo a cuánto las trabajaste en la actividad. Si consideras que un componente no se trabajó,

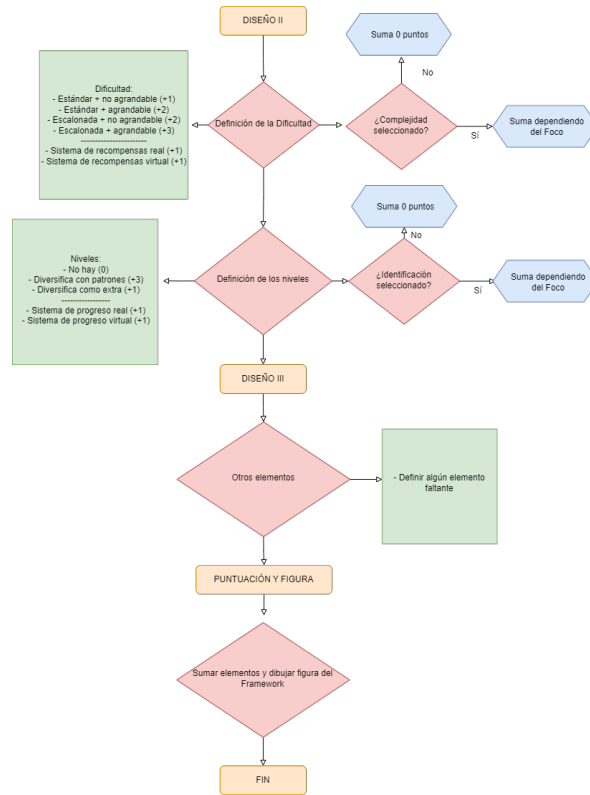


Figura 3.14: Flujo del Diseño II, III y Final.

Fuente: Elaboración propia.

entonces marca la opción "No se trabajó".

Sobre Trabajo Colaborativo (Segundo Nivel)

En cuanto a lo realizado, evalúa con nota entre 1 y 10 cuan presente y qué tan bien implementado estuvo el Trabajo Colaborativo en la actividad.

3.7.2. Sobre Gamification

Aquí, los usuarios contestan lo que percibieron en cuanto a los 4 elementos principales de Gamification utilizados en el Framework: niveles, dificultad, narrativa y las reglas.

Según tu criterio, evalúa las componentes del Pensamiento Computacional de acuerdo a cuánto las trabajaste en la actividad. Si consideras que un componente no se trabajó, entonces marca la opción "No se trabajó". *

	Mucho	Regular	Parcial	Casi nulo	No se trabajó
Abstracción	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Diseño de Algoritmos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Descomposición	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Identificación	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 3.15: Encuesta: Preguntas relacionadas a PC.

Fuente: Elaboración propia.

Si marcaste que la Abstracción estuvo presente, identifica brevemente de qué forma la trabajaste en el desarrollo de la actividad.

Tu respuesta _____

Figura 3.16: Encuesta: Ejemplo de pregunta para información adicional.

Fuente: Elaboración propia.

Debes marcar qué tan de acuerdo estás con las afirmaciones, donde 1 es "Muy en desacuerdo" y 5 es "Muy de acuerdo". En caso que la afirmación no aplique para la actividad, debes marcar 0.

Las preguntas se pueden ver en las Figuras 3.18 y 3.19.

3.7.3. Sobre la actividad

Esta sección busca que los participantes de la actividad entreguen feedback relacionado a cómo resultó la actividad en términos generales. Esta información servirá directamente al diseñador, pues permite ajustar el problema de la subjetividad en cuanto a la percepción propia, permitiéndole diseñar mejores actividades en el futuro. Un ejemplo se puede ver en la Figura 3.20.

En cuanto a lo realizado, evalúa con nota entre 1 y 10 cuan presente y qué tan bien implementado estuvo el Trabajo Colaborativo en la actividad. *

1 2 3 4 5 6 7 8 9 10

Deja tus comentarios sobre lo que crees que estuvo bien y lo que estuvo mal con respecto al Trabajo Colaborativo. *

Tu respuesta

Figura 3.17: Encuesta: Preguntas relacionadas al Trabajo Colaborativo.

Fuente: Elaboración propia.

Otras Preguntas

Estas preguntas son opcionales, y se pueden seleccionar si así el diseñador lo desea. Sirven para despejar algunas dudas que pueda tener el diseñador con respecto a elementos de Gamification específicos. Por ejemplo, se puede preguntar si los incentivos para resolver el problema eran o no adecuados, o si las mecánicas estaban bien o no.

Además, se pueden formular preguntas de estudio, por ejemplo para analizar el rango etario con respecto a la percepción de utilidad en este tipo de actividades.

3.7.4. Posibles Respuestas

Existen dos casos que son importantes para revisar de acuerdo a las posibles respuestas de los estudiantes en la encuesta.

Caso bueno

El caso bueno sería que el resultado sea una figura igual o muy similar a la que le resultó al diseñador. Lo más importante es que los focos principales y secundarios se mantengan, con quizás ± 1 punto, y ± 2 puntos en los otros vértices.

Los niveles de la actividad eran claros e identificables *

0 1 2 3 4 5

No aplica Muy de acuerdo

Si marcaste una opción entre 1 y 5, por favor deja tus comentarios sobre lo que crees que estuvo bien y lo que estuvo mal con respecto a los Niveles. (OPCIONAL)

Tu respuesta _____

La dificultad estaba bien balanceada y escalaba desde algo fácil hacia algo más difícil. *

0 1 2 3 4 5

No aplica Muy de acuerdo

Si marcaste una opción entre 1 y 5, por favor deja tus comentarios sobre lo que crees que estuvo bien y lo que estuvo mal con respecto a la dificultad. (OPCIONAL)

Tu respuesta _____

Figura 3.18: Encuesta: Preguntas relacionadas a Gamification.

Fuente: Elaboración propia.

Caso malo

Un mal caso sería cuando la figura resultado marca que los participantes percibieron algo que no tiene absolutamente nada que ver con lo que intentó hacer el diseñador. Por ejemplo, que el vértice Identificación tenga cerca de 10 puntos cuando al diseñador no le dieron más de 2 (lo que, además, significa que los participantes ni siquiera lograron reconocer de manera correcta los componentes del Pensamiento Computacional).

Las razones por las que esto podría pasar NO necesariamente son responsabilidad del diseñador con respecto a la actividad misma, puesto que puede que haya un problema más atrás: precisamente, que los participantes aún no entienden bien lo que es el Pensamiento Computacional y cómo se compone. En el caso de que sea cosa de la actividad, se le recomienda al diseñador volver a revisar el Framework y la utilidad de los elementos de Gamification y cómo potencian a los componentes del Pensamiento Computacional.

El contexto del problema me pareció interesante/entretenido y el espacio donde se realizó la actividad fue utilizado correctamente. *

0 1 2 3 4 5

No aplica Muy de acuerdo

Si marcaste una opción entre 1 y 5, por favor deja tus comentarios sobre lo que crees que estuvo bien y lo que estuvo mal con respecto al contexto y el uso de espacios. (OPCIONAL)

Tu respuesta

El sistema de reglas me pareció justo, completo y adecuado. *

0 1 2 3 4 5

No aplica Muy de acuerdo

Si marcaste una opción entre 1 y 5, por favor deja tus comentarios sobre lo que crees que estuvo bien y lo que estuvo mal con respecto al sistema de reglas. (OPCIONAL)

Tu respuesta

Figura 3.19: Encuesta: Preguntas relacionadas a Gamification.

Fuente: Elaboración propia.

3.8. El Framework en funcionamiento: un ejemplo demostrativo

Para aclarar las dudas que podrían surgir a la hora de leer y utilizar la guía de diseño de actividades basadas en el Framework de Pensamiento Computacional, se diseña una actividad a modo de ejemplo, mostrando paso a paso la lógica que se utiliza para reconocer elementos y luego puntuarlos.

La actividad que se va a desarrollar consiste en que los estudiantes, separados en grupos y compitiendo entre ellos, deban seguir una serie de secuencias de ejercicios, los cuales pueden ser aeróbicos, anaeróbicos y de estiramiento. Antes de realizar cada una de las series, deberán crear un algoritmo que explique cómo desarrollar cada uno de los ejercicios y,

En esta sección se consulta sobre aspectos generales de la actividad, los cuáles no están necesariamente relacionados con Pensamiento Computacional, Trabajo Colaborativo o Gamification.

¿Qué fue lo más entretenido de la actividad? *

Tu respuesta

¿Qué cosa resultó ser lo más aburrido, fome o tedioso en la actividad? *

Tu respuesta

¿Crees que la cantidad de tiempo para desarrollar la actividad fue suficiente? *
Justifica tu respuesta.

Tu respuesta

¿Cómo mejorarías la actividad? *

Tu respuesta

Si tuvieses que evaluar la actividad con una nota del 1 al 10, ¿Qué nota le pondrías? *

1 2 3 4 5 6 7 8 9 10

Figura 3.20: Encuesta: Preguntas relacionadas a la actividad en general.

Fuente: Elaboración propia.

además, deberán identificar a qué tipo de categoría de ejercicio pertenece.

La actividad se puede desarrollar en el gimnasio o en una cancha, y se necesitan implementos como colchonetas, balones y/o pesas pequeñas. Los estudiantes contarán con un tiempo específico para resolver la parte teórica, y luego con otra cantidad de tiempo para realizar el trabajo físico. En total serán tres secuencias de ejercicios, y ganarán puntos por terminarlas lo antes posible y por resolver bien la parte teórica.

3.8.1. Diseño de la actividad

Cuestionario

- **Qué quiero trabajar?**

Para esta actividad, se busca trabajar el Diseño de Algoritmos como foco principal, y la Identificación de Patrones como foco secundario.

- **¿Es una actividad individual o en grupo?**

Es una actividad grupal, donde participa todo el curso y son separados en equipos. Así, el Trabajo Colaborativo es también un foco.

- **¿Qué tipo de actividad es?**

Es una actividad teórica, práctica y unplugged.

- **¿Cuál es el nivel de Gamification que quiero incluir?**

Esta actividad posee un nivel de Gamification medio.

- **¿Cuanto es el tiempo que necesito para desarrollar la actividad?**

Esta actividad debería desarrollarse en un máximo de 50-60 minutos.

Diseño I

- **Reconocimiento de los vértices y nivel del Framework**

- Diseño de Algoritmos → Secuencia (Primario)
- Reconocimiento de Patrones → Identificación (Secundario)
- Trabajo Colaborativo → Conjunto → Se utiliza entonces el **Segundo Nivel** del Framework

- **Definición de la Narrativa**

- **Espacio Real:** La organización del Espacio Real se da en el gimnasio del colegio, o en su defecto, en una cancha con un espacio grande y adecuado para poder realizar actividad física. El área se divide en cuatro partes para que los cuatro grupos trabajen. Una representación se puede ver en la Figura 3.21.

- **Espacio Imaginario:** No hay.
- **Historia:** No hay.

Grupo 1	Grupo 2
Grupo 3	Grupo 4

Figura 3.21: División del espacio de trabajo (Espacio Real).

Fuente: Elaboración propia.

■ Definición de elementos Sociales

- **Definición de Equipos:** En total serán cuatro equipos, los cuales estarán compuestos por 1/4 de la cantidad de alumnos del curso. La selección de los estudiantes en cada equipo quedarán a cargo del profesor de la materia, de forma que pueda equilibrar los grupos.
- **Sistema de Roles:** Cada equipo tendrá un líder y un sub-líder. El líder se encargará de llevar la cuenta de los ejercicios y los tiempos de recuperación. El sub-líder colaborará con tutorías al realizar los ejercicios. Su trabajo consiste en estar pendiente en el desarrollo de los ejercicios por parte de sus compañeros de equipo, y corregirlos en caso que lo necesite.
- **Sistema de Puntaje:** Habrá dos formas de ganar puntos. La primera es a través de la competencia: luego de cada serie de ejercicios, los equipos se ordenan en función de la cantidad de tiempo que se demoraron en terminar, y se asignan puntos. La segunda es través de la teoría: el profesor se encargará de evaluar los algoritmos y las clasificaciones de ejercicios, y otorgará puntos por la cantidad de respuestas correctas.

■ Definición de las Reglas y Mecánicas

- **Reglas explícitas:**
 - El espacio de juego se divide en cuatro partes iguales, las cuáles serán utilizadas por los equipos de trabajo.

- Los equipos de trabajo estarán compuestos por 1/4 del total de estudiantes. El profesor será el encargado de construir los equipos.
 - Los equipos deben designar, entre sus participantes, a un líder y un sub-líder. El líder será el encargado de llevar la cuenta y los tiempos para resolver los ejercicios (también participa de los ejercicios). El sub-líder colabora como tutor dentro del equipo, ayudando a sus compañeros cuando ellos no saben como realizar los ejercicios de manera adecuada.
 - El profesor, que actúa como monitor durante todo el desarrollo de la actividad, puede detener una realización de ejercicios si nota que los estudiantes no la están realizando correctamente. También puede descontar series que estaban marcadas como realizadas, haciendo que los estudiantes repitan ese determinado ejercicio.
 - La actividad está dividida en tres fases, y cada una cuenta con una parte teórica y una parte de ejercicios. En cada una, los equipos deberán hacer algoritmos y categorizar los ejercicios de acuerdo al tipo: aeróbico, anaeróbico o de estiramiento. Luego, deben realizar los ejercicios.
 - Cada fase de la actividad cuenta con 10 minutos de tiempo máximo para su desarrollo. Si algún equipo no logra terminar ya sea la parte teórica o la parte práctica, verá perjudicado su puntaje.
 - La forma de asignar puntaje es la siguiente: En la parte teórica, cada algoritmo bien diseñado suma 100 puntos, y cada ejercicio bien categorizado suma 10 puntos. En la parte práctica, el puntaje se define de acuerdo al orden en que los equipos terminan las series de ejercicios. El primero en terminar recibe 1000 puntos, el segundo 700, el tercero 500 y el cuarto 300.
 - Dado que es importante cuidar la salud, si un estudiante no se siente bien o siente fatiga realizando la actividad, debe informarle de inmediato al profesor y dejar de realizar los ejercicios. En este tipo de casos, el equipo al que pertenece el estudiante NO se verá perjudicado en puntaje.
- **Reglas implícitas:**
 - Prohibido hacer trampa en cualquiera de sus fases.

- **Mecánicas:** Las mecánicas en este caso son simples, así que no necesitan de un listado. Los estudiantes pueden escribir (para la parte teórica) y pueden realizar los ejercicios. Además, el líder puede tomar ciertas decisiones y el sub-líder puede corregir y ayudar a sus compañeros.

- **Definición del Objetivo y/o Condición de Victoria**

- **Objetivo:** El objetivo de la actividad es lograr terminar las tres secuencias de ejercicios dictadas por el profesor en el tiempo asignado.
- **Condición de Victoria:** No hay. La etapa de Desarrollo de la actividad termina cuando se acaba el tiempo/cuando todos los equipos terminaron.

Diseño II

- **Definición de la Dificultad**

- **Dificultad:** La dificultad es escalonada (más repeticiones en las secuencias posteriores) y no agrandable, pues los ejercicios varían pero manteniendo su base (por ejemplo: abdominales simples y abdominales con peso).
- **Sistema de Recompensa:** El premio real queda a juicio del profesor (puede ser una nota, décimas para pruebas, etc.). El premio virtual es el puntaje obtenido después de cada serie.

- **Definición de los Niveles**

- **Niveles:** Los niveles son las distintas series de ejercicios diseñadas por el profesor.
- **Sistema de Progresión:** La progresión real es que el alumno mejora su estado físico, trabaja el desarrollo de algoritmos y la identificación de patrones, y aprende sobre distintos tipos de ejercicios y categorías en la materia de Educación Física. No hay progresión virtual en la actividad.

Diseño III

Para esta actividad en particular, no hay otro elemento gamificador que sea necesario agregar y definir.

3.8.2. Figura y Puntaje

Ahora corresponde sumar el puntaje de acuerdo a las reglas que utiliza el Framework.

Contexto

- No se trabaja Abstracción = 0 puntos.
- Se considera el Espacio Real = +1 punto.
- No hay otros elementos = 0 puntos.

Total: 1 punto.

Secuencia

- El Diseño de Algoritmos se trabaja y es foco principal = +5 puntos.
- El sistema de reglas no permite libertad a los estudiantes = +5 puntos.

Total: 10 puntos.

Complejidad

- No se trabaja Descomposición = 0 puntos.
- La dificultad es escalonada y no agrandable = +2 puntos.

- El sistema de recompensas contempla una recompensa real y una virtual = +2 puntos.

Total: 4 puntos.

Identificación

- El Reconocimiento de Patrones se trabaja y es foco secundario = +3 puntos.
- Los niveles diversifican y colaboran con patrones = +3 puntos.
- El sistema de progresión cuenta con un progreso real pero no uno virtual = +1 punto.

Total: 7 puntos.

Conjunto

- Se desarrolla el Trabajo Colaborativo = +5 puntos.
- Equipos, sistema de roles, puntaje y competencia = +8 puntos. (Recordar que por regla general, cada elemento suma 2 puntos).

Total: 10 puntos. (Por reglas, 10 puntos es el máximo).

En la Figura 3.22 se puede ver la representación de la actividad basada en el puntaje.

3.8.3. Encuesta de percepción

La actividad no requiere de ningún caso especial con respecto a la encuesta, por lo que se debe diseñar como aparece en el ejemplo de la sección anterior.

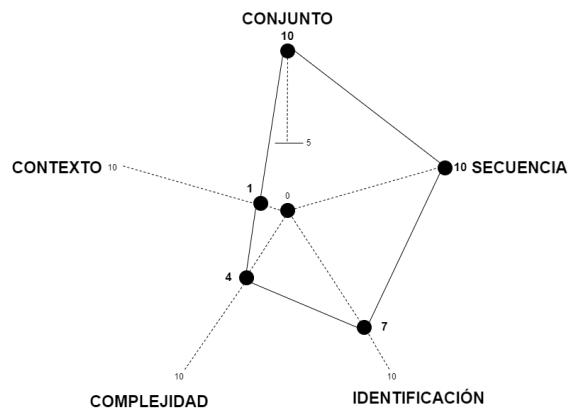


Figura 3.22: Figura de la actividad de para Educación Física basada en el Framework.
Fuente: Elaboración propia.

Capítulo 4

Implementación

En el capítulo anterior se hizo la propuesta del Framework, en donde se presenta su forma y componentes, y se explica el flujo de desarrollo junto con una guía teórica. El problema de esto es que, tal cómo se dice al final, es **teórico**, por lo que se obliga a los usuarios a leer prácticamente todo el trabajo de memoria para poder entender y utilizar el Framework. Además, el material y el flujo deberán llevarlo por su cuenta, lo que podría generar un caos que termine de manera inevitable en errores.

Así, se buscó generar una herramienta basada en el Framework que sirviera como **acercamiento** hacia los usuarios, de manera que bastase con tener los conocimientos básicos de Pensamiento Computacional, Trabajo Colaborativo (opcional) y Gamification. Además, será suficiente haber leído un resumen del Framework, así como los vértices de la figura, los niveles, y los componentes principales de Gamification que se seleccionaron para trabajar.

Dado que no tiene sentido generar un software muy complejo si el Framework aún no se encuentra validado, es que como requerimientos de esta herramienta se consideraron que sea local, que tenga una UI simple y que facilite todos los cálculos y flujo al usuario.

Para hacer esta herramienta, se experimentó con dos tecnologías distintas, siendo exitosa sólo una de ellas.

4.1. Experimentos

4.1.1. Primer experimento: Python 3

Lo primero a probar de forma natural resulta ser Python, pues es un lenguaje de programación que apunta a cubrir la mayoría de los paradigmas de la programación, además de las ventajas de ser un lenguaje de alto nivel y que la cantidad de librerías que posee lo transforma en un lenguaje muy dinámico. Manejar toda la lógica que va por detrás del Framework resultaba bastante simple, pues no había un manejo de datos muy complejo. El mayor de los problemas, y lo que gatilló desechar esta opción, fue el manejo de UI.

Librería 1: PySimpleGUI

PySimpleGUI es una librería de Python que sirve para levantar una interfaz de manera sencilla. Con unas simples líneas de código, se tiene una ventana a la cuál se le pueden colgar elementos para que el usuario interactúe. Además, dentro de los elementos se encuentran todas las cosas que uno usaría al crear un formulario: Checkbox, Textbox, etc.

El problema de esta librería resulta ser la navegación. En un formulario de múltiples páginas, los elementos que aparecen en la interfaz cambian dependiendo de la página seleccionada, para así mantener el flujo del diseñador. En cambio, esta librería no permite hacer cambio de elementos en pantalla “*on the go*”, por lo que la única solución al problema resulta ser levantar distintas ventanas en forma de “*pop up*”. Esto desecha totalmente la idea de utilizar esta librería, pues no es aceptable en términos de UI.

Librería 2: tkinter

tkinter es una librería más completa que PySimpleGUI (de hecho, es un wrapper de esta última) y no tiene los problemas de navegación mencionados anteriormente. El problema de esta librería es el paso de datos entre ventanas. La persistencia de datos es un elemento esencial dentro del formulario, dado que por diseño, las elecciones del usuario en las primeras partes del Framework influyen de manera sustancial en las secciones posteriores. Dado estos

problemas, también se desecha la idea de utilizar esta librería, y con esto, la idea de utilizar Python como tecnología para desarrollar la herramienta.

4.1.2. Segundo experimento: Microsoft Excel y Visual Basic

Con Microsoft Excel, la navegación y la persistencia de datos están cubiertas a través de las Hojas de Cálculo, pero el programa por sí solo no permite habilitar y bloquear elementos en base a las selecciones del usuario. Frente a este problema es que aparece Visual Basic, lenguaje que permite aplicar lógica al programa y manejar todo el sistema de widgets. Con esto, la implementación de la herramienta se puede realizar sin problemas, la cual se detalla a continuación.

4.2. Herramienta basada en el Framework

Habiendo seleccionado la tecnología a utilizar, corresponde desarrollar la herramienta pensando en el flujo que sigue el Framework. Obviamente, se considera agregar una parte con teoría previa (sobre la materia tratada y sobre el Framework) e instrucciones de uso.

La herramienta consta de tres partes: La parte teórica, la parte de diseño y la parte comparativa.

Todas las imágenes de esta sección están en el Anexo B del capítulo 6. Anexos.

4.2.1. Primera parte: Bienvenida, Teoría Previa e Instrucciones de Uso

Lo primero que visualiza el usuario al abrir el programa es una hoja de bienvenida, la cual cuenta con dos opciones. La primera es un botón para pasar por la parte de la teoría previa y las instrucciones de uso, indicando que es la opción recomendada para quienes utilizan el programa por primera vez. En cambio, la segunda opción permite al usuario saltar directamente al cuestionario inicial, que es el inicio del diseño (segunda parte del programa).

La hoja de bienvenida se ve como aparece en la imagen 6.1.

Si el usuario presiona el botón COMENZAR, entonces aparece la pantalla de teoría previa. En esta pantalla no hay ninguna interacción del usuario además de los botones para avanzar o continuar, y el objetivo de esta sección del programa es dotar al usuario de la información con la que se sustenta el Framework, tanto en Pensamiento Computacional como Gamification. La imagen 6.2 muestra cómo se ve la pantalla de teoría previa.

Al avanzar con el botón CONTINUAR, aparece la pantalla de teoría del framework. Esta sección del programa tiene un resumen de toda la información necesaria para entender cómo opera el Framework, sus niveles y la forma de puntuación. La primera parte resulta esencial para el usuario, pues sin entender de qué se trata el Framework, es poco probable que pueda sacarle provecho al programa (y a cualquier otro programa basado en el Framework). Esto se puede ver en las imágenes 6.3 y 6.4.

La última parte es de puntuación, y esta está ahí para ilustrar al usuario de dónde salen los números en la asignación de puntaje. Eso sí, es el programa quién se encarga de realizar la asignación de puntaje de manera automática, por lo que el usuario no tiene la necesidad de memorizar ningún elemento o regla de esta parte. Esto se ve en la imagen 6.5

Finalmente, con el botón CONTINUAR se llega a la pantalla de instrucciones de uso, la cual corresponde a la última pantalla de esta parte del programa. Esta se divide en dos partes: primero, aparece el flujo del programa en base a sus pantallas, considerando la hoja de cuestionario inicial como inicio. La imagen 6.6 muestra el flujo del programa.

La segunda parte de esta última pantalla son las instrucciones de uso como tal. Vienen incluidos “*dummies*” de los widgets que el usuario puede encontrar a lo largo del programa, así puede interactuar con ellos si es que no está familiarizado con este tipo de objetos en Excel. Esto se puede ver en la imagen 6.7.

4.2.2. Segunda parte: Diseño y Hoja de Puntaje

Esta es la parte donde inicia oficialmente la implementación del Framework como herramienta. La primera pantalla corresponde a la de cuestionario inicial, y es donde el usuario pone las bases del diseño de su actividad. Esto se puede ver en la imagen 6.8.

En la imagen 6.9 se puede ver la segunda pantalla de esta parte: pensamiento computacional y trabajo colaborativo. Dependiendo de las elecciones que haya realizado el usuario en la pantalla anterior, acá corresponde marcar cuales serán sus focos de trabajo con respecto a los vértices del Framework. En el caso del Trabajo Colaborativo, esto se hace de manera automática.

Luego vienen las partes de Gamification, las cuales se dividen en dos pantallas: diseño I y diseño II. El orden en que aparecen las secciones asociadas a cada vértice no sigue ninguna regla, y perfectamente podría haber estado una sección de diseño I en la de diseño II y viceversa. La razón de hacer una separación de los vértices en dos pantallas distintas reside en la necesidad de no exponer al usuario a un cuestionario extenso en una sola pantalla, sumado a que se debe decidir si seleccionar o no varias opciones distintas.

Las imágenes 6.10, 6.11 y 6.12 muestran las respectivas pantallas.

Finalmente, la segunda parte termina con la pantalla de hoja de puntaje. Esta pantalla se encarga de recolectar y sumar el puntaje de acuerdo a las opciones que el usuario fue marcando durante las etapas anteriores. Esta pantalla además muestra alertas (en forma de notas) cuando encuentra ciertos posibles errores (por ejemplo, que un foco de trabajo primario no tenga el mayor puntaje final) y la opción de generar un gráfico para tener una representación visual contrapuesta con la figura perfecta del nivel del Framework donde se está trabajando. Esto se puede ver en la imagen 6.13.

4.2.3. Tercera parte: Encuesta y Comparativa

Esta es la última parte del programa, y sirve para cumplir uno de los objetivos de esta propuesta. Cualquiera sea la actividad que diseñe el usuario, nunca sabrá si la actividad - en la práctica en el mundo real- funcionó como estaba planeada. Lo más importante para aprender y mejorar en el diseño de este tipo de actividades es con feedback directo de parte de los participantes de la actividad.

Así, esta parte se divide en tres pantallas, sumado a lo que sería una encuesta que se realiza fuera del programa (en este caso, utilizando Google Forms).

La primera parte es la pantalla de instrucciones de la encuesta. En esta se instruye al usuario de cómo realizar el ingreso de los datos en la pantalla siguiente, esto para que la macro que realiza el trabajo de obtener los promedios y otros cálculos no falle. Esto se puede ver en la imagen 6.14.

De acuerdo al flujo, en esta sección del Framework el usuario debería haber realizado la actividad en un escenario real, por lo que una vez finalizada, debería recolectar feedback a través de una encuesta. En la imagen 6.15 se puede ver una parte de una encuesta tipo del Framework, la cual es útil para el programa. Obviamente, la encuesta recorre todos los vértices, no sólo lo que sale en la imagen.

Teniendo los resultados y habiendo leído las instrucciones, el usuario debe ingresar los datos en la planilla y luego presionar el botón de CALCULAR PROMEDIOS para que la macro haga su trabajo, y así poder continuar a la última pantalla del programa. La imagen 6.16 muestra la planilla.

Y para terminar el programa, la última pantalla es la comparativa. En esta pantalla el usuario puede contrastar su percepción de la actividad en el momento del diseño de esta, contra la percepción promedio de los participantes de la actividad. Además, el gráfico que se puede generar ahora muestra la representación visual de ambas percepciones graficadas, junto con la figura perfecta del nivel correspondiente del Framework. La imagen 6.17 muestra la pantalla.

4.3. Aprendizajes y modificaciones

La primera versión del programa se desarrolló pensando exclusivamente en la funcionalidad de los elementos que lo integran, lo que decantó en serios problemas de UI/UX para cualquier usuario.

Luego de realizar un testeo para recopilar datos de uso (y no del Framework), se obtuvo lo siguiente:

- **Color:** Es necesario agregar un fondo de color para generar contraste con los espacios donde se ubican los elementos y la información en general.
- **Navegación espacial:** La disposición de los elementos obliga al usuario a usar tanto el eje X (horizontal) como el eje Y (vertical), teniendo que utilizar el mouse para generar ambos movimientos y poder cubrir todos los espacios de elementos.
- **Delimitación:** Al no existir una forma gráfica que defina los límites de cada pantalla, es el usuario quién debe verificar donde están los finales de cada eje, haciendo que en cada avance entre pantallas exista la duda de si efectivamente revisó todos los espacios o no.

Para ejemplificar, se muestra la imagen 4.1 que corresponde a la primera versión del programa con la que se realizó el testeo.

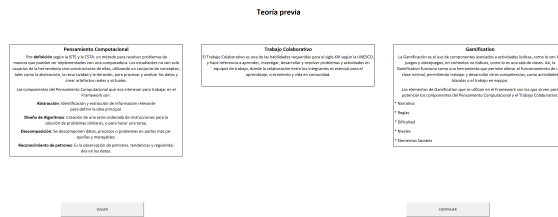


Figura 4.1: Pantalla Teoría Previa V1.0.

Para solucionar el tema del color, se asignó la tonalidad azul clara y se dejaron los espacios con fondo blanco y margen negro. Para la navegación, se desplazaron todos los espacios hacia la izquierda y hacia abajo, haciendo desaparecer el movimiento en el eje horizontal.

Finalmente, para la delimitación, se encerró todo el espacio que contiene información dentro de otro margen negro, marcando así los límites de la pantalla para el usuario. Como ejemplo, se puede ver cualquier imagen de este capítulo.

4.4. Validación del Trabajo

Para realizar la validación, se generaron dos instancias con usuarios distintos y con una mecánica distinta, para así obtener distinto feedback con respecto al Framework. En ambas instancias se utilizó el software en Excel, pero una consistió en una visita guiada mientras que la otra fue el usuario quién utilizó la herramienta posterior a la explicación del Framework y los componentes principales: Pensamiento Computacional y Gamification.

4.4.1. Primera instancia: Usuario utiliza la herramienta con estrategia Think Aloud

El método Think-Aloud es una herramienta que se utiliza con distintos propósitos. Uno de ellos es la enseñanza, otro es una técnica de inspección de usabilidad y otro es una técnica para realizar una validación. Consiste en que el usuario que está sometido a la prueba debe “pensar en voz alta”, esto significa que a medida que avanza en el desarrollo de la prueba, se le pide que entregue a viva voz sus comentarios y opiniones con respecto a lo que está realizando. Si el objetivo fuese testear usabilidad, el usuario, al contestar una encuesta creada en base a un formulario, podría comentar sobre el color del formulario, el tamaño y fuente de las letras, el color de las letras y el espaciado entre cada pregunta. Por otro lado, en el caso de la validación, el usuario podría comentar sobre las preguntas y el por qué está seleccionando determinada respuesta. Eso sí, aunque no son excluyentes entre ellas, la persona encargada debe conducir los comentarios hacia lo que quiere demostrar.

Esta metodología tiene beneficios muy buenos, entre ellos la simpleza de la prueba (pues sólo se necesita un usuario y acceso a lo que se quiere testear) y el hecho que permite entender -de cierta forma- el mapa mental del usuario a la hora de resolver la tarea. También hay que considerar los factores negativos, como son la poca objetividad cuando se revisan los

resultados, sumado a que todo se responde de manera perceptiva, por lo que al no haber una pauta directa de preguntas, lo que se obtienen son distintas aristas (que a veces coinciden, a veces no) sobre un tema en particular.

Para la primera instancia, el usuario fue un estudiante e último año de la carrera de Pedagogía en Música. Hombre, 25 años, y que además, fue estudiante de Ingeniería Civil Informática de esta misma casa de estudios (hasta tercer año), por lo que ya contaba con conocimientos sobre programación y tecnología en general. Eso último significó que la explicación asociada a Pensamiento Computacional quedó clara rápidamente (no más de 10 minutos). Sumado a esto, el usuario es, además, aficionado a los videojuegos, por lo que los conceptos de Gamification asociados al diseño de videojuegos estaban, también, en parte bien claros, y la explicación tampoco tomó más de 10 minutos. Lo que más tiempo utilizó fue la explicación del Framework, pues son varias cosas a revisar y tener en cuenta. En total, la explicación tomó unos 25 minutos. Todo esto se realizó por la plataforma Zoom.

Luego de la explicación, al usuario se le presentó un caso de prueba (una actividad asociada a su asignatura: música), con el que hizo uso de la herramienta en Excel. Mientras iba llenando la planilla con sus opciones, se le pidió que pensara en voz alta y justificara sus opciones. Esto en el software no es un requisito, pero se pidió para tener en claro que, efectivamente, el usuario entendía los conceptos y lo que estaba haciendo en el desarrollo. Esta parte de la prueba tomó aproximadamente 30 minutos más.

De esta parte, se detectó que el usuario tuvo algunos problemas para moverse dentro del programa. Los límites no estaban bien definidos y eso hacía que tuviese que revisar más de una vez si estaba dejando algo atrás al cambiar de página. En términos del Framework, en la mayoría de las veces entendía a la primera qué era lo que se le estaba preguntando o la referencia que había hacia cierto elemento, mientras que en algunas partes tuvo que ayudarse con los comentarios de guía que estaban asociados a las celdas. Además pasó que el usuario tenía, en algunos puntos, diferentes perspectivas a las del autor de este trabajo. Este punto es esencial, pues demuestra que la percepción es vital en el diseño de actividades.

Una vez terminado el uso de la herramienta, se le pidió al usuario que entregara feedback final sobre lo que él pensaba que estaba bien y mal. Esta parte duró 15 minutos.

De acuerdo a esto, se rescató lo siguiente:

- Desde su perspectiva pedagógica, el Framework resulta interesante para ser aplicado a distintas asignaturas.
- La dificultad para utilizarlo es baja, pero en los primeros intentos podría resultar tedioso y lento dada la cantidad de información que hay que manejar.
- Visualmente la herramienta necesitaba mejoras de contraste y delimitación de elementos (esta instancia se realizó con la versión 1.0 de la herramienta).
- La actividad (caso de prueba) era mala a nivel pedagógico. Aún así, dado que el diseño de la actividad fue llevado a cabo por una persona que no es profesor, se entendió que su uso está asociado a testear la herramienta y establecer la utilidad del Framework.

4.4.2. Segunda instancia: Usuarios ven la herramienta en uso a través de visita guiada y juicio experto

Los usuarios de esta instancia fueron profesores universitarios, hombre y mujer, los cuales son expertos en el tema de Pensamiento Computacional. Para poder realizar la visita guiada como corresponde, la validación se dividió en dos partes distintas y atemporales: una teórica y otra práctica. La parte teórica consistió en que los profesores tuvieron tiempo para leer el presente trabajo de memoria, con principal énfasis en el problema identificado y después el Framework, su funcionamiento y puntuación como propuesta de solución. Esta parte tuvo una duración de 2-3 semanas, pues había que leer y entender tantas cosas, sumado a que los profesores se encontraban en pleno desarrollo del semestre académico, por lo que su tiempo disponible también era acotado. Luego se inicia la parte práctica, en la que, utilizando la plataforma Zoom, se les explica (por si algo no hubiese quedado claro en la lectura) en qué consiste el Framework, los elementos de Pensamiento Computacional seleccionados para trabajar y los componentes de Gamification. A diferencia de la primera instancia de validación, acá la parte de Gamification requirió de un poco más de tiempo para la explicación, mientras que lo de Pensamiento Computacional fluyó de manera más expedita. Esta explicación tomó, en promedio, 30 minutos.

Una vez explicado todo, se procede a utilizar la herramienta de Excel con el mismo caso

de prueba que en la instancia anterior, pero en este caso quién utiliza la herramienta es el autor de este trabajo. La visita guiada se enfocó en explicar segmento por segmento y elemento por elemento las secciones del software, para que así quedara claro cómo es que opera el Framework al diseñar actividades y cuáles son los criterios para ir asignando puntaje, abstrayéndose de las reglas de puntuación. En este caso, y a diferencia de la técnica Think-Aloud, los profesores se mantuvieron en silencio hasta la finalización de la explicación, para luego tener micrófono abierto y entregar todo el feedback que estimasen necesario, además de realizar preguntas para generar una instancia de conversación con el autor. Esta sección de la parte práctica tomó entre 30-40 minutos.

A continuación se detalla lo rescatable de esta instancia de validación:

- El Framework hace algo que no existe actualmente, que es lograr unir los elementos de Pensamiento Computacional con los grupos de componentes de Gamification. Ahí es donde está lo más importante de este trabajo.
- Lograr entender el Framework en primera instancia toma mucho tiempo, lo que aumenta el tiempo de uso. Aún así, este tiempo debería reducirse a medida que aumenta el uso de la herramienta (u otra herramienta basada en el Framework).
- Por los problemas que hubo para hacer funcionar el software en Excel en diversos computadores, la herramienta podría haber sido desarrollada en otra plataforma o con otra tecnología. Se discute que esto está considerado para el trabajo futuro.

4.4.3. Otros

Dado que el foco oficial de este Framework es la educación básica (aunque está generalizado y debería funcionar para casi cualquier caso), se intentó contactar a profesores de enseñanza básica. Lamentablemente para este trabajo, los profesores que calificaban como sujetos de prueba eran, específicamente, los que tuviesen conocimiento de Pensamiento Computacional, y eso redujo ampliamente el espacio de candidatos. Aún así, se intentó contactar a profesores del norte del país, los cuales habían participado anteriormente en talleres de Pensamiento Computacional, junto con otros profesores de la V Región, pero ninguno

respondió a nuestra solicitud. Esto dejó parte de la validación en deuda, pero con lo que se alcanzó a testear (con esos sujetos de prueba), debería bastar como un mínimo para lo que se busca con la propuesta.

Capítulo 5

Conclusiones

Para finalizar este trabajo, resulta imperativo hablar acerca de los alcances y aprendizajes que dejó todo el diseño y desarrollo del Framework, además de proponer una mirada hacia el futuro con respecto a este trabajo.

5.1. Aprendizajes y limitantes

Al revisar el trabajo, lo primero que salta a la vista - y de forma obvia- resultan ser los conceptos de Pensamiento Computacional y de Gamification. El primero, en un principio, era totalmente desconocido por el autor, pese a que sus componentes (como la abstracción o el diseño de algoritmos) resultaron ser un elemento clave en el avance de la carrera. Esto se debe al desconocimiento general que existe debido a las razones expuestas en los primeros capítulos de este trabajo, siendo el más importante el que aún no hay un plan de inclusión en la malla regular de enseñanza en la educación básica, quedando -en algunos casos- marginado a una actividad opcional o extra curricular. Aún así, la exhaustiva investigación permitió no sólo conocer y comprender el concepto, sino también entender la importancia que tiene el desarrollar estas habilidades para la vida cotidiana. No son sólo problemas con un alto grado de complejidad, o los que requieren de una máquina para ser resueltos, sino que casi cualquier tipo de problema, de casi cualquier índole, utiliza al menos una componente del

Pensamiento Computacional.

En el caso de la Gamification, dado que su génesis ocurre en el diseño de juegos y videojuegos, y sumado a que es un concepto que está de moda hoy en día, resultó ser más fácil de investigar y entender, pero no así de trabajar. Al igual que cualquier área de estudio, no basta con sólo revisar la teoría. Es esencial poner en práctica los conocimientos, y en eso la Gamification no queda exenta. Y lamentablemente, por una cosa (probablemente) generacional, es que para comprender bien los conceptos es imperativo el tener que jugar. Al participar e interactuar en actividades de carácter lúdico es cuando se logra un dominio de los conceptos que se utilizan en Gamification, y esto ocurre debido a que se logra visualizar la forma en que estos se implementan y la sinergia que existe entre ellos cuando coexisten en una actividad.

Por otro lado, el Framework es la prueba de que se pueden integrar estos dos conceptos que pertenecen a mundos totalmente distintos, siguiendo una serie de instrucciones o patrones. La bibliografía ya había demostrado, a través de distintos experimentos, que estos dos conceptos podían mezclarse para generar instancias gamificadas en entornos serios, pero todos los autores utilizaban los elementos de Gamification de manera distinta y sin mucho énfasis en responder los *por qué*. Ahora, gracias a la existencia del Framework, existe una guía que le permite a los educadores saber qué elementos de Gamification utilizar dependiendo de lo que quieren trabajar con respecto a Pensamiento Computacional, y la forma en que estos ayudan a potenciar el trabajo. Además, el Trabajo Colaborativo como un extra permite generar instancias gamificadas que, según resultados de distintos experimentos revisados en la bibliografía, funcionan mejor y obtienen mejores resultados en el aprendizaje y en el desarrollo de las habilidades.

Finalmente para esta sección, corresponde hablar sobre los objetivos planteados al inicio del trabajo. El objetivo principal se completó en su totalidad, pues lo que se buscaba era diseñar un Framework y eso quedó demostrado a lo largo de este trabajo. Con respecto a los objetivos específicos, el Framework y su respaldo teórico demostraron que es posible encontrar un punto de encuentro -una sinergia- entre los elementos del Pensamiento Computacional y los de Gamification, pues gracias a la separación granular de los conceptos y luego re-unión entre ellos, se encontró una forma donde no sólo coexisten, sino que además se potencian. Por otro lado, la validación mostró que la herramienta basada en el Framework (y

creada en Excel) efectivamente facilita a los profesores a integrar los elementos de Gamification, aunque la validación dejó al debe una confirmación mejor elaborada. La herramienta también le permite al profesor contrastar lo que entiende de su propio diseño con lo que los estudiantes perciben, permitiéndole obtener feedback directo y una oportunidad de mejora en un trabajo que requiere de iteración para encontrar su mejor versión.

5.2. Trabajo Futuro

Dado que el Framework ya se encuentra desarrollado, sería interesante continuar con el trabajo en al menos tres ámbitos. En primer lugar, a nivel teórico, se podrían agrandar los subgrupos de Gamification seleccionados, o se podrían integrar grupos nuevos que permitan una mayor cohesión de las actividades gamificadas. Dado que siempre resulta riesgoso caer en lo repetitivo, añadir (o agrandar) subgrupos permite aumentar el abanico de opciones para los educadores. Eso sí, esto puede tener complicaciones, pues seguir añadiendo más cosas al Framework podría entorpecer el uso generar más problemas que soluciones. En segundo lugar, y algo que quedó al debe en este trabajo, corresponde a generar nuevas iteraciones con -ojalá- casos reales del Framework y profesores de la educación básica del país. Con esto, se podría pulir los vértices del Framework en cosas que, quizás, no se vieron en las iteraciones realizadas en este trabajo y en el desarrollo mismo. Finalmente, el último ámbito está asociado al software. La herramienta desarrollada en Excel cumple con lo necesario para trabajar con el Framework, pero claramente no es la mejor tecnología a utilizar, sobretodo considerando el potencial que tiene el Framework. Por ejemplo, una aplicación interesante sería una que haga lo mismo que abarca la herramienta en Excel, pero que funcione por web. Y, además, esta aplicación podría tener acceso con cuentas institucionales, permitiría compartir las actividades con sus respectivas descripciones y los resultados obtenidos, y levantaría la encuesta para los estudiantes dentro de la misma plataforma, de manera que no haya que operar con agentes externos. También podría recolectar datos de los estudiantes y reportar métricas a los educadores, para así obtener alertas de los estudiantes que estén teniendo problemas en las actividades.

Capítulo 6

Anexos

6.1. ANEXO A: Actividad-tipo de acuerdo al vértice

Las actividades tipo (o estándar) son instancias que ya se encuentran pre-diseñadas en su base, pero que permiten cierto grado de dinamismo para que el diseñador pueda hacer cambios en ellas sin alterar su núcleo principal, sobretodo en lo que respecta a los elementos de Gamification.

Para hacer más amena la búsqueda al lector, la separación se hace en base a tres categorías presentes en el aula de clase: Cursos científicos/matemáticos, Cursos humanistas/ciencias sociales, Arte/Tecnología y Educación Física.

- **Científicos/Matemáticos**

- Problemas de enunciado

La actividad más básica para trabajar Pensamiento Computacional, y también la más potente de todas. Permite potenciar las cuatro habilidades principales, pero tiene la desventaja que, así como se plantea, no tiene gamificación. Consiste en plantear al usuario de un problema camuflado dentro de un enunciado, haciendo que deba analizar y representar distintos tipos de información, crear algoritmos para resolver el problema, dividir el problema en diversos subproblemas más

abordables, y poder identificar patrones para utilizar la misma materia en distintas instancias.

En resumen:

- Tipo de actividad: teórica
- Nivel del Framework: ambos
- Componentes del PC: todos
- Requerimiento de tecnología: no
- Nivel de Gamification: alto
- Elementos de Gamification: narrativa, niveles, dificultad, reglas, social.

- Programación

Aprender a programar se ha vuelto esencial para el mundo moderno. Antiguamente, la programación era cosa de adultos, y sólo en los lugares con los mejores modelos educativos del mundo se enseñaba a los jóvenes, un poco antes que partieran a la universidad. Gracias al avance de la tecnología, existen diversos recursos online que permiten aprender a programar desde pequeños, en distintos niveles y con distintas directrices. Además, estos recursos permiten que los educadores sean los mismos diseñadores de las actividades, aplicando también elementos de gamification. También se pueden crear instancias más complejas al añadir otra clase de elementos, como por ejemplo, la programación sobre Arduino para aplicar en la robótica.

En resumen:

- Tipo de actividad: práctica
- Nivel del Framework: ambos
- Componentes del PC: todos
- Requerimiento de tecnología: sí
- Nivel de Gamification: alto
- Elementos de Gamification: narrativa, niveles, dificultad, reglas, social.

- Juego de preguntas

El juego de preguntas es una actividad que se puede desarrollar tanto individual como grupal (pero es mejor grupal) y consiste en una competencia, con puntaje,

donde los equipos deben diseñar un set de preguntas en base a cierto contenido dictado por el profesor, quién además cumple el rol de moderador. Una vez todos los equipos terminan de diseñar sus preguntas, el profesor las recopila, las lee, y selecciona las mejores de acuerdo a su juicio (buenos criterios de selección es que no sean preguntas *rebuscadas*, que estén bien planteadas y que sean acordes a la materia en cuestión). Por cada pregunta seleccionada, la otorga un punto al grupo que diseñó la pregunta. Finalmente, interroga a los grupos, uno por uno, con una pregunta que haya diseñado un equipo distinto. Por cada respuesta correcta otorga un punto, y gana el equipo que junta la mayor cantidad de puntos. Existe un recurso online para realizar esta actividad utilizando tecnología.

- Tipo de actividad: teórica/práctica
- Nivel del Framework: ambos
- Componentes del PC: abstracción, diseño de algoritmos, reconocimiento de patrones
- Requerimiento de tecnología: no
- Nivel de Gamification: medio
- Elementos de Gamification: narrativa, niveles, reglas, social.

■ **Humanistas y Ciencias Sociales**

- Juego de preguntas

Explicado en el punto anterior.

- Mezcla de poesía

Esta es una actividad práctica de Segundo Nivel, y consiste en que distintos equipos deben trabajar con distintos tipos de poesía (por lo que la materia sobre poesía y los distintos estilos ya debe estar revisada). Una vez conformados los equipos, el profesor, quién también actúa como moderador, le asigna a cada grupo un estilo de poesía (EN SECRETO) y le asigna a todos los equipos EL MISMO tema para trabajar. Los equipos deben hacer dos cosas: primero, deben escribir una secuencia de pasos para detallar cómo se compone el estilo que les tocó trabajar. Cuando terminan, deben pedirle al profesor que lo revise. Luego, en segundo

lugar, deben componer una poesía, entre todos, utilizando los elementos del estilo correspondiente. Cuando todos los equipos terminan, el profesor intercambia las poesías entre los grupos y los integrantes deben hacer dos cosas: primero, identificar el estilo de la poesía, y segundo, escribir una secuencia de pasos para escribir poesía con ese estilo. Finalmente, los equipos comparan las secuencias de pasos.

- Tipo de actividad: teórica/práctica
- Nivel del Framework: segundo nivel
- Componentes del PC: abstracción, diseño de algoritmos
- Requerimiento de tecnología: no
- Nivel de Gamification: medio
- Elementos de Gamification: narrativa, reglas, social.

■ **Arte/Tecnología**

- Juego de apreciación

Esta actividad se puede desarrollar tanto de forma individual como grupal. Consiste en una competencia donde el profesor, luego de haber revisado la materia de distintos estilos de arte de acuerdo a la época, le muestra a los participantes un set de obras de distintas épocas, y estos últimos deben identificar el estilo al que pertenecen.

- Tipo de actividad: práctica
- Nivel del Framework: ambos
- Componentes del PC: reconocimiento de patrones
- Requerimiento de tecnología: no
- Nivel de Gamification: bajo
- Elementos de Gamification: niveles, reglas, social.

- Dibujo Abstracto

Esta actividad se puede desarrollar tanto como una competencia como una actividad con baja gamificación. Consiste en que los participantes deben crear un

dibujo utilizando arte abstracto, ya sea con un tema libre o un tema general dado por el profesor. Una vez que todos terminen su dibujo, la clase se dedica a apreciar las obras de todos los compañeros y deben intentar entender el tema del dibujo y la intención del autor al realizar la obra.

- Tipo de actividad: práctica
- Nivel del Framework: primer nivel
- Componentes del PC: abstracción, reconocimiento de patrones
- Requerimiento de tecnología: no
- Nivel de Gamification: bajo/medio
- Elementos de Gamification: narrativa, niveles, reglas.

■ Educación Física

- Escape Room (Actividad kinestésica y unplugged)

Las Escape Rooms se han puesto de moda en el último tiempo, y consisten en un juego con límite de tiempo donde los participantes quedan encerrados en una pieza cerrada con un candado. La única forma de escapar de la sala es abriendo el candado, y para eso, deben resolver distintos problemas y misterios dentro de la sala hasta encontrar con el paradero de la llave.

En este caso, la Escape Room puede ser el gimnasio del colegio o un sector delimitado al aire libre (y no, no es necesario el candado), y los estudiantes (todo el curso participa como un equipo) deben realizar ciertas actividades seleccionadas por el profesor, para con eso ganar puntos. Los estudiantes logran escapar de la Escape Room si juntan, al menos, la cantidad de puntos definidos por el diseñador. Las actividades que pueden realizar para ganar puntos son las mismas que se realizarían en cualquier clase de Educación Física (correr, elongar, trabajo físico, etc) pero antes de realizarlas, los jugadores deben diseñar secuencias de ejercicios, llamados desafíos, y secuencias de pasos de cómo realizar esos ejercicios. Además, para entusiasmar al trabajo en equipo, hay premio doble en puntos si quienes realizan los desafíos son los estudiantes que en general no tienen buen rendimiento en esta clase.

- Tipo de actividad: práctica

- Nivel del Framework: segundo nivel
- Componentes del PC: diseño de algoritmos, subdivisión de problemas
- Requerimiento de tecnología: no
- Nivel de Gamification: alto
- Elementos de Gamification: narrativa, niveles, dificultad, reglas, social.

Todas estas actividades se pueden utilizar tal cual están escritas, editadas e incluso mezcladas. Se pueden crear otras actividades basadas en estas actividades también. Lo importante siempre es considerar los componentes del PC a las que apuntan a trabajar.

6.2. ANEXO B: Imagenes de la herramienta basada en el Framework

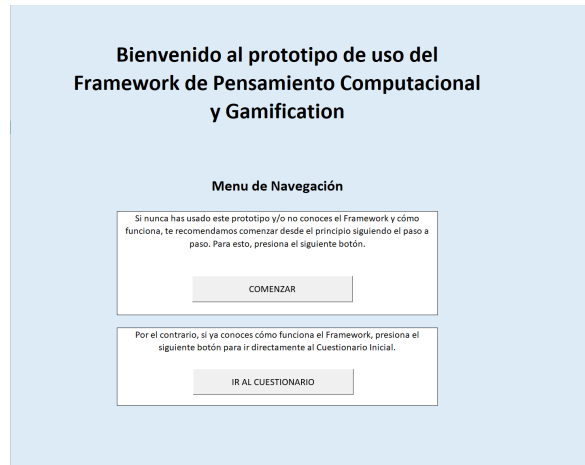


Figura 6.1: Pantalla de Bienvenida.

Teoría previa

Pensamiento Computacional

Por **definición** según la ISTE y la CSTA: un método para resolver problemas de manera que puedan ser implementados con una computadora. Los estudiantes no son solo usuarios de la herramienta sino constructores de ellas; utilizando un conjunto de conceptos, tales como la abstracción, la recursividad y la iteración, para procesar y analizar los datos y crear artefactos reales y virtuales.

Las componentes del Pensamiento Computacional que nos interesan para trabajar en el Framework son:

Abstracción: Identificación y extracción de información relevante para definir la idea principal.

Diseño de Algoritmos: Creación de una serie ordenada de instrucciones para la solución de problemas similares, o para hacer una tarea.

Descomposición: Se descomponen datos, procesos o problemas en partes más pequeñas y manejables.

Reconocimiento de patrones: Es la observación de patrones, tendencias y regularidades en los datos.

Trabajo Colaborativo

El Trabajo Colaborativo es una de las habilidades requeridas para el siglo XXI según la UNESCO, y hace referencia a aprender, investigar, desarrollar y resolver problemas y actividades en equipos de trabajo, donde la colaboración entre los integrantes es esencial para el aprendizaje, crecimiento y vida en comunidad.

Para que el trabajo sea verdaderamente colaborativo (y no una instancia de pseudoaprendizaje o trabajo competitivo) se debe cumplir con la presencia de los elementos cooperativos, los cuales son: interdependencia positiva, responsabilidad individual, interacción estimuladora técnicas interpersonales y de equipo, y evaluación grupal.

Gamification

La Gamificación es el uso de componentes asociados a actividades lúdicas, como lo son los juegos y videojuegos, en contextos no lúdicos, como lo es una sala de clases. Así, la Gamification funciona como una herramienta que permite alterar el funcionamiento de una clase normal, permitiendo trabajar y desarrollar otras competencias, como actividades blandas o el trabajo en equipo.

Los elementos de Gamification que se utilizan en el Framework son los que sirven para potenciar los componentes del Pensamiento Computacional y el Trabajo Colaborativo:

- * Narrativa
- * Reglas
- * Dificultad
- * Niveles
- * Elementos Sociales

VOLVER

CONTINUAR

Figura 6.2: Pantalla de Teoría Previa.

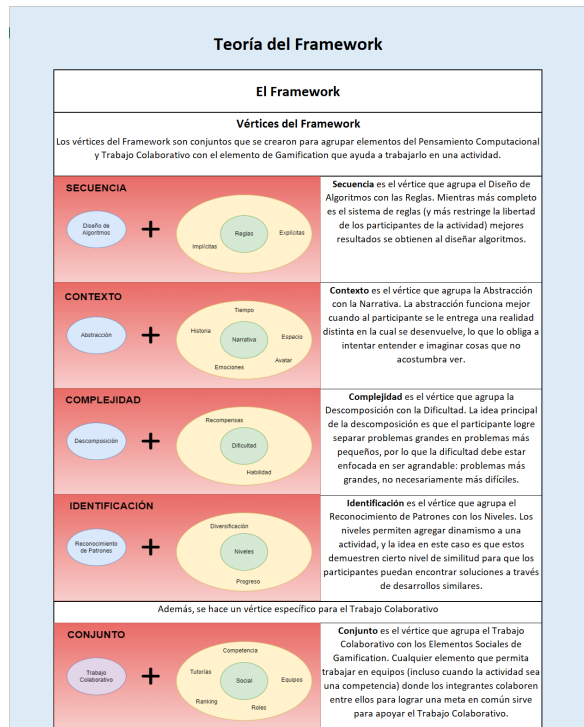


Figura 6.3: Pantalla de Teoría del Framework (1/3).

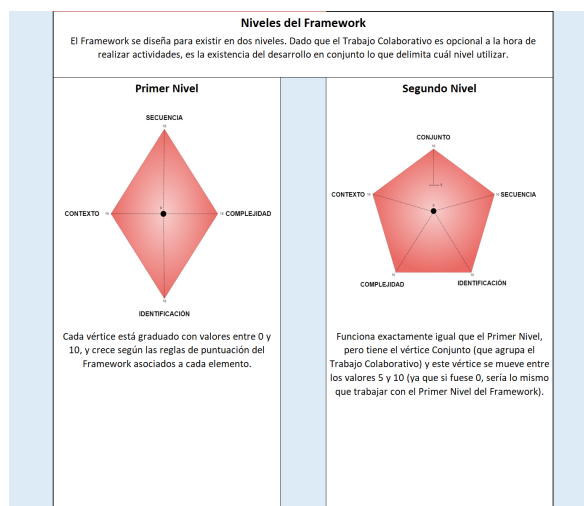


Figura 6.4: Pantalla de Teoría del Framework (2/3).

Sistema de Puntuación del Framework

Dependiendo de las características de una actividad, los vértices del Framework se movieron acumulando un determinado puntaje. Para esto, existe una guía que marca la forma en que se asigna puntaje dependiendo de esas características.

<p>Focos de Pensamiento Computacional</p> <p>Una actividad puede trabajar desde uno de los componentes del Pensamiento Computacional hasta los cuatro de ellos. Aunque se recomienda no trabajar más de dos en una misma actividad, es posible diseñar actividades donde se utilicen los cuatro, sobretodo en las asociadas a la programación.</p> <p>Para esto, el Framework propone que el diseñador considere focos de trabajo, los cuales sirven para dirigir los esfuerzos en generar una actividad bien definida y sólida en torno a ellos componentes de Pensamiento Computacional seleccionados.</p> <p>Así, dependiendo de la prioridad del foco y de la existencia (o no) de un componente del Pensamiento Computacional, es cómo los vértices aumentan su puntaje, siguiendo la fórmula:</p> <p>El componente del PC es foco Primario → 5 puntos al vértice correspondiente</p> <p>El componente del PC es foco Secundario → 3 puntos al vértice correspondiente</p> <p>El componente del PC es foco Terciario → 2 puntos al vértice correspondiente</p> <p>El componente del PC es foco Cuaternario → 1 punto al vértice correspondiente</p> <p>El componente del PC no se trabaja → 0 puntos al vértice correspondiente</p> <p>Por ejemplo, una actividad que huya trabajar el Diseño de Algoritmos como foco Primario y la abstracción como foco Secundario tendrá, como base, 5 puntos asignado al vértice de Secuencia, 3 puntos al vértice de Contexto y 0 puntos para Complejidad e Identificación.</p>	
<p>Conjunto</p> <p>Este vértice aparece cuando las actividades se desarrollan de forma grupal, por lo que recibe como base 5 puntos directamente al estar presente el Trabajo Colaborativo (que es independiente de los focos de trabajo del PC).</p> <p>La regla de puntuación para Conjunto es tan sencilla como que, por cada elemento de Gamificación que corresponda a un Elemento Social, se suman 2 puntos. Si la suma total de puntos es mayor a 10, el puntaje se ajusta a 10.</p>	

Gamification	
Los elementos de Gamificación asociados a cada vértice del Framework permiten que dicho vértice suba desde su puntaje base (punto anterior) a 5 puntos más arriba.	
<p>Reglas</p> <p>Las Reglas, que son el elemento que está asociado al vértice Secuencia, suman puntaje de acuerdo a lo robusto que resulta ser el sistema en cuanto al control que se tiene sobre los participantes de la actividad. En otras palabras, mientras menos libertad hay, mayor es el puntaje asociado.</p> <p>El puntaje se asigna de una forma completamente subjetiva por parte del diseñador, otorgando 0 puntos si no hay sistema de reglas, un intervalo entre 1 y 4 según su criterio, y 5 si el sistema está completo y tiene control total.</p>	<p>Niveles</p> <p>Los Niveles son el elemento que está asociado al vértice Identificación y suman puntaje de acuerdo al tipo de diversificación que proponen en el desarrollo de la actividad, siendo el máximo si están enfocados en generar recompensas para que los usuarios logren reconocer elementos.</p> <p>Si los niveles diversifican permitiendo la identificación de patrones → 3 puntos</p> <p>Si no hay niveles → 0 punto</p> <p>Además, es esencial que frente a los niveles exista un sistema de progresión. Este sirve para evitar la sensación de estancamiento, y el progreso existe tanto en la vida real como dentro del mundo imaginario de la actividad.</p> <p>Si hay progresión real → 1 punto</p> <p>Si hay progresión virtual/imaginaria → 1 punto</p>
<p>Dificultad</p> <p>La Dificultad, que es el elemento que está asociado al vértice Complejidad, suma puntaje en base a la forma en que esta está implementada. Para que la dificultad influya en la Descomposición, esta debe ser agradable. Si decir, debe buscar agrandar los problemas en cuanto a contenido, en vez de buscarlo hacerlos más complejos.</p> <p>Así, la fórmula es la siguiente:</p> <p>Si la dificultad es estándar y no agrandable → 1 punto</p> <p>Si la dificultad es estándar y agrandable → 2 puntos</p> <p>Si la dificultad es escalonada y no agrandable → 2 puntos</p> <p>Si la dificultad es escalonada y agrandable → 3 puntos</p> <p>Además, para que la dificultad no sea un factor de disensión en el desarrollo de la actividad, el diseñador debe implementar un sistema de recompensas que haga llamativo avanzar en la actividad</p> <p>Si existe una recompensa real → 1 punto</p> <p>Si existe una recompensa virtual/imaginaria → 1 punto</p>	<p>Narrativa</p> <p>La Narrativa es el elemento asociado al vértice Contexto. Aunque existen algunas excepciones para trabajar la abstracción, es la presencia de un ambiente imaginario generado por un enunciado lo que ayuda a potenciar el trabajo de esta.</p> <p>Para trabajar, se consideran 3 elementos como los más importantes dentro de la Narrativa: el trabajo en el espacio real, el espacio imaginario y el enunciado/historia.</p> <p>Si hay una organización del espacio real → 1 punto</p> <p>Si hay un espacio imaginario → 2 puntos</p> <p>Si hay un enunciado o historia → 2 puntos</p>

Figura 6.5: Pantalla de Teoría del Framework (3/3).

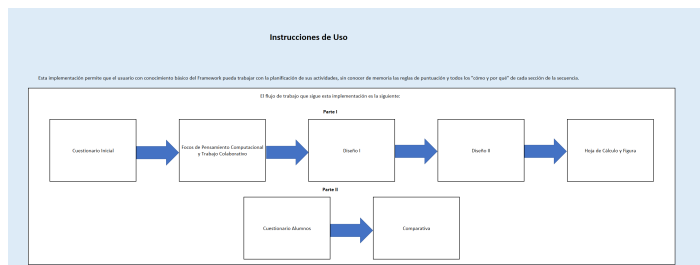


Figura 6.6: Pantalla de Instrucciones de uso (1/2).

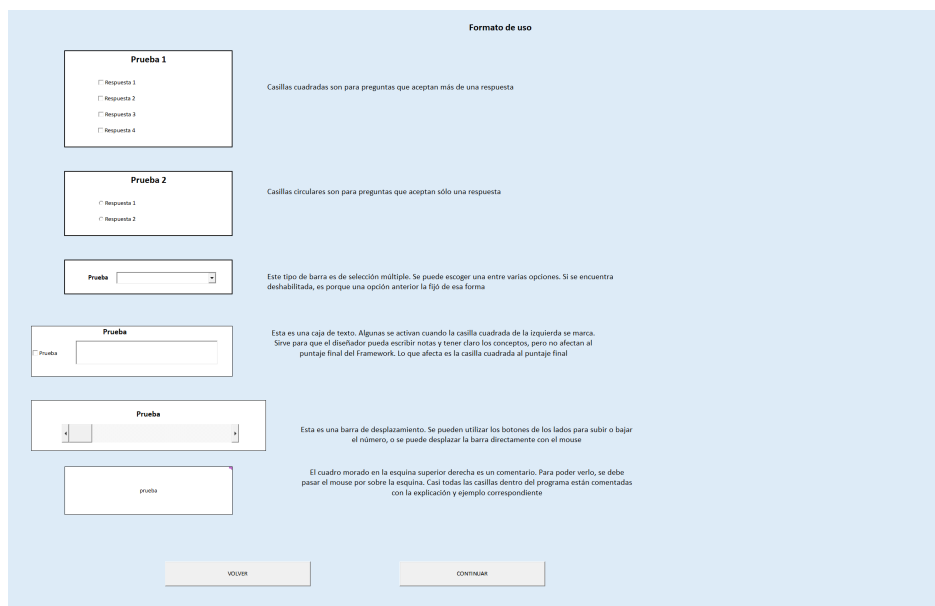


Figura 6.7: Pantalla de Instrucciones de uso (2/2).

Cuestionario Inicial (Aquí empieza el Framework)

1. ¿Qué quiero trabajar?

- Abstracción
- Descomposición de problemas
- Diseño de Algoritmos
- Reconocimiento de Patrones

2. ¿Actividad individual o grupal?

- Individual
- Grupal

3. ¿Qué tipo de actividad es?

- Teórica
- Práctica
- Unplugged (Desconectada)

4. Nivel de Gamification

- Baja
- Media
- Alta

5. Duración aproximada de la actividad

- 30 minutos
- 60 minutos
- 90 minutos

LIMPIAR TODO

VOLVER CONTINUAR

Figura 6.8: Pantalla de Cuestionario Inicial.

Pensamiento Computacional y Trabajo Colaborativo

1. Selección de los componentes y prioridad

Abstracción	<input type="text"/>	0	PUNTOS
Descomposición	<input type="text"/>	0	PUNTOS
Diseño de Algoritmos	<input type="text"/>	0	PUNTOS
Identificación de Patr.	<input type="text"/>	0	PUNTOS

2. Trabajo Colaborativo

PUNTOS

3. Nivel del Framework

La actividad es individual. Se utiliza el Primer Nivel del Framework.

VOLVER

CONTINUAR

Figura 6.9: Pantalla de PC y TC.

Diseño de la Actividad I

1. Definición de la Narrativa

<input type="checkbox"/> Espacio Real	1. Espacio Real	PUNTOS	<input type="text" value="0"/>
<input type="checkbox"/> Espacio Imaginario	2. Espacio Imaginario	PUNTOS	<input type="text" value="0"/>
<input type="checkbox"/> Enunciado	3. Enunciado	PUNTOS	<input type="text" value="0"/>
<input type="checkbox"/> Otros	4. Otros	TOTAL	<input type="text" value="0"/>

2. Definición de los Elementos Sociales (Sólo en Segundo Nivel)

<input type="checkbox"/> Equipos	1. Equipos (configuración)	PUNTOS	<input type="text" value="0"/>
<input type="checkbox"/> Roles	2. Sistema de Roles	PUNTOS	<input type="text" value="0"/>
<input type="checkbox"/> Puntaje	3. Sistema de Puntaje	PUNTOS	<input type="text" value="0"/>
<input type="checkbox"/> Otros	4. Otros	PUNTOS	<input type="text" value="0"/>
		TOTAL	<input type="text" value="0"/>

Figura 6.10: Pantalla de Diseño I (1/2).

3. Definición de las Reglas y Mecánicas

1. Reglas implícitas

2. Reglas explícitas

3. Mecánicas

4. Libertad vs Control

En base a lo establecido arriba, asigne un puntaje de acuerdo a su percepción de las reglas.

Libertad

Control

4. Definición del Objetivo y Condición de Victoria

1. Objetivo

2. Condición de Victoria

VOLVER

CONTINUAR

Figura 6.11: Pantalla de Diseño I (2/2).

Diseño de la Actividad II

1. Definición de la Dificultad y Sistema de Recompensa

1. Dificultad

Aggravado

PUNTOS

2. Recompensa real

 Recompensa real

3. Recompensa virtual

 Recompensa virtual

TOTAL

2. Definición de los Niveles y Sistema de Progreso

1. Niveles

PUNTOS

2. Progreso real

 Progreso real

3. Progreso virtual

 Progreso virtual

TOTAL

VOLVER

CONTINUAR

Figura 6.12: Pantalla de Diseño II.

118

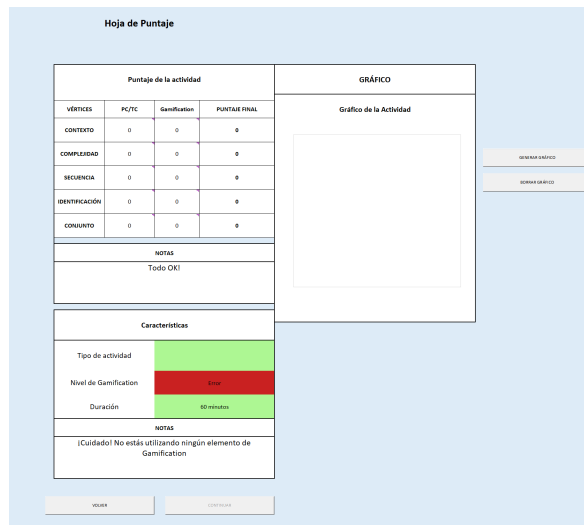


Figura 6.13: Pantalla Hoja de Puntaje.

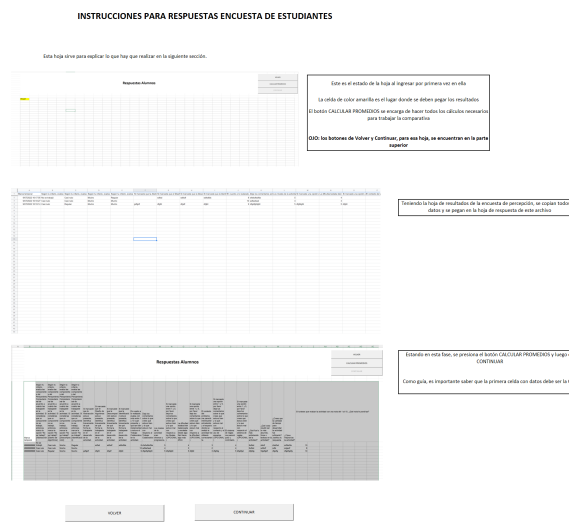


Figura 6.14: Pantalla Instrucciones Encuesta.

Encuesta de Percepción y Feedback (Primer Nivel)

jose.docmac@sansano.usm.cl (no compartidos)
[Cambiar de cuenta](#)

*Obligatorio

Sobre Pensamiento Computacional

En esta sección, las preguntas hacen referencia exclusivamente a elementos del Pensamiento Computacional que estuvieron o pudieron estar presentes en el desarrollo de la actividad.

Según tu criterio, evalúa las componentes del Pensamiento Computacional de acuerdo a cuánto las trabajaste en la actividad. Si consideras que un componente no se trabajó, entonces marca la opción 'No se trabajó'.

	Mucho	Regular	Parcial	Casi nulo	No se trabajó
Abstracción	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Diseño de Algoritmos	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Descomposición	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Identificación	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Si marcaste que la Abstracción estuvo presente, identifica brevemente de qué forma la trabajaste en el desarrollo de la actividad.

Tu respuesta

Si marcaste que el Diseño de Algoritmos estuvo presente, identifica brevemente de qué forma la trabajaste en el desarrollo de la actividad.

Tu respuesta

Si marcaste que la Descomposición estuvo presente, identifica brevemente de qué forma la trabajaste en el desarrollo de la actividad.

Tu respuesta

Si marcaste que la Identificación estuvo presente, identifica brevemente de qué forma la trabajaste en el desarrollo de la actividad.

Tu respuesta

[Atrás](#)
[Siguiente](#)
[Borrar formulario](#)

Figura 6.15: Encuesta en Google Forms.

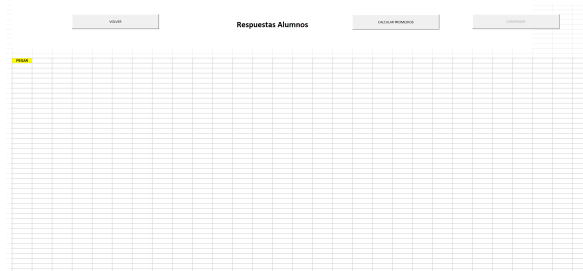


Figura 6.16: Pantalla Planilla Respuestas.



Figura 6.17: Pantalla Comparativa.

Bibliografía

- [1] La brecha digital en Chile. <https://www.conaccion.cl/la-brecha-digital-en-chile-2/>, 2020. Último acceso: 23 July 2021.
- [2] Fundación Google for Education. <https://edu.google.com/resources/programs/exploring-computational-thinking/index.html>, 2021. Último acceso: 30 July 2021.
- [3] Plan nacional de lenguajes digitales. <https://sitios.mineduc.cl/lenguajesdigitales/que-es-el-plan.html>, 2021. Último acceso: 23 July 2021.
- [4] Program.ar. <https://program.ar/>, 2021. Último acceso: 30 July 2021.
- [5] Claudio A Agostini, Manuel Willington, et al. Radiografía de la brecha digital en Chile: ¿se justifica la intervención del estado? *Estudios públicos*, (119), 2010.
- [6] Valerie Barr and Chris Stephenson. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1):48–54, February 2011.
- [7] Maira Isbeth Sarmiento Bolívar. Experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en Iberoamérica. *Pensamiento Actual*, 19(32):12–27, 2019.
- [8] Peter J. Denning. Ubiquity symposium 'what is computation?': Opening statement. *Ubiquity*, 2010(November), November 2010.
- [9] Sebastian Deterding, Rilla Khaled, Lennart Nacke, and Dan Dixon. Gamification: Toward a definition. pages 12–15, 01 2011.
- [10] Oxford Dictionary, 2022. Último acceso: 14 February 2022.
- [11] María Ros Esteve, Emilia López-Iñesta, and Pascual D. Diago. Medición del impacto del pensamiento computacional en la resolución de problemas con herramientas de gamificación. *Actas de las V Jornadas sobre Sistemas de Votación Electrónica 2019*, page 15.

- [12] J. Huizinga. *Homo Ludens*. Alianza Editorial, 1972.
- [13] David Johnson and Roger Johnson. Making cooperative learning work. *Theory Into Practice - THEORY PRACT*, 38:67–73, 03 1999.
- [14] Ralph Johnson and Brian Foote. Designing reusable classes. *Journal of Object-Oriented Programming*, 1:22textendash35, 06 1988.
- [15] Yu kai Chou. <https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>, 2012. Último acceso: 16 February 2022.
- [16] Nicole Lazzaro. Why we play games: Four keys to more emotion without story. *Game Dev Conf*, 01 2004.
- [17] Joey Lee and Jessica Hammer. Gamification in education: What, how, why bother? *Academic Exchange Quarterly*, 15:1–5, 01 2011.
- [18] Andrzej Marczewski. <https://www.gamified.uk/2014/05/07/game-design-process-framework/>, 2014. Último acceso: 17 February 2022.
- [19] Robert Michael Panoff. Computational thinking for all: The power and the peril. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE '14*, page 1–2, New York, NY, USA, 2014. Association for Computing Machinery.
- [20] Nick Pelling. The (short) prehistory of “gamification”.... <https://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification/>, 2011. Último acceso: 13 November 2021.
- [21] Rosabel Roig-Vila and Víctor Moreno-Isac. El pensamiento computacional en educación. análisis bibliométrico y temático. *Revista de Educación a Distancia (RED)*, 20(63), abr. 2020.
- [22] Rosana Rosas. <https://rosanarosas.com/que-es-gamificacion-como-funciona/>, 2017. Último acceso: 28 February 2022.
- [23] K. Salen and E. Zimmerman. *Rules of Play: Game Design Fundamentals*. MIT Press, 2004.
- [24] J. Schell. *The Art of Game Design: A Book of Lenses*. CRC Press, 2015.
- [25] Cynthia Luna Scott. El futuro del aprendizaje 2 ¿qué tipo de aprendizaje se necesita en el siglo xxi? *Investigación y prospectiva en educación: documentos de trabajo, UNESCO*, 14(41), 2015.
- [26] Jesús Valverde Berrocoso, María Rosa Fernández Sánchez, and María del Carmen Garrido Arroyo. El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia (RED)*, (46), Octubre 2015.

- [27] Diego Vergara and J.M. Mezquita. Design of serious games to reinforce knowledge: An educational experience in high school. *Profesorado. Revista de Curriculum y Formación del Profesorado*, 20:238–255, 01 2016.
- [28] Diego Vergara, José Mezquita, and Ana Gomez. Metodología innovadora basada en la gamificación educativa: Evaluación tipo test con la herramienta quizizz. *Profesorado. Revista de Curriculum y Formación del Profesorado*, 23, 09 2019.
- [29] Kevin Werbach. *For the Win: How Game Thinking can Revolutionize your Business*. 01 2012.
- [30] Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.
- [31] Jeannette M. Wing. *Computational thinking: What and why?* 2010.
- [32] Diego Vergara y Ana Isabel Gómez. Origen de la gamificación educativa. <http://espacioeniac.com/origen-de-la-gamificacion-educativa-por-diego-vergara-rodriguez-y-ana-isabel-gomez-vallecillo-universidad-catolica-de-avila/>. Último acceso: 30 July 2021.
- [33] G. Zichermann and C. Cunningham. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O’Reilly Media, 2011.