Universidad Nacional de Misiones. Facultad de Ciencias Exactas, Químicas y Naturales. Secretaría de Investigación y Postgrado. Maestría en Tecnología de la Información

Maestrando
Alejandro Micheloud

Aplicación de Design Thinking para el desarrollo de entornos educativos interactivos en educación superior

Trabajo final de Maestría presentada para obtener el título de "Magíster en Tecnología de la Información"

"Este documento es resultado del financiamiento otorgado por el Estado Nacional, por lo tanto, queda sujeto al complimiento de la Ley N° 26.899".

Director
Dr. Eduardo Zamudio
Co-Director
Dr. Joaquín Sotelo González

Año 2022

Esta obra está licenciado bajo Licencia Creative Commons (CC) Atribución-NoComercial-CompartirIgual 4.0 Internacional. https://creativecommons.org/licenses/by-nc-sa/4.0/

rid.unam.edu.ar

Universidad Nacional de Misiones

Facultad de Ciencias Exactas, Químicas y Naturales

Trabajo Final de Maestría en Tecnologías de la Información

Aplicación de Design Thinking para el desarrollo de entornos educativos interactivos en educación superior

Autor: Alejandro Micheloud

Director: Dr. Eduardo Zamudio

Co-Director: Dr. Joaquín Sotelo González

Año 2022

Dedicado a la memoria de Alfredo Horna.

Innovador incansable, motor de cambio en tecnología educativa.

Gracias por brindar energía y entusiasmo.

Agradecimientos

A mi querida familia, Ana Luz, Lucas y Rosana, por apoyarme y acompañarme en todo momento. A mis queridos padres, Carmen y Raúl.

A mis compañeros de Oneclick, por su gran labor profesional, por el compañerismo y energía empeñadas en mejorar la educación.

A mis compañeros de maestría, profesores y el personal involucrado, por darme la posibilidad de seguir creciendo.

A Eduardo Zamudio, por brindarme su conocimiento, tiempo y guiarme durante el desarrollo de este trabajo.

Resumen

A lo largo del tiempo, el Design Thinking viene proporcionando una manera de abordar problemáticas en distintas disciplinas para centrarse en el ser humano, y en los últimos años ha evolucionado adaptándose a diversos ámbitos. Mediante Design Thinking, se propone diseñar una solución a una problemática de la manera más óptima posible, con los recursos que estén a disposición. Los marcos de trabajo construidos sobre Design Thinking proponen diversas etapas y herramientas para construir soluciones, crear prototipos, validarlos e implementarlos en la vida real.

El objetivo del presente trabajo es adecuar el Design Thinking para aplicarlo en el proceso de definición de un producto software, en el ámbito específico de la Educación Superior.

La aplicación de la adaptación Design Thinking propuesta en este trabajo tiene como objetivo diseñar una plataforma educativa accesible, que promueva una dinámica colaborativa para añadir valor en los procesos educativos y asistir a los usuarios de diferentes roles que conforman el sistema educativo.

En el proceso de adaptación de Design Thinking propuesto, están involucrados alumnos, docentes, instituciones educativas, editoriales y un equipo de desarrollo tecnológico. Como resultado de la propuesta, se presenta un caso de estudio donde se desarrolla un producto software que brinda soporte a procesos educativos interactivos mediado por tecnología.

Palabras claves: Design Thinking, Desarrollo de software, Principios ágiles, Tecnología en Educación.

Abstract

Over time, Design Thinking has been providing a way to solve problems in different disciplines to focus on the human being, and in recent years it has evolved and adapted to various fields. Design Thinking aims to design a solution to a problem in the most optimal way possible, with the resources available. The frameworks built on Design Thinking propose several stages and tools to build solutions, create prototypes, validate them and implement them in real life.

The objective of the present work is to adapt Design Thinking to be applied in the process of defining a software product, in the specific field of Higher Education.

The application of the Design Thinking adaptation proposed in this work aims to design an accessible educational platform, which promotes a collaborative dynamic to add value in the educational processes and assist users of different roles that make up the educational system.

In the proposed Design Thinking adaptation process, students, teachers, educational institutions, publishers and a technological development team are involved. As a result of the proposal, a case study is presented where a software product is developed to support interactive educational processes through technology.

Keywords: Design Thinking, Software Development, Agile Principles, Technology in Education.

Índice

1. Introducción.....18

1.1. Conceptualización del problema.....18

1.2. Objetivo general.....19

1.3. Objetivos específicos.....19

1.4. Contribuciones del trabajo.....20

1.5. Contexto del presente trabajo.....20

1.6. Organización del documento.....20

2. Marco teórico.....23

2.1. Design Thinking.....23

2.1.1. Procesos Principales de Design Thinking.....24

2.1.1.1. El espacio del problema.....25

2.1.1.2. El espacio de la solución.....25

2.1.2. Marcos de trabajo en Design Thinking.....26

2.1.2.1. Hasso Plattner Institute of Design at Stanford.....26

2.1.2.2. IDEO, International Design and Consulting.....27

2.1.2.3. Google Design Sprints.....28

2.1.2.4. DEEP Design Thinking.....28

2.1.2.5. SAP Design Thinking.....29

2.1.2.6. Modelo Double Diamond, Design Council UK.....30

2.1.2.7. Design for America.....31

2.1.2.8. Austin Center for Design.....32

2.1.3. Tabla comparativa entre marcos de trabajo.....34

2.1.4. Herramientas y artefactos de los marcos de trabajo.....36

2.1.4.1. Tabla de herramientas.....36

2.1.4.2. Empatizar - Mapa de empatía.....36

2.1.4.3. Empatizar - User Journey Map.....37

2.1.4.4. Empatizar - Entrevista cualitativa.....39

2.1.4.5. Empatizar - Análisis DAFO.....40

2.1.4.6. Idear - Bocetos en papel.....42

2.1.4.7. Idear - Wireframes.....43

2.1.4.8. Idear - Storyboard.....44

2.1.4.9. Prototipos.....48

2.1.4.10. Prototipos en software multiusuario.....50

2.1.4.11. Testear - Validar prototipos con usuarios.....51

2.1.5. Aplicaciones de Design Thinking.....52

2.1.6. Design Thinking aplicado al desarrollo de software.....53

2.2. Desarrollo de tecnologías de información aplicado a la educación.....54

2.2.1. Entornos educativos interactivos.....54

2.2.2. Aprendizaje mejorado con tecnología.....54

2.2.3. Orquestación en el aula.....55

2.2.3.1. Diseño de actividades.....56

2.2.3.2. Factores intrínsecos y extrínsecos.....56

2.2.3.3. Consideraciones para el diseño de actividades.....57

2.3. Metodología de desarrollo ágil.....59

2.3.1. Scrum como marco de trabajo.....60

2.3.1.1. Modelo agile de Spotify.....61

2.4. Producto de software.....64

2.4.1. Demos.....64

2.4.2. Desarrollo y operaciones.....65

2.4.3. Integración continua y entrega continua.....65

2.4.4. Help Desk y Service Desk.....66

3. Propuesta de adaptación de Design Thinking para el desarrollo de software educativo 69

3.1. Design Thinking en desarrollo de producto software.....69

3.1.1. Proceso de Design Thinking.....71

3.1.1.1. Responsable de producto.....71

3.1.1.2. Responsable de UX.....71

3.1.1.3. Equipo multidisciplinar.....71

3.1.1.4. PRD.....71

3.1.1.5. Prototipo.....72

3.1.1.6. Roadmap.....72

3.1.1.7. Épicas.....72

3.1.1.8. Historias de usuario.....72

3.1.2. Desarrollo tecnológico ágil.....72

3.1.2.1. Equipo Tech....73
3.1.2.2. Responsable QA....73
3.1.2.3. Scrum Master....74
3.1.2.4. Sprint Planning....74
3.1.2.5. Sprint....74
3.1.2.6. Daily Meeting....74
3.1.2.7. Retrospective Meeting....75
3.1.3. Producto software....75
3.1.3.1. Demos....75
3.1.3.2. Desarrollo y operaciones....76
3.1.3.3. Help desk....77
3.1.3.4. Service Desk....78
3.2. Etapas del proceso Design Thinking....78
3.2.1. Etapa I - Empatizar....81
3.2.1.1. Conocer a los usuarios....82
3.2.1.2. Definición de la propuesta de producto software....88
3.2.1.3. Roles....91
3.2.1.4. Dinámica....92
3.2.2. Etapa II - Prototipar....92
3.2.2.1. Prototipos de baja fidelidad....93
3.2.2.2. Prototipos en alta fidelidad....93
3.2.3. Etapa III - Validar....97

4. Caso de estudio.....99

4.1. Etapa I - Empatizar.....99

4.2. Etapa II - Prototipado.....102

4.3. Etapa III - Validación.....103

4.4. Artefactos resultantes del proceso Design Thinking.....104

4.5. Épicas.....104

4.6. Historias de usuario.....105

4.7. Gráfico de control.....106

4.8. Sprints.....108

4.9. Burndown.....110

4.10. Encuesta al equipo técnico.....112

4.10.1.1. Resultados.....114

4.11. Aplicación en contexto de educación superior.....117

4.11.1. Vídeo presentación Go STEAM.....117

4.11.2. Encuesta a docentes de educación superior.....118

4.11.2.1. Resultados.....119

5. Conclusiones.....123

Bibliografía.....125

Índice de figuras

Figura 1: Espacio del problema y espacio de solución, ciclo iterativo en Design Thinking. *Adaptado de* [9].....25

Figura 2: Proceso de Design Thinking según Hasso Plattner Institute of Design at Stanford [Elaboración propia] [7].....27

Figura 3: Proceso de Design Thinking IDEO [Elaboración propia] [12].....27

Figura 4: Proceso de Design Thinking según Google Design Sprint [Elaboración propia] [63].....28

Figura 5: Proceso de Design Thinking según DEEP DT [Elaboración propia] [15].....29

Figura 6: Proceso de Design Thinking según SAP [Elaboración propia] [16].....30

Figura 7: Proceso de Double Diamond según Design Council [Elaboración propia] [62] 31

Figura 8: Proceso de Design Thinking según Design for America [Elaboración propia] [19].....32

Figura 9: Proceso de Design Thinking según Austin Center for Design [Elaboración propia] [20].....33

Figura 10: Mapa de empatía [Elaboración propia] [21].....37

Figura 11: Mapa de recorrido de usuario [Elaboración propia] [22].....38

Figura 12: Matriz DAFO [Elaboración propia] [25].....41

Figura 13: Bocetos y prototipado en papel [Elaboración propia].....43

Figura 14: Prototipado con wireframes [Elaboración propia].....44

Figura 15: Storyboard - Etapa uno, comenzar con la historia [Elaboración propia] [29] 46

Figura 16: Storyboard - Etapa dos, ilustrar la historia [Elaboración propia] [29].....47

Figura 17: Storyboard - Etapa tres, reproducir y discutir [Elaboración propia] [29].....48

Figura 18: Elementos del marco de trabajo Scrum para el desarrollo ágil [Elaboración propia] [43].....61

Figura 19: Marco de trabajo propuesto por Spotify para escalar la metodología ágil [Elaboración propia] [44].....64

Figura 20: CI/CD modelo de integración continua [Elaboración propia] [48].....66

Figura 21: Propuesta de Design Thinking como proceso integrado en el desarrollo de producto software [Elaboración propia].....70

Figura 22: Flujo de trabajo Gitflow [Elaboración propia] [49].....77

Figura 23: Etapas propuestas en el proceso Design Thinking [Elaboración propia].....79

Figura 24: Etapas The value proposition canvas [Elaboración propia] [50].....81

Figura 25: The value proposition canvas – Gains [Elaboración propia] [50].....83

Figura 26: The value proposition canvas – Pains [Elaboración propia] [50].....85

Figura 27: The value proposition canvas – Gains [Elaboración propia] [50].....88

Figura 28: The value proposition canvas - Products & services [Elaboración propia] [50].....89

Figura 29: The value proposition canvas - Pains relivers [Elaboración propia] [50].....90

Figura 30: The value proposition canvas - Gains creators [Elaboración propia] [50].....91

Figura 31: Diseño de prototipos en baja fidelidad [Elaboración propia].....93

Figura 32: Diseño de prototipos en alta fidelidad en Adobe XD [Elaboración propia]. .94

Figura 33: Diseño de prototipos en alta fidelidad en Figma [Elaboración propia].....95

Figura 34: Publicación de diseños en Zeplin [Elaboración propia].....96

Figura 35: Publicación de diseños en Zeplin, pantalla de mensajes del curso [Elaboración propia].....96

Figura 36: Workshop aplicando The value proposition canvas, identificación de elementos [Elaboración propia].....100

Figura 37: Taller The value proposition canvas, Post its, valoraciones [Elaboración propia].....101

Figura 38: Miro, lienzo colaborativo en línea [Elaboración propia].....102

Figura 39: Prototipos navegables en Zeplin [Elaboración propia].....103

Figura 40: Prototipo navegable con zonas interactivas [Elaboración propia].....104

Figura 41: Listado de épicas en Jira Software [Elaboración propia].....105

Figura 42: Listado de historias de usuario en Jira Software [Elaboración propia].....106

Figura 43: Gráfica de Control [Elaboración propia].....107

Figura 44: Caso atípico [Elaboración propia].....108

Figura 45: Burndown - Sprint MVP 1 [Elaboración propia].....111

Figura 46: Burndown - Sprint MVP 2.....111

Figura 47: [Elaboración propia] Burndown - Sprint MVP 2.....111

Figura 48: Burndown - Sprint MVP 3 [Elaboración propia].....112

Figura 49: Resultados de la encuesta al equipo técnico [Elaboración propia].....115

Figura 50: Grado de conformidad por pregunta [Elaboración propia].....115

Figura 51: Grado de conformidad agrupado por respuesta [Elaboración propia].....116

Figura 52: Grado de conformidad general [Elaboración propia].....116

Figura 53: Vídeo de presentación GOSTEAM [Elaboración propia].....117

Figura 54: Resultados de la encuesta a docentes de educación superior [Elaboración

propia].....120

Figura 55: Grado de conformidad por pregunta [Elaboración propia].....120

Figura 56: Grado de conformidad agrupado por respuesta [Elaboración propia].....121

Figura 57: Grado de conformidad general [Elaboración propia].....121

Índice de tablas

Tabla 1: Tabla comparativa entre marcos de trabajo Design Thinking. [Elaboración propia].....34

Tabla 2: Herramientas empleadas en fases de Design Thinking [Elaboración propia]…36

Tabla 3: Fases de una entrevista.....39

Tabla 4: Consideraciones en el uso de la matriz DAFO.....41

Tabla 5: Factores intrínsecos en la orquestación en el aula.....57

Tabla 6: Factores extrínsecos en la orquestación en el aula.....57

Tabla 7: The value proposition canvas - Pains.....84

Tabla 8: The value proposition canvas - Gains.....86

Tabla 9: Informe de Sprints.....109

Lista de abreviaturas

API: Interfaz de programación de aplicaciones

CI/CD: Integración continua / Desarrollo continuo

CSD: Desarrollo continuo de software

DAFO: Debilidades, Amenazas, Fortalezas y Oportunidades

DevOps: Desarrollo y Operaciones

HTTP: Protocolo de Transferencia de HiperTexto

JSON: Notación de Objeto de JavaScript

JWT: JSON Web Token

K12: Designación utilizada en algunos sistemas educativos para la escolarización primaria y secundaria

MVP: Producto Mínimo Viable

PO: Dueño del Producto

PRD: Definición de Requisitos del Producto

QA: Aseguramiento de la Calidad

REST: Transferencia de Estado Representacional

SLA: Acuerdo de Nivel de Servicio

TEL: Aprendizaje Mejorado por la Tecnología

TIC: Tecnologías de la Información y Comunicación

UI: Interfaz de Usuario

UX: Experiencia de Usuario

CAPÍTULO I

Introducción

1. Introducción

El presente Trabajo Final de Maestría consiste en adaptar Design Thinking para su uso en procesos de diseño de producto de software, en particular en el marco de la creación de plataformas educativas de nivel superior cuyo desarrollo tecnológico se realiza mediante principios ágiles.

La adaptación de Design Thinking en la producción de software en un entorno educativo pretende aportar mejoras en la captura de requisitos de software.

En este capítulo se presenta una conceptualización de la problemática del trabajo, los objetivos del mismo, sus contribuciones y el contexto en el que se desarrolla el caso de estudio.

1.1. Conceptualización del problema

Los equipos tecnológicos que desarrollan software necesitan como material de entrada especificaciones, que tiene su propia área de conocimiento denominada requisitos del software o gestión de requisitos de software.

Los equipos de desarrollo tecnológico han adoptado principios ágiles, desde hace décadas. En la creación de ciertos productos de software es necesario utilizar principios ágiles para poder lidiar con requisitos cambiantes.

Los principios ágiles promueven la naturalización del cambio. El manifiesto ágil [40] prefiere a individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación extensiva, colaboración con el cliente sobre negociación contractual, y respuesta ante el cambio sobre seguir un plan estricto.

En el sector educativo, las plataformas de aprendizaje se enfrentan a problemas complejos, con escenarios dinámicos, que tienen como centro al ser humano y pueden ir cambiando con el tiempo.

En procesos tradicionales de toma de requerimientos en los que no se contempla el factor humano, los requerimientos se definen de manera secuencial generando un conjunto de información extensiva que pretende ser muy precisa, exacta e inmutable en el tiempo.

En muchas ocasiones el software resultante no es utilizado por los usuarios finales, no cumple las expectativas, y no logra aportar valor en los procedimientos donde interviene.

Detectar las falencias en la captura de requerimientos, empatizar con los usuarios y adaptarse al cambio, son factores a tener en cuenta en las organizaciones que pretendan desarrollar productos de software para humanos.

1.2. Objetivo general

En el presente trabajo, el objetivo general es adaptar Design Thinking para capturar requisitos y definir el producto de software que se debe implementar, haciendo foco en las problemáticas a resolver, teniendo al ser humano como el centro de estudio.

En las últimas décadas, se han incorporado herramientas tecnológicas en los procesos educativos. En Latinoamérica se ha reducido la brecha digital y existen diversos estilos de adopción tecnológica [1]. Tanto dentro como fuera de las instituciones educativas, los protagonistas de la Educación Superior utilizan tecnología de manera habitual.

El desarrollo de una plataforma educativa que será utilizada por diferentes tipos de usuario, representa un desafío en lo que concierne a la toma de requerimientos y definición de producto de software.

1.3. Objetivos específicos

- Adaptar Design Thinking para el diseño e implementación de soluciones tecnológicas educativas.
- Determinar las fases y actividades que deben componer el proceso adaptado de Design Thinking.
- Indicar las herramientas necesarias para la aplicación Design Thinking como así también las especificaciones resultantes del proceso de definición de producto.
- Desarrollar una plataforma de educación interactiva como caso de estudio, para un contexto de educación superior utilizando como entrada las especificaciones

obtenidas por el proceso adaptado de Design Thinking.

- Evaluar la aplicación del proceso adaptado de Design Thinking en la plataforma educativa desarrollada.

1.4. Contribuciones del trabajo

- Abordaje de Design Thinking en el contexto del diseño de soluciones tecnológicas aplicadas a educación superior.
- Relevamiento de marcos de trabajos actuales basados en Design Thinking en desarrollo tecnológico.
- Promoción de diseño e implementación de productos de software centrados en el ser humano.
- Desarrollo de una plataforma educativa interactiva de aprendizaje basado en proyectos.

1.5. Contexto del presente trabajo

El presente trabajo corresponde al requerimiento para obtener el título de Magister en Tecnologías de la Información. El mismo se desarrolla dentro del ámbito de la FCEQyN, Facultad de Ciencias Exactas, Químicas y Naturales, y cuenta con el aval institucional para su desarrollo.

El caso de estudio escogido se desarrolla dentro de la empresa Oneclick S.L. radicada en Madrid, España, en la cual el maestrando es Software Engineer y Tech Lead de producto software.

1.6. Organización del documento

El resto del presente documento se encuentra dividido en cuatro capítulos.

El capítulo 2 brinda el marco teórico utilizado como base para el desarrollo del presente Trabajo Final de Maestría. Inicialmente se introduce al concepto de Design Thinking y sus procesos fundamentales, se recopilan varios de los marcos de trabajo construidos sobre Design Thinking y se identifican las herramientas utilizadas en cada una

de las fases. Luego se define el desarrollo tecnológico aplicado a la educación, el aprendizaje mejorado por la tecnología y los principios de la orquestación en el aula. Finalmente, se describen aspectos del desarrollo de software mediante metodologías de desarrollo ágil.

El capítulo 3 desarrolla la propuesta de adaptación de Design Thinking para el desarrollo de entornos interactivos de nivel superior, se definen las etapas, procesos y herramientas que lo conforman haciendo foco en empatizar, prototipar y validar para lograr definir un producto de software.

El capítulo 4 describe la aplicación de la adaptación de Design Thinking en un caso de estudio. Se describe la experiencia en cada una de las etapas y se detallan artefactos resultantes que sirven de entrada para la implementación tecnológica siguiendo principios ágiles.

Finalmente el capítulo 5 presenta las conclusiones con relación a los objetivos que han sido establecidos, las contribuciones realizadas y las futuras líneas de investigación que pueden llevarse a cabo.

CAPÍTULO II

Marco teórico

2. Marco teórico

Este capítulo desarrolla el marco teórico en el que se determina cómo se encuentra en la actualidad el estado de las cuestiones que se abordan en el trabajo.

Inicialmente se introduce el concepto de Design Thinking a partir de sus definiciones teóricas fundamentales, la caracterización de sus principales propulsores, etapas involucradas, herramientas y consideraciones generales.

Por otra parte, se establece el foco en las características del aprendizaje mejorado por el uso de la tecnología, que implica la orquestación del aprendizaje y diseño de actividades con fines pedagógicos.

Finalmente, se introducen los aspectos fundamentales de la producción de software, metodología ágil, desarrollo y operaciones, integración continua y entrega continua, help desk y service desk.

2.1. Design Thinking

El Design Thinking aporta varios beneficios asociados al incluir al usuario final en el proceso de diseño del producto, a una mejor toma de decisiones, a la innovación. Aporta empatía con el usuario final para localizar lo que el usuario final busca y necesita [2].

El Design Thinking es un proceso centrado en la resolución de problemas, con un enfoque en la generación de soluciones creativas teniendo en cuenta un proceso social, que consiste en pensar y trabajar a través de diferentes perspectivas. A menudo implica considerar conflictos y negociación.

El Design Thinking es un proceso que reúne todo el espectro de actividades de innovación con una filosofía de diseño centrada en las personas. La innovación está impulsada por una comprensión sólida, mediante la observación directa, de lo que las personas quieren y necesitan para su vida, de lo que les gusta o disgusta respecto a cómo se fabrican, empacan, comercializan y venden [3].

El Design Thinking ayuda a crear una gran satisfacción en el cliente [4].

Se considera a Edison como uno de los primeros genios en utilizar Design Thinking aunque sin saberlo ya que en aquel entonces no existía como tal [5]. Edison creó la bombilla eléctrica y luego generó toda una industria a su alrededor, la bombilla por sí sola no sería útil a los usuarios finales sin un sistema de transmisión de energía eléctrica. Edison fue capaz de prever cómo la gente querría utilizar su creación ideando una solución completa para los usuarios finales. Edison no era un científico estrictamente especializado, sino un amplio generalista con un agudo sentido para los negocios. En su laboratorio de Menlo Park, Nueva Jersey, se rodeó de talentosos restauradores, improvisadores y experimentadores. De hecho, rompió el molde de genio inventor solitario al crear un enfoque basado en el trabajo en equipo para la innovación, su método no busca validar hipótesis preconcebidas sino ayudar a los experimentadores a aprender algo nuevo de cada intento iterativo. La innovación es una tarea ardua y Edison la convirtió en profesión que combina arte, destreza, ciencia, conocimiento del negocio y una comprensión sagaz de los clientes y mercados [5].

El Design Thinking se utiliza en diversos sectores que en muchos casos son lejanos entre sí como ser la medicina, diseño industrial, tecnología de la información, negocios, entre otros [6].

2.1.1. Procesos Principales de Design Thinking

El proceso de Design Thinking es incremental, altamente interactivo, e impulsado por personas con diferentes antecedentes y experiencias. Esta heterogeneidad es importante para capturar tantos puntos de vista como sea posible durante el diseño, no solo los de expertos técnicos e ingenieros [7].

La innovación guiada por Design Thinking complementa la visión tradicional que normalmente se enfoca en el desarrollo e integración de nuevas tecnologías. El Design Thinking aporta innovación principalmente dotando a productos y servicios de nuevos significados, las cosas deben tener una forma para ser vistas, pero deben tener sentido para ser entendidas y utilizadas. El diseño por naturaleza es una disciplina que trata con los significados [8].

Al desafiar patrones de pensamiento y comportamiento, Design Thinking permite generar soluciones con nuevos significados, que activan diversos elementos cognitivos,

emocionales y sensoriales, que están involucrados en la experiencia humana.

Para comprender cómo funciona la Design Thinking es útil trabajar con dos espacios fundamentales, el espacio de problemas y soluciones, por un lado, y el pensamiento divergente / convergente, por el otro [9].

2.1.1.1. El espacio del problema

Cuando se explora el espacio del problema, Design Thinking adquiere una comprensión intuitiva, principalmente se concentra en observar casos de uso general o escenarios representativos.

2.1.1.2. El espacio de la solución

Design Thinking promueve evaluar un gran número de ideas alternativas en paralelo, se crean bocetos y prototipos de solución, de esta manera, las ideas se transforman en representantes tangibles.

En la Figura 1 se representa el espacio del problema, en la mitad de la izquierda, y el espacio de la solución a la derecha, dividiéndose a su vez cada uno en un proceso divergente y convergente.

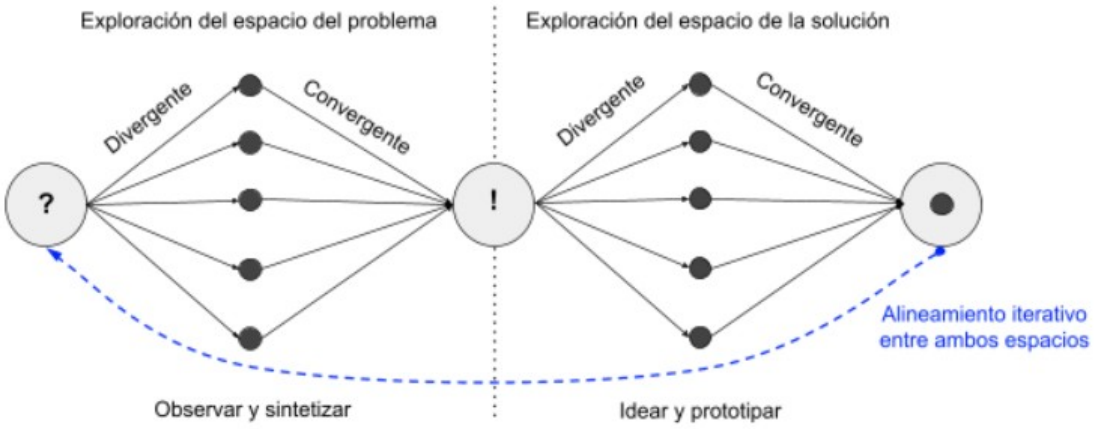

Figura 1: Espacio del problema y espacio de solución, ciclo iterativo en Design Thinking. *Adaptado de* [9]

La alineación iterativa de ambos espacios representa las ideas y conceptos que facilitan la comunicación, no solo en el equipo de diseño, sino también con usuarios,

clientes y expertos. El Design Thinking ayuda a mantenerse en contacto con el entorno relevante para el problema y puede usar esta información para refinar y revisar alternativas de solución.

El Design Thinking produce un sistema de frenos y contrapesos para garantizar que la solución concluyente sea equilibrada, innovadora y adecuada [10].

2.1.2. Marcos de trabajo en Design Thinking

Los marcos de trabajo son instancias basadas en Design Thinking definidos para estandarizar las etapas y procesos, pueden ser de propósito general o diseñados para trabajar en un campo específico.

Los marcos de trabajo que se analizan en este apartado provienen principalmente de instituciones educativas y empresas de tecnología.

Inicialmente se enumeran los marcos de trabajo a estudiar, se añade un esquema y una breve introducción en cada uno de ellos, seguidamente se identifican las etapas de cada marco de trabajo en una tabla comparativa, finalmente se realiza un análisis para determinar las etapas fundamentales del Design Thinking.

2.1.2.1. Hasso Plattner Institute of Design at Stanford

El Hasso Plattner Institute of Design at Stanford, comúnmente llamado “d.school”, es un instituto de Design Thinking fundado en el año 2010 basado en la Universidad de Stanford, San Francisco, Estados Unidos de América [11].

La Figura 2 sintetiza las etapas de la propuesta de Hasso Plattner Institute of Design at Stanford. Las etapas son empatizar, definir, idear, prototipar y testear.

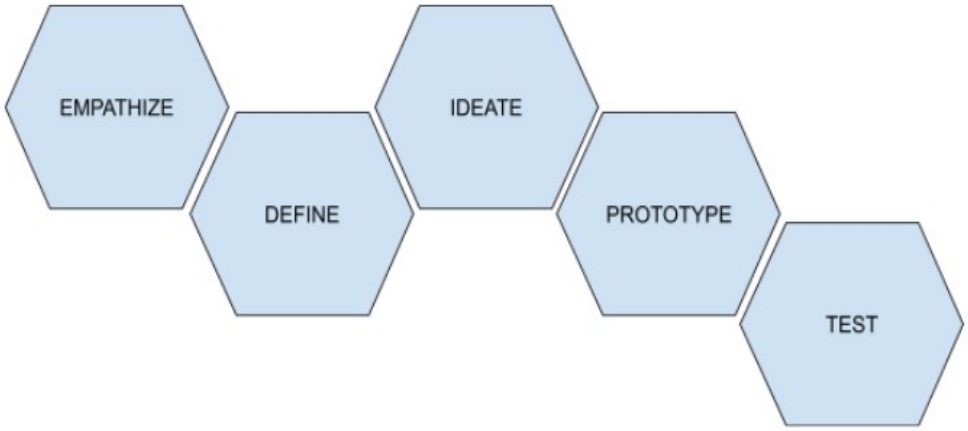

Figura 2: Proceso de Design Thinking según Hasso Plattner Institute of Design at Stanford [Elaboración propia] [7]

2.1.2.2. IDEO, International Design and Consulting

IDEO es una empresa de diseño y consultoría. Fue fundada en Palo Alto, California, Estados Unidos de América en el año 1991. La compañía utiliza el enfoque de Design Thinking para conceptualizar productos, servicios, entornos y experiencias digitales [12].

En la Figura 3 se representa la secuencia de las cinco etapas Design Thinking en la propuesta IDEO.

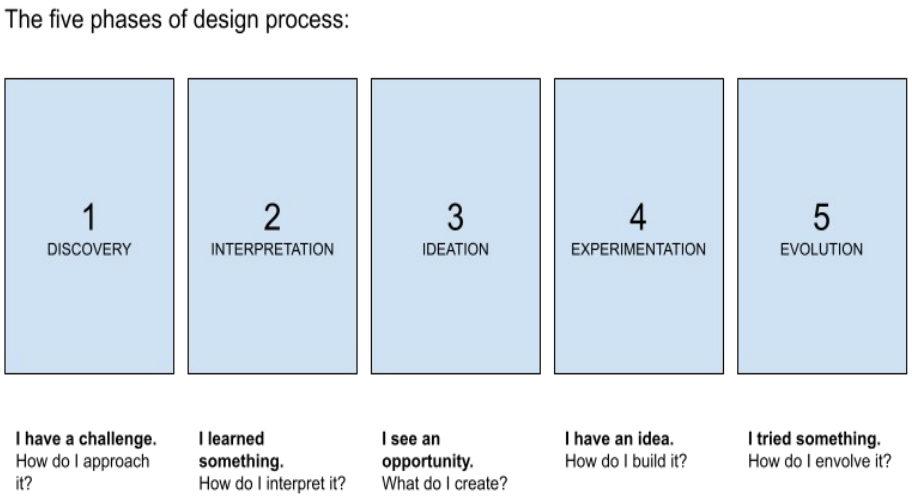

Figura 3: Proceso de Design Thinking IDEO [Elaboración propia] [12]

2.1.2.3. Google Design Sprints

Design Sprint de Google se creó en el 2010 por Google Ventures en Mountain View, California, Estados Unidos de América [13], el marco de trabajo propuesto es un proceso de cinco días que trata de responder las cuestiones críticas de un negocio a través del diseño, el prototipado y el testeo con los clientes. En esencia es un proceso de brainstorming estructurado para responder preguntas críticas de negocios a través del diseño, la creación de prototipos y la prueba de nuevas ideas con los usuarios a lo largo de un taller [14].

En la Figura 4 se encuentran los tres rombos principales que representan tres etapas generales, haciendo analogía a procesos divergentes y convergentes en cada uno de los triángulos. En total la propuesta suma seis fases que son comprender, definir, bosquejar, decidir, prototipar y validar.

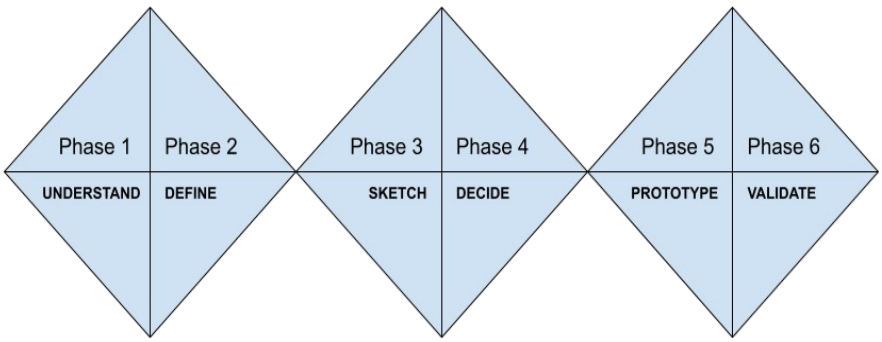

Figura 4: Proceso de Design Thinking según Google Design Sprint
[Elaboración propia] [63]

2.1.2.4. DEEP Design Thinking

DEEP es una propuesta para aplicar Design Thinking con estudiantes, creada por Mary Cantwell en el año 2010, en Atlanta, Estados Unidos de América [15].

En la Figura 5 queda plasmada la propuesta compuesta por cuatro etapas, la

primera consiste en descubrir, la segunda empatizar, la tercera experimentar y la cuarta producir.

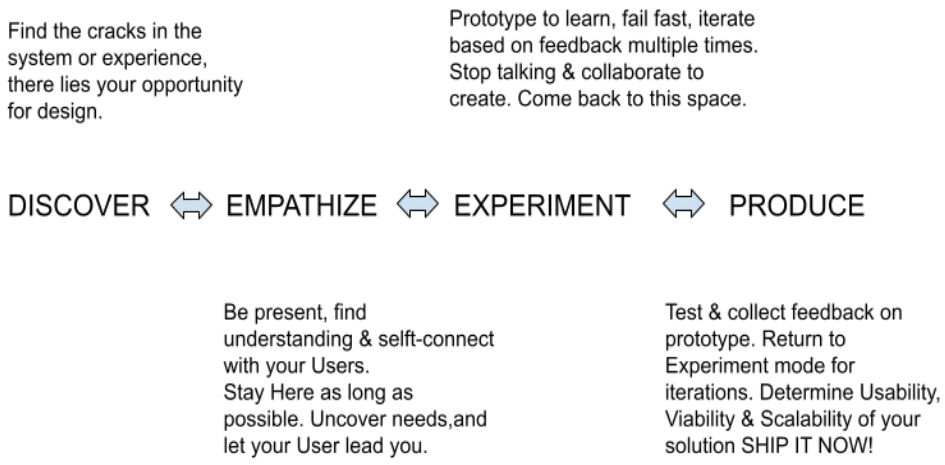

Figura 5: Proceso de Design Thinking según DEEP DT [Elaboración propia] [15]

2.1.2.5. SAP Design Thinking

SAP es una empresa multinacional alemana dedicada al diseño de productos informáticos de gestión empresarial, tanto para empresas como para organizaciones y organismos públicos.

SAP utiliza Design Thinking como marco de trabajo, la promueve a través de workshops y certificaciones oficiales [16]. Desde el año 2010 difunden el marco de trabajo desde lo que denominan comercialmente las AppHaus, en diferentes ubicaciones alrededor del mundo.

La Figura 6 es un esquema de la propuesta SAP de Design Thinking que está compuesta por etapas y actividades. La primer etapa es descubrir, que tiene como actividades determinar el alcance alcance, investigación 360º y sintetizar. La segunda etapa es diseñar, con las actividades idear, prototipar y validar. La tercer y última etapa es entregar, con las actividades de implementar, testear y desplegar.

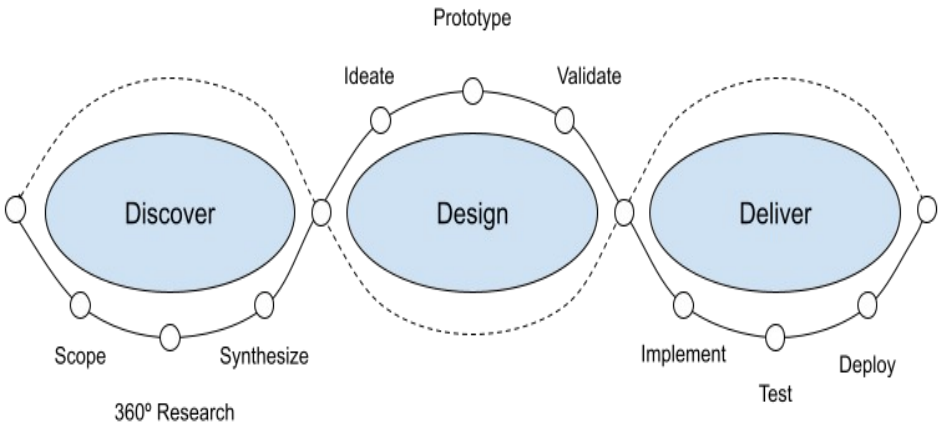

Figura 6: Proceso de Design Thinking según SAP [Elaboración propia] [16]

2.1.2.6. Modelo Double Diamond, Design Council UK

El modelo Double Diamond es un framework de trabajo creado por el Design Council basado en Design Thinking [17].

Design Council fue establecido por el gobierno de guerra de Winston Churchill en diciembre de 1944 para apoyar la recuperación económica de Gran Bretaña. El Consejo de Diseño Industrial tenía el propósito fundamental de promover por todos los medios posibles la mejora del diseño en los productos de la industria británica [18].

En la Figura 7 está representado el modelo de doble diamante, es de destacar que los vértices horizontales de los diamantes representan entradas y salidas, en concreto, existe una entrada inicial denominada problema, la salida del primer diamante es lograr una definición del problema al tiempo que constituye la entrada del segundo diamante para llegar a la salida del segundo y último diamante que representa la solución al problema. Para llegara la solución del problema es necesario pasar por dos etapas divergentes, representadas por las etapas descubrir del primer triángulo y desarrollar del segundo, y dos etapas convergentes representadas por las etapas definir del primer triángulo y entregar del segundo triángulo.

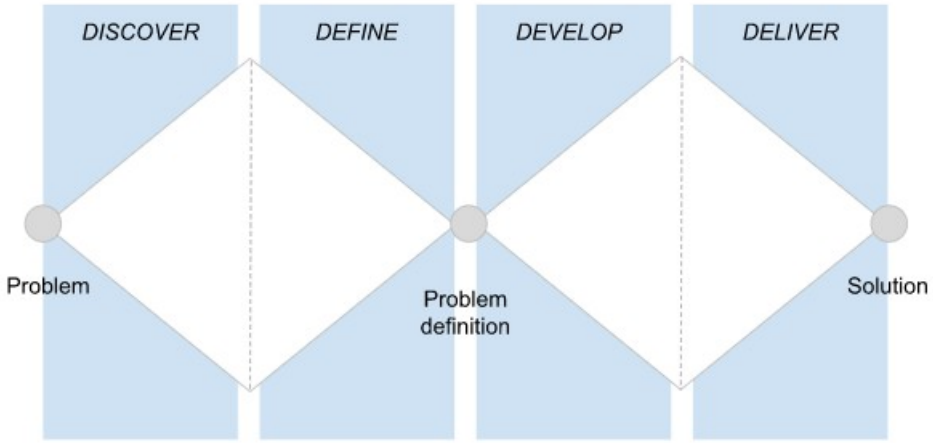

Figura 7: Proceso de Double Diamond según Design Council [Elaboración propia] [62]

2.1.2.7. Design for America

Design for America es una incubadora de ideas, una forma de abordar desafíos complejos y una comunidad de aprendices para toda la vida. Creado por la fundación Thomas J. Watson Foundation [19].

Design for America guía a los equipos dirigidos por estudiantes a medida que aprenden y lideran un proceso de innovación centrado en la comunidad que enfatiza el diseño participativo, la interacción inmediata y el trabajo en equipo interdisciplinario. Fuera del aula, pero aún dentro del entorno universitario, los equipos trabajan hasta que sus soluciones se implementan en sus comunidades [19].

La fundación está ubicada en Chicago, Estados Unidos de América, la Figura 8 representa la tercer propuesta Design Thinking de Design for America, publicada en el año 2014. Está compuesta por dos grandes etapas que son comprender, ubicada en la parte superior y crear, con fondo gris, en la parte inferior. De manera secuencial, las fases son identificar, sumergirse, reformular, idear, construir y testear.

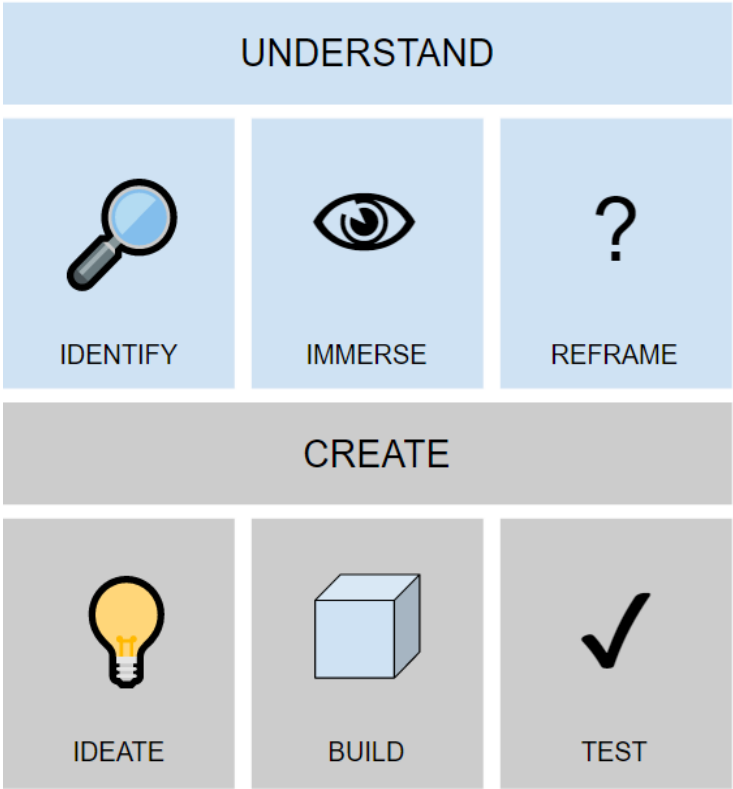

Figura 8: Proceso de Design Thinking según Design for America
[Elaboración propia] [19]

2.1.2.8. Austin Center for Design

Austin Center for Design es una organización educativa sin fines de lucro, creada en el año 2010 por Jon Kolko en Austin, Texas, Estados Unidos. Es una escuela destinada a ayudar a desarrollar la autonomía a través del diseño [20].

En la Figura 9 están representadas las etapas y actividades de la propuesta Design Thinking de Austing Center for Design, la propuesta se centra en tres etapas principales y una secuencia de actividades que son identificar el problema o el área de oportunidad, descubrir, idear, iterar, redefinir, validar e implementar.

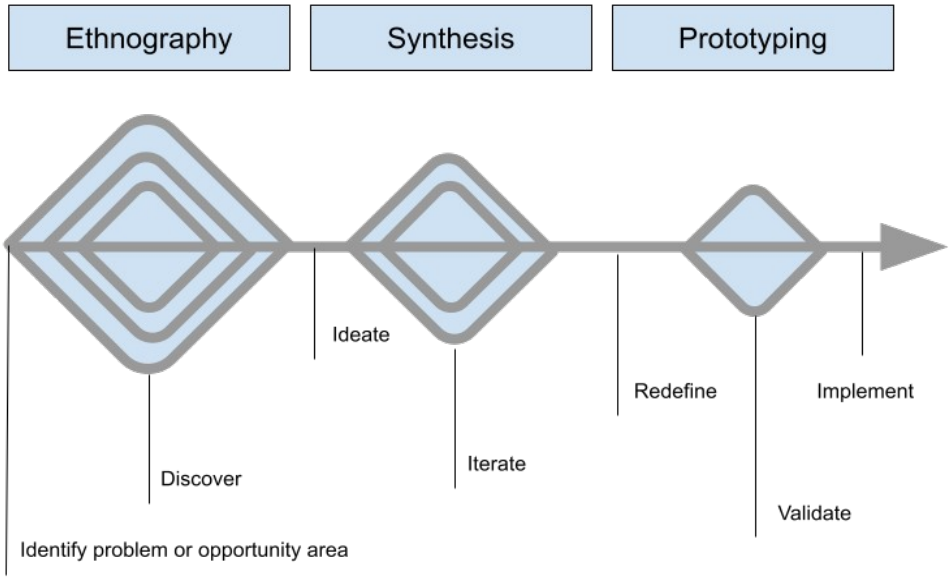

Figura 9: Proceso de Design Thinking según Austin Center for Design [Elaboración propia] [20]

Larry Page y Sergey Brin, fundadores de Google en 1998, fueron alumnos de la universidad de Stanford. Hoy en día, Standford y Google tienen marcos de trabajo para aplicar Design Thinking, y aunque comparten el fundamento, son diferentes entre sí. La aplicación Design Thinking propuesta por Standford está dirigida al sector denominado K12, escolarización primaria y secundaria, con la finalidad de proporcionar herramientas a los docentes y alumnos en la resolución de problemas de contexto general. En cambio, la aplicación Design Thinking propuesta por Google, está dirigida al diseño y desarrollo de producto, en lo que denominan Design Sprints [14], que son jornadas dedicadas al diseño y desarrollo de producto utilizando Design Thinking, normalmente requieren 5 días continuados junto a un equipo interdisciplinario, el marco de trabajo está dirigido principalmente a emprendedores, que normalmente se denominan Startups.

Uno de los grandes promotores de la Design Thinking de los últimos tiempos es

Tim Brown, CEO de IDEO [12]. El artículo que escribió para Harvard Business Review despertó un interés por el uso de Design Thinking [5]. IDEO es una firma internacional de consultoría y diseño. La propuesta Design Thinking fomenta una transformación en las organizaciones, en el desarrollo de productos, servicios, procesos y estrategias.

2.1.3. Tabla comparativa entre marcos de trabajo

En el apartado anterior se describe un conjunto de propuestas de Design Thinking, cada una de ellas puede ser considerada como un marco de trabajo para la aplicación de Design Thinking en un contexto dado.

En la Tabla 1 se sintetizan los marcos de trabajo para facilitar su análisis y comparación pudiendo inferir posteriormente las etapas fundamentales que son compartidas entre los marcos de trabajo con mayor frecuencia.

Tabla 1: Tabla comparativa entre marcos de trabajo Design Thinking. [Elaboración propia]

Framework	**Etapas**	**Descripción**
Hasso Plattner Institute of Design at Stanford [9][11]	1. Empatizar 2. Definir 3. Idear 4. Prototipar 5. Testing	Ideado para la educación primaria y secundaria (sector K12), para ser aplicada por docentes y estudiantes en la resolución de problemas de índole general.
IDEO [12]	1. Descubrir 2. Interpretar 3. Idear 4. Experimentar 5. Evolucionar	Centrado en las necesidades de las personas, las posibilidades tecnológicas y los negocios.
Google Design Sprints [14]	1. Comprender 2. Definir 3. Bosquejar 4. Decidir 5. Prototipar 6. Validar	Aplicado al diseño y desarrollo de producto, optimizada para Startups y emprendedores.
Deep Design Thinking	1. Descubrir	Aplicado en el contexto del aula

[15]	2. Empatizar 3. Experimentar 4. Producir	como herramienta pedagógica dentro del sector K12.
SAP design [16]	1. Descubrir: alcance, investigación, síntesis. 2. Diseño: idear, prototipar, validar. 3. Entregar: implementar, testear, desplegar.	Utilizado como enfoque de innovación en la relación con clientes.
Double diamond, Design Council UK [18]	1. Descubrir: identificar, investigar y comprender el problema inicial. 2. Definir: delimitar y definir un problema a resolver. 3. Desarrollar: centrarse y desarrollar una solución. 4. Entregar: probar y evaluar, producir y lanzar.	Proporciona una representación gráfica del proceso de diseño.
Design for America [19]	1. Identificar 2. Sumergirse 3. Replantear 4. Idear 5. Construir 6. Testear	Promovido por universidades con campus de estudios distribuidos en Estados Unidos de América.
Austin Center for Design [20]	1. Identificar 2. Descubrir 3. Idear 4. Iterar 5. Refinar 6. Validar 7. Implementar	Enfocado en la resolución de problemas sociales a través del diseño y la educación en diseño.

Los marcos de trabajo propuestos en Design Thinking tienen etapas en común. Algunas etapas se comparten entre todas las propuestas.

Los marcos de trabajo proponen un desarrollo ágil e iterativo cuyas etapas se pueden generalizar en **empatizar**, incluyendo las tareas de comprender, identificar, descubrir; **idear**, incluyendo las tareas de construir, experimentar, diseñar, prototipar,

producir; y **testear**, incluyendo las tareas de implementar y evolucionar.

La generalización de etapas y la reducción en la cantidad de las mismas podría contribuir en la adopción de Design Thinking.

2.1.4. Herramientas y artefactos de los marcos de trabajo

Cada una de los marcos de trabajo ofrece un conjunto de herramientas y artefactos que representan actividades propuestas que deben llevarse a cabo en cada una de las fases.

Se recopilan aquí algunas herramientas con el objetivo de conocerlas en detalle, se realiza un análisis las características destacadas y se dimensiona el trabajo que representa la aplicación de Design Thinking.

2.1.4.1. Tabla de herramientas

En la Tabla 2 se representan las tres fases fundamentales de Design Thinking en la primer columna, y en la segunda columna, las herramientas elegidas agrupadas en cada etapa.

Tabla 2: Herramientas empleadas en fases de Design Thinking [Elaboración propia]

Fase	**Herramienta**
Empatizar	● Mapas de empatía ● User Journey map ● Entrevista cualitativa ● Encuestas ● Análisis DAFO
Idear	● Bocetos en papel ● Wireframes ● Storyboard
Testear	● Prototipos ● Test con usuarios

2.1.4.2. Empatizar - Mapa de empatía

El mapa de empatía es una herramienta que permite explorar las actitudes y comportamientos de los usuarios dentro del dominio de intereses, ayuda al equipo de

experiencia del usuario a comprender al usuario de una manera más profunda. El proceso de mapeo también revela cualquier brecha en los datos de los usuarios existentes [21].

La Figura 10 se representa un mapa de empatía que está delimitado por siete zonas, cada zona tiene como propósito identificar una característica del usuario.

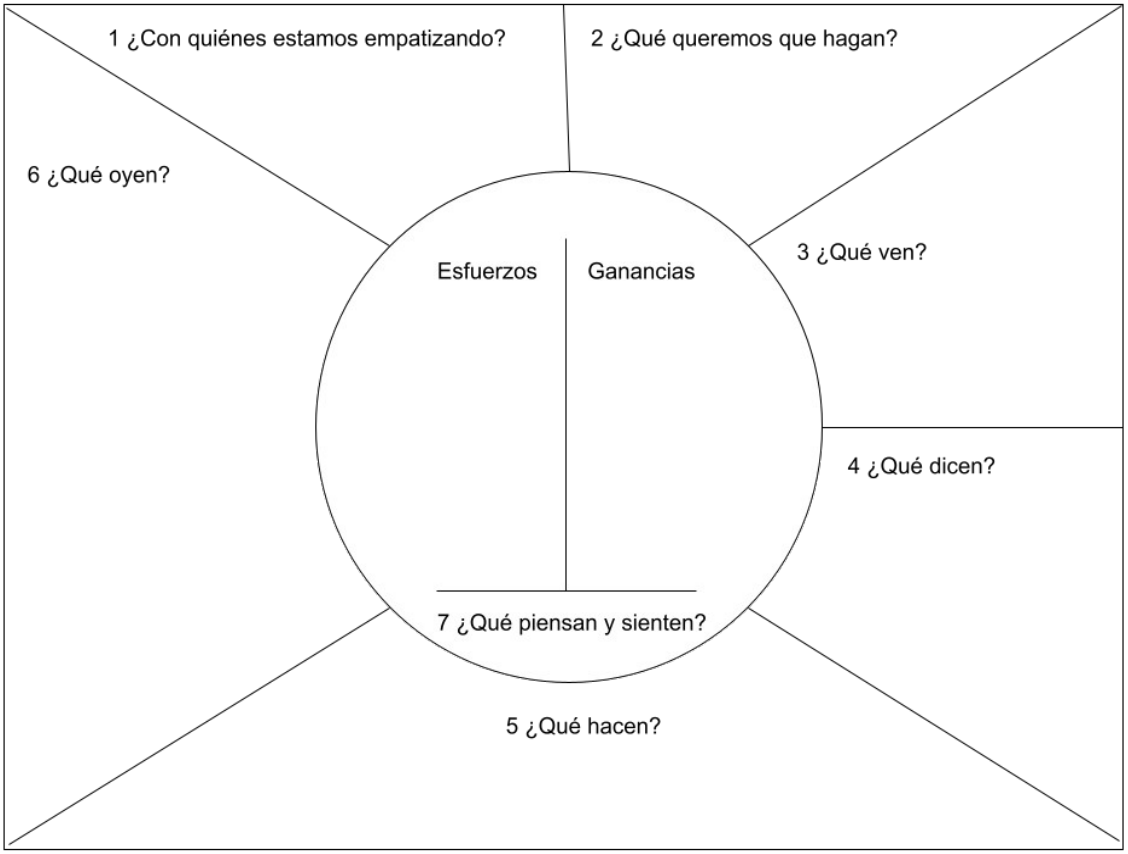

Figura 10: Mapa de empatía [Elaboración propia] [21]

2.1.4.3. Empatizar - User Journey Map

Un User Journey Map es una visualización del proceso que atraviesa una persona para lograr una meta.

En su forma más básica, el mapeo del recorrido comienza compilando una serie de acciones del usuario en una línea de tiempo. A continuación, la línea de tiempo se completa con los pensamientos y emociones del usuario para crear una narrativa. Esta

narrativa está condensada y pulida, lo que en última instancia conduce a una visualización [22].

En la Figura 11 se representa un modelo de User Journey Map compuesto por tres fases, la cantidad de actividades que realiza un usuario en su viaje pueden variar en cada mapa que se confeccione.

Usuario específico + Escenario + Objetivos		
Fase 1	Fase 2	Fase 3
1. 2. 3.	4. 5. 6. 7.	8. 9. 10.
Oportunidades + propiedades internas		

Figura 11: Mapa de recorrido de usuario [Elaboración propia] [22]

La mayoría de los mapas de recorrido siguen un formato similar: en la parte superior, un usuario específico, un escenario específico y las expectativas u objetivos. En el medio, las fases que se componen de acciones, pensamientos y emociones del usuario; en la parte inferior, las conclusiones: oportunidades, indicadores y conocimiento interno [22].

2.1.4.4. Empatizar - Entrevista cualitativa

Las entrevistas cualitativas son entrevistas con usuarios o potenciales usuarios de un determinado producto o servicio con el objetivo de relevar su modelo mental. El modelo mental de los usuarios nos permite conocer cómo las personas creen que las cosas funcionan o deberían funcionar, en qué contexto y para satisfacer qué objetivos.

La mejor forma de preparar una entrevista es en equipo. Los miembros del equipo podrán hacer una lluvia de ideas sobre cuáles serán los temas a tratar. Después se definirán las preguntas y se eliminarán aquellas que sean redundantes. Las entrevistas deben ir de lo racional a lo emocional. Empieza preguntando sobre características demográficas, sobre preguntas concretas acerca de su rutina, o el uso de un producto o un servicio, para terminar con preguntas evocadoras del tipo "¿Si pudieras..., cómo...?" "¿Cómo te imaginas...?", “¿Qué cosa te facilitaría....?”. Es importante saber que lo mejor de una entrevista es favorecer la espontaneidad, pero es importante poder apoyarse sobre un guión [23].

2.1.4.4.1. Fases de una entrevista

En cuanto al guión de la entrevista, la organización ThinkerCO [24] propone estructurar la entrevista en cuatro fases descritas a continuación en la Tabla 3.

Tabla 3: Fases de una entrevista

1. Introducción	Presentación, explicación de los motivos por los cuales se realiza la entrevista.
2. Calentamiento	De cara a romper el hielo y generar confianza se puede hablar de su experiencia previa en este contexto, buscar anécdotas. Es recomendable ir a lo sencillo, tirar del baúl de los recuerdos, ir generando confianza y para contestar no es necesario construir algo desde cero. Son cosas que nos han pasado.
3. Preguntas complejas	En esta parte preguntaremos sobre aspectos más motivacionales, aspiracionales y filosóficos del marco de trabajo. Interesa saber cómo los define y entiende el usuario. Esta parte es la más intensa para el entrevistado y para el entrevistador porque el primero, tiene que estar

	pendiente de cómo continuar si aparece un hilo del que tirar o como modificar la pregunta para inspirar al entrevistado si se bloquea. Y el segundo tiene que viajar a su interior para dar respuesta. Es necesario saber aguantar los silencios durante las entrevistas, ya que muchas veces el interlocutor necesita asimilar la pregunta y construir la respuesta.
4. Cierre	Espacio para que el entrevistado añada lo que considere, si es que tiene algo que añadir. Es importante finalizar con un agradecimiento.

2.1.4.5. Empatizar - Análisis DAFO

El análisis DAFO (Debilidades, Amenazas, Fortalezas, Oportunidades) es una herramienta clásica que el método Design Thinking emplea para analizar cuál es la situación de un producto/servicio con respecto al mercado [25].

En la Figura 12 se representa la matriz DAFO conformada por cuatro rectángulos contiguos.

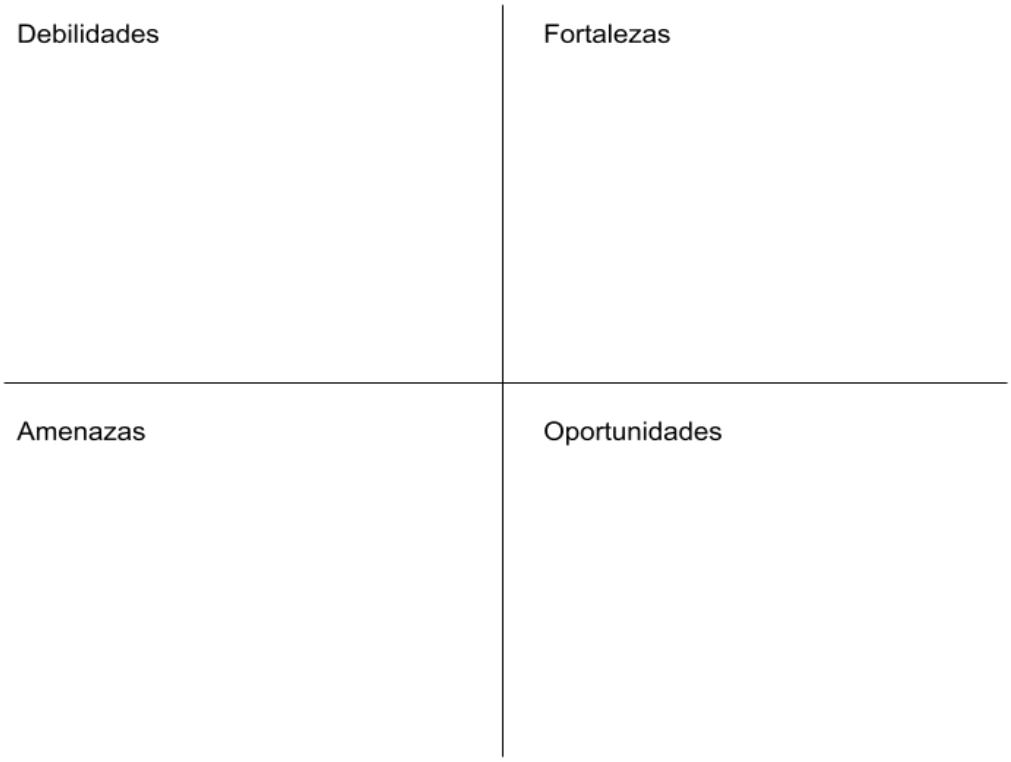

Figura 12: Matriz DAFO [Elaboración propia] [25]

Según la comunidad en línea de Design Thinking [26], en el análisis DAFO se deben realizar las siguientes consideraciones.

Tabla 4: Consideraciones en el uso de la matriz DAFO

Oportunidades	¿A qué buenas oportunidades nos enfrentamos?; ¿de qué tendencias del mercado se tiene información?; ¿qué cambios de tecnología se están presentando en el mercado?; ¿qué cambios en la normatividad legal y/o política se están presentando?; ¿qué cambios en los patrones sociales y de estilos de vida se están presentando?
Amenazas	¿A qué obstáculos nos enfrentamos?; ¿qué están haciendo los competidores?; ¿se tienen problemas de recursos de capital?; ¿se identifican amenazas que puedan impedir totalmente la actividad de la organización?
Fortalezas	¿Qué ventajas existen?; ¿qué se está haciendo mejor que en otras organizaciones?; ¿a qué recursos de bajo costo o de manera única se tiene acceso?; ¿qué perciben los usuarios

	como una fortaleza?; ¿qué elementos facilitan obtener el éxito?
Debilidades	¿Qué se puede mejorar?; ¿que se debería evitar?; ¿qué percibe el usuario como una debilidad?; ¿qué factores reducen el éxito del proyecto?

2.1.4.6. Idear - Bocetos en papel

Los bocetos en papel se utilizan para prototipar construyendo una representación de la solución que se ha diseñado.

Los bocetos son prototipos que se hacen rápidamente y con pocos recursos, no son necesarios complicados dispositivos tecnológicos o materiales de última generación. Con un lápiz y algo de papel se pueden diseñar experiencias de usuario.

En la Figura 13 se bocetan tres interfaces de usuario que representan la experiencia de un usuario al corregir el trabajo de otro usuario de un sistema informático del sector educativo.

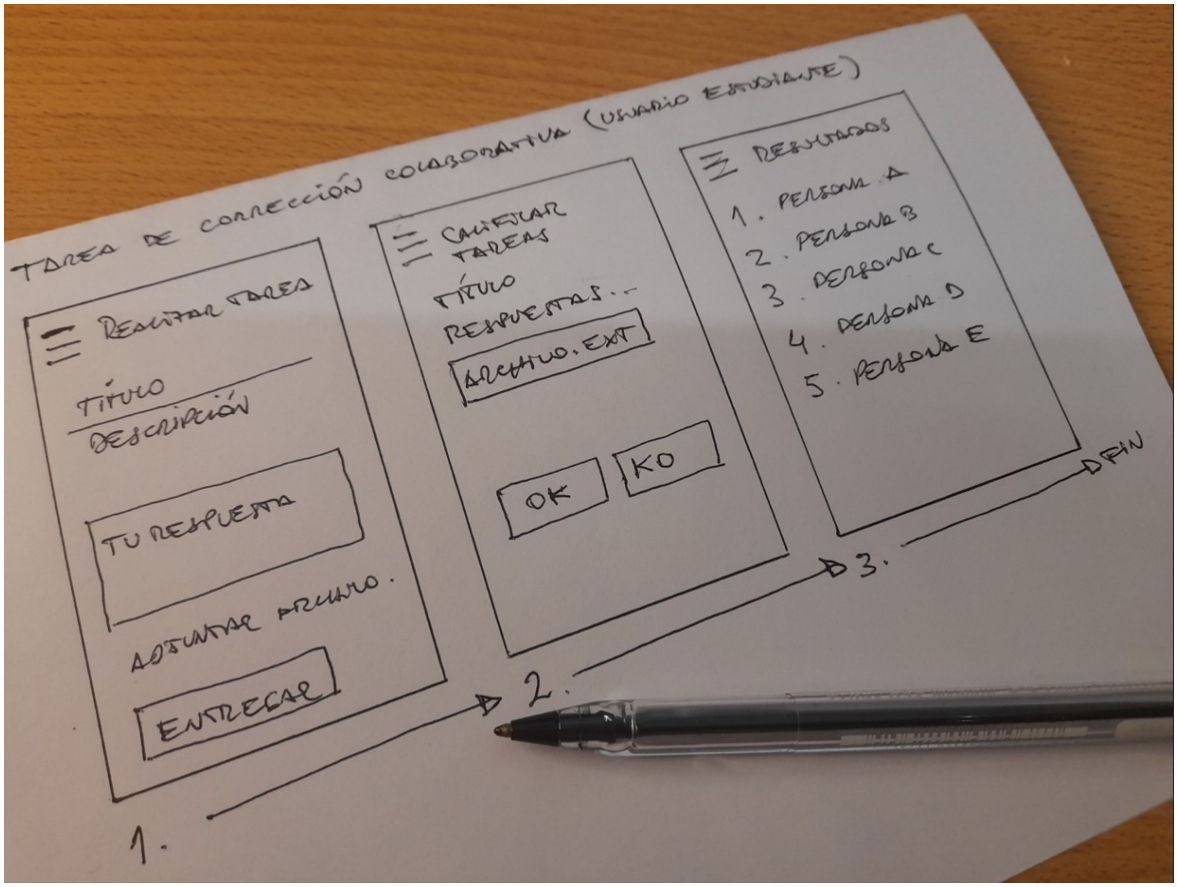

Figura 13: Bocetos y prototipado en papel [Elaboración propia]

2.1.4.7. Idear - Wireframes

Un wireframe es el plano básico que ilustra la forma y las funciones principales que se encuentran en una sola pantalla de la página web o aplicación. La fidelidad de los wireframes aumentará en detalle a medida que se perfeccionen. Sin embargo, es probable que la primera versión solo utilice contornos y formas básicas en blanco y negro para indicar dónde se colocarán los elementos de navegación, el texto y los gráficos en la pantalla. La colección de wireframes debe brindar una vista estructural integral de todo el producto software [27].

A continuación, en la Figura 14 se presenta un wireframe de ejemplo donde el contenido de la de la interfaz de usuario tiene una tarea principal: asistir al usuario para crear una nueva tarea en el contexto de un sistema informático educativo.

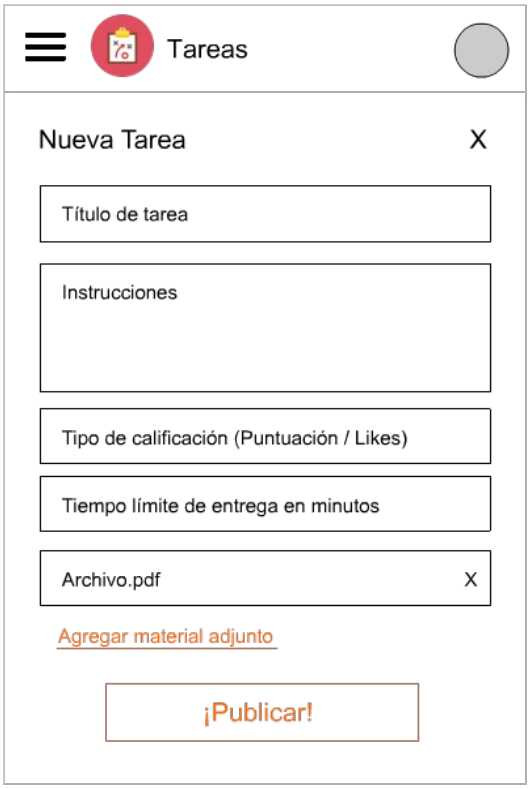

Figura 14: Prototipado con wireframes [Elaboración propia]

En los wireframes se definen puntos de acceso que se muestran como imágenes, encabezados y enlaces. Sin embargo, no se definen los textos finales ni los elementos de los cuales estará compuesta la navegación, ni que aspecto final tendrán las imágenes. Los elementos finales se definirán más adelante, en esta fase se especifican los elementos y puntos de acceso.

2.1.4.8. Idear - Storyboard

Los Storyboards son guiones gráficos que muestran a los actores involucrados en una serie de acciones con un propósito en mente, a menudo en forma de bocetos simples [28].

A diferencia de los wireframes, los Storyboards incluyen a los actores, involucrados en una tarea o proceso, los wireframes en cambio, se centran en el aspecto, el contenido y las conexiones entre interfaces de usuario.

Al hacer un guión gráfico, se deben tener en cuenta los principales artefactos, como

tecnologías o entornos, que se utilizan para representar una narrativa para un diseño. Incluidas en artefactos, las soluciones se pueden recopilar y conectar entre sí para ilustrar una serie de acciones clave. Estos artefactos reflejan una funcionalidad vital y, por lo tanto, pueden ser fundamentales para el diseño de un sistema.

El Storyboard se suele ilustrar como un proceso creativo. Se centra en describir la interacción de un usuario con un sistema en un lapso de tiempo a través de una serie de imágenes, a menudo bocetos, y una narrativa textual. Los Storyboards se utilizan para representar el flujo de una historia, reducir costosos elementos de diseño y proponer ideas a las partes interesadas. Por lo tanto, se pueden considerar como los primeros prototipos de baja fidelidad en el proceso de diseño.

IBM recomienda el uso de Storyboards en tres etapas [29].

1. Comenzar con la historia: Identificar un personaje, un escenario y una trama. Luego elegir las escenas que muestran el desarrollo de la trama de principio a fin.

En la figura 15 se disponen en papeles de colores rectangulares pegados en una pizarra los elementos a identificar, los elementos a identificar son el personaje, ajustes y trama, en el ejemplo de la Figura 15 el personaje es el estudiante, el ajuste representa en contexto, en este caso el estudiante está en su casa, la trama consiste en que el estudiante, en su casa, decide revisar si tiene tareas pendientes de ser entregadas, realizar entregas de trabajos y/o actividades y planificar las futuras entregas.

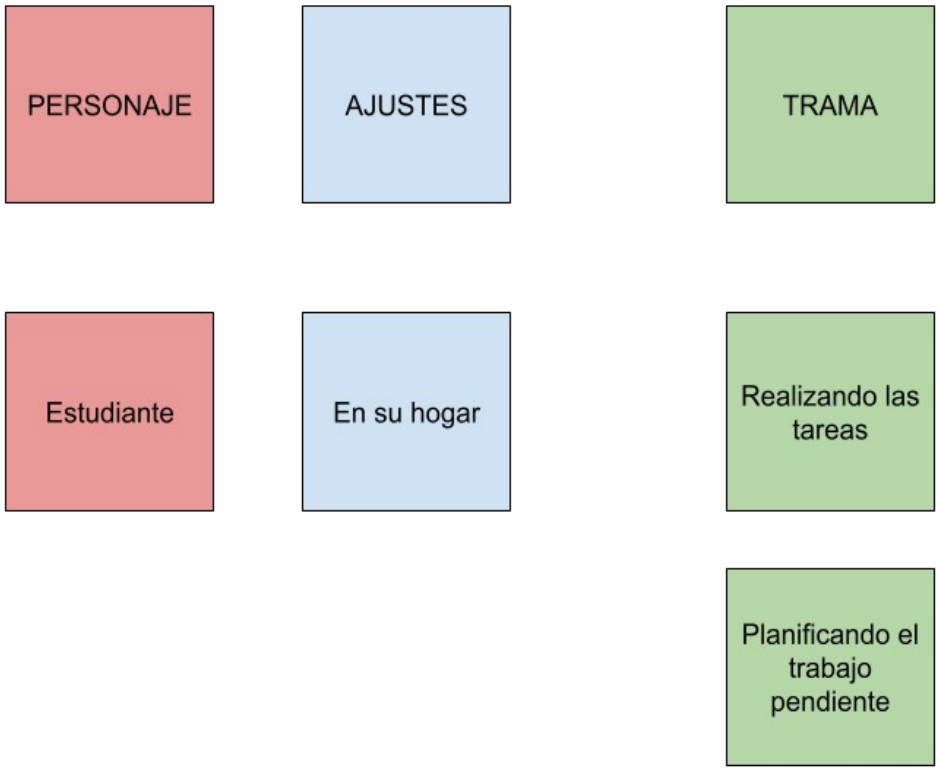

Figura 15: Storyboard - Etapa uno, comenzar con la historia [Elaboración propia] [29]

2. Ilustrar la historia: Intentar pensar en el guión gráfico como una tira cómica. Combinar bocetos rápidos con bocadillos y burbujas de pensamiento, ráfagas de acción, subtítulos y narración.

En la Figura 16 se ilustra la historia del estudiante que decide realizar la tarea desde su hogar a través de un sistema informático de gestión del aprendizaje, la secuencia diseñada incluye cuatro fases:

1. El estudiante se conecta a la plataforma educativa
2. La plataforma muestra las notificaciones ordenadas por prioridad.
3. El estudiante realiza las actividades.
4. La plataforma recibe las entregas realizadas por el alumno.

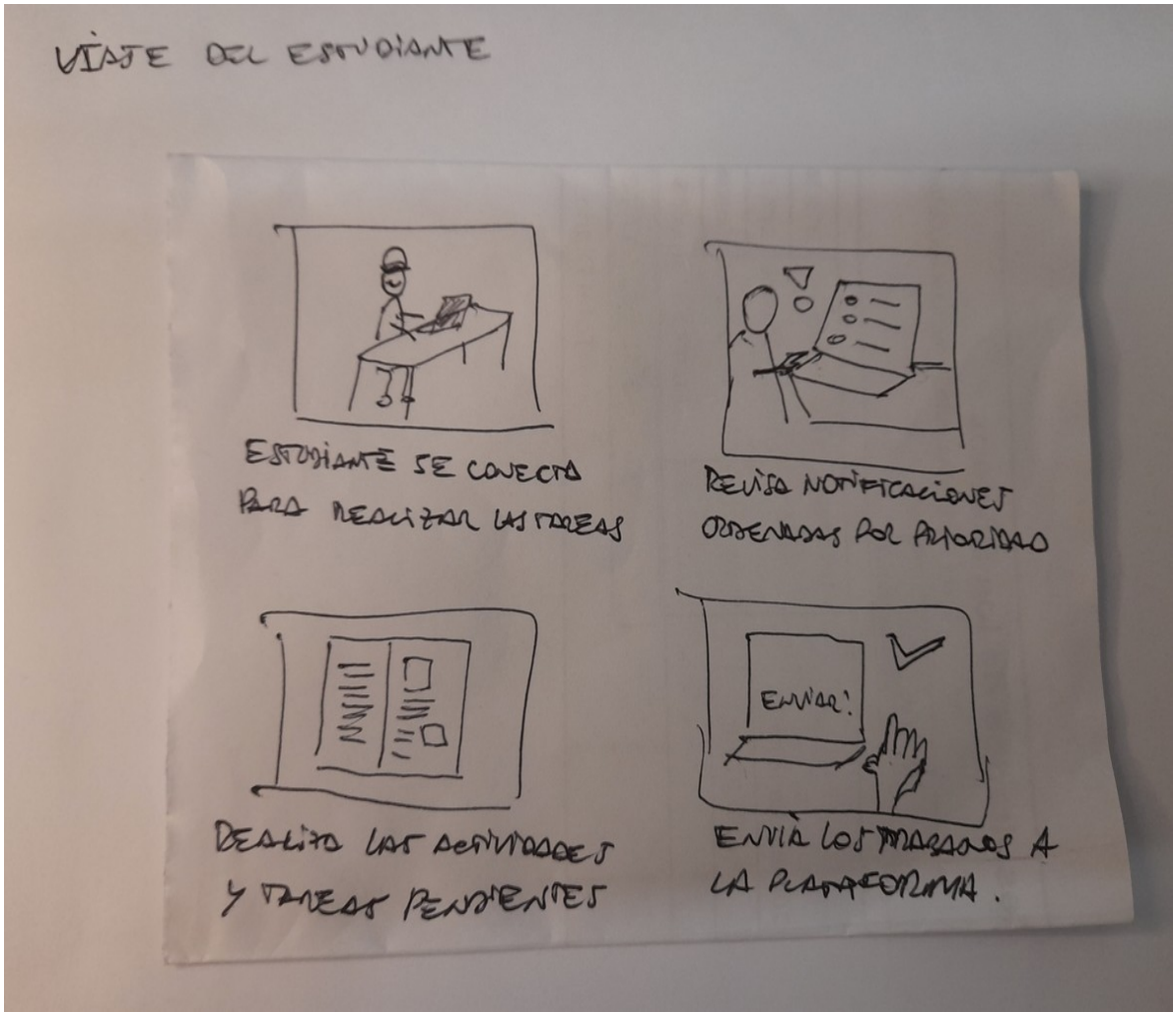

Figura 16: Storyboard - Etapa dos, ilustrar la historia [Elaboración propia] [29]

3. Reproducir y discutir: Compartir la historia con el equipo de trabajo, se debe tener en cuenta que el objetivo no es juzgar la capacidad de dibujar, el objetivo es comunicar una idea en detalle.

En la Figura 16 representa la reproducción de la historia, narrada por su creador o el equipo que la ha confeccionado con el objetivo de que el público presente pueda valorarla.

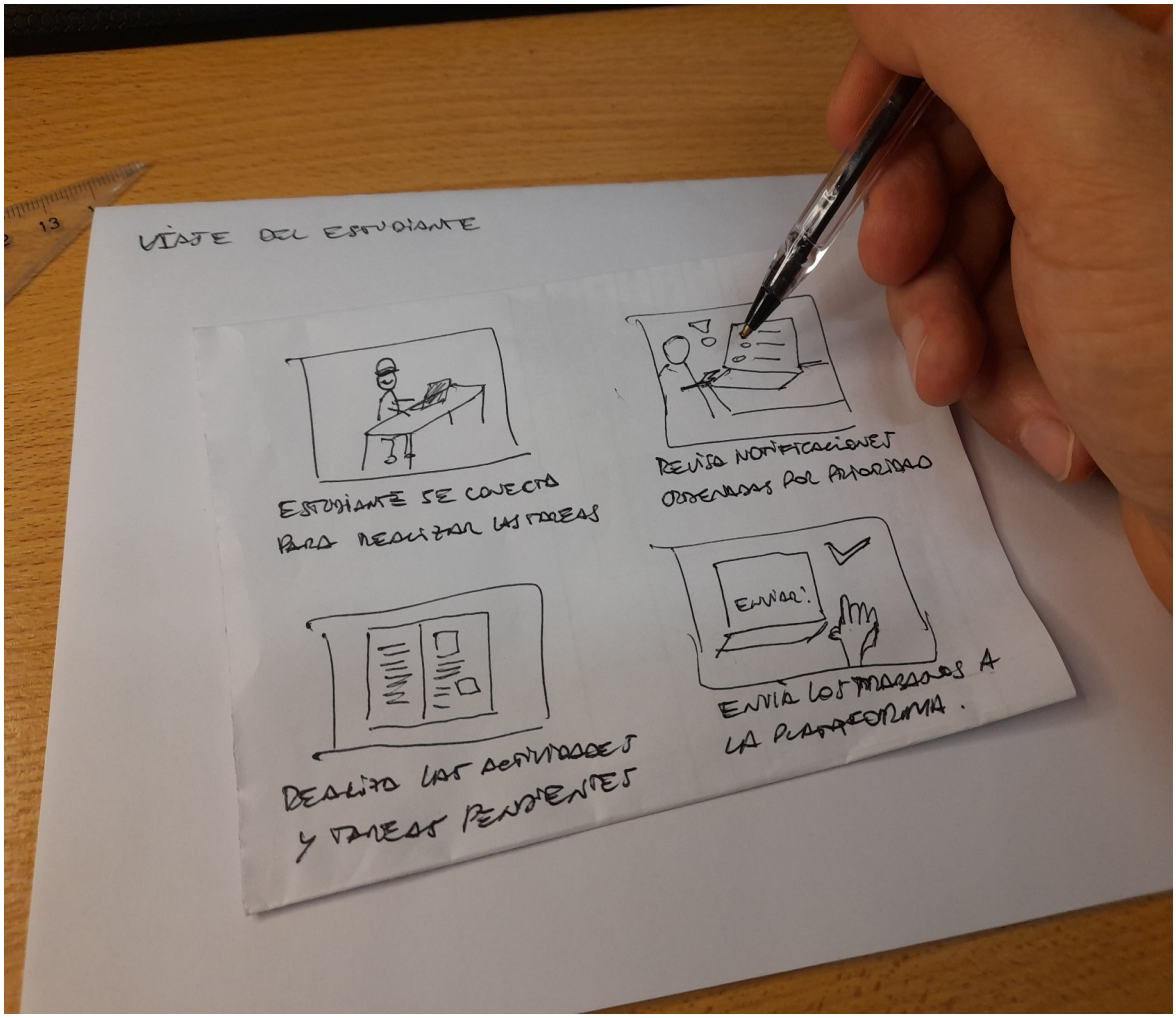

Figura 17: Storyboard - Etapa tres, reproducir y discutir [Elaboración propia] [29]

2.1.4.9. Prototipos

La aplicación de Design Thinking propone, en todas las versiones estudiadas en el presente trabajo, el diseño e implementación de prototipos como una etapa fundamental. Para la construcción de prototipos se deben tener en cuenta las fases de empatizar con el usuario e ideación, lo realizado en dichas etapas constituyen el material de entrada, son especificaciones necesarias para la construcción del prototipo que se validará en la siguiente fase.

Un prototipo es todo aquello tangible que nos permite explorar una idea, evaluarla y llevarla adelante [5]. Los prototipos en Design Thinking tienen el propósito principal de

apoyar el aprendizaje sobre el concepto de producto subyacente [9].

En informática, los modelos juegan un papel clave en el diseño de sistemas complejos. Un modelo es un artefacto que está diseñado para permitir la observación de aquellos aspectos del sistema final que se consideran relevantes.

El modelado no está restringido a la informática. Si el objetivo del proceso de diseño es un producto físico, los prototipos pueden estar hechos de papel, madera o algún otro material físico con el que sea fácil trabajar. Un prototipo de madera de un porta vasos, por ejemplo, sirve como modelo, porque los usuarios finales pueden usar el porta vasos de madera para observar el mismo comportamiento o similar que el producto final. Los prototipos permiten estudiar los aspectos del producto físico en el que está interesado el proceso de diseño. Los usuarios finales pueden tomar el prototipo, experimentarlo y proponer formas de mejorarlo. Esto funciona bien, porque los prototipos son tangibles. Son bastante similares al producto final en el sentido de que exponen un comportamiento similar y permiten observaciones que son relevantes para mejorar el diseño. Al diseñar una interfaz gráfica de usuario de un sistema de software, la situación es algo diferente. La razón es que el producto final no es tan tangible como lo es un producto físico. Podemos argumentar que la interfaz de usuario sigue siendo el aspecto más tangible de un sistema de software, y esa puede ser una razón por la que frecuentemente se encuentra en el centro de los procesos de Design Thinking relacionados con el software [7].

Para hacer tangible la interfaz de usuario e invitar a los usuarios a formular comentarios, se desarrollan prototipos en papel, llamados prototipos de baja fidelidad. Estos prototipos en papel permiten discutir el comportamiento del sistema final con los usuarios finales. Una vez que el usuario está satisfecho con el prototipo de baja fidelidad, se desarrolla un prototipo de alta fidelidad, por ejemplo, utilizando interfaces de usuario navegables, con zonas calientes. Este prototipo es una maqueta del producto final, por lo que se pueden discutir problemas de UI, según sus siglas en inglés User Interface.

Finalmente, el diseño se completa, y la ingeniería se hace cargo. Con respecto al modelado, un prototipo en papel de una interfaz gráfica de usuario sirve como modelo.

Un prototipo más interactivo de una interfaz gráfica de usuario sirve como modelo y posibilita observaciones más detalladas. En muchas disciplinas de ingeniería, los prototipos y la simulación se emplean de manera regular para comprender mejor el

problema en cuestión e identificar soluciones innovadoras.

En situaciones donde se busque resolver problemas complejos a través de la aplicación de Design Thinking, la creación de prototipos y la simulación ayudan a resolver conflictos de diseño en una etapa temprana, se puede validar el diseño de solución con equipos multidisciplinarios, pueden ser también los mismos usuarios finales, al exponer a los usuarios al prototipo o simulación se plantea la situación "qué pasaría si", habilitando la retroalimentación inmediata que estimula la innovación y capacidad de resolución de problemas complejos a través del diseño del producto [30].

Para el software, el pionero Fred Brooks declaró en el año 1986 que "La parte más difícil de construir un sistema de software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados… Ninguna otra parte afecta tanto al sistema resultante si se hace incorrectamente" [31], los prototipos pueden ser las herramientas más efectivas para determinar en una etapa temprana qué software se debe construir.

2.1.4.10. Prototipos en software multiusuario

Un sistema multiusuario puede tener gran cantidad de escenarios de uso durante el proceso de Design Thinking. Estos escenarios pueden considerarse como instancias de proceso, y deben ser analizados y relacionados entre sí. Por ejemplo, es importante encontrar inconsistencias entre los escenarios o validarlos al pasar por los procesos respectivos. El modelado de escenarios y procesos, así como las técnicas de simulación, son conceptos importantes para estudiar las relaciones entre los escenarios de uso. Otro problema clave para diseñar sistemas complejos multiusuario es la trazabilidad.

La información recopilada por los diferentes miembros del equipo que implementa el software generalmente no se interpreta bien y no se vincula correctamente con las soluciones previstas en los prototipos.

Por lo tanto, el diseño y la ingeniería deben estar integrados. Esto significa que los artefactos de diseño que documentan el proceso de diseño y sus decisiones deben estar disponibles durante la ingeniería. Los prototipos juegan un papel clave en la implementación final del software [7].

2.1.4.11. Testear - Validar prototipos con usuarios

Durante la fase de prueba y validación del prototipo, se buscan comentarios valiosos de los miembros del equipo y los usuarios. Los prototipos deben ser fáciles de entender y exponer el mismo comportamiento que el producto final [7].

El testing es el proceso de validación del prototipo con usuarios reales. Durante la fase de prueba, se verá cómo los usuarios interactúan con el prototipo y se recopilarán comentarios valiosos. Se identificará dónde tiene éxito el prototipo y dónde debe mejorarse. Los conocimientos recopilados durante la fase de prueba permitirán iterar en su prototipo [32].

Algunos de los objetivos que se persiguen en el proceso de testing son los siguientes:

- **Ahorro de tiempo y costos**: al detectar errores y problemas de usabilidad desde el principio, el producto será más fácil de usar y con menos errores. Al detectar falencias en el diseño en fases tempranas ahorrará la refactorización de código en la programación.
- **Revelar información inesperada:** no importa qué tan exhaustiva fue la investigación inicial o qué tan convencido se esté sobre el diseño de la solución óptima, siempre hay nuevos indicadores por descubrir.
- **Mejorar la satisfacción del usuario:** el proceso Design Thinking trata de poner al usuario en primer lugar. Al recopilar comentarios de primera mano de los usuarios, nos mantendremos enfocados en el usuario, mejorando su nivel de satisfacción.

Existen diferentes métodos y técnicas para validar los prototipos. Las pruebas de usuario pueden realizarse en persona o de forma remota, pueden ser moderadas o no moderadas.

En las pruebas de usuario en persona se compartirá un mismo espacio de trabajo con el usuario mientras que prueba el prototipo, esto tiene varias ventajas. Se puede controlar el entorno y mantener las distracciones al mínimo, se puede observar directamente al usuario, estaremos al tanto de las expresiones faciales, el lenguaje corporal

y cualquier comentario verbal que haga el usuario mientras interactúa con el prototipo, lo que brinda información valiosa y de primera mano sobre la experiencia. Las pruebas en persona pueden ser costosas y requerir mucho tiempo.

Las pruebas de usuarios remotos ofrecen una alternativa menos costosa y más conveniente, pero se tiene poco o ningún control sobre el entorno de prueba del usuario. Las pruebas en remoto posibilitarán realizar el testing en un mayor número de usuarios finales al mismo tiempo.

Las pruebas de usuarios remotos moderadas son un buen término medio entre las pruebas en persona y las pruebas remotas completamente no moderadas. Las pruebas remotas en vivo permiten observar a los usuarios a través de una video llamada, por ejemplo. Se puede usar una aplicación de grabación de pantalla para capturar la prueba, y ciertos programas también rastrearán y resaltarán dónde el usuario hace clic en el prototipo digital.

Las pruebas no moderadas se pueden realizar a través de plataformas de prueba de usuarios como UserZoom [33], Loop11 [34] y UserTesting [35]. Si se dispone de poco tiempo, estas herramientas facilitan la realización de pruebas de usuario de forma rápida y con un menor esfuerzo, aunque, no se tendrá la oportunidad de observar a los usuarios o hacerles preguntas.

Realizar pruebas de usuario en persona o remotas, moderadas o no moderadas, dependerá del presupuesto, las limitaciones de tiempo y el prototipo que se esté validando. Los prototipos de papel se prueban mejor en persona, mientras que los prototipos digitales se pueden probar tanto de forma remota como de manera personal.

2.1.5. Aplicaciones de Design Thinking

En contraste con el pensamiento analítico en la ciencia, el proceso Design Thinking asume un rol de coordinación dentro del contexto de desafíos múltiples, porque tienen que confiar en el conocimiento de los demás siempre que quieran abordar la viabilidad general de las soluciones. Para ello, surgieron estrategias de aprendizaje profesional en el diseño: patrones cognitivos para captar conocimiento múltiple y perspectivas múltiples de otros con el propósito de sintetizar y transformar creativamente el conocimiento a nuevos conceptos de servicio o producto [5].

El objetivo de aplicar la Design Thinking al desarrollo tecnológico requiere establecer un estilo de pensamiento complementario, que amplíe las capacidades de resolución de problemas de los equipos de desarrollo tecnológico con el propósito de hacer que sus resultados sean más innovadores [9].

2.1.6. Design Thinking aplicado al desarrollo de software

Los marcos de trabajo sobre Design Thinking de Google Design Sprints y SAP fueron específicamente diseñados para ser aplicados en el desarrollo de software.

Según un estudio [9] sobre la aplicación de Design Thinking en el sector IT, según sus siglas en inglés Information Technology, se ha revelado que aquellos empleados que reportan niveles de jerarquía más altos perciben al Design Thinking como un riesgo. Para ellos, es más seguro planificar procesos de desarrollo basados en hitos (con un enfoque principal en el pensamiento convergente), incluso si los resultados no son tan innovadores como podrían ser. En este contexto, no es sorprendente que las organizaciones centradas en Design Thinking se hayan creado primero en los laboratorios y agencias universitarias. Cuando se trata de la decisión de integrar Design Thinking dentro de la estructura organizativa interna de una empresa, es bastante probable que se considere como una técnica de frontend, relacionada a mejorar lo que se denomina UX, según sus siglas en inglés User Experience.

Tradicionalmente, los procesos de desarrollo de IT están orientados hacia la tecnología y tienen lugar dentro de las esferas de competencia de los ingenieros de software y hardware. La integración de la perspectiva del usuario normalmente sólo tiene lugar justo antes del trabajo de desarrollo real (fase de especificación) y en las etapas finales de desarrollo (eliminación de errores de software). Por lo tanto, ni las opiniones de los usuarios, ni la colaboración multidisciplinaria desempeñan un papel central en el proceso de desarrollo real. Como resultado, las funcionalidades y las interfaces de usuario se pueden planificar de una manera que no tenga en cuenta las necesidades del usuario. Asimismo, los procesos de creatividad que pueden mejorar el carácter innovador de un producto, no se realizan [9].

2.2. Desarrollo de tecnologías de información aplicado a la educación

2.2.1. Entornos educativos interactivos

Al momento de desarrollar la plataforma interactiva, se presenta una diferencia considerable en los procesos donde una sola persona es el centro de atención.

En el contexto de aula interviene más de una persona, por lo que se considera un sistema multiusuario. En los sistemas multiusuario al tratarse de varias personas involucradas, cada una de ellas normalmente tendrá un punto de vista diferente. Estos puntos pueden exponer comportamientos conflictivos. Es necesario identificar los aspectos importantes intrínsecos a la actividad que se llevará a cabo y asignar prioridades a las necesidades.

2.2.2. Aprendizaje mejorado con tecnología

TEL según sus siglas en inglés Technology-Enhanced Learning, es un campo que estudia el aprendizaje mejorado por tecnología como un tema relevante, cuya importancia es reconocida por investigadores educativos, profesionales, diseñadores de software y entidades gubernamentales [36].

El aprendizaje mejorado con tecnología es un campo de investigación en el que se interceptan diferentes disciplinas, como la informática, la educación, la psicología, la filosofía, la pedagogía y la comunicación.

Considera el aprendizaje mejorado con tecnología un tema multifacético y multidisciplinario desde cuatro puntos de vista diferentes [36]:

1. Principios de aprendizaje.
2. Aprendizaje en dominios específicos.
3. Forma del entorno de aprendizaje.
4. Tecnologías especiales.

La creciente dependencia de la tecnología de la información y comunicación (TIC) para mejorar el aprendizaje y la enseñanza ha introducido un factor de complejidad en los

entornos de aprendizaje de todo tipo. Esta complejidad tecnológica, combinada con la diversidad pedagógica natural de los entornos educativos plantea una variedad de desafíos para profesionales y estudiantes por igual. Esto es especialmente cierto en entornos educativos reales, donde se aplican restricciones adicionales (logística, currículo, restricciones de tiempo, etc.). En el campo del aprendizaje mejorado con tecnología, la metáfora de "orquestación del aprendizaje" se ha utilizado con frecuencia para abordar los desafíos de coordinar la complejidad de realizar actividades de aprendizaje en estos entornos. Sin embargo, la orquestación ha sido definida por los investigadores y existen pocas definiciones concretas del concepto [37].

2.2.3. Orquestación en el aula

Prieto [37] define la orquestación como “Los métodos y estrategias empoderadas por aulas equipadas con tecnología que un educador puede adoptar cuidadosamente para involucrar a los estudiantes en actividades conducentes al éxito y aprendizaje”.

La orquestación se refiere a cómo un profesor administra, en tiempo real, las actividades en un contexto de aula. Muchos escenarios pedagógicos integran actividades individuales, como la lectura, el trabajo en equipo, la resolución de problemas; y actividades en todo el aula, como las conferencias.

Algunas de estas actividades están basadas en dispositivos, otras no. Algunas son cara a cara, mientras que otras están en línea. Esta integración pedagógica se refleja en la aplicación técnica de diferentes herramientas, como simulaciones, actividades digitales, wikis, etc., distribuidas en múltiples dispositivos, como computadoras portátiles, tabletas, y teléfonos inteligentes [38].

Las aulas donde se imparte enseñanza son complejas, con múltiples propósitos, roles, recursos y actividades.

El rol del docente es de particular importancia. El docente decide qué se implementa o no en las aulas. Los docentes pueden percibir más dificultades relacionadas al uso de tecnologías en relación a potenciales beneficios.

La tecnología se desarrolla en el laboratorio en entornos idealizados, y luego se transfiere al aula, a los entornos de práctica.

El entorno del laboratorio facilita la investigación científica, pero es difícil generar impacto desde allí. En la práctica pueden presentarse varios problemas. Las aulas son complejas y altamente variables, son impredecibles, donde el recurso más crítico es el tiempo.

En muchas ocasiones, los docentes no adoptan tecnologías de aprendizaje basadas en la investigación. Tales tecnologías frecuentemente agregan demasiada complejidad, no logran ser lo suficientemente flexibles en el mundo impredecible del aula, o simplemente toman demasiado tiempo [39].

2.2.3.1. Diseño de actividades

Diseñar para la orquestación implica facilitar el flujo dual de información en un aula a través de contenedores de información digital y física. El flujo dual se refiere al flujo de información digital y el flujo de información analógica. Este flujo de trabajo integra actividades de aprendizaje junto a otras actividades en el aula. Se tiene en cuenta que, además de aprender, los docentes enfrentan muchas limitaciones: tiempo, disciplina, currículo, energía, etcétera. Algunos principios de diseño parecen cumplir con estos requisitos: flexibilidad, control, visibilidad, minimalismo y dimensión física.

El objetivo de la usabilidad a nivel individual es minimizar la carga cognitiva. En el contexto de aula se trata de minimizar la carga de orquestación.

El trasfondo pedagógico es el empoderamiento de los docentes como impulsores de las actividades en el aula. El trasfondo tecnológico es el minimalismo. Tecnologías que no llegan a ser inteligentes, en el sentido de inteligencia artificial, pero que tienen efectos simples como hacer visible cosas, pensamientos, interacciones, que de otra manera serían invisibles [38].

A continuación, en la Tabla 5, se identifican factores intrínsecos, en la Tabla 6, los factores extrínsecos y consideraciones útiles para el diseño de actividades en un contexto de aula.

2.2.3.2. Factores intrínsecos y extrínsecos

En la orquestación, el docente debe tener en cuenta tres factores intrínsecos y seis factores extrínsecos [38].

Tabla 5: Factores intrínsecos en la orquestación en el aula

1. Qué	Los contenidos que debe enseñar
2. Quién	Las características del alumno
3. Cómo	Cómo la mente del alumno construirá el conocimiento

Tabla 6: Factores extrínsecos en la orquestación en el aula

1. Tiempo	El tiempo es un recurso escaso en la enseñanza, una preocupación permanente de los docentes.
2. Relevancia del plan de estudio	Los docentes deben enseñar lo que está contemplado en el plan de estudios, el diseño tecnológico debe ser consecuente.
3. Disciplina	Los directivos y padres esperan que los docentes tengan el control de sus estudiantes teniendo un nivel razonable de disciplina. Se espera cierta animación y motivación en el aula, pero en ningún caso el caos.
4. Evaluación	Las instituciones educativas están impulsadas por la necesidad de proporcionar calificaciones. Un buen escenario pedagógico puede ser abandonado si la evaluación no está presente. En ocasiones los docentes no promueven los proyectos colaborativos por la dificultad de proporcionar finalmente calificaciones individuales.
5. Energía	El esfuerzo total que un docente puede invertir es limitado (tiempo para preparar la clase, para evaluar, para enviar devoluciones). Hay que tenerlo en cuenta para no perder escalabilidad y sostenibilidad.
6. Espacio	Es importante determinar si hay suficiente espacio en el aula para organizar actividades. Además, evaluar si es compatible con las interacciones sociales y el formato de trabajo.

Existen más restricciones extrínsecas, como son las finanzas, la autoestima de los docentes, la seguridad, la cultura, entre otros. Sin embargo, teniendo en cuenta los seis factores nombrados anteriormente en el diseño tecnológico, se puede considerar un paso importante [38].

2.2.3.3. Consideraciones para el diseño de actividades

Según experimentos realizados utilizando orquestación en contexto del aula, Pierre Dillenbourg [38] enumera consideraciones importantes para diseñar actividades que

requieren orquestación conducidas por docentes.

2.2.3.3.1. Control

El docente es el conductor, controla lo que sucede en el aula. En términos de diseño, el control se relaciona con la flexibilidad, la decisión del docente debe reemplazar cualquier decisión del sistema. El control se refiere a la consciencia de lo que está sucediendo en el aula, no solo dentro del entorno de aprendizaje diseñado, sino también en el contexto físico y real del aula.

2.2.3.3.2. Visibilidad

Se refiere a la visibilidad de lo que está ocurriendo del entorno de aprendizaje diseñado, se considera a su vez un factor de control. La visibilidad no es un proceso en una sola dirección, desde el alumno hacia el docente, como el caso de las herramientas de monitoreo, sino más bien en ambas direcciones, visibilidad para todos los actores del aula.

2.2.3.3.3. Flexibilidad

Se refiere a la posibilidad de que un docente cambie sobre la marcha cualquier decisión tomada por él o por el diseñador al preparar la actividad. La flexibilidad no solo adapta el escenario a las necesidades de los alumnos, sino que también reacciona de manera adecuada a los eventos y factores extrínsecos.

Puede existir un conflicto de intereses entre la flexibilidad y el flujo de trabajo, ya que el diseño de la actividad trata de automatizar los procesos, pero fundamentalmente, la orquestación solo puede ocurrir si el docente tiene la libertad de cambiar las actividades sobre la marcha.

2.2.3.3.4. Dimensión física

Se debe tener en cuenta la forma de los dispositivos, como así también que cualquier otro elemento físico pueda intervenir en la actividad. La dimensión física se revela a través del uso de materiales tangibles. La orquestación es física, se trata de la movilidad, la mirada y la distancia entre los actores.

2.2.3.3.5. Minimalismo

Es un principio que se aplica a todos las consideraciones antes mencionadas, el minimalismo se aborda teniendo en cuenta que cada herramienta debe estar diseñada para el uso en tiempo real. Se trata de minimizar la carga de factores extrínsecos y fomentar los recursos de los estudiantes sobre los factores intrínsecos y pertinentes. Las herramientas de orquestación deben minimizar la carga de factores extrínsecos. Es decir, cualquier complejidad creada accidentalmente por la tecnología. De esta manera, llegamos a la noción de usabilidad en el aula.

2.3. Metodología de desarrollo ágil

En el año 2001 se definió el manifiesto de desarrollo ágil de software que sentó las bases para nuevas y mejores maneras para desarrollar software donde se propone lo siguiente [40].

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

La metodología de desarrollo ágil promueve una entrega rápida, se ajusta a cambios de requerimientos y asegura la satisfacción del cliente [41]. A si mismo, promueve procesos iterativos e incrementales en el desarrollo de software, cuando se compara el marco de trabajo ágil con el desarrollo de software planificado, la metodología ágil tiene un mejor desempeño, reduce el número de fallas causadas por los cambios de requerimiento [42].

En lugar de centrarse en un lanzamiento de gran envergadura, un equipo ágil entrega el trabajo en incrementos pequeños, pero que se pueden consumir. Los requisitos, los planes y los resultados se evalúan de forma continua, de modo que los equipos disponen de un mecanismo natural para responder con rapidez ante los cambios [43].

2.3.1. Scrum como marco de trabajo

Se suele pensar que Scrum y la metodología ágil son lo mismo porque el Scrum se centra en la mejora continua, que es un principio básico de la metodología de desarrollo ágil. Sin embargo, Scrum es un marco para realizar el trabajo, mientras que la metodología de desarrollo ágil es una mentalidad. En realidad, uno solo no puede "adoptar una metodología ágil", ya que requiere la dedicación de todo el equipo para cambiar la mentalidad a la hora de ofrecer valor a los clientes. Lo que sí que se puede usar es un marco de trabajo como Scrum para ayudar al equipo de desarrollo tecnológico a empezar a pensar de esa manera y poner en práctica la aplicación de principios de metodología ágil en la comunicación y el trabajo diario.

El marco de trabajo de Scrum es heurístico. Se basa en el aprendizaje continuo y en la adaptación a los factores fluctuantes. Reconoce que el equipo no lo sabe todo al inicio de un proyecto y evolucionará a través de la experiencia. Scrum está estructurado para ayudar a los equipos a adaptarse de forma natural a las condiciones cambiantes y a los requisitos de los usuarios, con el cambio de prioridades integrado en el proceso y ciclos de lanzamiento breves para que tu equipo pueda aprender y mejorar constantemente [43].

En la Figura 18 se representa el marco de trabajo Scrum como un círculo, se identifican a su vez dentro de él las partes que la componen y que se describen a continuación.

El **Sprint**: es un período breve de tiempo fijo en el que un equipo de Scrum trabaja para completar una cantidad de trabajo establecida.

La **planificación de sprint** es el evento en Scrum que inicia el sprint. El objetivo de la planificación de sprints es definir lo que se puede entregar en el sprint y cómo se conseguirá ese trabajo.

La **daily** es una reunión diaria en la que se implica al equipo básico, normalmente cada asistente responde a las preguntas ¿En qué estuve trabajando ayer?, ¿En qué estoy trabajando hoy? Y ¿Qué problemas me están bloqueando?

Una **revisión de sprint** consiste en demostrar el trabajo de todo un equipo: diseñadores, desarrolladores y el propietario del producto.

En la reunión **retro** el equipo debe reflexionar sobre el comportamiento para ser más eficaces y, después, equilibrarlo y ajustarlo en consecuencia.

El **backlog de sprint** es un conjunto de tareas del backlog del producto que se han impulsado para desarrollarse durante el siguiente incremento del producto.

El **backlog del producto** es una lista de nuevas funciones, mejoras, correcciones de errores, tareas o requisitos de trabajo necesarios para crear un producto.

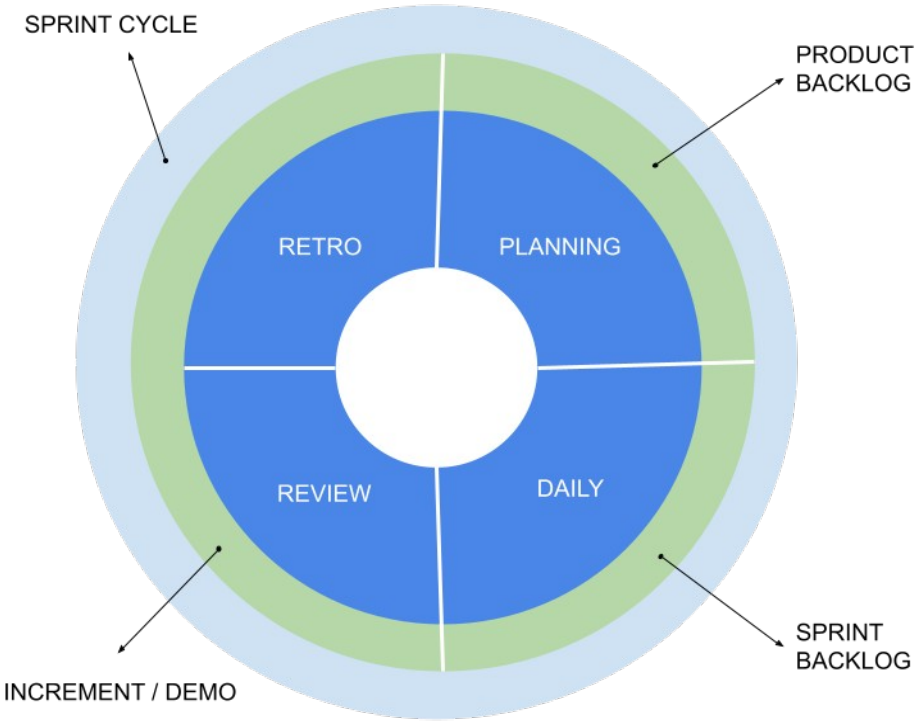

Figura 18: Elementos del marco de trabajo Scrum para el desarrollo ágil [Elaboración propia] [43]

2.3.1.1. Modelo agile de Spotify

Cuando se necesita escalar la metodología ágil y existen en la organización los recursos necesarios, se puede optar por una organización basada en Squads [44], utilizada por Spotify para escalar la metodología ágil, la compañía sueca de transmisión de música Spotify propone escalar la metodología ágil para lograr que las unidades de desarrollo sean más dinámicas y usar las habilidades de una manera más específica mediante la formación

de equipos pequeños, multifuncionales y auto organizados.

La organización para escalar la metodología ágil propone una división de los empleados en Tribes, Squads, Chapters & Guilds. Este modelo ahora también se conoce como modelo de Spotify y se considera uno de los modelos de trabajo ágil más avanzados de la industria. En particular, su escalabilidad es una gran ventaja para las empresas en crecimiento.

Squads (Equipos, Escuadrones), podría equivaler al Scrum Team de Scrum. Consiste en un pequeño equipo multifunctional y auto-organizado de hasa 8 personas. Es un equipo con responsabilidad de principio a fin y trabajan juntos hacia una meta a largo plazo. Su clave es la autonomía.

Cada Squad tiene autonomía para decider qué hacer y cómo hacerlo, así como la forma de trabajar juntos para conseguirlo.

A pesar de contar con autonomía, es necesario tenerlos alineados mediante la misión, la estrategia de producto utilizando metas a corto plazo.

La figura del líder se destaca, es el encargado de comunicar el problema que necesita ser resuelto. Así mismo es responsabilidad del líder el explicar por qué es necesario resolver dicho problema.

Una vez puestos en situación, es el turno de los Squads el colaborar entre sí para encontrar la mejor solución.

Tribes (Tribus), son una estructura matricial bastante ligera que lo que pretende es agrupar a los Squads y mantenerlos bajo el foco de un producto. Podemos decir que agrupa a los Squads bajo un paraguas de negocio, por ejemplo, el desarrollo de funcionalidades relativas a la aplicación móvil que la compañía ofrece a sus clientes.

Las tribus se colocalizan bajo el mismo espacio físico y el número de Squads por tribu no debe de exceder las 100, aunque generalmente hablamos de números bastante inferiores.

El liderazgo de la tribu es llevado a cabo por los líderes de que estén dentro de la misma tribu, está formado por personas representativas del producto, ingeniería, diseño, negocios, entre otros, cuya formación dependerá de las propias necesidades de la tribu. La

clave para el liderazgo de la tribu es que, al igual que el liderazgo del capítulo, se trata de un liderazgo de servicio.

Chapter (División, Departamento, Sección), el chapter es otro tipo de agrupación cross-squad, que se focaliza en áreas de competencia, como por ejemplo, la calidad, el desarrollo frontend, backend, el ux/ui, entre otros.

Los chapters están dirigidos por un chapter lead, cuyo propósito principal es apoyar y enfocar el crecimiento de los miembros del chapter, tanto como individuos como grupo.

Guild (Gremio), esta organización, más parecida a una comunidad de interés, agrupa personas de toda la compañía que quieren compartir conocimiento en un área específica. Es una comunidad abierta, cualquiera puede unirse al gremio o salir de él en cualquier momento.

Las personas pueden pertenecer a uno o varios gremios. Cada persona puede decidir cuán de activo o inactivo es en relación a cada uno de los gremios a los que está adscrito.

Los gremios también son orgánicos. Los gremios se forman en torno a cualquier interés común, incluso aquellos que no tienen que ver con el propio producto o negocio.

En la Figura 19 se representa el esquema propuesto por Spotify para trabajar con metodología ágil.

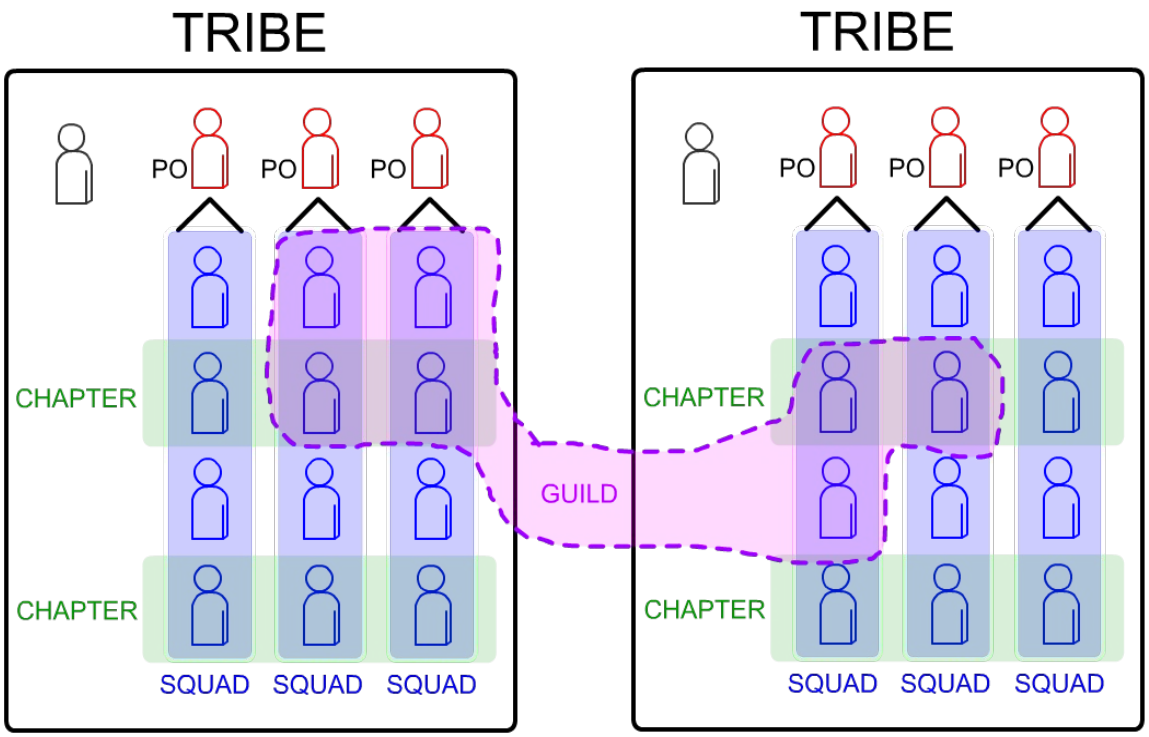

Figura 19: Marco de trabajo propuesto por Spotify para escalar la metodología ágil [Elaboración propia] [44]

2.4. Producto de software

El software es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación [45].

Definimos como producto software al conjunto de programas y recursos que representan un producto utilizable y disponible para los usuarios finales.

2.4.1. Demos

Las Demos o Demo [43] como se suele decir, son programas de demostración, tratan de visualizar algo para que el público objetivo pueda comprender. En el proceso de

desarrollo de producto software, las demos se pueden realizar internamente, dentro de la organización, o externamente, con por ejemplo, clientes, potenciales clientes y usuarios finales.

En el desarrollo interno, en el marco de la aplicación de metodología de desarrollo ágil, las demos se realizan al finalizar el Sprint.

2.4.2. Desarrollo y operaciones

También llamado DevOps, es un conjunto de prácticas que combina el desarrollo de software (Dev) y las operaciones de TI (Ops). Su objetivo es acortar el ciclo de vida del desarrollo de sistemas y proporcionar una entrega continua con software de alta calidad. DevOps es complementario con el desarrollo de software ágil.

La principal característica del movimiento DevOps es defender la automatización y el monitoreo en todos los pasos de la construcción del software, desde la integración, las pruebas, la liberación hasta la implementación y la administración de la infraestructura. DevOps apunta a ciclos de desarrollo más cortos, mayor frecuencia de implementación, lanzamientos más confiables, en estrecha alineación con los objetivos de las organizaciones o empresas. Con un aumento en el uso de las metodologías de software ágil, Lean y DevOps en los últimos años se pasó denominar estas prácticas colectivamente como desarrollo continuo de software, o simplemente CSD por sus siglas en inglés Continuous Software Development [46].

2.4.3. Integración continua y entrega continua

La integración continua es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas. La integración continua se refiere en su mayoría a la fase de creación o integración del proceso de publicación de software y conlleva un componente de automatización (p. ej., CI o servicio de versiones) y un componente cultural (p. ej., aprender a integrar con frecuencia). Los objetivos clave de la integración continua consisten en encontrar y arreglar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo que se tarda en validar y publicar nuevas actualizaciones de software [47].

El modelo de integración continua es conocido como CI/CD. La sigla CI/CD tiene diferentes significados. La CI se refiere a la integración continua, que es un proceso de automatización para los desarrolladores. Si la CI tiene éxito, los cambios del código nuevo en una aplicación se diseñan, se prueban y se combinan periódicamente en un repositorio compartido. Esto soluciona el problema de que se desarrollen demasiadas divisiones de una aplicación al mismo tiempo, porque podrían entrar en conflicto entre sí.

La CD se refiere a la distribución o la implementación continuas, los cuales son conceptos relacionados que suelen usarse indistintamente. Ambos conceptos se refieren a la automatización de las etapas posteriores del canal, pero a veces se usan por separado para explicar la cantidad de automatización que se está incorporando [48].

En la Figura 20 se representa el modelo de integración continua y entrega continua de Red Hat compuesto por tres fases principales: integración, entrega y despliegues.

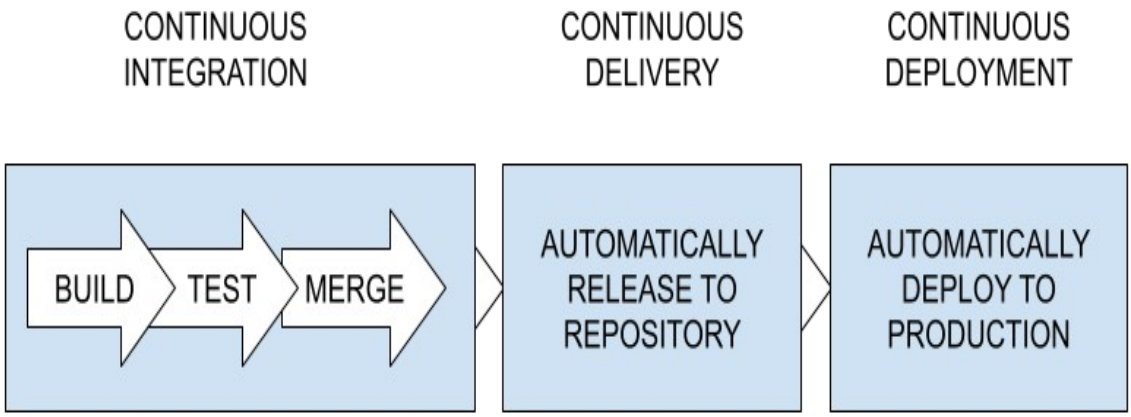

Figura 20: CI/CD modelo de integración continua [Elaboración propia] [48]

2.4.4. Help Desk y Service Desk

Cuando el producto software llega a ser utilizado por los usuarios finales es recomendable establecer un soporte para usuarios que usualmente se organiza en dos niveles:

1. Help Desk: soporte de primer nivel.
2. Service Desk: Soporte de segundo nivel.

Help Desk significa literalmente Mesa de Ayuda. Es allí que se realiza el primer contacto entre cliente y la organización. Los clientes acceden a este primer nivel de atención con dudas o problemas acerca de productos o servicios adquiridos, la atención que el técnico de Help Desk brinda en estos momentos es fundamental para captar, mantener y fidelizar clientes.

Dentro de su desempeño profesional, los técnicos de Help Desk conocen en detalle el producto o servicio que la organización ofrece, y tienen capacidad para resolver problemas comunes o frecuentes entre los usuarios.

Si no consiguen brindar una respuesta, su tarea es elevar la solicitud a un segundo nivel de atención, el Service Desk.

El Service Desk es el equipo técnico capacitado para dar respuestas acerca de la lógica de negocio, estado del servicio y eventualmente efectuar correcciones en el producto de software.

CAPÍTULO III

Adaptación de Design Thinking

3. Propuesta de adaptación de Design Thinking para el desarrollo de software educativo

Este capítulo desarrolla la propuesta de adaptación e integración de Design Thinking para el desarrollo de entornos educativos interactivos en educación superior.

Inicialmente se realiza la propuesta de adaptación, se definen las etapas involucradas junto con una descripción de las entradas, salidas, artefactos, equipos y roles.

Se describen las consideraciones en roles, recursos, elementos propios de Design Thinking en conjunto con prácticas ágiles y procesos de la ingeniería del software.

Se propone el uso de "The value proposition canvas" para empatizar con los usuarios, capturar los requisitos funcionales y conseguir prototipar. Finalmente, se propone una fase para prototipar y validar la definición de producto software resultante.

3.1. Design Thinking en desarrollo de producto software

En la producción de software, la aplicación de Design Thinking puede representar un eslabón importante en el desarrollo de productos de software. En este contexto, su objetivo final es crear una especificación de producto, que se representa mediante un conjunto de artefactos. Esos artefactos pueden incluir un PRD por sus siglas en inglés Product Requirement Document, un prototipo, y artefactos de principios ágiles como ser roadmap, un conjunto de épicas e historias de usuario del producto software que se pretende construir.

La especificación del producto es importante tanto en productos nuevos, que aún no han sido desarrollados, como así también en productos existentes.

En productos nuevos, la aplicación de Design Thinking tiene como objetivo especificar un Producto Mínimo Viable, también llamado MVP por sus siglas en inglés Minimum Viable Product, mientras que para los productos de software existentes, su objetivo podrá ser incorporar nuevas funcionalidades y experiencias de usuario.

El proceso Design Thinking se puede añadir a la producción de software, incluso, si la producción de software se fundamenta en principios ágiles.

En la Figura 21 se integra el proceso de Design Thinking dentro de los diferentes eslabones que componen la producción de software con principios ágiles.

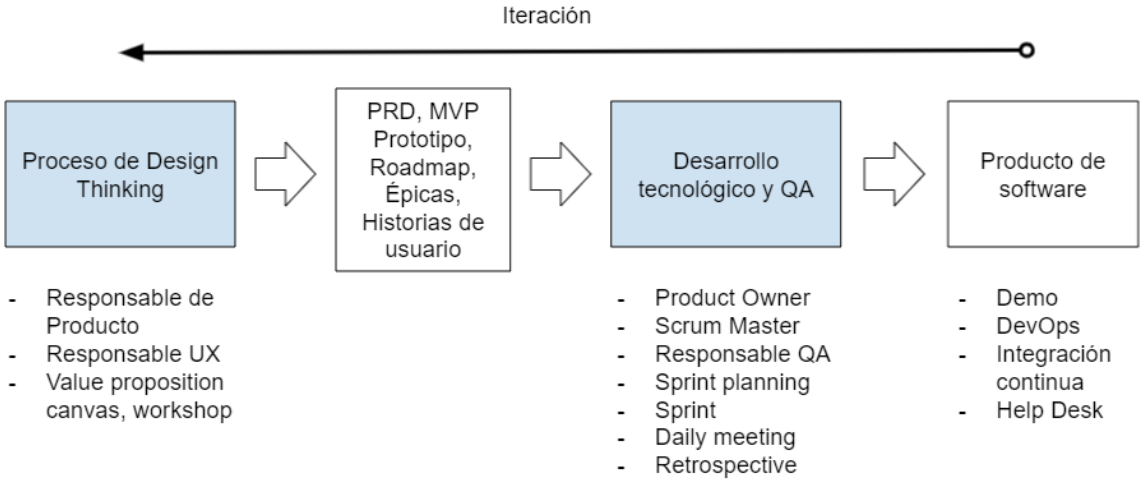

Figura 21: Propuesta de Design Thinking como proceso integrado en el desarrollo de producto software [Elaboración propia]

Los equipos de desarrollo de software se sienten cómodos utilizando principios ágiles.

El Design Thinking se vincula con principios ágiles mediante la definición del MVP, ya que la misma permite identificar un conjunto de artefactos de utilidad para comenzar con el desarrollo del producto software en un escenario de menor incertidumbre.

Los artefactos generados en la aplicación de Design Thinking pueden ser un PRD MVP, prototipo, roadmap, épicas e historias de usuario.

En la etapa de desarrollo de software los actores principales son el propio equipo tecnológico junto al equipo de calidad. En esta fase se realiza la programación e integración del producto de software propiamente dicho, dependiendo de la naturaleza del producto podrá requerir desarrollo backend, frontend, aplicaciones nativas, aplicaciones de escritorio, bases de datos, almacenamiento de archivos, entre otros.

A continuación se describen las fases, roles y artefactos que componen la propuesta de adaptación de Design Thinking para la producción de un producto software.

3.1.1. Proceso de Design Thinking

El proceso de Design Thinking es el primer eslabón en la propuesta, es recomendable que esté conformado por un equipo multidisciplinario conformado por técnicos, diseñadores , responsable de producto y lo más importante: los usuarios finales.

Los roles y artefactos se detallan a continuación.

3.1.1.1. Responsable de producto

El responsable de producto es considerado el dueño del producto software, también llamado PO por sus siglas en inglés Product Owner, está realmente inmerso y comprende el problema que el producto de software resuelve. En determinadas situaciones tiene que tomar decisiones importantes. Es el principal interesado en que el producto consiga tener éxito, consiga lograr su misión.

3.1.1.2. Responsable de UX

Trabaja de manera estrecha con el equipo de producto, se encarga de diseñar las experiencias de usuario del producto de software generando wireframes, storyboards e interfaces de usuario interactivas.

3.1.1.3. Equipo multidisciplinar

El equipo multidisciplinar son las personas que realizan los talleres para empatizar, descubrir e idear soluciones en Design Thinking.

Es recomendable que esté conformado por diversos perfiles que estén dispuestos a aportar conocimiento acerca de características requeridas en el producto de software, las problemáticas a resolver, el panorama del estado del arte, historias de usuario, y consideraciones de interés.

3.1.1.4. PRD

Un PRD es un documento que contiene todos los requerimientos de un producto, está escrito de manera tal que las personas puedan comprender lo que debe hacer un producto de software.

MVP

Un MVP es un conjunto de funcionalidades mínimas que debe contener un software para considerarse producto software. Es lo mínimo de funcionalidades que debe tener el producto de software para resolver la problemática y cumplir con su misión.

3.1.1.5. Prototipo

Normalmente definen las interfaces de usuarios y funcionalidades del producto de software, pueden ser de baja o alta fidelidad. Si fueran de baja fidelidad pueden estar diseñados directamente en papel o bocetos digitales simples, si se producen en calidad media o alta serán interfaces digitales navegables, en una versión demostración, pudiendo incluir las interacciones que realice el usuario con los elementos de la interfaz de usuario, representando así de manera más fiel, la experiencia que tendrá el usuario final cuando utilice el producto software.

3.1.1.6. Roadmap

El Roadmap, conocido en español como hoja de ruta, representa un cronograma para la planificación del proyecto. Se trata de un verdadero manifiesto ágil, un documento directivo que enumera la lista de tareas a ejecutar en el marco del proyecto según un principio de iteración e integración continua.

3.1.1.7. Épicas

Las épicas son un conjunto de trabajo grande que puede dividirse en tareas específicas (denominadas “historias de usuario”) en función de las necesidades o solicitudes de los clientes o usuarios finales. Las épicas son una práctica importante para los equipos ágiles que trabajen con integración continua.

3.1.1.8. Historias de usuario

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al usuario.

3.1.2. Desarrollo tecnológico ágil

El desarrollo tecnológico es la etapa de programación del producto software

utilizando principios ágiles, en caso de que el equipo de programación disponga de varios perfiles técnicos se crearán equipos según su especialidad.

3.1.2.1. Equipo Tech

El equipo tech son los profesionales informáticos que codifican el software. Normalmente para integrar producto de software se requiere trabajar en diferentes áreas, a saber:

- Construcción de interfaces de usuario
- Programación de frontend
- Programación de backend y microservicios
- Administración de base de datos relacionales y no relacionales
- Programación de aplicaciones nativas
- Programación de aplicaciones de escritorio

Dependiendo del tamaño, tiempos y recursos disponibles para integrar el producto software se podrán asignar varios recursos técnicos a un área, o un solo recurso para varias áreas, esto dependerá de las características del producto software y la envergadura de la organización o empresa.

3.1.2.2. Responsable QA

El responsable de QA se encarga de validar el funcionamiento de cada épica e historia de usuario que el Equipo Tech implementa. Habitualmente se testean y validan dos tipos de tareas:

1. Features, nuevas funcionalidades que no existían previamente en el producto de software, normalmente vinculado a épicas e historias de usuario.
2. Hotfixes, solución a problemas detectados en el producto de software existente, los errores, llamados “bugs” pueden ser encontrados en producción o en entornos de preproducción como suelen ser entorno de desarrollo o entorno QA.

3.1.2.3. Scrum Master

El Scrum Master es la persona designada para llevar a cabo las reuniones de planificación, actualizaciones diarias y retrospectivas.

3.1.2.4. Sprint Planning

El Sprint Planning es una reunión para planificar y proyectar el Sprint. Al Sprint Planning estarán invitados el Product Owner, Scrum Master, Responsable de QA y los integrantes del equipo tecnológico. La duración aproximada es de una hora.

En la planificación se indica la fecha de inicio y fin del Sprint, el objetivo y las historias de usuario con las cuales se pondrá cada integrante del equipo tecnológico. La sesión de planificación se inicia definiendo el orden del día y el punto de atención, se intenta crear un entorno ameno para que el equipo esté motivado.

Al analizar las historias de usuarios es recomendable realizar una estimación de esfuerzo para planificar la carga de trabajo asumible.

3.1.2.5. Sprint

En Scrum, el sprint es una iteración, un periodo definido de tiempo en el que se intenta realizar una cantidad de trabajo establecida en el Sprint Planning.

El Sprint es el corazón de la metodología Scrum que aplica principios ágiles. La duración del Sprint ronda de una a cuatro semanas.

3.1.2.6. Daily Meeting

El objetivo de la Daily Meeting es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar la productividad, al poner de manifiesto puntos en que se pueden ayudar unos a otros.

Cada miembro del equipo inspecciona el trabajo que el resto está realizando, dependencias entre tareas, progreso hacia el objetivo del Sprint, obstáculos que pueden impedir este objetivo.

La duración del Daily Meeting es de 15 minutos. Está dirigida por el Scrum Master.

Cada miembro del equipo debe responder las siguientes preguntas.

- ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
- ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
- ¿Qué impedimentos tengo o voy a tener que puedan transformarse en obstáculos para conseguir nuestro objetivo?

3.1.2.7. Retrospective Meeting

La Retrospective Meeting se lleva a cabo al finalizar el Sprint. La reunión trata sobre la reflexión de hechos pasados para aplicar mejoras en el futuro. La duración aproximada es de una hora. Se identifica lo que fue bien, lo que se puede mejorar y acciones concretas a realizar para mejorar en el futuro inmediato.

3.1.3. Producto software

El producto software es el objetivo final buscado. Es lo que genera valor en la organización.

Por más que estemos hablando de software, que es algo intangible por naturaleza, y además estamos trabajando en las primeras iteraciones de desarrollo, siempre se recomienda tener una versión entregable al final de un Sprint, es algo “tangible” que puede ser utilizado por un usuario o interesado. Por este motivo, normalmente se organiza una reunión de demostración al finalizar el Sprint.

3.1.3.1. Demos

Las demos, son demostraciones del producto de software que se está construyendo. Las demos suelen ser reuniones programadas con anticipación, con fecha posterior o igual a la finalización del Sprint.

En la reunión de demostración se enseña el producto software, por ejemplo, a través de pantalla compartida, se realiza el seguimiento de las historias de usuario en las que se ha trabajado.

Si el producto de software abarca gran cantidad de lógica de negocio, que se traduce a muchas historias de usuario, en las demos se intentará ver las historias de usuario más relevantes del Sprint, en caso contrario, si el producto de software es acotado o un MVP, la demostración será completa, incluyendo todas las historias de usuario.

3.1.3.2. Desarrollo y operaciones

El equipo encargado de desarrollo y operaciones, o simplemente DevOps, siempre intentará garantizar la integración continua / entrega continua. Se busca garantizar que el equipo tecnológico pueda integrar de manera continua, teniendo siempre la posibilidad de realizar entregas en todos los entornos establecidos.

Es recomendable definir un conjunto de entornos, por ejemplo, entorno de desarrollo, entorno de calidad, entorno de producción. Dependerá del producto de software, sus características y su presupuesto la cantidad de entornos de trabajo necesarios. El mínimo suelen ser dos entornos, entorno de desarrollo y entorno de producción.

En lo que refiere al código fuente y el control de versiones, es recomendable definir flujos estándar de trabajo, es importante que todo el equipo técnico lo conozca y adopte.

En la Figura 22 se representa Gitflow [48]. Gitflow es un ejemplo de flujo de trabajo para el control de versiones del código fuente del producto software, donde se destaca la diferenciación entre feature y hotfix, es importante destacar la diferencias. La feature, puede iniciarse como una rama del código que se encuentra en desarrollo, sin embargo, un hotfix, debe ser una rama del código publicado en ambiente producción, esto evitará que cuando se publique el hotfix, llegue a producción código que aún está en desarrollo.

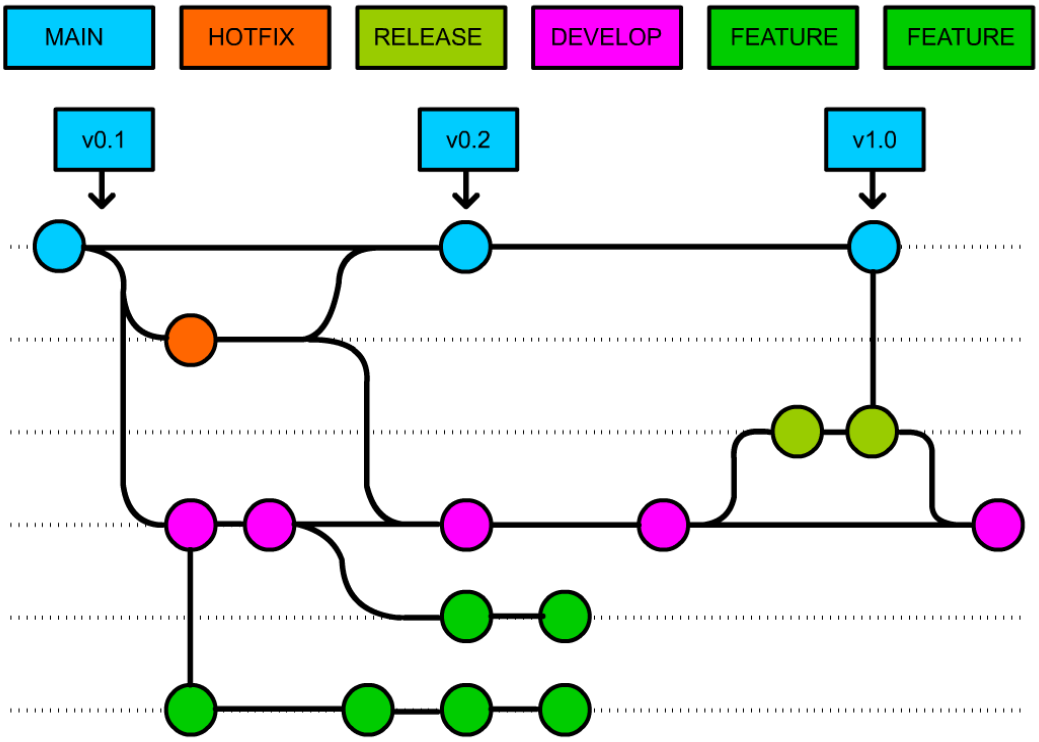

Figura 22: Flujo de trabajo Gitflow [Elaboración propia] [49]

3.1.3.3. Help desk

El equipo de Help Desk, también llamado servicio de primer nivel, es el encargado de proporcionar respuestas a las dudas que tengan los usuarios sobre la utilización del producto software que no se encuentren respondidas en las preguntas frecuentes.

En caso de que se identifique un error en la plataforma de software, se deberá informar al Service Desk a través de una herramienta de gestión de incidencias para darle seguimiento y notificar al usuario final cuando la incidencia se encuentre resuelta.

Los encargados de prestar servicio de Help Desk deben poseer conocimientos sólidos del producto software, conocer las políticas de la organización, disponer de capacidades comunicacionales idóneas tales como saber escuchar y comprender la información.

Para los casos donde exista un SLA por sus siglas en inglés Service Level Agreement, podrán existir cláusulas de contrato a cumplir.

El SLA es un acuerdo escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio. Es una herramienta que ayuda a ambas partes a llegar a un consenso en términos del nivel de calidad del servicio, en aspectos tales como tiempo máximo de respuesta, disponibilidad horaria, documentación disponible y personal asignado al servicio.

3.1.3.4. Service Desk

El Service Desk es el servicio de segundo nivel. El soporte de primer nivel es el primero en entrar en contacto con las incidencias reportadas por los usuarios del producto software.

En caso de que la incidencia lo requiera, es decir, que el soporte de primer nivel no pueda resolver por sí mismo, será escalado al soporte de segundo nivel, donde el equipo tecnológico de desarrollo de software y/o infraestructura se hará cargo.

El proceso del Service Desk puede ser el siguiente:

- El usuario notifica su problema a través de uno de los canales de acceso al Help Desk como ser un número telefónico, website, portal, servicio de email o chat online.
- El soporte de primer nivel genera un ticket que contiene los detalles del problema.
- El soporte de primer nivel analiza el problema y lo intenta resolver. Si es así, el ticket se cierra y se actualiza con la documentación de la solución. Esto permite que otros técnicos de servicio tengan una referencia.
- Si el problema necesita ser escalado será derivado al segundo nivel.
- Finalmente, el segundo nivel analizará y resolverá el problema cerrando el ticket correspondiente actualizando la documentación de la solución para permitir a otros técnicos de servicio tener una referencia.

3.2. Etapas del proceso Design Thinking

En la presente propuesta de adaptación de Design Thinking, como resultado de analizar los diferentes marcos de trabajo Design Thinking podemos llegar a la conclusión

de que existen tres etapas fundamentales que son Empatizar, Prototipar y Validar, representadas secuencialmente en la Figura 23.

La ejecución de las etapas se puede realizar de manera iterativa cuando el producto software a desarrollar es extenso, es decir, intervienen varios roles de usuario o bien, está compuesto por un gran número de funcionalidades.

Realizar Design Thinking de manera iterativa posibilita a su vez enfocarse en una problemática específica a la vez, por ejemplo, se pueden organizar procesos de Design Thinking para el rol docente, alumno, o para la funcionalidad proyectar en clase, asignación y corrección de asignaciones, entre otros.

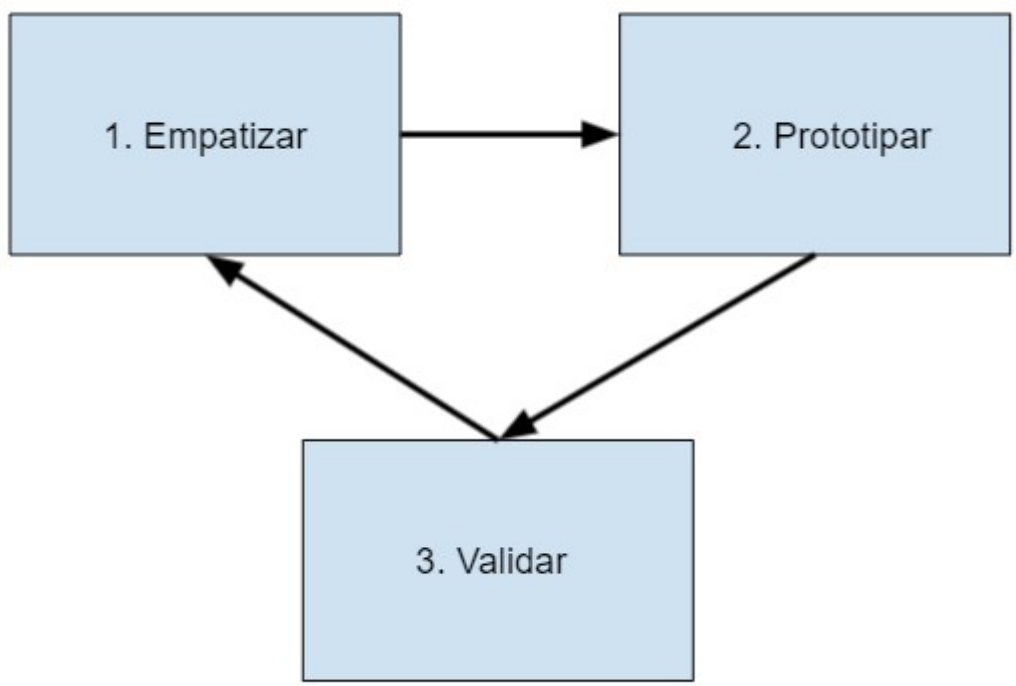

Figura 23: Etapas propuestas en el proceso Design Thinking [Elaboración propia]

En cada una de las etapas del proceso Design Thinking se deben tener en cuenta los objetivos finales que se persiguen, en este caso de estudio, desarrollar una plataforma educativa interactiva, que posibilite el aprendizaje colaborativo mediante la utilización de diversos dispositivos, en un contexto de educación superior.

Teniendo en cuenta los objetivos buscados podemos definir requisitos indispensables, como por ejemplo:

- La plataforma educativa deberá ser accesible desde cualquier dispositivo con Internet de uso habitual por los usuarios del sistema, es decir, docentes y estudiantes. Contemplamos aquí a desktops, notebooks, tablets, smartphones. Las interfaces de usuario se deberán ajustar a distintas resoluciones de pantalla. La tecnología utilizada en las aplicaciones cliente deberá ejecutarse en todos los dispositivos mencionados de manera fluida teniendo en cuenta que los usuarios podrán disponer de una conexión a Internet inestable o lenta (3G, 4G, banda ancha).
- Los usuarios tendrán experiencias interactivas, los docentes podrán ir definiendo materiales de estudio, tareas, los estudiantes podrán ir entregando las tareas, será necesario definir un soporte tecnológico que posibilite las notificaciones síncronas y asincrónicas en tiempo real.
- El enfoque colaborativo deberá permitir a los docentes proponer dinámicas de trabajo que requieran interacción social y ayuda mutua entre los estudiantes.

Cada una de las etapas de Design Thinking tiene un propósito, de manera resumida es el siguiente:

- **Empatizar**: conocer la problemática a resolver, conocer a los usuarios finales, detectar oportunidades de mejoras.
- **Prototipar**: buscar y construir alternativas de solución basadas en prototipos respetando las especificaciones del proyecto.
- **Validar**: presentar los prototipos a los usuarios finales e interesados para determinar si resuelven la problemática inicialmente planteada.

Una vez que conocemos el propósito de cada etapa necesitamos determinar las actividades y dinámicas que tienen cada una de ellas.

3.2.1. Etapa I - Empatizar

Para empatizar con los usuarios y descubrir los requisitos funcionales se propone utilizar la herramienta "The value proposition canvas", que permitirá conocer a nuestros usuarios mediante el desarrollo de un Workshop, que podrá realizarse de manera presencial o virtual, los integrantes serán el equipo multidisciplinar y un líder encargado de coordinar la actividad.

La herramienta "The value proposition canvas" fue desarrollada inicialmente por el Dr. Alexander Osterwalder [50] como un marco para garantizar que haya un ajuste o acuerdo entre el producto y el mercado.

Inicialmente el lienzo de la propuesta de valor se utilizaba para perfeccionar una oferta de producto o servicio.

La Figura 24 es el lienzo de trabajo para trabajar la propuesta de valor, cada una de las partes que lo conforman serán analizados a continuación.

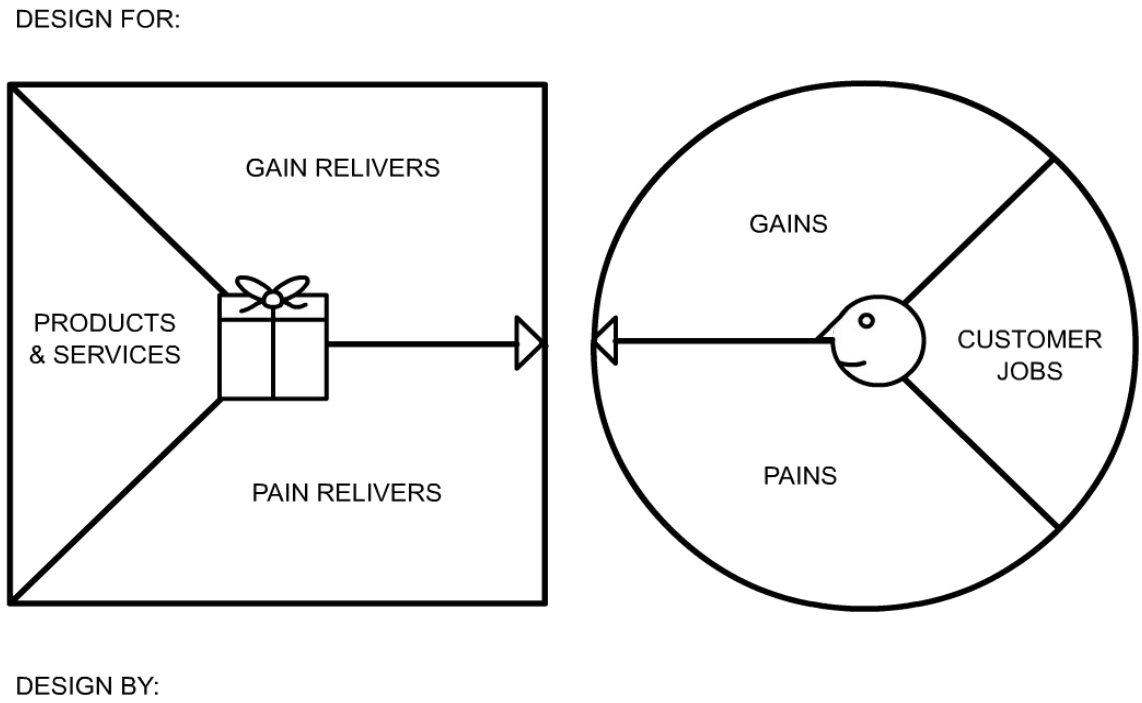

Figura 24: Etapas The value proposition canvas [Elaboración propia] [50]

3.2.1.1. Conocer a los usuarios

Para lograr que un producto de software cumpla con su misión es importante conocer a los usuarios finales, sus hábitos, los problemas y beneficios que esperan obtener al utilizar la plataforma educativa.

El conocimiento de los usuarios se propone mediante la identificación de las actividades que los usuarios realizan habitualmente en su entorno educativo.

La herramienta "The value proposition canvas" propone identificar los siguientes elementos de cada rol de usuario.

3.2.1.1.1. Customer Jobs, actividades habituales o diarias

Son las actividades relacionadas con el producto de software que deben llevar a cabo los usuarios finales de manera habitual. Por ejemplo, un docente podría requerir: crear una tarea, corregir exámenes, organizar clases, etc. un estudiante en cambio podría requerir: revisar mis tareas pendientes, ver mi libro de calificaciones, comunicarse con el docente, etcétera.

Los Customers Jobs en la dinámica "The value proposition canvas" se escriben en "post its" y se pegan en el área indicada dentro del canvas, en un proceso posterior se podrán transformar en historias de usuarios que se integrarán en el producto de software.

Un conjunto de historias de usuario puede conformar una épica, por consiguiente, la épica representará un objetivo mayor.

En la Figura 25, se resalta el área asignada a los Customer Jobs dentro de "The value proposition canvas".

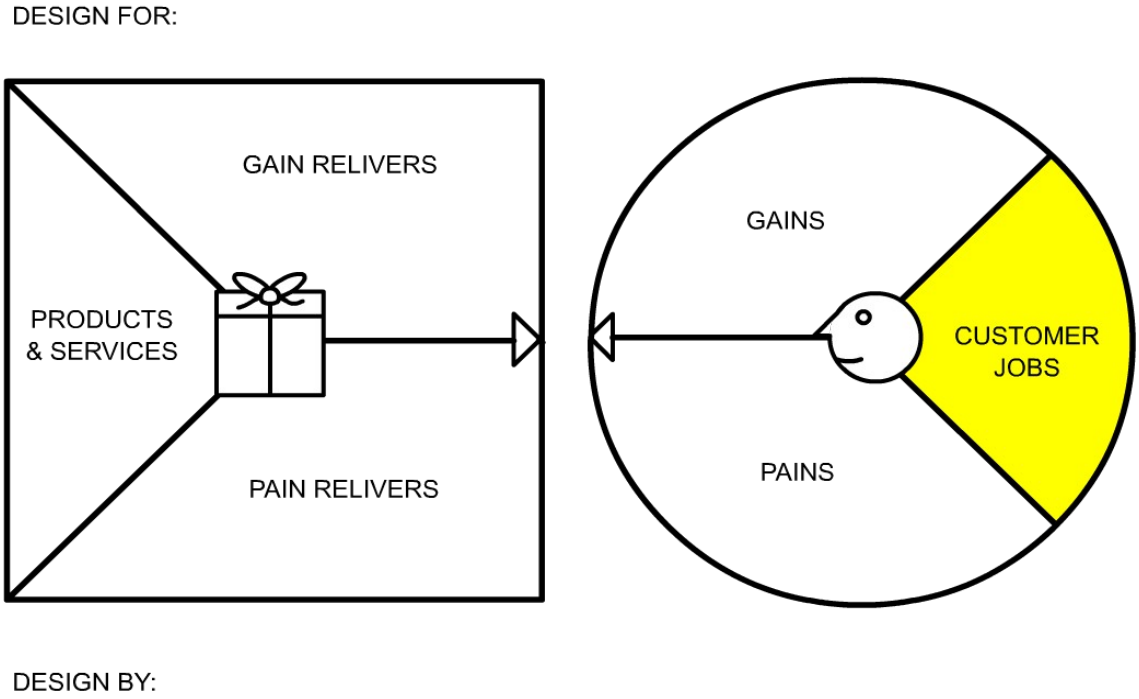

Figura 25: The value proposition canvas – Gains [Elaboración propia] [50]

3.2.1.1.2. **Pains,** dificultades

Los Pains son las situaciones o costes no deseados que experimentan los usuarios al utilizar un producto de software semejante. Los Pains describen cualquier cosa que moleste a los usuarios antes, durante o luego de realizar los Customer Jobs, también los Pains podrían ser situaciones que imposibiliten que las tareas se lleven a cabo.

La severidad de los Pains identificados por los usuarios podrá ser catalogada como extrema o moderada.

A continuación, en la Tabla 7, algunos ejemplos de dificultades que se pueden encontrar usuarios de un producto de software educativo.

Tabla 7: The value proposition canvas - Pains

Pains (dificultades)	**Rol**
Los estudiantes presentan dificultades para conocer las tareas que añade el docente a la plataforma educativa, el sistema de notificaciones de las plataformas educativas tiene pocas opciones de notificación. Es habitual que un estudiante desconozca la fecha de entrega de las tareas que debe realizar.	Estudiante
Los estudiantes cuyos docentes no utilizan la plataforma educativa quedan privados de acceder a los contenidos digitales propuestos por las editoriales.	Estudiante
A los estudiantes les molesta mucho cuando la plataforma educativa pierde la entrega de su tarea por un error de sistema, o por un pico de sobrecarga, por problemas de seguridad, conectividad, etcétera.	Estudiante
Los docentes, en tiempos de pandemia, echan en falta herramientas de videoconferencia integradas en las propias plataformas. Es habitual que tengan que utilizar herramientas externas.	Profesor
A los docentes les lleva mucho esfuerzo corregir las entregas de los estudiantes porque deben revisarlas una por una.	Docentes
A los docentes que llevan muchas asignaturas les llega innumerable cantidad de notificaciones.	Docentes
Los administradores de centros educativos consideran que el costo de disponer de una plataforma educativa es alto y no siempre está justificado el desembolso monetario.	Administradores de centros educativos
A los administradores de centros educativos les lleva mucho tiempo el proceso de SETUP, esto es, añadir todos los usuarios a la plataforma, configurar el ciclo lectivo, añadir los cursos, asignaturas, etcétera.	Administradores de centros educativos
A los usuarios les cuesta invertir tiempo en el aprendizaje sobre el uso de plataformas educativas, en ocasiones las interfaces de usuario son complejas.	Todos
Los usuarios perciben problemas de rendimiento, la plataforma educativa puede quedar fuera de servicio en horarios críticos del día o momentos del año, por ejemplo, antes de cerrar los semestres.	Todos

En la Figura 26, se resalta el área asignada a los Pains del usuario dentro de “The value proposition canvas”.

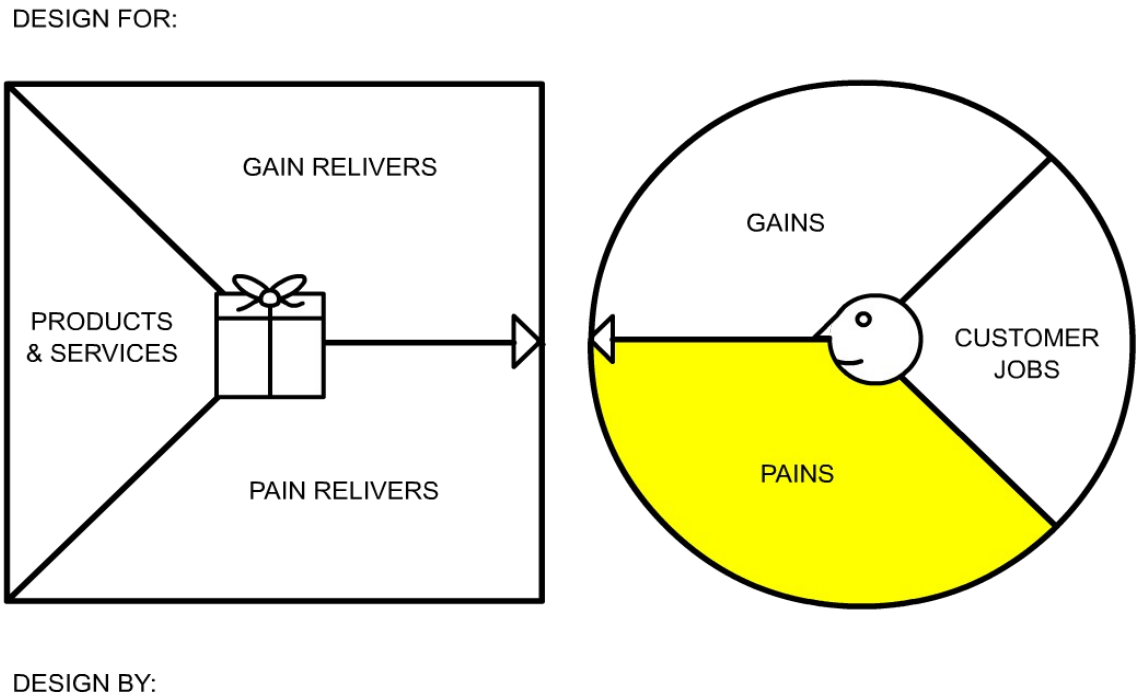

Figura 26: The value proposition canvas – Pains [Elaboración propia] [50]

3.2.1.1.3. **Gains,** beneficios

Son los beneficios que esperan obtener los usuarios al utilizar el producto de software. Los Gains describen los resultados y beneficios que desean obtener los usuarios.

Los Gains se pueden catalogar según su nivel de importancia entre requerido, esperado o deseado.

Los Gains incluyen las funcionalidades útiles, ganancias sociales, emociones positivas y ahorro de costes.

De manera general, la relevancia de los Gains se determina por sí los usuarios lo

consideran esencial o deseable de tener.

En la Tabla 8 se identifican, a modo de ejemplo, beneficios que espera tener el usuario del producto o servicio.

Tabla 8: The value proposition canvas - Gains

Gains (beneficios)	**Rol**
Como estudiante espero poder repasar los contenidos de manera autónoma y entregar las tareas cuando me sea posible.	Estudiante
Como estudiante necesito recibir las notificaciones para realizar las tareas y entregarlas a tiempo.	Estudiante
Como docente me haría feliz ver el progreso global de cada curso, pudiendo a su vez ver el progreso de cada lección y contenido que lo compone.	Docente
Como docente necesito revisar los trabajos de los estudiantes de manera ágil, debería ser trivial corregir trabajos entregados digitalmente.	Docente
Como docente me gustaría poder añadir nuevas evaluaciones y contenidos en el curso propuesto por la editorial. Sería interesante poder añadir contenidos SCORM y xAPI de terceros, pudiendo ver el desempeño de los estudiantes dentro de los contenidos educativos importados.	Docentes
Como administrador de centro educativo necesito realizar un SETUP sencillo, que permita la carga masiva de usuarios desde planillas de cálculo.	Administrador de centro educativo
Como administrador de centro educativo me gustaría que la plataforma educativa pueda reportar la actividad de los usuarios	Administrador de centro educativo

a un LRS, por sus siglas en inglés Learning Record System, centralizado mediante el estándar xApi.	
Como administrador de centro educativo me gustaría que el SETUP se realice de manera automática, mediante el estándar ONEROSTER.	Administrador de centro educativo
Como centro educativo me gustaría pagar únicamente por los usuarios que realmente utilizan la plataforma, por ejemplo, usuarios que han iniciado sesión al menos dos veces en el período académico.	Centro educativo

En la Figura 27, se resalta el área asignada a los Gains que espera el usuario dentro de “The value proposition canvas”.

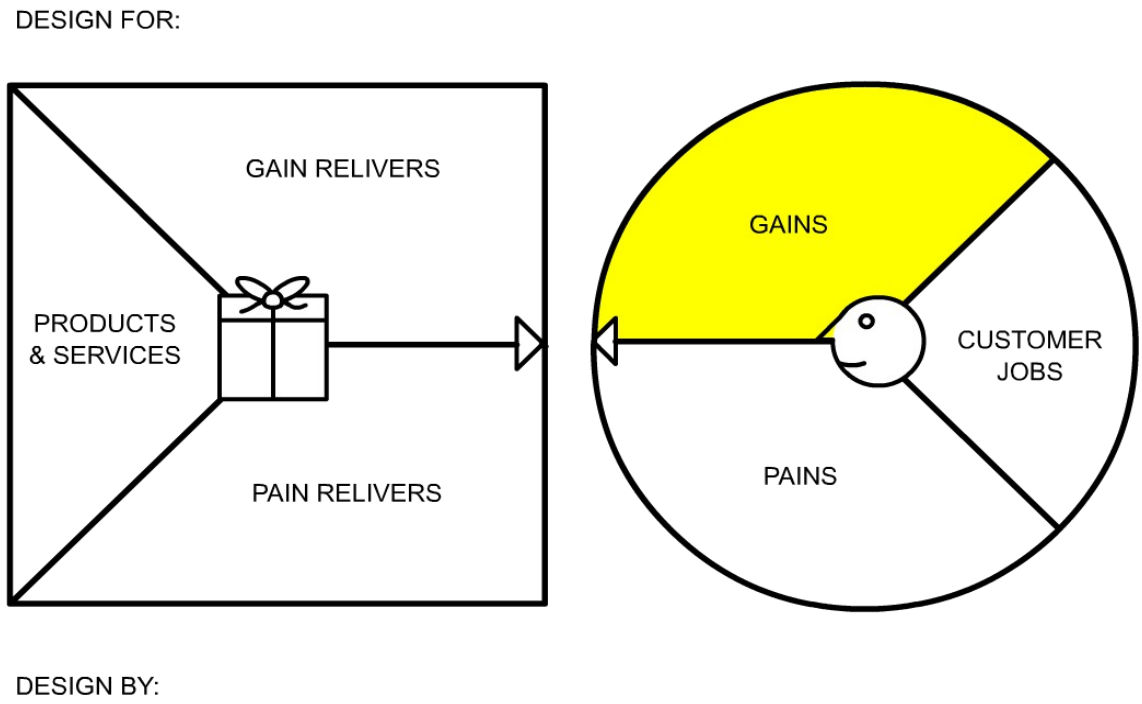

Figura 27: The value proposition canvas – Gains [Elaboración propia] [50]

3.2.1.2. Definición de la propuesta de producto software

Una vez que se empatiza con el usuario, conocemos las actividades que debe realizar, sus dificultades y expectativas, es decir, lo que engloba la parte derecha del lienzo de propuesta de valor, estamos en condiciones de analizar la propuesta del producto software.

El producto software está representado en forma de cuadrado dentro de la propuesta de valor, contiene en su interior tres categorías que se trabajarán.

Las categorías a trabajar dentro del contexto de propuesta de producto software son Products and services, Gains creators y Pain relievers, a continuación se describen cada una de ellas.

3.2.1.2.1. Products and services

Son los servicios y productos que se ofrecen a los usuarios para ayudarlos con las actividades que deben realizar. La identificación de servicios y productos es la contraparte de Customer Jobs. En la Figura 28 se resalta la propuesta de producto y servicios.

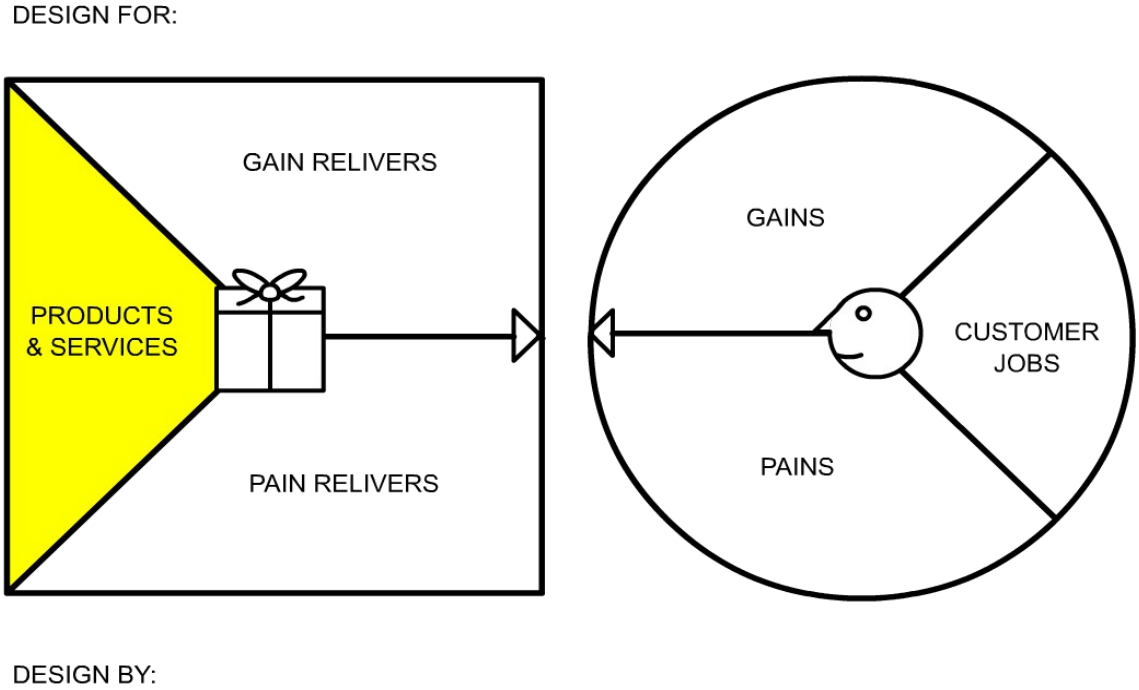

Figura 28: The value proposition canvas - Products & services [Elaboración propia] [50]

3.2.1.2.2. Pain relievers

Se trata de determinar la manera en que se resuelven los problemas o necesidades de los usuarios del producto de software. La identificación de los Pains relivers la contraparte de los Pains, en la Figura 29 se resalta el áre de los Pains dentro del lienzo de la propuesta de valor.

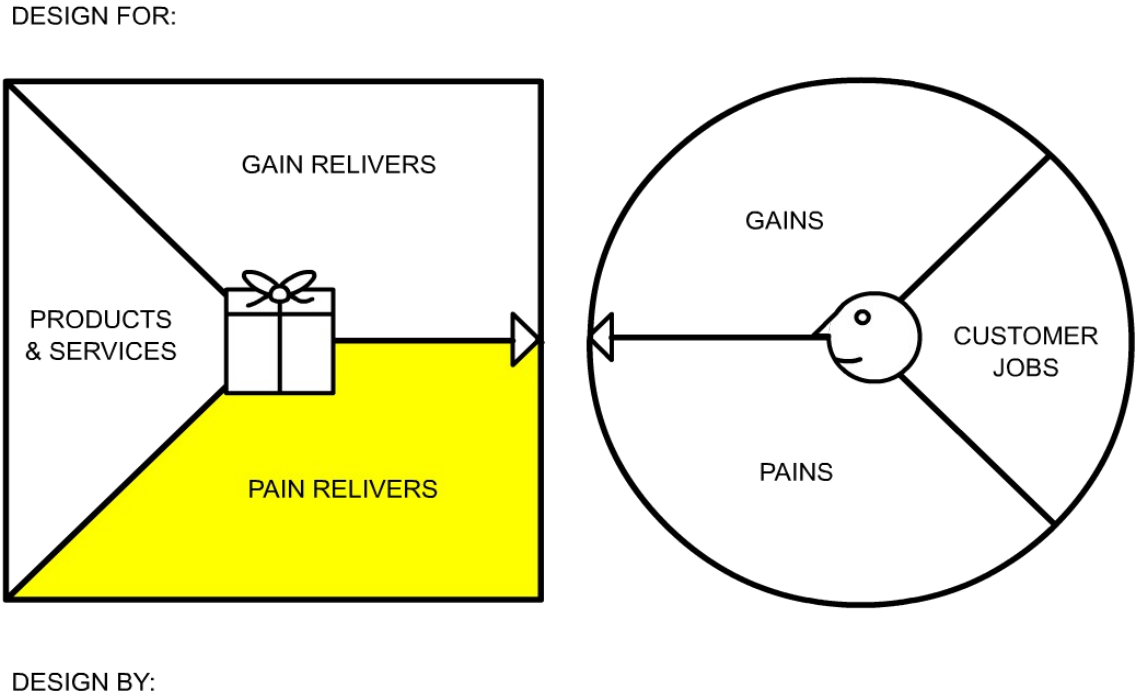

Figura 29: The value proposition canvas - Pains relivers [Elaboración propia] [50]

3.2.1.2.3. Gain creators

La identificación de Gains creators se traduce a cómo se aportan beneficios a los usuarios en base a las expectativas que los usuarios tienen.

La identificación de Gains creators es la contraparte de los Gains que los usuarios esperan del producto software. El área de Gains dentro del lienzo está resaltada en la Figura 30.

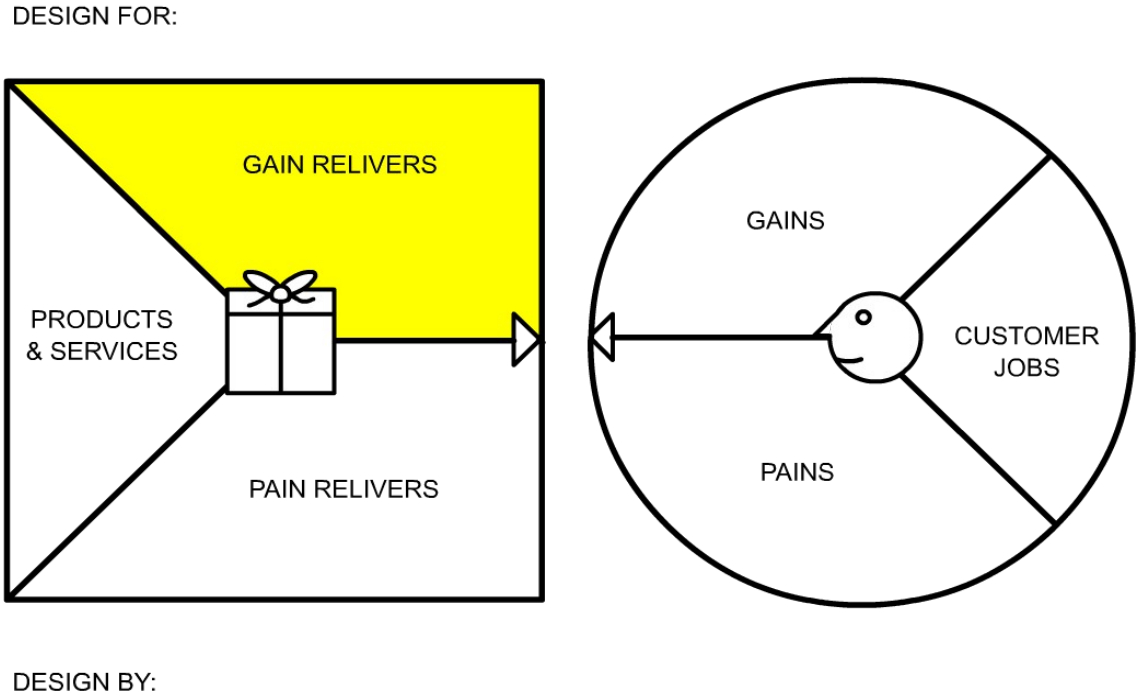

Figura 30: The value proposition canvas - Gains creators [Elaboración propia] [50]

Hay que considerar que en esta fase estamos manejando hipótesis que tendremos que validar posteriormente con los usuarios finales a través de prototipos de producto software.

3.2.1.3. Roles

Para que el taller de The value proposition canvas se lleve a cabo es importante prever los roles de usuario y funciones que deben cumplir.

El **líder** es el encargado de dirigir el taller, debe establecer los tiempos y trabajos a realizar en cada etapa. Es importante que disponga de habilidades de comunicación para dirigir el taller con celeridad. Debe conseguir que el equipo de trabajo esté enfocado y motivado.

Miembro del equipo de trabajo, pueden ser usuarios finales del producto de software, docentes, directivos, alumnos, entre otros. El rol del miembro del equipo de

trabajo será aportar criterio pudiendo identificar Customer Jobs, Pains, Gains, Products & Services, Gains creators y Pains relievers. Además de identificar, podrá exponer sus aportes al equipo de trabajo y ponderar la importancia de otros elementos que se identifiquen en la propuesta de valor.

3.2.1.4. Dinámica

El líder inicia el taller explicando en qué consiste The value proposition canvas y cuál es el producto software objetivo. La propuesta de valor es un lienzo al que todos los integrantes del equipo deben tener acceso para poder añadir los Post its y ponderaciones.

En cada ventana de tiempo los integrantes del equipo definirán los elementos para el área de la propuesta de valor que se esté trabajando, cada miembro del equipo escribirá en Post its los Customer Jobs, Pains, Gains, Products & Services, Gains creators y Pains relievers que considere relevantes.

Completado el tiempo asignado, cada miembro del equipo pasará a explicar los elementos que logró identificar a los demás integrantes del taller, seguidamente, añadirá los Post its a la propuesta de valor.

Completada la fase de exposición, el líder deberá agrupar los Post its que se repitan y dará paso a la votación.

En la votación cada integrante dispondrá de una cantidad determinada de votos, por ejemplo, tres votos por cada categoría.

Finalizada la votación se podrán identificar cuáles son los elementos identificados más relevantes en cada categoría.

3.2.2. Etapa II - Prototipar

Para conseguir materializar las especificaciones de requerimientos capturadas en los procesos de empatía se puede recurrir al prototipado, que a su vez, se divide según su naturaleza en prototipos de baja y alta fidelidad.

Es recomendable que no pase mucho tiempo entre el taller de The value

proposition canvas y la fase de prototipado.

3.2.2.1. Prototipos de baja fidelidad

Los prototipos de baja fidelidad son los que no tienen el aspecto final de la interfaz de usuario del producto de software, aunque pueden contemplar las funcionalidades finales del producto de software.

Los prototipos de baja fidelidad deben ser sencillos de integrar, pueden realizarse con bocetos a mano alzada, lo importante es que se produzcan sin mucho esfuerzo.

En la Figura 31 se ejemplifica un prototipo de baja fidelidad dibujado en forma de bocetos sobre una pizarra blanca.

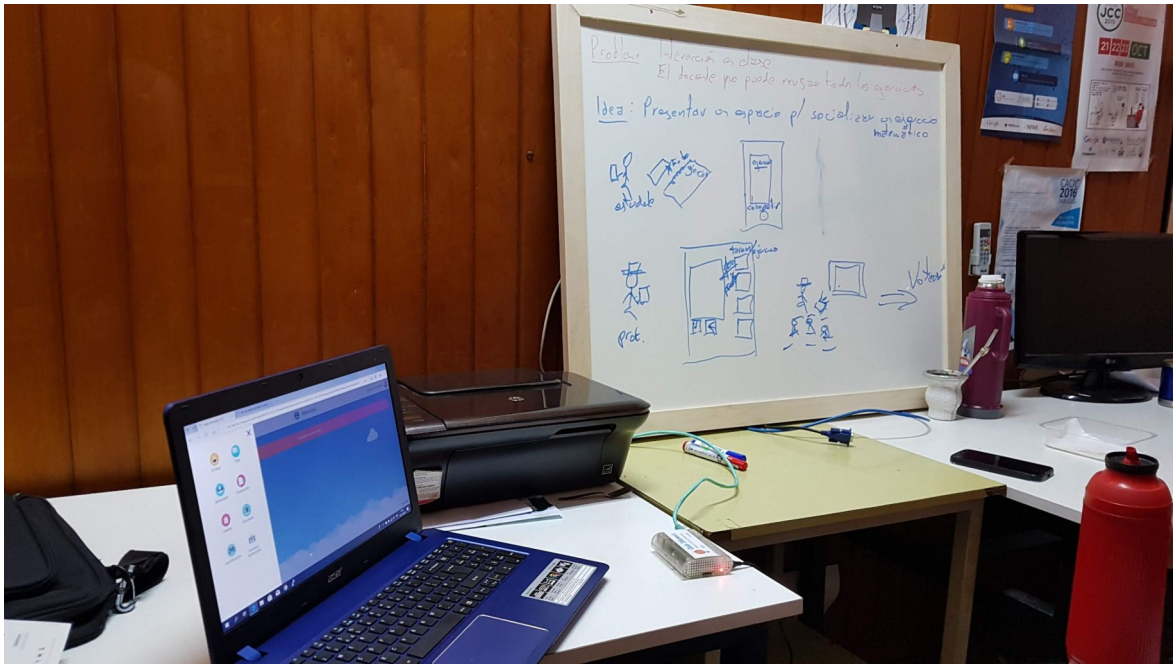

Figura 31: Diseño de prototipos en baja fidelidad [Elaboración propia]

3.2.2.2. Prototipos en alta fidelidad

En los prototipos de alta fidelidad los diseños intentarán ser lo más realistas posibles, esto puede incluir interfaces navegables, elementos interactivos, simulación de entrada de mouse y teclado, en resumen, se busca que el prototipo sea lo más real posible.

Algunas de las herramientas de software utilizadas para crear prototipos de alta

fidelidad son las siguientes.

- Figma [51]
- Sketch [52]
- Adobe XD [53]

Los programas informáticos para la generación de prototipos son herramientas que producen una experiencia de usuario para nuevos productos de software.

Los prototipos de alta fidelidad pueden ser navegables, se pueden crear de manera ágil y colaborativa, simulan el comportamiento del nuevo software que se pretende construir enlazando pantallas y funcionalidades a través de zonas calientes accionables por el usuario, permiten validar en corto plazo la experiencia final del usuario en el nuevo producto de software.

En la Figura 32 se muestra un diseño de prototipo de alta fidelidad de un banco de preguntas integrado en un sistema de gestión del aprendizaje.

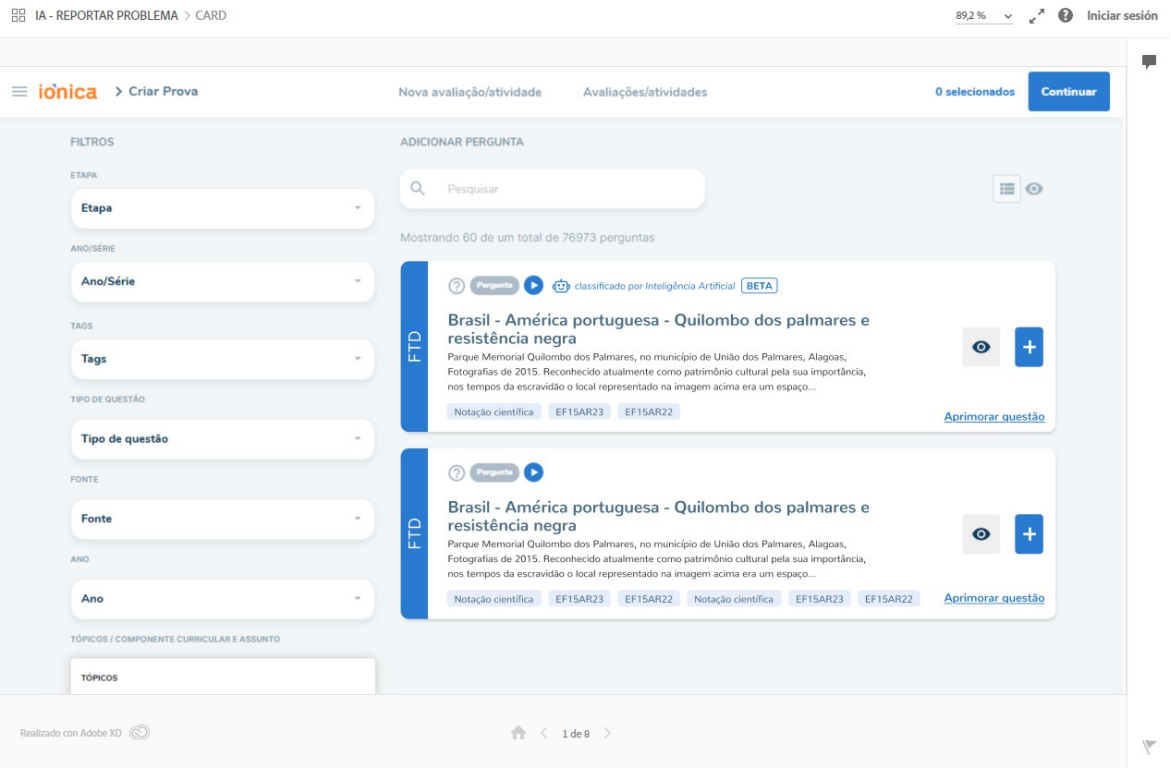

Figura 32: Diseño de prototipos en alta fidelidad en Adobe XD [Elaboración propia]

En la Figura 33 se puede observar otro prototipo de alta fidelidad, también dentro de un sistema de gestión del aprendizaje, pero en esta ocasión dentro de un flujo diferente, se trata de una funcionalidad para crear clases con contenidos editoriales predefinidos como plantilla.

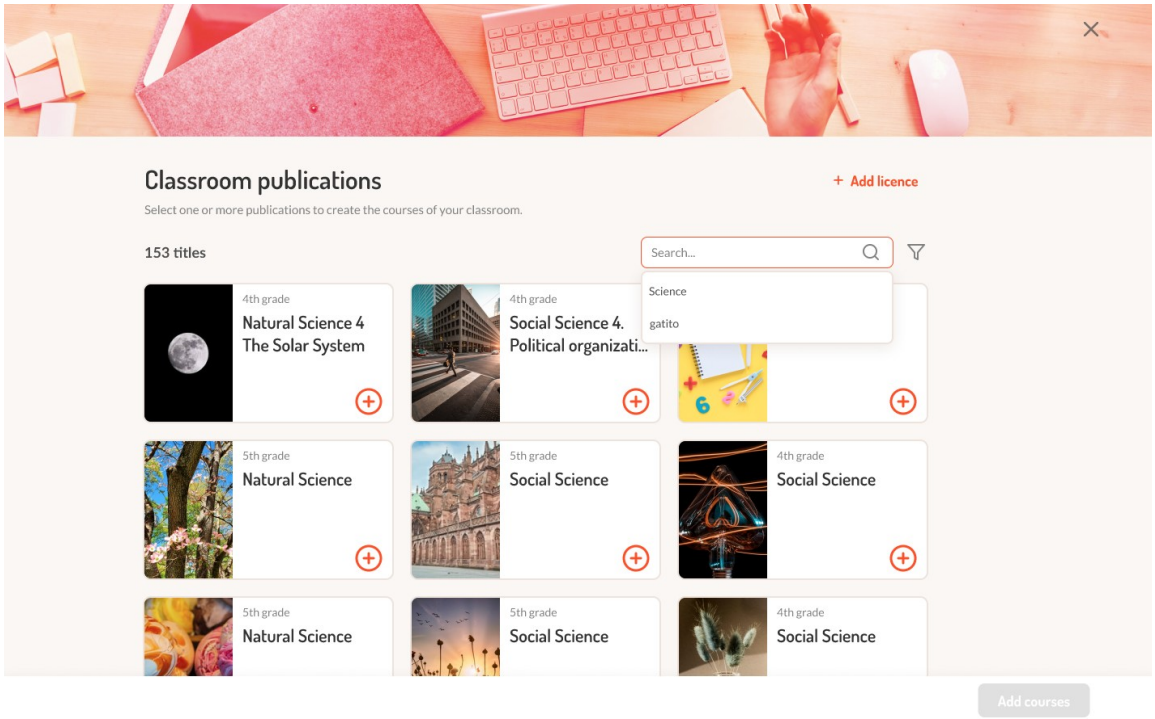

Figura 33: Diseño de prototipos en alta fidelidad en Figma [Elaboración propia]

Una vez validados los prototipos de experiencia de usuarios se puede considerar que están listos para el desarrollo, esto es, la integración final, en las tecnologías que correspondan (Frontend, Backend, Apps, React, Android Studio, Node JS, etc).

Existen herramientas como Zeplin [54] que facilitan el traspaso del Prototipo UX al equipo de desarrollo tecnológico. Toman como origen los diseños creados en software de diseño gráfico como ser, Figma, Sketch, Adobe XD, Photoshop, se exportan a una versión web con especificaciones concretas de cada pantalla y elemento de pantalla, añadiendo además, las líneas código de hoja de estilos CSS correspondientes. Este proceso de exportación reduce tiempo de integración y garantiza que las interaces de usuario se vean como el diseño original.

En la Figura 34 se muestran los detalles y propiedades de la interfaz gráfica del usuario antes de ser implementados en el software definitivo.

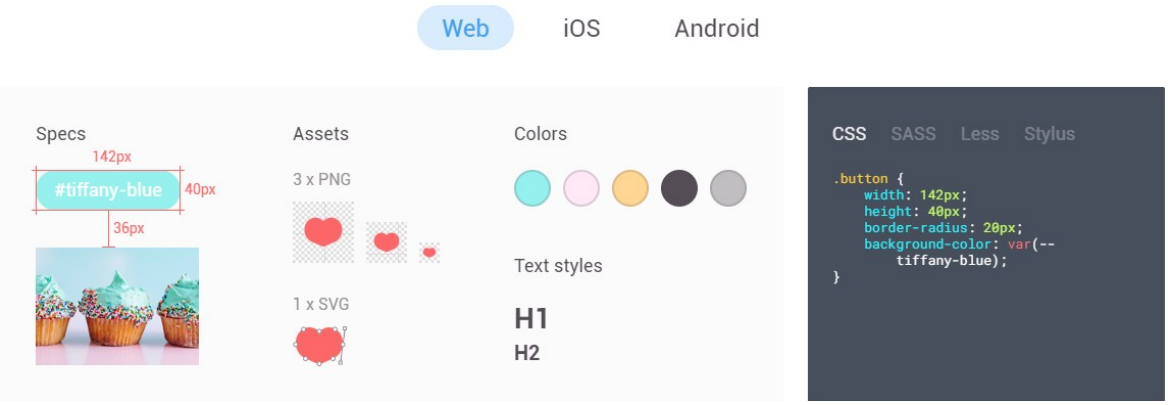

Figura 34: Publicación de diseños en Zeplin [Elaboración propia]

En la Figura 35 se presenta una pantalla completa conformada por varios componentes, todos los competentes pueden ser inspeccionados y exportados para ser integrados en el software definitivo, de esta manera se reducen las incidencias de integración en el proceso de maquetación web o integración definitiva de interfaces de usuario.

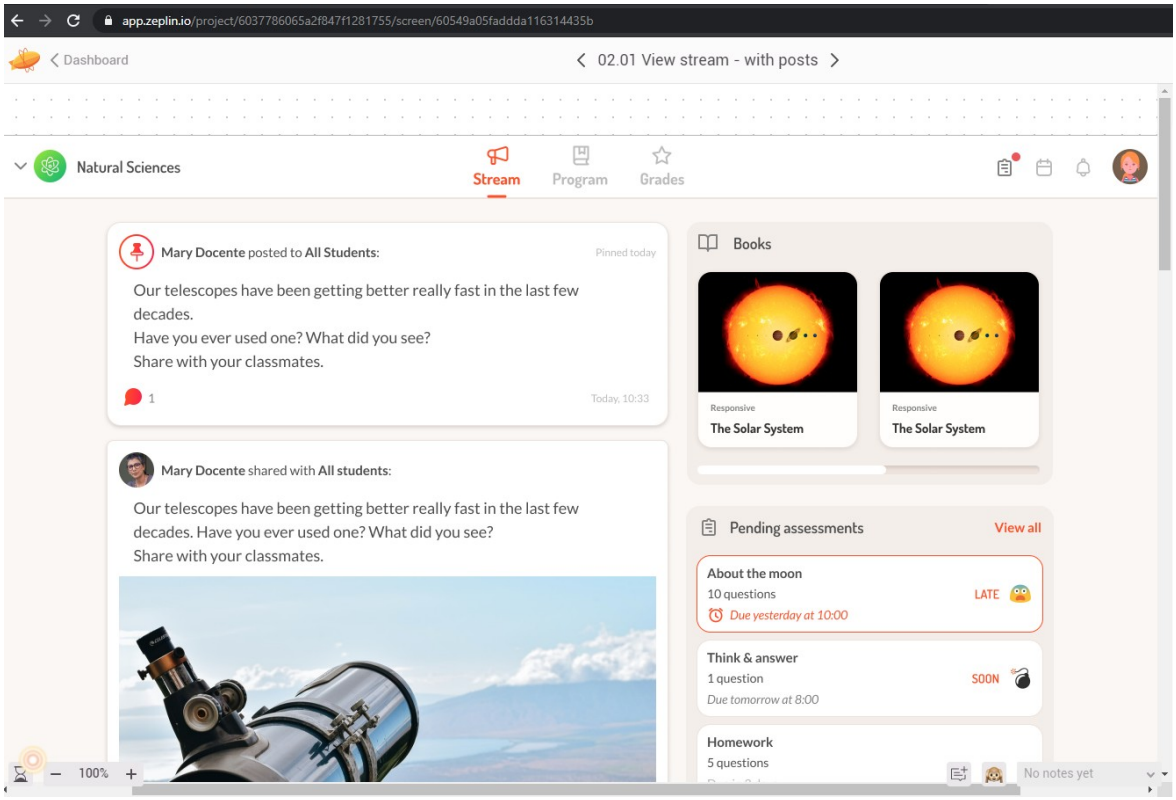

Figura 35: Publicación de diseños en Zeplin, pantalla de mensajes del curso [Elaboración propia]

3.2.3. Etapa III - Validar

La etapa de validación consiste en presentar prototipos a los usuarios finales y responsables del proyecto para conseguir la validación del mismo, recopilar comentarios, mejoras y correcciones que retroalimentan la especificación del producto software.

Una vez aprobados los prototipos se puede pasar a la etapa del desarrollo tecnológico donde se hará una integración tecnológica final, con las normas del buen arte de la ingeniería de software.

CAPÍTULO IV

Aplicación de Design Thinking

4. Caso de estudio

En este capítulo se mencionan las evidencias y resultados de las tareas realizadas en la aplicación de Design Thinking para la definición de plataforma educativa.

La aplicación de la propuesta adaptada de Design Thinking en el presente trabajo define el producto software resultante.

Oneclick S.L. es la empresa tecnológica involucrada en todo el proceso de producción. El producto de software resultante forma ahora parte de su base de conocimiento.

4.1. Etapa I - Empatizar

Para empatizar y conocer a los usuarios del producto software se propuso el taller de "The value proposition canvas" como se ve en la Figura 36. En el workshop participaron docentes, directivos de instituciones educativas, diseñadores, editores y responsables técnicos de diversas áreas.

Workshop realizado en las instalaciones de Editorial FTD, San Pablo, Brasil. En la Figura 36, los participantes del workshop escriben fragmentos de textos que se colocan en la propuesta de valor que se encuentra en la pizarra.

Figura 36: Workshop aplicando The value proposition canvas, identificación de elementos [Elaboración propia]

Como resultado del workshop se obtienen las historias de usuario más relevantes con las cuales se especificará el MVP y Roadmap del producto software.

Las historias de usuario más relevantes son las que más puntuación recibieron por parte de los miembros del equipo multidisciplinar que participa en el taller.

El resultado del trabajo realizado se presenta en la Figura 37, la propuesta de valor se completa a través de fragmentos de textos de distinto color, los requisitos son valoradas con puntos verdes o rojos.

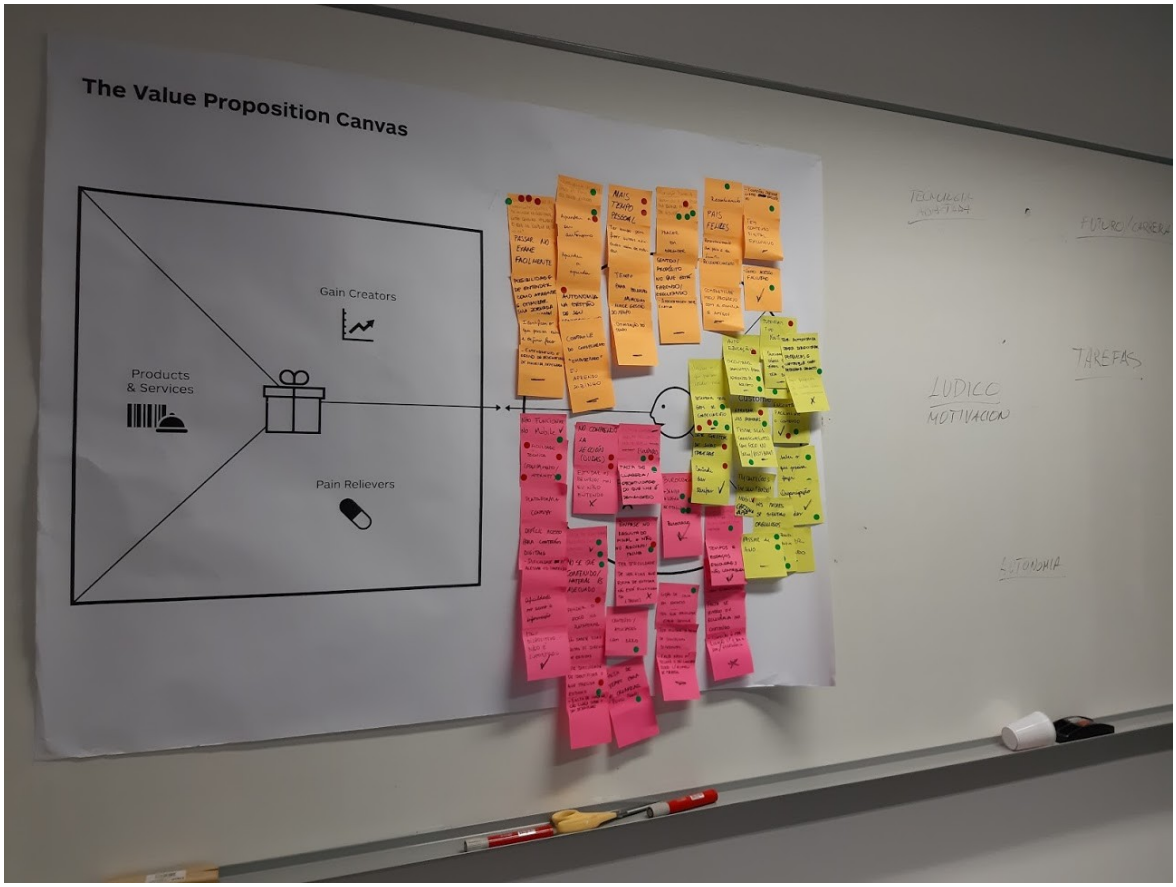

Figura 37: Taller The value proposition canvas, Post its, valoraciones [Elaboración propia]

Cuando el taller no se puede realizar de manera presencial una buena alternativa es la utilización de herramientas interactivas colaborativas que funcionan a través de Internet, un ejemplo de ellas es Miro [55].

En la Figura 38 se definen historias de usuario catalogadas dentro de la herramienta en línea Miro.

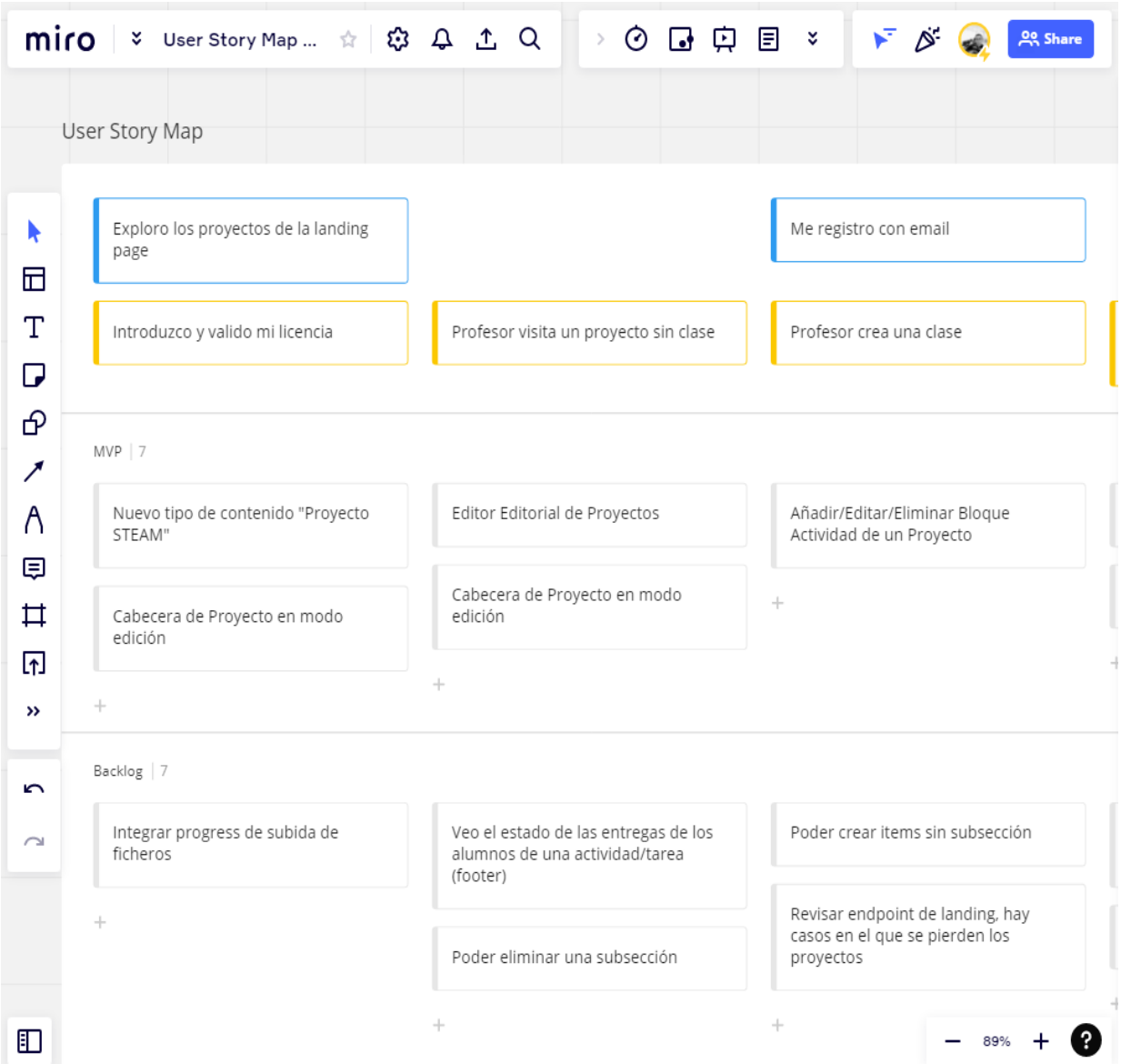

Figura 38: Miro, lienzo colaborativo en línea [Elaboración propia]

4.2. Etapa II - Prototipado

El prototipo resultante incluyó 304 pantallas en total que abarcaban las experiencias de los usuarios con rol docente y estudiante para navegadores de escritorio, dispositivos móviles y aplicaciones nativas.

En la Figura 39 se listan el diseño de algunas pantallas preparadas para dispositivos móviles y navegadores de escritorio.

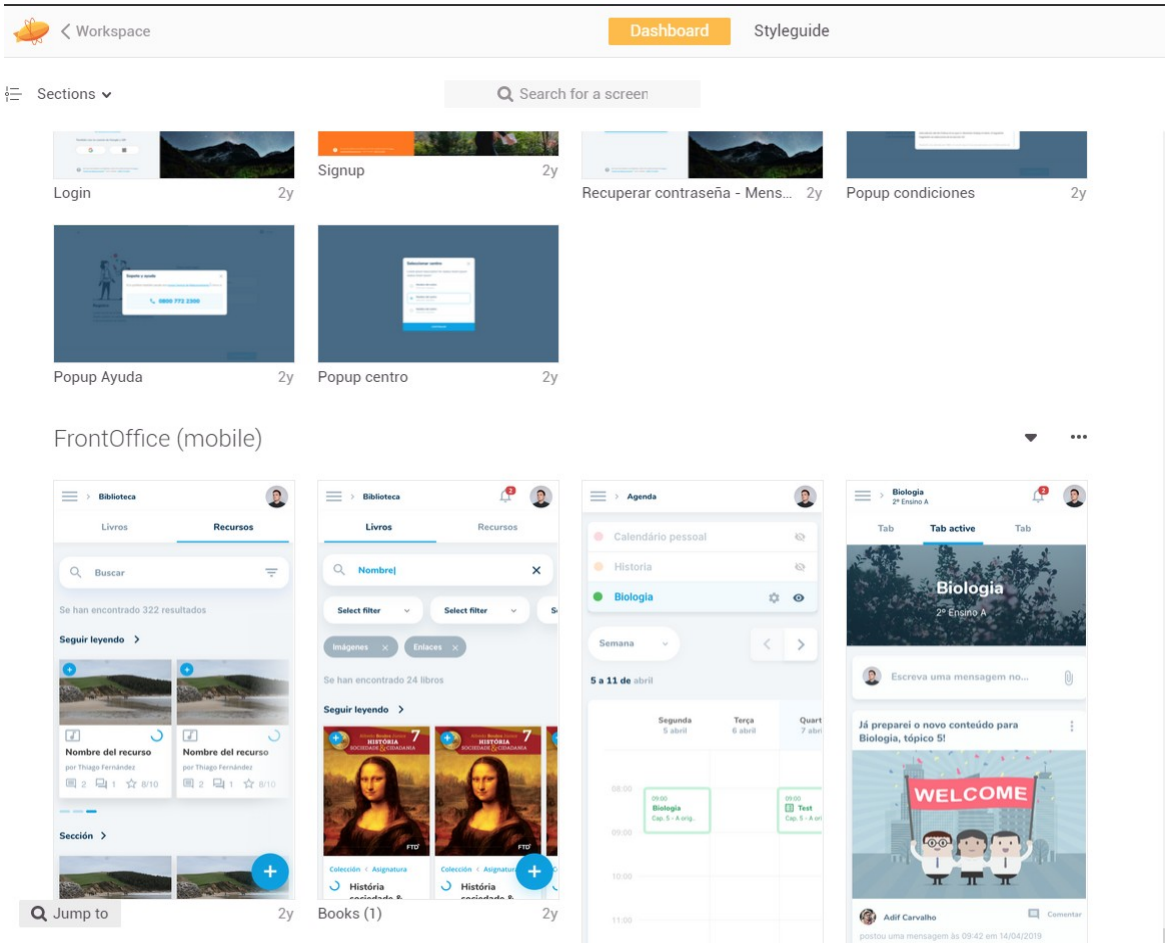

Figura 39: Prototipos navegables en Zeplin [Elaboración propia]

4.3. Etapa III - Validación

Para validar la definición del producto software se utilizan las versiones de alta fidelidad interactivas de los prototipos, la validación consiste en realizar una demostración a pantalla compartida simulando el uso del producto software por parte de los diferentes roles de usuario en un recorrido por las diferentes historias de usuario.

Como resultado del proceso de validación se obtienen comentarios y correcciones que permitirán depurar los prototipos consiguiendo de esta manera satisfacer de mejor manera las necesidades y requerimientos de los usuarios.

Los prototipos navegables, como el mostrado en la Figura 40, hace posible experimentar el flujo de pantallas que recorrerá el usuario final al realizar una tarea. Las

zonas resaltadas de color celeste son áreas interactivas que llevan a otra pantalla.

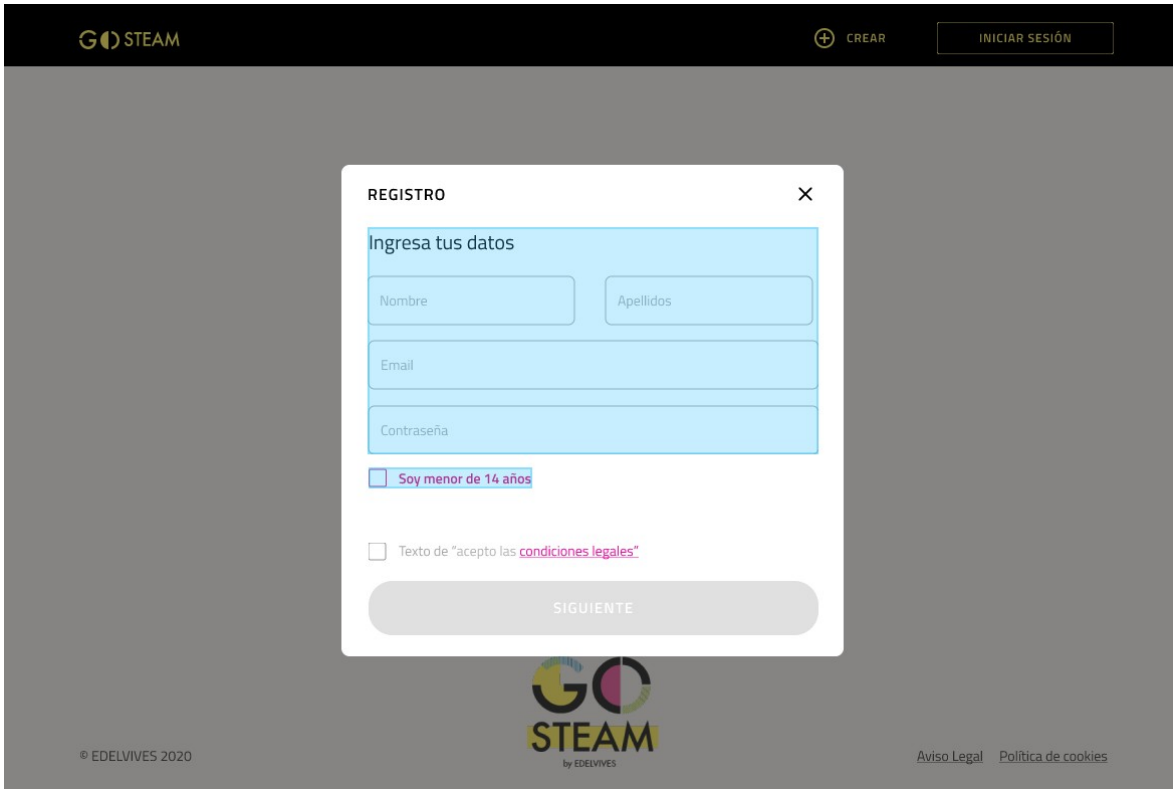

Figura 40: Prototipo navegable con zonas interactivas [Elaboración propia]

4.4. Artefactos resultantes del proceso Design Thinking

En esta sección se detallan los artefactos producidos que conforman la salida del proceso Design Thinking y a su vez se convierten en las especificaciones de entrada para la implementación del producto software.

Los artefactos ágiles se registran en una herramienta de gestión de equipos ágiles, por ejemplo, Jira [55].

4.5. Épicas

Las épicas obtenidas por el proceso Design Thinking fueron 32. Algunas de las cuales fueron las siguientes:

- Un usuario anónimo visita la web.
- Proceso de registro de un profesor o alumno.
- Un alumno se une y gestiona las clases a las que pertenece.
- Un alumno revisa qué tareas tiene pendientes de realizar.
- Un profesor interactúa con un proyecto asignado a una clase.

Al iniciar el desarrollo, todas las épicas están sin asignar, como se muestra en la Figura 41, al comenzar el trabajo de implementación, se eligen las primeras épicas a realizar y se designa una persona responsable.

Tipo	Clave	Resumen	Persona asignada	Informador	Pr
	GOS-794	Autoevaluación / Heteroevaluación	Sin asignar	Francisco Cuenca	
	GOS-117	Un profesor usa el Authoring Tool para personalizar un proyecto	Sin asignar	Camilo López	
	GOS-112	Herramienta para facilitar al profesor la proyección de material en clase	Sin asignar	Camilo López	
	GOS-108	Un alumno o profesor explora los recursos disponibles en GoSteam	Sin asignar	Camilo López	
	GOS-104	Un alumno revisa la evaluación de su profesor sobre una tarea/actividad	Sin asignar	Camilo López	
	GOS-91	Un profesor revisa las entregas de los alumnos y pone una evaluación	Sin asignar	Camilo López	
	GOS-88	Un alumno se une y gestiona las clases a las que pertenece	Sin asignar	Camilo López	
	GOS-83	Un admin editorial administra los distintos perfiles de la plataforma y la relación entre ellos	Sin asignar	Camilo López	
	GOS-79	Un usuario anónimo visita la web de GoSTEAM	Sin asignar	Camilo López	
	GOS-77	Un alumno revisa que tareas tiene pendientes de realizar	Sin asignar	Camilo López	
	GOS-76	Un profesor revisa que tareas tiene pendientes de corregir	Sin asignar	Camilo López	
	GOS-71	Un profesor interactua con un proyecto asignado a una clase	Sin asignar	Camilo López	

Figura 41: Listado de épicas en Jira Software [Elaboración propia]

4.6. Historias de usuario

Las historias de usuario generadas en el proceso Design Thinking fueron en total 109. Algunas de las cuáles se listan a continuación.

- Veo la entrega de un alumno.

- Evalúo la entrega de un alumno.
- Devuelvo la entrega a un alumno.
- Permito que un alumno pueda rehacer una tarea/actividad.
- Veo las tareas que ya he corregido.

Las historias de usuario normalmente pertenecen, o se corresponden a una épica, aunque no es un requisito obligatorio. En la Figura 42 se listan historias de usuarios, cada una de las cuales está representada con un identificador único, nombre de historia de usuario, persona asignada, informador, prioridad y el estado actual.

Tipo	Clave	Resumen	Persona asignada	Informador	Pr	Estado
	GOS-773	Al menos un proyecto cuando no estás registrado no devuelve ningún item	Alejandro Micheloud	Agustín Ortega		TEST
	GOS-731	Soporte multi-language en canales	Eduardo Aranda	Alejandro Micheloud		FINALIZADA
	GOS-275	En explorar, Ver todos los proyectos de una etapa	Eduardo Aranda	Francisco Cuenca		FINALIZADA
	GOS-183	Profesor desvincula una clase de un proyecto	Eduardo Aranda	Francisco Cuenca		FINALIZADA
	GOS-143	Importador en bloque de contenidos multimedia	Eduardo Aranda	Camilo López		FINALIZADA
	GOS-116	Controlo la reproducción de un proyecto desde mi teléfono	Janire Noguerol	Camilo López		FINALIZADA
	GOS-114	Proyecto contenido en la pizarra	Eduardo Aranda	Camilo López		FINALIZADA
	GOS-103	Dejo un comentario privado a un alumno	Eduardo Aranda	Camilo López		FINALIZADA
	GOS-96	Permito que un alumno pueda rehacer una tarea/actividad	Eduardo Aranda	Camilo López		FINALIZADA
	GOS-95	Devuelvo la entrega de un alumno	Eduardo Aranda	Camilo López		FINALIZADA
	GOS-94	Evaluo la entrega de un alumno	Eduardo Aranda	Camilo López		FINALIZADA
	GOS-89	Alumno ve los proyectos de una clase	Eduardo Aranda	Camilo López		FINALIZADA

Figura 42: Listado de historias de usuario en Jira Software [Elaboración propia]

4.7. Gráfico de control

La gráfica de control distribuye las incidencias y clúster de incidencias a través del tiempo indicando en el eje Y el tiempo en días invertido en la realización de la misma, hay que tener en cuenta que lo que se denomina incidencia son realmente tareas de historias de usuario, es decir, nuevas funcionalidades que se añaden al software.

El promedio general para la realización de una tarea es de 10 días, en un Sprint de dos semanas se abordan varias tareas en simultáneo.

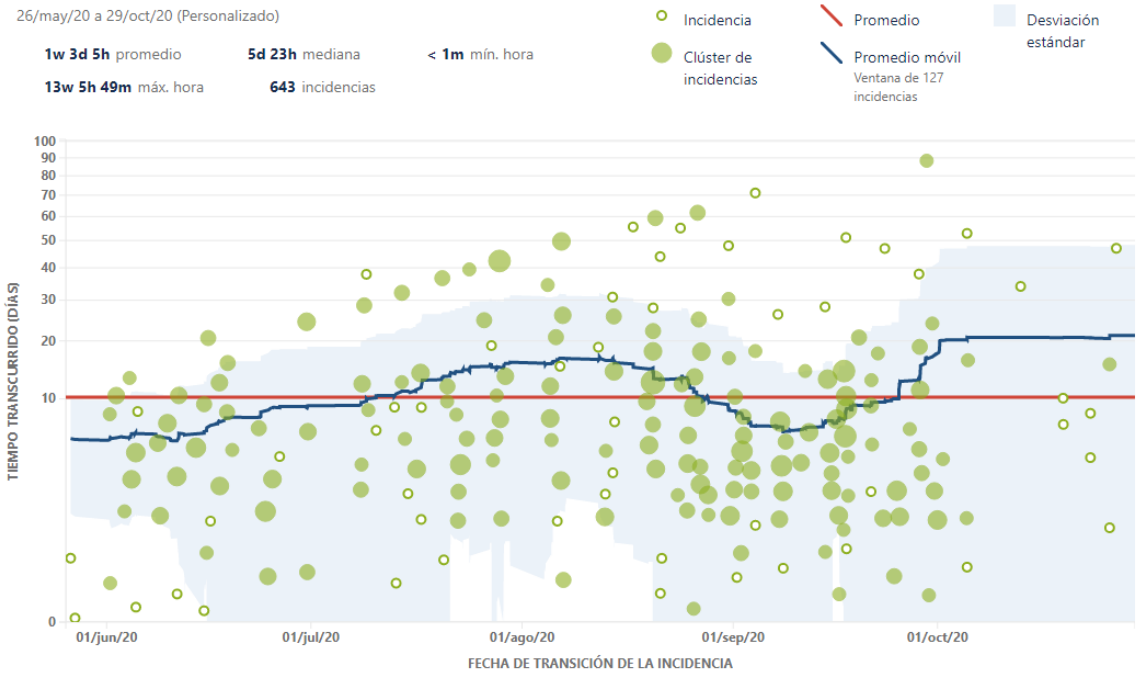

Figura 43: Gráfica de Control [Elaboración propia]

El gráfico de control, como el mostrado en la Figura 43, proporciona una visión general que permite identificar elementos de interés como lo son la eficiencia y valores atípicos.

La eficiencia se puede leer a través del promedio móvil, que está representado en el gráfico con la línea azul, se alcanza mayor nivel de eficiencia cuando menor sea el tiempo utilizado en realizar las tareas. La mayor eficiencia se logró cuando en promedio se utilizaban de 5 a 6 días por tarea, y la mayor ineficiencia se alcanzó al requerir en promedio 20 días por incidencia.

Los valores atípicos son las incidencias o clúster de incidencias que están muy por encima de la media, es recomendable identificarlos, una vez identificados se pueden investigar las causas para reducirlos en el futuro.

Un ejemplo de caso atípico es la incidencia GOS-347, de la Figura 44, donde se indica que los usuarios no pueden iniciar sesión con cuentas de Google y Microsoft si

navegan en modo incógnito, como es una incidencia que ocurre en un escenario atípico se decidió no atenderla de inmediato. No obstante, se puede crear una base de conocimiento de incidencias frecuentes para reaccionar más rápidamente en el futuro cuando se presente una incidencia de estas características en otro proyecto.

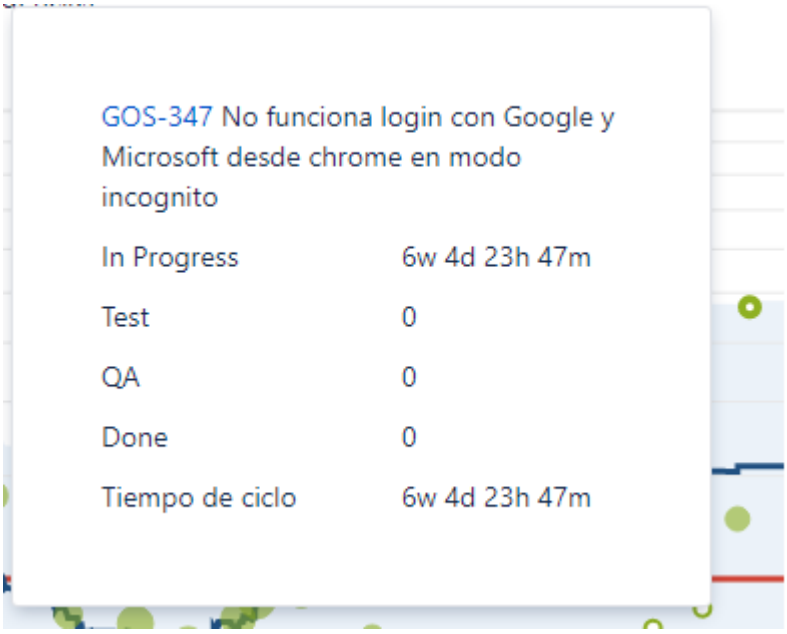

Figura 44: Caso atípico [Elaboración propia]

4.8. Sprints

En total se han llevado a cabo 33 Sprints, en la tabla 9 se indican los nombres de los sprints, las fechas de inicio y fin junto a la cantidad de tareas en el tablero ágil.

A partir del Sprint 14 el producto de software comenzó a ser utilizado por los usuarios finales, no obstante, desde el Sprint 3 comenzó a ser utilizado por el equipo editorial con el uso de la herramienta de autor para proyectos STEAM.

La cantidad de personas y equipos abocados a los Sprints es dinámico, al inicio no había equipo de calidad asignado al proyecto, luego del Sprint MVP se comienza a trabajar con el equipo de calidad añadiéndose al tablero incidencias reportadas internamente.

Tabla 9: Informe de Sprints

#	Nombre del Sprint	Fecha de inicio y fin	Tareas
1	Authoring Tool Proyectos	20/may/20 - 01/jun/20	10
2	Parte pública y AT Proyectos	01/jun/20 - 08/jun/20	16
3	MVP 1	08/jun/20 - 15/jun/20	17
4	MVP 2	15/jun/20 - 22/jun/20	14
5	MVP 3	22/jun/20 - 02/jul/20	17
6	Revisión Actividades	02/jul/20 - 13/jul/20	28
7	ToDo y Notificaciones	13/jul/20 - 20/jul/20	35
8	Apps terceros	20/jul/20 - 28/jul/20	51
9	Proyectar Bugs	28/jul/20 - 03/ago/20	44
10	Proyectar Bugs 2	03/ago/20 - 10/ago/20	34
11	Proyectar Bugs 3	10/ago/20 - 17/ago/20	39
12	Rematar Core y Bugs	17/ago/20- 24/ago/20	49
13	Rematar Core y Bugs 2	24/ago/20- 31/ago/20	66
14	Rematar Core y Bugs 3	31/ago/20 - 07/sep/20	59
15	Rematar Core y Bugs 4	07/sep/20 - 14/sep/20	65
16	Rematar Core y Bugs 5	14/sep/20 - 21/sep/20	55
17	Rematar Core y Bugs 6	21/sep/20 - 29/sep/20	29
18	Rematar Core y Bugs 7	29/sep/20 - 08/oct/20	21
19	Rematar Core y Bugs 8	13/oct/20- 19/oct/20	14

20	Rematar Core y Bugs 9	19/oct/20 - 26/oct/20	15
21	Rematar Core y Bugs 10	26/oct/20 - 21/dic/20	12
22	Go Steam Catalan 1	21/dic/20 - 27/ene/21	20
23	Rematar Core y Bugs 12	22/mar/21 - 13/abr/21	6
24	Go Steam Fase 2 - 1	13/abr/21 - 17/may/21	12
25	Diana - Licenciamiento Publica	17/may/21 - 30/jun/21	19
26	Recomendador	30/jun/21 - 12/jul/21	8
27	Go Steam Fase 2 - 4	12/jul/21 - 30/ago/21	16
28	Issues varias	30/ago/21 - 05/oct/21	19
29	Go Steam Fase 2 - 6	06/oct/21 - 09/nov/21	15
30	Go Steam Fase 2 - 7	10/nov/21 - 23/nov/21	16
31	Go Steam Gradebook	23/nov/21 - 14/dic/21	16
32	Go Steam 2022 - 1	16/dic/21 - 25/feb/22	24
33	Go Steam Fase 2 - Duplicar	02/mar/22 - 29/sep/22	17

4.9. Burndown

El gráfico de Burndown es el gráfico del trabajo pendiente que tiene el equipo tecnológico durante el desarrollo de un Sprint. El eje Y representa la cantidad de tareas pendientes, el eje X representa el paso del tiempo. El gráfico permite determinar la cantidad de trabajo que asume el equipo de desarrollo tecnológico en el Sprint Planning y la capacidad para ir realizado las tareas en la medida que pasa el tiempo.

El gráfico de Burndown permite realizar un seguimiento del trabajo total restante y pronostica la probabilidad de alcanzar el objetivo del Sprint. Esto ayuda al equipo a

gestionar sus propios avances y responder en consecuencia.

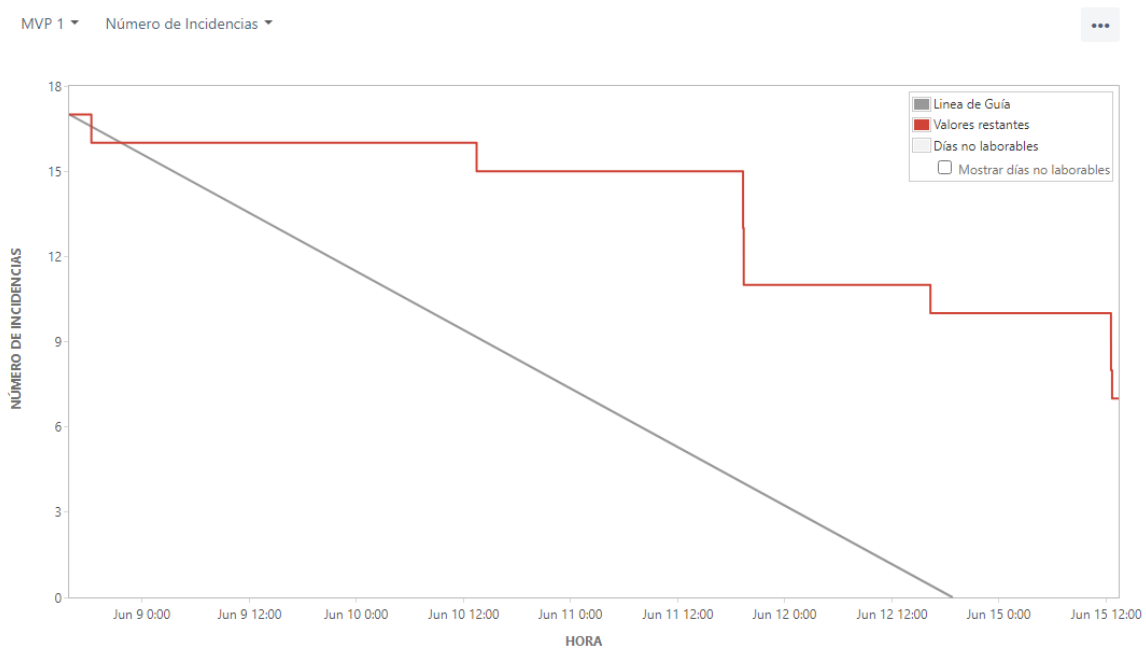

Figura 45: Burndown - Sprint MVP 1 [Elaboración propia]

En la Figura 45 Burndown, del “Sprint MVP 1” se observa que comenzó el Sprint con 18 tareas y se finalizó con 7 tareas sin terminar, en total se realizaron 11 tareas en el Sprint.

En la Figura 46 “Burndown – Sprint MVP 2” se comenzó el Sprint con 14 tareas y se finalizó con 7 tareas sin terminar, en total se realizaron 7 tareas en el Sprint.

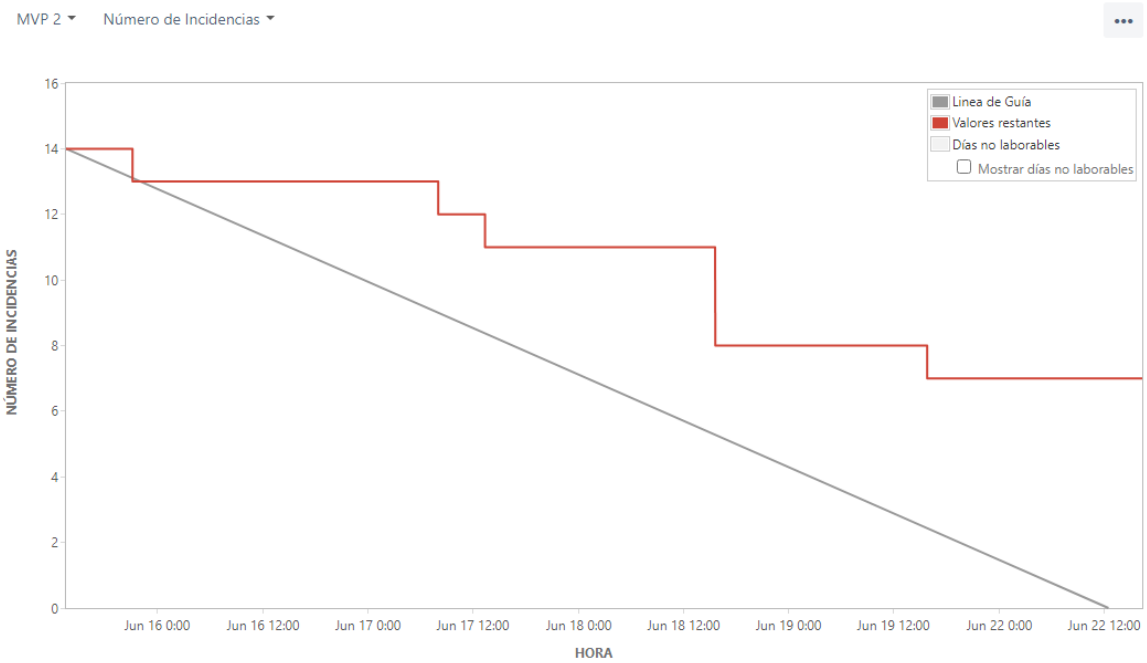

Figura 46: Burndown - Sprint MVP 2

En la Figura 47 del "Burndown – Sprint MVP 3" se comenzó el Sprint con 14 tareas, luego se cambió el alcance del Sprint añadiéndose más tareas que sumaron 17 en total, se finalizó con 6 tareas sin terminar, en total se realizaron 11 tareas en el Sprint.

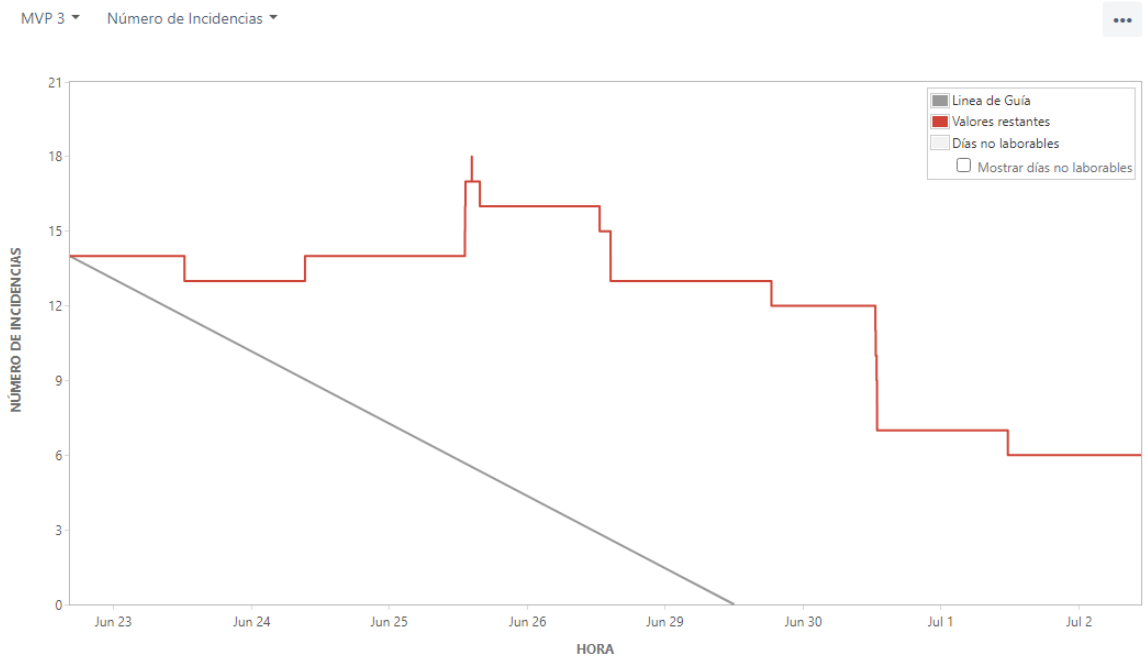

Figura 48: Burndown - Sprint MVP 3 [Elaboración propia]

4.10. Encuesta al equipo técnico

Se propone llevar a cabo una encuesta al equipo técnico involucrado en el desarrollo tecnológico del producto de software.

Tiene como objetivo principal analizar la propuesta Design Thinking en el proceso de desarrollo de software y detectar oportunidades de mejora.

La encuesta utiliza la escala de Likert [57], las escalas Likert son instrumentos psicométricos donde el encuestado debe indicar su acuerdo o desacuerdo sobre una afirmación, ítem o reactivo, lo que se realiza a través de una escala ordenada y unidimensional.

La encuesta con método Likert tiene por objetivo recopilar el grado de satisfacción

acerca de cada item que se evalúa. Las opciones de respuesta incluyen:

1. Totalmente en desacuerdo
2. En desacuerdo
3. Ni de acuerdo ni en desacuerdo
4. De acuerdo
5. Totalmente de acuerdo

Los ejes principales de la encuesta son:

- Design Thinking en la producción de software.
- Proceso de Design Thinking propuesto.

Título de la encuesta: Evaluación de desempeño de Design Thinking en el proceso de definición de producto software.

Encuesta

Eje temático	**Encuesta**
Design Thinking	El Design Thinking puede ser utilizado para definir un producto de software.
	El Design Thinking ayuda a empatizar con los usuarios y permite detectar las necesidades del usuario final.
	El Design Thinking ayuda a construir y validar prototipos del producto software.
	La adopción de Design Thinking en empresas de producto de software es de vital importancia.
Propuesta de Design Thinking	El Design Thinking aplicado en proyectos como GOSTEAM ha mejorado la calidad percibida por los usuarios.
	El Design Thinking permitió identificar las historias de usuario

más relevantes para conseguir un MVP.

La adopción de Design Thinking al inicio del proceso de desarrollo ha permitido un mejor entendimiento de los requerimientos del producto software.

La adopción de Design Thinking ha contribuido a la empatía de los usuarios con el producto.

La adopción de Design Thinking ha contribuido a afrontar adecuadamente los cambios en los requerimientos.

La adopción de Design Thinking ha contribuido para que el producto de software sea mantenible.

La adopción de Design Thinking ha contribuido a resolver errores con celeridad.

4.10.1.1. Resultados

Se encuestaron a seis de los técnicos encargados de implementar y validar el software en la empresa Oneclick, en total los técnicos han respondido las once preguntas.

En la Figura 49 las preguntas se encuentran en la primer fila, con el prefijo P y valores P1 al P11. Los encuestados se encuentran en la primer columna con el prefijo E, valores E1 al E6.

El grado de satisfacción por encuestado se encuentra en la última columna en tanto que el promedio conformidad por pregunta se encuentra en la última fila.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	**Total**	Promedio
E1	5	5	5	3	5	3	5	5	5	5	3	**49**	4,45
E2	5	4	5	4	5	5	5	4	5	5	4	**51**	4,64
E3	5	5	5	4	5	4	5	5	4	5	4	**51**	4,64
E4	4	4	4	5	4	4	5	5	5	3	4	**47**	4,27
E5	5	5	5	5	5	4	4	4	4	4	4	**49**	4,45
E6	5	5	5	5	4	4	5	4	4	4	5	**50**	4,55
Total	**29**	**28**	**29**	**26**	**28**	**24**	**29**	**27**	**27**	**26**	**24**		
Promedio	4,14	4,00	4,14	3,71	4,00	3,43	4,14	3,86	3,86	3,71	3,43		

Figura 49: Resultados de la encuesta al equipo técnico [Elaboración propia]

A partir de los resultados se puede conocer el grado de conformidad general por ténico como así también el grado de conformidad por pregunta realizada.

Para determinar el grado de conformidad por pregunta realizaremos un gráfico totalizando la puntuación obtenida por pregunta.

En la Figura 51, el eje X representa las preguntas realizada en la encuesta y el eje Y muestra la sumatoria del valor de todas las respuestas, de todos los docentes, a esa pregunta.

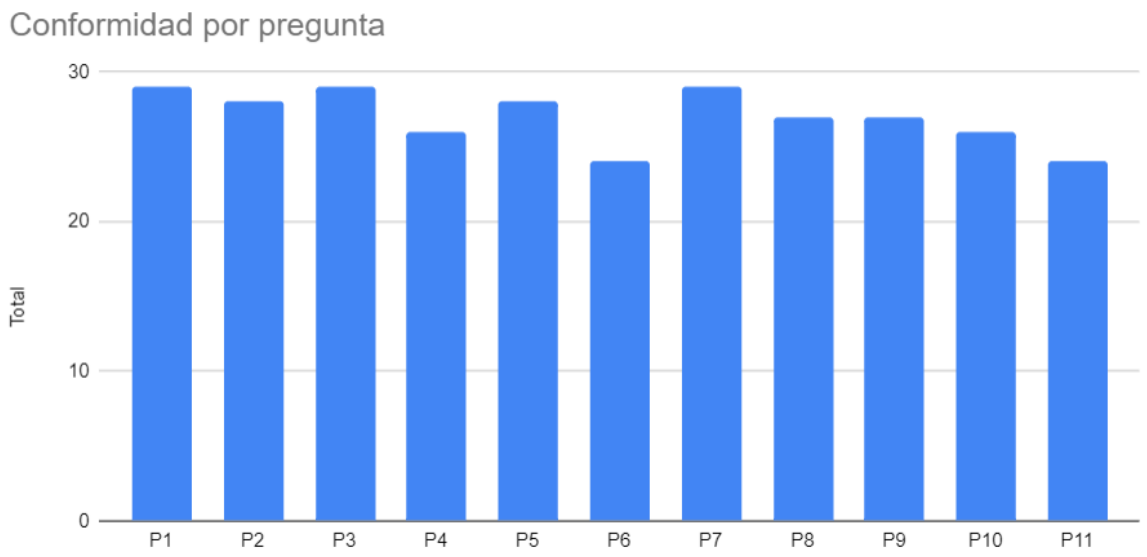

Figura 50: Grado de conformidad por pregunta [Elaboración propia]

Las preguntas 6 y 11 son las que tienen menos grado de conformidad, las preguntas 1, 3 y 7 en cambio, son las que mayor grado de conformidad tienen según el equipo técnico.

Para tener una visión general de conformidad, se contabilizan los grados de conformidad por pregunta, agrupando las respuestas iguales en un solo indicador.

Grado de conformidad												Total
G1	0	0	0	0	0	0	0	0	0	0	0	0
G2	0	0	0	0	0	0	0	0	0	0	0	0
G3	0	0	0	1	0	1	0	0	0	1	1	4
G4	1	2	1	2	2	4	1	3	3	2	4	25
G5	5	4	5	3	4	1	5	3	3	3	1	37
	6	6	6	6	6	6	6	6	6	6	6	66

Figura 51: Grado de conformidad agrupado por respuesta [Elaboración propia]

La Figura 52 nos revela que no hay ningún técnico que esté en desacuerdo con lo planteado en todas las preguntas.

El grado 3 es el puntaje con menor grado de conformidad. El 6,1% de los encuestados no está de acuerdo o ni en desacuerdo.

El 37% de los encuestados está de acuerdo y el 57% totalmente de acuerdo.

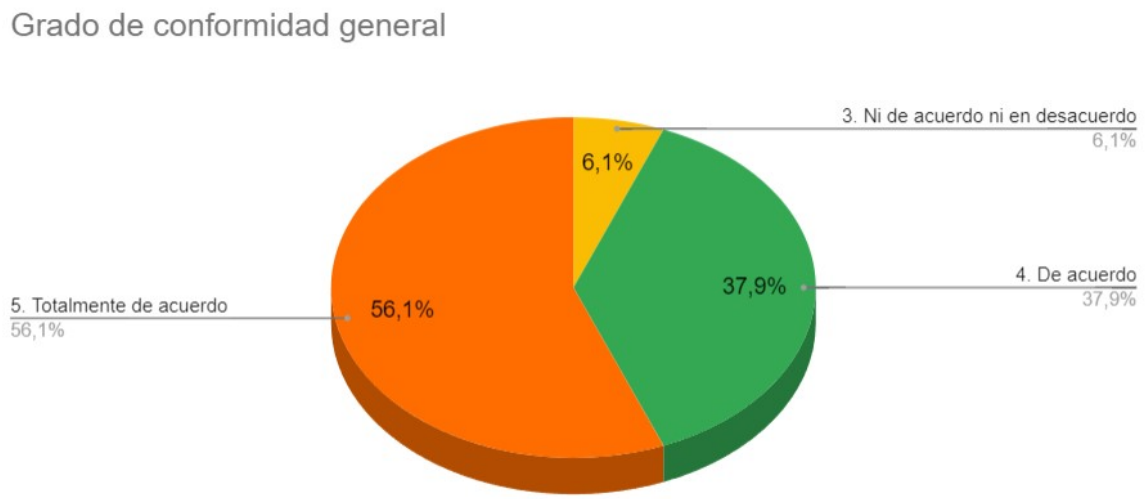

Figura 52: Grado de conformidad general [Elaboración propia]

4.11. Aplicación en contexto de educación superior

Con la finalidad de analizar una posible integración de la plataforma educativa en un contexto de educación superior se realizó un vídeo de presentación que describe las funcionalidades centrales de la plataforma.

El vídeo fue distribuido a docentes de educación superior junto con una encuesta que releva información sobre dos temáticas:

1. Evaluación objetiva de la plataforma.

2. Potencial uso de la plataforma en educación superior.

4.11.1. Vídeo presentación Go STEAM

Se produce un vídeo que sintetiza en poco más de 12 minutos el funcionamiento general de la plataforma GOSTEAM para docentes y estudiantes. Inicialmente se realiza una introducción general del contexto de la metodología de aprendizaje basado en proyectos y el marco del presente trabajo de investigación.

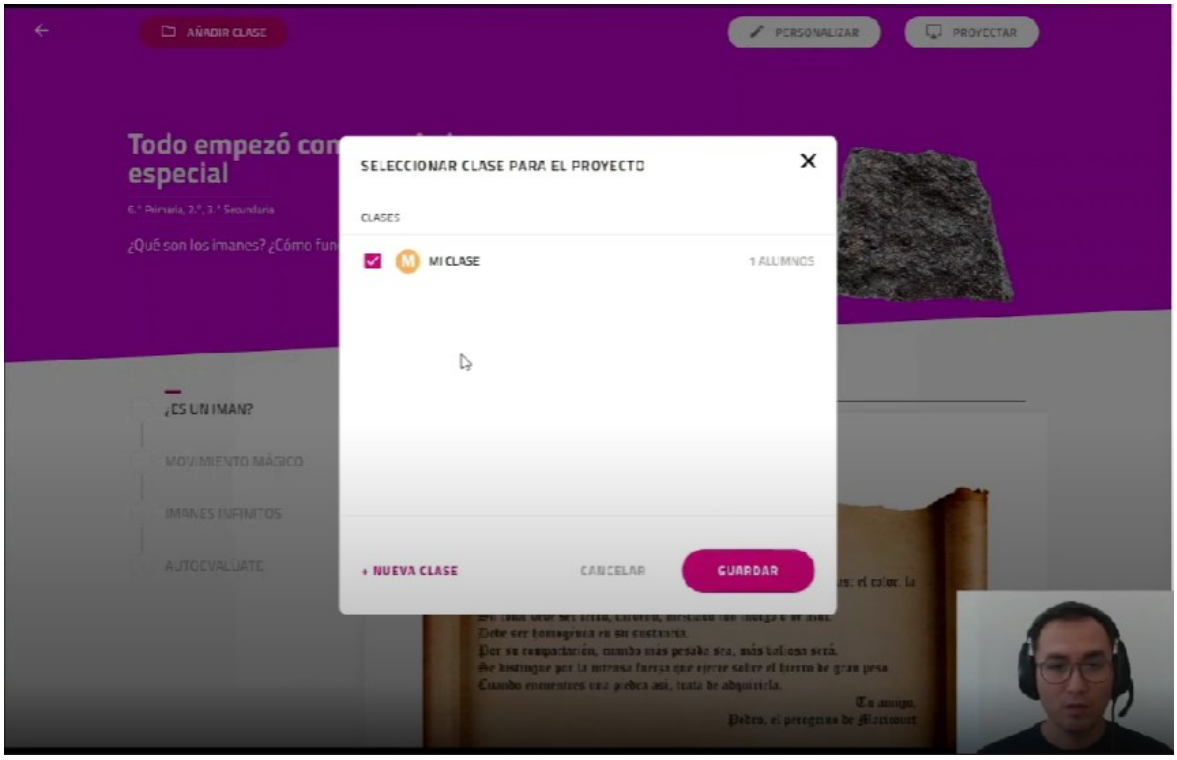

Figura 53: Vídeo de presentación GOSTEAM [Elaboración propia]

Enlace al vídeo: https://www.youtube.com/watch?v=d38skhaqNCo

4.11.2. Encuesta a docentes de educación superior

Las respuestas a la encuesta tendrán las siguientes opciones de respuesta:

1. Totalmente en desacuerdo
2. En desacuerdo
3. Ni de acuerdo ni en desacuerdo
4. De acuerdo
5. Totalmente de acuerdo

Nro.	Eje temático	Encuesta
1	Evaluación general de plataforma	GOSTEAM podría ser apropiada para su aplicación en "Entornos educativos interactivos mediados por Tecnologías de Información (TI)" en educación superior
2	Orquestación	GOSTEAM contribuiría a la orquestación del aula mediante la incorporación de actividades de lectura
3	Orquestación	GOSTEAM contribuiría a la orquestación del aula mediante la incorporación de actividades de trabajo en equipo
4	Orquestación	GOSTEAM contribuiría a la orquestación del aula mediante la incorporación de actividades de resolución de problemas
5	Roles de usuario	GOSTEAM permitiría discriminar claramente los roles de estudiantes y docentes
6	Impacto pedagógico	GOSTEAM permitiría una transferencia de las actividades al aula física y entornos de presenciales

		práctica
7	Impacto pedagógico	GOSTEAM permitiría el desarrollo de actividades científicas con los estudiantes
8	Gestión del aprendizaje	GOSTEAM vista como producto de software: podría ayudar y asistir a los docentes en la realización de proyectos
9	Metodología de aprendizaje	GOSTEAM podría ayudar y asistir a los estudiantes en el proceso de aprendizaje basado en proyectos*
10	Auto percepción	Como docente, poseo una adecuada formación en aprendizaje basado en proyectos
11	Auto percepción	Como docente, poseo una adecuada formación en entornos educativos mediados por TI
12	Propuesta de valor	GOSTEAM me permitiría desarrollar normalmente mis actividades docentes
13	Propuesta de valor	GOSTEAM permitiría a mis estudiantes mejorar su aprendizaje y trabajar de manera autónoma
14	Metodología de aprendizaje	Como docente, considero que el aprendizaje basado en proyectos debe tener lugar en educación superior
15	Intención de uso	¿Desearía realizar una prueba de GOSTEAM?

4.11.2.1. Resultados

Se encuestaron a siete docentes de educación que han respondido un total de 15 preguntas.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	**Total**	Promedio
E1	1	1	3	3	3	5	3	3	3	5	5	3	2	5	1	**46**	3,07
E2	5	4	5	5	5	4	5	4	4	3	4	4	3	4	5	**64**	4,27
E3	4	2	3	4	5	2	2	4	3	5	4	4	3	5	5	**55**	3,67
E4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	**75**	5,00
E5	5	5	5	5	5	5	5	3	5	3	3	5	3	5	5	**67**	4,47
E6	5	4	4	4	5	4	5	4	5	4	4	5	3	5	5	**66**	4,40
E7	4	4	5	5	5	3	4	5	5	5	5	5	5	5	5	**70**	4,67
Total	**29**	**25**	**30**	**31**	**33**	**28**	**29**	**28**	**30**	**30**	**30**	**31**	**24**	**34**	**31**		
Promedio	4,14	3,57	4,29	4,43	4,71	4,00	4,14	4,00	4,29	4,29	4,29	4,43	3,43	4,86	4,43		

Figura 54: Resultados de la encuesta a docentes de educación superior [Elaboración propia]

A partir de los resultados se puede conocer el grado de conformidad general por docente como así también el grado de conformidad por pregunta realizada.

Para determinar el grado de conformidad por pregunta realizaremos un gráfico totalizando la puntuación obtenida por pregunta.

En la Figura 51, el eje X representa las preguntas realizada en la encuesta y el eje Y muestra la sumatoria del valor de todas las respuestas, de todos los docentes, a esa pregunta.

Figura 55: Grado de conformidad por pregunta [Elaboración propia]

Las preguntas 2 y 13 son las que menos grado de conformidad acumulan, las preguntas 5 y 14 en cambio, son las que más grado de conformidad tienen entre los docentes de educación superior.

Para tener una visión general de conformidad, se contabilizan los grados de conformidad por pregunta, agrupando las respuestas iguales en un solo indicador.

Grado de conformidad																Total
G1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	3
G2	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	4
G3	0	0	2	1	1	1	1	2	2	2	1	1	4	0	0	18
G4	2	3	1	2	0	2	1	3	1	1	3	2	0	1	0	22
G5	4	2	4	4	6	3	4	2	4	4	3	4	2	6	6	58
	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	105

Figura 56: Grado de conformidad agrupado por respuesta [Elaboración propia]

La Figura 56 nos lleva a la conclusión de que 3 docentes han respondido con grado 1, es decir, están totalmente en desacuerdo, 4 de ellos están en desacuerdo, 18 ni en acuerdo ni en desacuerdo, 22 en acuerdo y 58 totalmente de acuerdo.

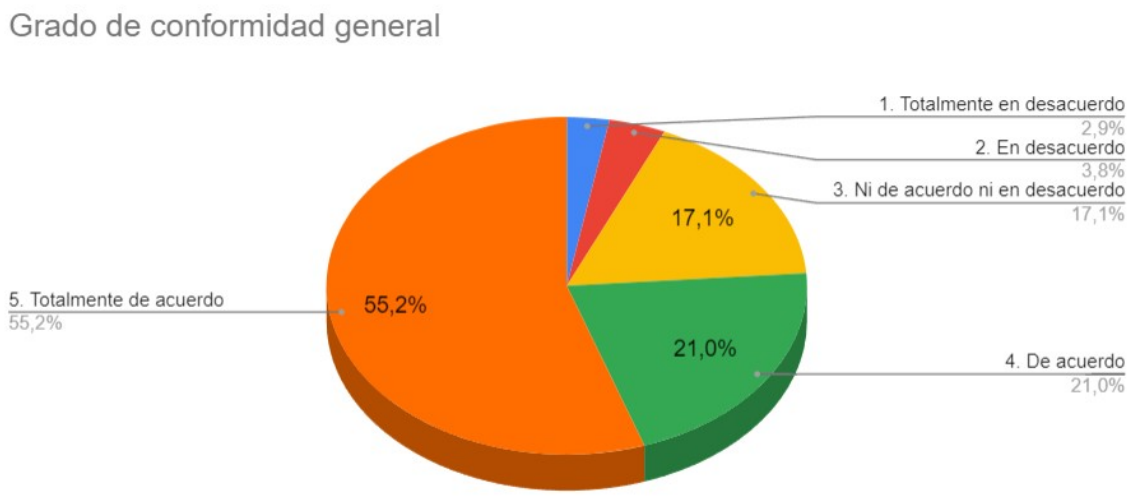

Figura 57: Grado de conformidad general [Elaboración propia]

En la Figura 57 se concluye que el 55,2% está totalmente de acuerdo, el 21% de acuerdo, y el 17,1% ni de acuerdo ni en desacuerdo.

Capítulo V

Conclusiones

5. Conclusiones

En este capítulo se describen las conclusiones alcanzadas en base a los objetivos planteados y las principales contribuciones del trabajo. Finalmente se describen las futuras líneas de investigación.

La propuesta de este trabajo basada en la adaptación de Design Thinking al desarrollo de productos software, se presenta como una opción para obtener productos alineados a los requerimientos de los usuarios.

La propuesta de Design Thinking se implementó con éxito en un caso de estudio dentro de la empresa Oneclick. Se realizaron validaciones para determinar el grado de aceptación de Design Thinking en el equipo tecnológico como así también el grado de aceptación de la metodología de enseñanza basada en proyectos en el contexto de educación de nivel superior.

La adecuación y aplicación de Design Thinking permite establecer los requisitos para el desarrollo de plataforma educativas centrada en el ser humano.

La evaluación del caso de estudio proporciona evidencia de que el Design Thinking produce un alineamiento de especificaciones y objetivos del producto de software de manera interna, dentro de la organización, a través de la generación de artefactos ágiles como ser las épicas e historias de usuarios. Además permite establecer una terminología en común entre el equipo de producto y el equipo de desarrollo mejora la comunicación.

La adopción de Design Thinking permite dar paso a la implementación tecnológica de un producto software. Los prototipos, la definición de producto y roadmap son artefactos valiosos para la organización y para el desarrollo tecnológico que se construye de manera ágil.

De acuerdo al caso de estudio evaluado, como resultado de empatizar con los usuarios finales y conocer sus necesidades, al haber diseñado el producto de software en conjunto con ellos, se consiguen beneficios como: experiencias de usuario y funcionalidades que se enfocan en el ser humano y sus problemáticas cotidianas; y se

consiguen automatizar tareas permitiendo ahorrar tiempo y esfuerzo.

Analizar y seleccionar tecnologías de comunicación que den soporte a procesos interactivos y colaborativos de enseñanza es un desafío importante en el contexto educativo, por lo que los dispositivos de los usuarios y conectividad a Internet son factores clave a considerar.

Con objeto de sumar evidencia, es conveniente evaluar la adaptación de este procedimiento Design Thinking en otras situaciones problemáticas relacionadas con la definición de producto para desarrollo tecnológico en educación, en otros niveles educativos y/o áreas de conocimiento.

La propuesta Design Thinking se puede seguir adaptando para cubrir de manera eficiente la toma de requerimientos y definición de producto en otros contextos relacionados con la tecnología de la información.

Bibliografía

[1] A. Artopoulos y D. Kozak, «Tsunami 1:1: estilos de adopción de tecnología en la educación latinoamericana», *Revista iberoamericana de ciencia tecnología y sociedad*, vol. 6, n.º 18, pp. 137-171, ago. 2011.

[2] R. Razzouk y V. Shute, «What Is Design Thinking and Why Is It Important?», *Review of Educational Research*, vol. 82, n.º 3, pp. 330-348, 2012, doi: 10.3102/0034654312457429.

[3] T. Brown, «Design Thinking», *IEEE Software*, vol. 37, n.º 2, pp. 21-24, 2020.

[4] R. I. Beverly, *Design Thinking for Entrepreneurs and Small Businesses*, 1.ª ed. Apress, Berkeley, CA, 2013.

[5] T. Brown, «Design Thinking», *Harvard Business Review*, vol. 86, n.º 6, 2008.

[6] G. Melles, Z. Howard, y S. Thompson-Whiteside, «Teaching Design Thinking: Expanding Horizons in Design Education», *Procedia - Social and Behavioral Sciences*, vol. 31, pp. 162-166, 2012, doi: 10.1016/j.sbspro.2011.12.035.

[7] G. Gabrysiak, H. Giese, y A. Seibel, «Towards Next Generation Design Thinking: Scenario-Based Prototyping for Designing Complex Software Systems with Multiple Users», en *Design Thinking*, C. Meinel, L. Leifer, y H. Plattner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 219-236. doi: 10.1007/978-3-642-13757-0_13.

[8] «The semantic turn: A new foundation for design: Klaus Krippendorff (Taylor & Francis, 2005)», *Artifact*, vol. 1, n.º 1, pp. 56-59, abr. 2007, doi: 10.1080/17493460600844157.

[9] T. Lindberg, C. Meinel, y R. Wagner, «Design Thinking: A Fruitful Concept for IT Development?», *Design Thinking*, pp. 3-18, 2011, doi: 10.1007/978-3-642-13757-0_1.

[10] S. Grönman y E. Lindfors, «The Process Models of Design Thinking», *Design and Technology in Education*, vol. 28 (2), n.º 2016, pp. 110-118, 2021.

[11] D. M. Murray, S. P. Varnell, y J. Blitstein, «Stanford dSchool», *American Journal of Public Health*, vol. 94, n.º 3. pp. 423-432, 2014. Accedido: 23 de octubre de 2021. [En línea]. Disponible en: dschool.stanford.edu

[12] «IDEO». Accedido: 23 de octubre de 2021. [En línea]. Disponible en: https://www.ideo.com

[13] «Google Ventures». [En línea]. Disponible en: https://www.gv.com

[14] J. Knapp, B. Kowitz, y J. Zeratsky, *Sprint : how to solve big problems and test new ideas in just five days*. Simon & Schuster, 2016.

[15] M. Cantwell, «DEEP Design Thinking». EDUCOLLABORATORS.

[16] A. Johnson, *Design Thinking with SAP*, SAP Press. Rheinwerk.

[17] J. Kwon, Y. Choi, y Y. Hwang, «Enterprise Design Thinking: An Investigation on User-Centered Design Processes in Large Corporations», *Designs*, vol. 5, n.º 3, p. 43, jul. 2021, doi: 10.3390/designs5030043.

[18] P. John Clarkson y R. Coleman, «History of inclusive design in the UK», *Applied Ergonomics*, vol. 46, n.º PB, pp. 235-247, 2015, doi: 10.1016/j.apergo.2013.03.002.

[19] E. Gerber, «Design for America: organizing for civic innovation», *interactions*, vol. 21, n.º 2, pp. 42-47, mar. 2014, doi: 10.1145/2567840.

[20] J. Kolko, «Austin center for design www.ac4d.com», *Interactions*, vol. 20, n.º 3, pp. 84-87, 2013, doi: 10.1145/2451856.2451877.

[21] W. V. Siricharoen, «Using Empathy Mapping in Design Thinking Process for Personas Discovering», *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 343, pp. 182-191, 2021, doi: 10.1007/978-3-030-67101-3_15.
[22] P. Fehér y K. Varga, «The Value of Customer Journey Mapping and Analysis in Design Thinking Projects», *Lecture Notes in Business Information Processing*, vol. 361, pp. 333-336, 2019, doi: 10.1007/978-3-030-30429-4_25.
[23] J. M. Carraro y Y. Duarte, *Diseño de Experiencia de Usuario (UX)*. Autores de Argentina, 2015.
[24] «Thinkers Co». Accedido: 31 de octubre de 2021. [En línea]. Disponible en: https://thinkersco.com/
[25] D. Sánchez Huerta, *ANÁLISIS FODA O DAFO*. Bubox Publishing S.L., 2020.
[26] «Design Thinking Services». Accedido: 31 de octubre de 2021. [En línea]. Disponible en: https://www.designthinking.services/
[27] M. J. Hamm, *Wireframing essentials: an introduction to user experience design ; learn the fundamentals of designing the user experience for applications and websites*. Birmingham Mumbai: Packt Publishing, 2014.
[28] S. Wahid, D. S. McCrickard, J. DeGol, N. Elias, y S. Harrison, «Don't drop it!: pick it up and storyboard», en *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver BC Canada, may 2011, pp. 1571-1580. doi: 10.1145/1978942.1979171.
[29] «IBM Design Thinking». Accedido: 6 de noviembre de 2021. [En línea]. Disponible en: https://www.ibm.com/design/thinking/
[30] M. Schrage, *Serious Play: How the World's Best Companies Simulate to Innovate*. Harvard Business Review Press, 1999.
[31] F. Brooks Jr, «No Silver Bullet Essence and Accidents of Software Engineering», *IEEE Computer*, vol. 20, pp. 10-19, abr. 1987, doi: 10.1109/MC.1987.1663532.
[32] C. Hass, *A Practical Guide to Usability Testing*. Springer, Cham, 2019.
[33] «UserZoom, User research, UX testing solution». Accedido: 6 de noviembre de 2021. [En línea]. Disponible en: https://www.userzoom.com
[34] «Loop11, Usability Testing Tool». Accedido: 6 de noviembre de 2021. [En línea]. Disponible en: www.loop11.com
[35] «UserTesting, Human Insight Platform». Accedido: 6 de noviembre de 2021. [En línea]. Disponible en: https://www.usertesting.com
[36] N. Balacheff, S. Ludvigsen, T. de de Jong, A. Lazonder, y S. Barnes, *Technology-Enhanced Learning*. 2009.
[37] L. P. Prieto, M. H. Dlab, I. Gutié, N. A. rrez, M. Abdulwahed, y W. Balid, «Orchestrating technology enhanced learning: a literature review and a conceptual framework», *IJTEL*, vol. 3, n.º 6, p. 583, 2011, doi: 10.1504/IJTEL.2011.045449.
[38] P. Dillenbourg, «Design for classroom orchestration», *Computers & Education*, vol. 69, pp. 485-492, nov. 2013, doi: 10.1016/j.compedu.2013.04.013.
[39] J. Roschelle, Y. Dimitriadis, y U. Hoppe, «Classroom orchestration: Synthesis», *Computers & Education*, vol. 69, pp. 523-526, nov. 2013, doi: 10.1016/j.compedu.2013.04.010.
[40] K. Beck *et al.*, «Manifesto for Agile Software Development». 2001. Accedido: 13 de noviembre de 2021. [En línea]. Disponible en: http://AgileManifesto.org
[41] P. Abrahamsson, O. Salo, y J. Ronkainen, «Agile Software Development Methods: Review and Analysis», p. 113, 2002.
[42] University of Houston-Clear Lake,Houston, Texas, USA., M. Sameen Mirza, y S.

Datta, «Strengths and Weakness of Traditional and Agile Processes - A Systematic Review», *JSW*, vol. 14, n.º 5, pp. 209-219, may 2019, doi: 10.17706/jsw.14.5.209-219.
[43] «Atlassian Agile Coach». Accedido: 13 de noviembre de 2021. [En línea]. Disponible en: https://www.atlassian.com/es/agile
[44] H. Kniberg y A. Ivarsson, «with Tribes, Squads, Chapters & Guilds», p. 14.
[45] S. IEEE, *IEEE Software Engineering Standards (ANSI/IEEE Std; V. 729, 730, 828, 829, 830)*. John Wiley & Sons Inc.
[46] T. Theunissen, U. van Heesch, y P. Avgeriou, «A mapping study on documentation in Continuous Software Development», *Information and Software Technology*, vol. 142, p. 106733, feb. 2022, doi: 10.1016/j.infsof.2021.106733.
[47] «Amazon Web Services Continuous Integration». Accedido: 13 de noviembre de 2021. [En línea]. Disponible en: https://aws.amazon.com/es/devops/continuous-integration/
[48] «Redhat - What is CI-CD». Accedido: 13 de noviembre de 2021. [En línea]. Disponible en: https://www.redhat.com/es/topics/devops/what-is-ci-cd
[49] «Gitflow workflow». Accedido: 17 de noviembre de 2021. [En línea]. Disponible en: https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow
[50] A. Osterwalder, Y. Pigneur, G. Bernarda, y A. Smith, *Value Proposition Design*. John Wiley & Sons.
[51] «Figma, herramienta de generación de prototipos». Accedido: 24 de noviembre de 2021. [En línea]. Disponible en: https://www.figma.com/
[52] «Sketch, editor de gráficos vectoriales». Accedido: 24 de noviembre de 2021. [En línea]. Disponible en: https://www.sketch.com/
[53] «Adobe XD, editor de gráficos vectoriales para diseñar prototipos de experiencia de usuario». Accedido: 24 de noviembre de 2021. [En línea]. Disponible en: https://www.adobe.com/la/products/xd.html
[54] «Zeplin, espacio de trabajo para publicar diseños». Accedido: 24 de noviembre de 2021. [En línea]. Disponible en: https://zeplin.io/
[55] «Miro, online whiteboard tool». Accedido: 24 de noviembre de 2021. [En línea]. Disponible en: https://miro.com/
[56] «Jira, gestión de proyectos ágiles». Accedido: 24 de noviembre de 2021. [En línea]. Disponible en: https://www.atlassian.com/es/software/jira
[57] D. Bertram, «Likert Scale», p. 11.
[58] «Node Js». Accedido: 25 de noviembre de 2021. [En línea]. Disponible en: https://nodejs.org/es/
[59] «Swagger, OpenAPI specification». Accedido: 25 de noviembre de 2021. [En línea]. Disponible en: https://swagger.io/
[60] «Mysql - base de datos relacional». [En línea]. Disponible en: https://www.mysql.com/
[61] «REDIS in-memory data structure». Accedido: 26 de noviembre de 2021. [En línea]. Disponible en: https://redis.io/
[62] S. Wright, «Design Council», *Paper Europe*, vol. 10, n.º 10. p. 27, 1999. Accedido: 23 de octubre de 2021. [En línea]. Disponible en: https://www.designcouncil.org.uk
[63] L. Vinet y A. Zhedanov, «Design Sprint». doi: 10.1088/1751-8113/44/8/085201.