



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE CIENCIAS EMPRESARIALES

ESCUELA DE INGENIERÍA CIVIL INFORMÁTICA

30 de Agosto, 2021

Concepción, Chile

**Desarrollo de una solución tecnológica a empresas
Essbio, utilizando la metodología Design-Led
Development (DLD) y las herramientas Build y
Business Technology Platform (BTP) de SAP.**

Proyecto de Título presentado en conformidad a los requisitos para obtener el título de Ingeniero
Civil en Informática

Patricio Pardo Novoa

Profesora Guía:

PhD. Elizabeth Grandón Toledo

Agradecimientos

En primer lugar, agradezco a la profesora guía PhD Elizabeth Grandón Toledo, flamante directora del nuevo Doctorado en Economía y Gestión de la Información de la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío, por su excelente disposición como profesional y como persona. Gracias a ella se formó el vínculo con la empresa en donde se me abrieron las puertas para desarrollar el proyecto, constantemente aportó con sus conocimientos y experiencia en la investigación, y lo más importante, me apoyó a pesar del fracaso e incentivó a no rendirme. Por esto y más, ha sido el pilar fundamental en la última parte de mi formación de pregrado.

En segundo lugar debo agradecer a empresa Essbio, específicamente al Jefe del Departamento de Sistemas Operacionales y BI, Carlos Ceballos Pérez, por compartir uno de los proyectos en los que actualmente hoy la empresa se mantiene interesada. Gracias a él pude ingresar a la empresa y generar los lazos con las personas indicadas para poder ejecutar el proyecto satisfactoriamente.

En tercer lugar agradezco a mis padres, por siempre haber confiado en mí y mi capacidad, son ellos los que han mantenido motivado mi constante esfuerzo y auto superación. Gracias por darme la mejor educación que pudieron, tanto escolar como en casa. Los amo a los dos de corazón, gracias a ustedes soy la persona que soy.

En tercer lugar agradezco a mi novia y compañera, con quien he convivido los últimos años de mi vida. Es la persona con la que he compartido mis éxitos y mis fracasos, y mi soporte emocional más fuerte. Gracias por tu apoyo incondicional, te amo.

Resumen

Normalmente, al desarrollar software, los equipos utilizan metodologías tradicionales o ágiles. Las metodologías ágiles actualmente están al alza y cada vez son más utilizadas a la hora de llevar a cabo un proyecto de desarrollo de software. Sin embargo, en busca de adecuarse cada vez más al concepto de experiencia de usuario y generar soluciones significativas, los desarrolladores de software experimentan con metodologías basadas en diseño.

La metodología basada en diseño usada en este proyecto es el Design-Led Development o DLD, el cual sigue los lineamientos del Design Thinking. Estas metodologías buscan generar ideas innovadoras, centran sus esfuerzos en entender y dar solución a las necesidades reales de los usuarios, y por esta razón, su principal característica es centrarse en el humano, en donde el equipo del proyecto realiza un trabajo colaborativo.

En vinculación con una reconocida empresa de servicios sanitarios de la región, se realizó un proyecto de coinnovación con el objetivo de disminuir el tiempo que les requiere a los operadores de planta subir datos de mediciones de parámetros al sistema de información que apoya su proceso. El resultado de este proyecto, que se ajustó al tiempo limitado para su ejecución, es la implementación de un prototipo del sistema de toma de mediciones en terreno usando la herramienta SAP Build, la cual basa sus diseños en SAPUI5 de Fiori, el que posteriormente fue subido y ejecutado en el entorno de desarrollo integrado web, proporcionado por SAP Business Technology Platform para su posterior finalización.

A modo de conclusión, como primera experiencia en un proyecto de desarrollo aplicando una metodología basada en diseño, se destaca la vinculación con los usuarios finales durante las etapas y el uso de la herramienta SAP Build para iniciar un desarrollo rápido. Esta herramienta permite respaldar los entregables del procedimiento de la metodología, convirtiéndose en una ventaja a la hora de desempeñarse en el área de desarrollo en entornos SAP.

Abstract

Typically, when developing software, work teams use traditional or agile methodologies. Agile methodologies are currently on the rise and are increasingly used when carrying out a software development project. However, in search of increasingly adapting to the user experience and generating meaningful solutions, software developers experiment with design-based methodologies.

The design-based methodology selected in this project is Design-Led Development or DLD, which follows the guidelines of Design Thinking. These methodologies seek to generate innovative ideas, focus their efforts on understanding and solving the real needs of users, and for this reason, their main characteristic is to focus on the human, where the project team performs collaborative work.

In conjunction with a renowned services company in the region, a co-innovation project was carried out with the aim of reducing the time it takes for plant operators to upload parameter measurement data to the information system that supports their process. The result of this project was adjusted to the limited time for its execution, and it was possible to implement a prototype of the system for taking measurements in the field using the SAP Build tool that bases its designs on SAPUI5 Fiori, and upload it to the web integrated development environment provided by Business Technology Platform for later completion.

In conclusion, as the first experience in a development project applying a design-based methodology such as DLD, the link with the end users during all stages is highlighted, and using the Build tool to start a rapid development and support the deliverables of the procedure of the methodology, becomes an advantage when working in the SAP environment development area.

Índice General

ÍNDICE DE TABLAS.....	8
ÍNDICE DE FIGURAS.....	9
GLOSARIO DE CONCEPTOS Y SIGLAS.....	11
1. INTRODUCCIÓN.....	13
1.2 OBJETIVO GENERAL.....	14
1.3 OBJETIVOS ESPECÍFICOS.....	14
2. MARCO TEÓRICO.....	15
2.1 METODOLOGÍAS DE DESARROLLO SOFTWARE.....	15
2.1.1 METODOLOGÍAS TRADICIONALES.....	15
2.1.1.1 Lineales.....	16
2.1.1.2 Incrementales.....	17
2.1.1.3 Evolutivas.....	18
2.1.2 METODOLOGÍAS ÁGILES.....	19
2.1.2.1 Manifiesto Ágil.....	20
2.1.2.2 Scrum.....	21
2.1.2.3 Programación Extrema (XP).....	23
2.1.2.4 Kanban.....	25
2.1.2.5 Lean.....	27
2.1.3 USO DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE EN CHILE.....	28
2.1.4 COMPARACIÓN ENTRE METODOLOGÍAS TRADICIONALES Y ÁGILES.....	29
2.2 DESIGN THINKING.....	34
2.2.1 EVOLUCIÓN.....	36

2.2.2 PREMISAS DEL PROCESO CREATIVO.....	37
2.2.3 FASES	38
2.2.3.1 Empatía	41
2.2.3.2 Definir.....	42
2.2.3.3 Idear	42
2.2.3.4 Prototipar	43
2.2.3.5 Evaluar.....	44
2.2.4 DESIGN THINKERS.....	45
2.3 DESIGN-LED DEVELOPMENT	46
2.3.1 DESCUBRIR	47
2.3.1.1 Alcance	47
2.3.1.2 Investigar.....	48
2.3.1.3 Sintetizar	49
2.3.2 DISEÑAR	52
2.3.2.1 Idear	52
2.3.2.2 Prototipar	56
2.3.2.3 Validar	58
2.3.3 DESARROLLAR	59
2.3.3.1 Implementar.....	59
2.3.3.2 Probar.....	60
2.3.3.4 Desplegar	61
2.4 RESUMEN DE LA METODOLOGÍA.....	62
2.5 TRABAJOS E INVESTIGACIONES SIMILARES	63
<u>3. CONTEXTO DEL DESAFÍO.....</u>	<u>65</u>
3.1 DATOS GENERALES DE LA EMPRESA.....	65
3.1.1 MISIÓN.....	65
3.1.2 VISIÓN	65
3.1.3 VALORES CORPORATIVOS.....	66
3.1.4 ORGANIGRAMA GENERAL	66

3.2 GERENCIA DE FINANZAS Y CONTROL DE GESTIÓN	68
3.2.1 DATOS GENERALES SUBGERENCIA DE INFORMÁTICA	70
3.3 PROBLEMÁTICA	72
3.3.1 DESCRIPCIÓN GENERAL.....	72
3.3.2 ENTORNO TECNOLÓGICO.....	73
3.3.3 OBJETIVO EMPRESARIAL DEL PROYECTO	77
<u>4. DESARROLLO DE UNA SOLUCIÓN</u>	<u>78</u>
4.1 DESCUBRIR.....	80
4.1.1 ALCANCE	80
4.1.2 INVESTIGAR.....	83
4.1.3 SINTETIZAR	84
4.2 DISEÑAR.....	91
4.2.1 IDEAR.....	91
4.2.2 PROTOTIPAR	94
4.2.3 VALIDAR.....	103
4.3 DESARROLLAR.....	109
4.3.1 IMPLEMENTAR.....	109
4.3.2 PROBAR	110
4.3.3 DESPLEGAR	111
<u>5. DISCUSIÓN Y CONCLUSIONES</u>	<u>115</u>
5.1 ANÁLISIS DE OBJETIVOS.....	115
5.2 PRINCIPALES APORTES	117
5.3 TRABAJOS FUTUROS	119
5.5 EXPERIENCIA GENERAL.....	120
<u>6. BIBLIOGRAFÍA</u>	<u>122</u>
<u>7. ANEXOS</u>	<u>124</u>

7.1 ANEXO 1 – DOCUMENTO DE ALCANCE DE PROYECTO	124
7.2 ANEXO 2 – SCRIPT ENTREVISTA	125
7.3 ANEXO 3 – HALLAZGOS POR USUARIO	128
7.2 ANEXO 4 – HALLAZGOS GENERALES.....	130
7.5 ANEXO 5 – PERSONA.....	131
7.6 ANEXO 6 – STORYBOARDS	132
7.7 ANEXO 7 – CASOS DE USO	135
7.8 ANEXO 8 – BOCETOS	141
7.9 ANEXO 9 – MAPA DE NAVEGACIÓN DEL PROTOTIPO DE BAJA FIDELIDAD	144
7.10 ANEXO 10 – CAPTURAS.....	145
7.11 ANEXO 11 – MAPA DE NAVEGACIÓN DEL PROTOTIPO DE ALTA FIDELIDAD	148
7.12 ANEXO 12 – CAPTURAS PROTOTIPO DE ALTA FIDELIDAD EN VIVO	149
7.13 ANEXO 13 – CONFIRMACIÓN DE PARTICIPACIÓN EN EL CURSO EN OPEN\$AP.....	152

Índice de Tablas

TABLA 1 - INFORMACIÓN DEMOGRÁFICA ENCUESTADOS SOBRE USO DE METODOLOGÍAS DE DESARROLLO (GRANDÓN ET. AL., 2018).	29
TABLA 2 - COMPARACIÓN DE ASPECTOS CLAVES ENTRE METODOLOGÍAS TRADICIONALES VS METODOLOGÍAS AGILES (NAVARRO, FERNÁNDEZ & MORALES, 2013)	30
TABLA 3 - DIFERENCIAS ENTRE METODOLOGÍAS DE DESARROLLO TRADICIONAL Y ÁGIL (STOICA, MIRCEA, & GHILIC-MICU, 2013).....	33
TABLA 4 - DEFINICIONES DE DESIGN THINKING (SCHALLMO, WILLIAMS, & LANG, 2018)	35
TABLA 5 - ÁREAS EJECUTORAS Y ÁREA DE NEGOCIOS	72
TABLA 6 – RIESGOS, MITIGACIONES Y CONTINGENCIAS	79
TABLA 7 – PLANIFICACIÓN INICIAL DEL PROYECTO	80
TABLA 8 - REGISTRO DE ENTREVISTAS	83
TABLA 9 - AGRUPACIÓN GENERAL DE HALLAZGOS.....	86
TABLA 10 - PARTES DEL ARTEFACTO PERSONA.....	89
TABLA 11 - LISTA DE CASOS DE USO.....	93
TABLA 12 - MAPEO DE HISTORIAS DE USUARIO.....	109
TABLA 13 - OBJETIVOS INICIALES VS CUMPLIMIENTO	116

Índice de Figuras

FIGURA 1 - MODELO CASCADA (SOMMERVILLE, 2010).....	16
FIGURA 2 - MODELO INCREMENTAL (SOMMERVILLE, 2010)	17
FIGURA 3 - MODELO ESPIRAL (SOMMERVILLE, 2010)	18
FIGURA 4 - IMAGEN USADA AL LEVANTAR EL SITIO WEB WWW.AGILEMANIFIESTO.ORG (2001)	20
FIGURA 5 - ESQUEMA PROCEDIMIENTO SCRUM (NAVARRO, FERNÁNDEZ, & MORALES, 2013)	22
FIGURA 6 - EJEMPLO DE TABLERO KANBAN.....	26
FIGURA 7 - GRÁFICO DE USO DE METODOLOGÍAS EN CHILE (GRANDON, CAMPOS, FICA, & ÁVILA, 2018).....	28
FIGURA 8 - PROCEDIMIENTO DESIGN THINKING (BROWN, 2008)	39
FIGURA 9 - PROCEDIMIENTO DESIGN THINKING (PLATTNER, MEINEL, & WEINBERG, 2009).....	40
FIGURA 10 – PROCEDIMIENTO DESIGN THINKING (D.SCHOOL, 2010)	41
FIGURA 11 - ESQUEMA DEL DLD: ETAPAS Y SUB-ETAPAS. FUENTE: HTTPS://WWW.BUILD.ME/LEARNING (ADAPTADO)	46
FIGURA 12 - DIAGRAMA DE PROCEDIMIENTOS DESIGN-LED DEVELOPMENT DLD (ELABORACIÓN PROPIA)	62
FIGURA 13 - ORGANIGRAMA GENERAL DE ESSBIO.	66
FIGURA 14 - DEFINICIONES DE ÉXITO DE LA ESTRATEGIA ESSBIO (2021).	67
FIGURA 15 - MAPA ESTRATÉGICO ESSBIO (2021).	68
FIGURA 16 – ORGANIGRAMA GERENCIA DE FINANZAS Y CONTROL DE GESTIÓN ESSBIO.....	68
FIGURA 17 - ORGANIGRAMA SUBGERENCIA DE INFORMÁTICA ESSBIO.....	70
FIGURA 18 - VISIÓN TRANSVERSAL: ALINEAMIENTO ESTRATÉGICO. SUBGERENCIA DE INFORMÁTICA ESSBIO.....	71
FIGURA 19 - PORTAL DEL SISTEMA WEB SAP BUILD.....	73
FIGURA 20 - ESPACIO DE TRABAJO DE SAP BUILD	74
FIGURA 21 - CAPTURA DE UN DIRECTORIO EN SAP BUSINESS APPLICATION STUDIO.....	75
FIGURA 22 - CREACIÓN DE CUENTA EN SAP BUILD.....	94
FIGURA 23 - VERIFICACIÓN DE CORREO SAP BUILD.....	95
FIGURA 24 - ESPACIO DE TRABAJO PROYECTO ESSBIO ES BUILD.....	95
FIGURA 25 - ARTEFACTOS EN EL ESPACIO DE TRABAJO SAP BUILD.....	97
FIGURA 26 - BOCETOS PORTAL DE ACCESO Y MENÚ PRINCIPAL	98
FIGURA 27 - MAPA DE NAVEGACIÓN DEL PROTOTIPO DE BAJA FIDELIDAD EN BUILD	99
FIGURA 28 – SELECCIÓN DE MARCO RESPONSIVO.....	100
FIGURA 29 - CREACIÓN PROTOTIPO DE ALTA FIDELIDAD EN BUILD	100

FIGURA 30 - MAPA DE NAVEGACIÓN DEL PROTOTIPO DE ALTA FIDELIDAD EN BUILD	102
FIGURA 31 - INICIO DE LA CREACIÓN DE UN ESTUDIO EN BUILD	103
FIGURA 32- GENERACIÓN DE ESTUDIO EN BUILD.	104
FIGURA 33 - SELECCIÓN DE FORMATO PARA EL ESTUDIO.	104
FIGURA 34 - CONFIGURACIÓN DE PREGUNTA PARA ESTUDIO EN BUILD	105
FIGURA 35 - ACCESO AL ESTUDIO DE PRUEBA	105
FIGURA 36 - PARTICIPACIÓN EN EL ESTUDIO CONFIGURADO EN BUILD	106
FIGURA 37 - EJEMPLO DE UN COMENTARIO EN UN ESTUDIO	106
FIGURA 38 - FINALIZACIÓN DEL ESTUDIO EN BUILD.....	107
FIGURA 39 - RESULTADOS DEL ESTUDIO EN BUILD	107
FIGURA 40 - COMENTARIO REALIZADO EN EL MAPA DE NAVEGACIÓN DEL PROTOTIPO DE ALTA FIDELIDAD EN BUILD	108
FIGURA 41 - DESCARGA DEL CÓDIGO DEL PROTOTIPO DE ALTA FIDELIDAD DE BUILD	111
FIGURA 42 - SAP BUSINESS TECHNOLOGY PLATFORM.	112
FIGURA 43 - CREACIÓN DE ESPACIO DE DESARROLLO EN SAP BUSINESS APPLICATION STUDIO	112
FIGURA 44 - IMPORTACIÓN DEL PROYECTO BUILD AL BUSINESS APPLICATION STUDIO	113
FIGURA 45 - DIRECTORIO PROYECTO BUILD	113
FIGURA 46 - PORTAL DE LA APLICACIÓN EN VIVO.....	114
FIGURA 47 - MAPA GLOBAL DE D.SCHOOL (OBTENIDO DE LA WEB OFICIAL DE LA UNIVERSIDAD MAYOR)	118
FIGURA 48 - CURSO DE DESIGN THINKING Y BUILD EN OPENSAP.COM.....	120
FIGURA 49 - AVANCE DEL CURSO EN OPENSAP.COM	121

Glosario de conceptos y siglas

- DLD: o Design-Led Development es la metodología basada en diseño utilizada para la ejecución de este proyecto. Es una adaptación del Design Thinking enfocada en el desarrollo de software y propuesta por SAP para el uso de la herramienta Build.
- DT: o Design Thinking es una metodología basada en diseño impulsada por la Universidad de Stanford y el instituto Hasso Plattner.
- BTP: o Business Technology Platform es una de las herramientas de la suite de SAP y permite configurar espacios de trabajo para realizar proyectos informáticos en entornos SAP. Esta herramienta incluye un útil entorno de desarrollo integrado llamado Business Application Studio.
- SAP: viene de la traducción al alemán de “Desarrollo de sistemas de análisis” y es uno de los principales productores mundiales de software para gestión de procesos de negocio, y desarrolla soluciones que facilitan el procesamiento eficaz de datos y el flujo de información entre las organización (definición tomada de la página principal de SAP).
- Experiencia de Usuario (UX): este término engloba todas las interacciones de un usuario con una marca. Se puede ver como un conjunto de factores y elementos relativos a las interacciones del usuario con un entorno o dispositivo, dando como resultado la percepción positiva o negativa de dicho producto o servicio.
- Software: conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.
- Metodología: conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.
- Coinnovación: es la cooperación y colaboración para innovar y es un elemento crucial de creación de valor para cualquier organización.
- Operadores: trabajadores en terreno de la empresa que verifican la operatividad de las maquinarias de AP y AS.
- AP: producción de agua potable.
- AS: producción de aguas servidas.
- Planta: espacio físico que contiene las maquinarias revisadas por los operadores.

- SAP UI5: es el nueva parte gráfica desarrollada por SAP, y que mediante la combinación de las tecnologías HTML5, CSS3 y JavaScript, tiene como objetivo principal mejorar la interfaz de usuario y experiencia de usuario.
- Fiori: es la nueva experiencia de usuario propuesta por SAP. Se basa en la utilización de los componentes de SAP UI5.
- IDE: o entorno de desarrollo integrado es un sistema de software para el diseño de aplicación que combina herramientas para desarrolladores en una sola interfaz gráfica y está compuesto por un editor de código fuente, un compilador y un depurador.
- Ágil: enfoque iterativo de la gestión de proyectos y desarrollo de software. Es visto como una filosofía que supone una forma distinta de trabajar y organizarse a las antes establecidas.
- Product Owner: o dueño del producto, es el profesional encargado de maximizar el valor de una solución generada y es el representante del cliente en el equipo.
- Prototipo: o Wireframe, es un boceto donde se representa visualmente, de una forma sencilla y esquemática, la estructura de una vista de una web.
- HMW: significa “How Might We” y es una técnica utilizada para reformular sentencias.

1. Introducción

Hoy en día, los sistemas de información son una parte esencial de la cotidianidad. En conjunto con las tecnologías de la información y comunicación, han extendido sus redes a gran parte de la superficie del planeta, incluyendo las profundidades del océano ^[1], la atmósfera e incluso el espacio exterior ^[2].

Cuesta trabajo pensar en una persona que no use dispositivos móviles, un Smart TV o que no esté registrado en ninguna red social o sistema informático. Según la SUBTEL (Subsecretaría de Telecomunicación, organismo dependiente del Ministerio de Transportes y Telecomunicaciones de Chile), los número de teléfonos móviles vigentes, ascienden a una cifra de 26 millones, y solo el año 2019 fueron habilitados 1,5 millones de nuevos puntos de acceso a internet ^[3].

Los sistemas que frecuentemente usamos poseen características de valor; cómo ser responsivos, móviles, multiplataforma, orientados a la experiencia de usuario, multitarea, con inteligencia artificial, entre otras. Sin embargo, el recorrido para llegar hasta acá ha sido largo y no exento de complicaciones.

Inicialmente, el desarrollo de sistemas informáticos fue rústico e intuitivo, pero la ingeniería de software y metodologías de desarrollo mejoraron este proceso en eficacia y eficiencia. Lamentablemente, esto no fue suficiente, ya que según los reportes de caos del Standish Group (Chaos Report, 2015), entre el año 2011 y 2015, en promedio solo 39% de los proyectos de desarrollo de software fue exitoso, mientras que el resto fueron desafiados o abandonados. Este indicador responde a la tónica que se ha repetido desde el inicio del reporte del caos el año 1994, por lo que el año 2001 fue publicado el manifiesto ágil como respuesta a esta problemática, en el que los creadores de las principales metodologías definieron los valores y principios de la filosofía ágil. Actualmente, el desarrollo de software ha sido adaptado y probado en metodologías de diseño como el Design Thinking. El objetivo de estas adaptaciones es generar productos significativos, útiles y validados por los propios usuarios antes de su lanzamiento, reduciendo costos por fallas de diseño.

El capítulo 2 de este documento, que corresponde al marco teórico, aborda la evolución mencionada con anterioridad, respaldándose de una revisión de literatura. La información recopilada se centra en; las

¹https://www.nationalgeographic.com.es/ciencia/internet-red-miles-cables-submarinos_13997

²https://www.bbc.com/mundo/ciencia_tecnologia/2010/01/100122_0129_internet_nasa_twitter_jrg

³<https://www.subtel.gob.cl/usuarios-de-internet-movil-consumen-10-gb-en-promedio-al-mes-y-numero-de-telefonos-moviles-sube-a-26-millones/>

metodologías tradicionales y ágiles; el Design Thinking y Design-Led Development; y algunas comparaciones relevantes entre grupos de metodologías.

El capítulo 3 presenta las características organizacionales de la empresa en donde se desarrolló este proyecto, y en particular, del departamento que lo impulsó. Además, se describe el problema a enfrentar, el entorno tecnológico utilizado y el objetivo empresarial que busca alcanzar el proyecto. En este capítulo se destaca la principal herramienta que apoyó el proceso de desarrollo: SAP Build.

El capítulo 4 detalla cada etapa de la ejecución del proyecto utilizando la metodología Design-Led Development, propuesta por SAP para el uso de la herramienta Build. Finalmente, el capítulo 5 contiene la discusión y conclusiones que se derivan del proyecto, se destacan los principales aportes y se propone un camino a seguir.

1.2 Objetivo General

Entregar una solución tecnológica para la empresa Essbio, la cual consiste en el desarrollo de una aplicación con interfaz Fiori, a través de la metodología DLD (Design-Led Development) que cuenta con lineamiento de DT (Design Thinking), usando las herramientas de SAP: Build y Business Technology Platform (BTP).

1.3 Objetivos Específicos

- Investigar los fundamentos de las metodologías DLD y DT; sus pasos, el valor de su uso y sus puntos de convergencia.
- Aplicar la metodología DLD, para el desarrollo de una solución tecnológica de Software, generando prototipos de baja y alta fidelidad por medio de la herramienta de SAP: Build, los que serán desafiados por sus futuros usuarios antes de su aceptación.
- Implementar funcionalidades sobre prototipos aceptados, exportados desde la herramienta Build, usando el entorno de desarrollo integrado proporcionado por BTP.
- Desplegar el producto de software a producción en un entorno SAP.

2. Marco Teórico

Este capítulo contiene el marco teórico, el cual contextualiza la selección de la metodología utilizada en el desarrollo del proyecto, y además sintetiza los hallazgos encontrados al consultar en la literatura.

Primero, se describe el concepto de metodología de desarrollo de software, para luego mencionar las metodologías tradicionales y ágiles. Después, se presenta una metodología basada en diseño llamada Design Thinking, que tiene características útiles y eficientes a la hora de generar productos o servicios, lo que es perfecto para desarrollar software.

Este capítulo finaliza describiendo la metodología seleccionada para este proyecto: Design-Led Development, recomendada por SAP para el uso de su herramienta Build.

2.1 Metodologías de Desarrollo Software

Las metodologías de desarrollo de software proponen un procedimiento disciplinado sobre el desarrollo de software, dan un enfoque sistemático y están compuestas por un conjunto de guías, actividades, técnicas y herramientas basadas en una filosofía de desarrollo (Grandón, et al., 2018).

Hoy en día, las metodologías previas a las ágiles se denominan tradicionales, ambos tipos de metodologías de desarrollo de software son descritos a continuación.

2.1.1 Metodologías Tradicionales

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de licitación de requerimientos, previo a etapas de análisis y diseño. El objetivo es asegurar resultados con alta calidad, relacionados a una planificación calendarizada. En estas metodologías se concibe un solo proyecto con una estructura definida; se sigue un proceso secuencial en una sola dirección; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado el proceso (Khurana & Sohal, 2011). Las metodologías tradicionales se pueden dividir en 3 grupos: lineales, incrementales y evolutivas (Grandón, et al., 2018).

2.1.1.1 Lineales

Dentro del grupo de metodologías lineales se encuentran las basadas en el modelo cascada, que siguen una secuencia ordenada de pasos que guían el desarrollo. En 1970 Royce (Royce, 1970) definió el modelo cascada y en 1976 Boehm (Boehm, 1988) mejoró el modelo para hacer frente a la complejidad de los proyectos de desarrollo de software. Sommerville (Sommerville, 2010) define el modelo cascada, como una metodología que toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, para luego representarlo como fases separadas del proceso, tales como; especificación de requerimientos, diseño de software, implementación, pruebas, entre otras. La Figura 1 esquematiza un modelo cascada más perfeccionado, que cuenta con la capacidad de retroceder etapas.

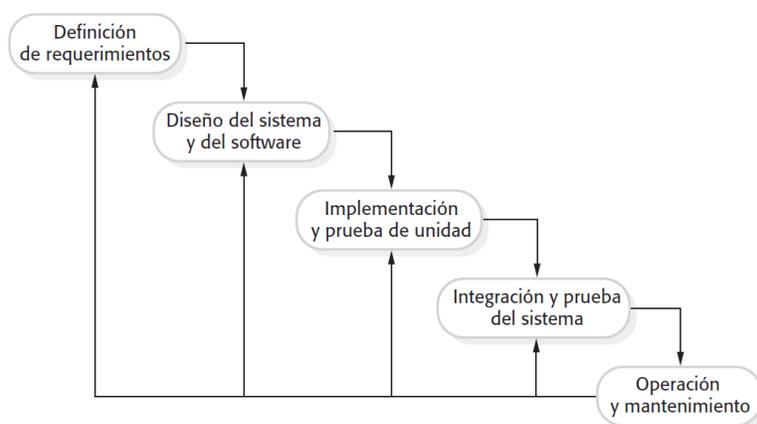


Figura 1 - Modelo cascada (Sommerville, 2010)

Este tipo de modelos permitió a los administradores de proyectos realizar seguimientos y revisar el progreso del desarrollo con exactitud. Además, se exige que el proceso de desarrollo genere una serie de documentos, los cuales son utilizados para testear y mantener el sistema. Dentro de las variantes del modelo cascada existe la cascada con iteración, cascada con subproyectos, Structured System Analysis and Design Method (SSADM) y el modelo Case de Oracle (Grandón, et al., 2018).

2.1.1.2 Incrementales

El grupo de metodologías incrementales busca construir una implementación parcial de todo el sistema y lentamente agregar mayor funcionalidad o rendimiento (Davis, Bersoff & Comer, 1988). Las metodologías incrementales o iterativas provienen del trabajo de Shewhart, quien propuso un ciclo que posteriormente fue denominado con su mismo apellido (Larman & Basili, 2003) y se resume en 4 partes que iteran; “Planifique-Haga-Chequee-Actué”. Esto produce un producto operacional más rápido y reduce la probabilidad de que las necesidades del usuario cambien durante el proceso de desarrollo.

Sommerville (Sommerville, 2010) declara que el desarrollo incremental se basa en la idea de diseñar una implementación inicial, exponerla al comentario del usuario y luego desarrollar diversas versiones hasta producir un sistema adecuado. La Figura 2 esquematiza una versión estandarizada del modelo incremental. Se inicia con un bosquejo de descripción y se pasa al proceso de desarrollo, donde a través de la especificación de requerimientos se logra una versión inicial, con el desarrollo se logran versiones intermedias y se realizan validaciones para generar una versión final.

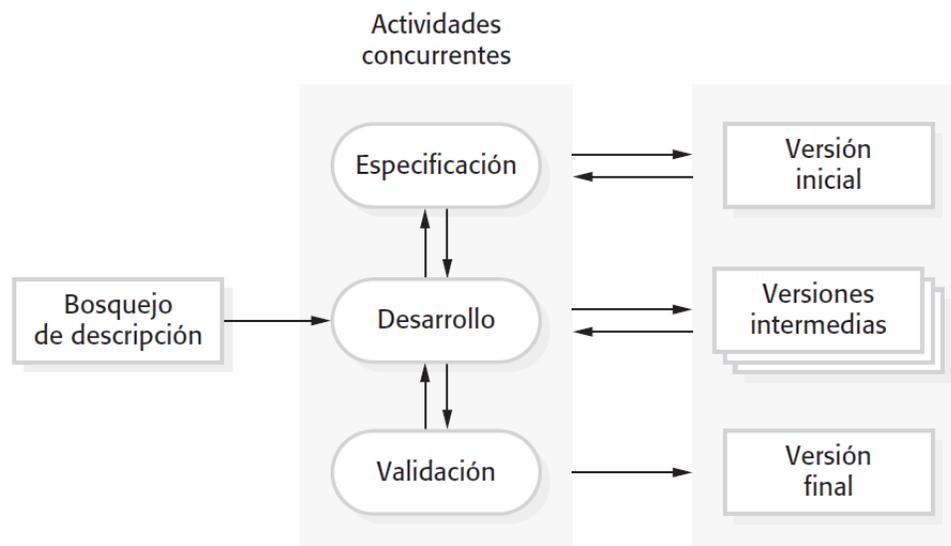


Figura 2 - Modelo incremental (Sommerville, 2010)

2.1.1.3 Evolutivas

La metodología que guía el grupo de evolutivas es el prototipado evolutivo. En este caso, los desarrolladores construyen una implementación parcial del sistema que cumple con requerimientos conocidos. Luego, el prototipo se utiliza por los usuarios para entender el resto de los requerimientos. Este tipo de metodologías alcanzó popularidad masiva en los años 80 con la aparición de desarrollo de sistemas de inteligencia artificial (Larman & Basili, 2003).

Dentro de este grupo de metodologías se encuentran las metodologías de entrega por etapas, modelo espiral y cascada con reducción de riesgo; siendo el modelo espiral el máximo exponente. Este modelo contiene 4 etapas definidas: determinación de objetivos, alternativas y restricciones; evaluación de alternativas e identificación y manejo de riesgos; desarrollo y verificación del próximo nivel del producto; y finalmente, planificación de las siguientes etapas (Boehm, 1988). El manejo de los riesgos es el elemento diferenciador de esta metodología. La Figura 3 esquematiza el modelo espiral basado en Boehm, propuesto por Sommerville.

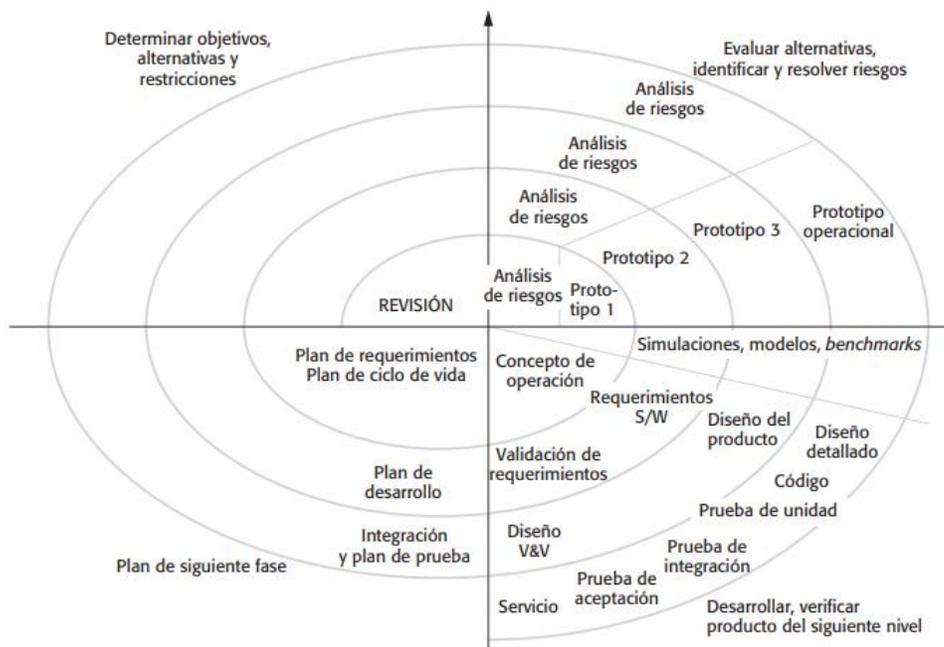


Figura 3 - Modelo Espiral (Sommerville, 2010)

2.1.2 Metodologías Ágiles

Las metodologías ágiles surgieron en los años noventa, inicialmente eran denominadas metodologías ligeras y su principal objetivo era reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software (Navarro, Fernández & Morales, 2013). Estas metodologías nacieron como respuesta a las metodologías existentes (tradicionales), con el propósito de disminuir la burocracia que implicaba la aplicación de las metodologías anteriores en proyectos de pequeña y mediana escala (Fowler, 2005).

De acuerdo a Abrahamsson (Abrahamsson, Warsta, Siponen & Ronkainen, 2003), las metodologías ágiles son flexibles, ya que pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto. Se caracterizan por el desarrollo iterativo e incremental; la simplicidad de la implementación; las entregas frecuentes; la priorización de los requerimientos o características a desarrollar a cargo del cliente; y la cooperación entre desarrolladores y clientes. Las metodologías ágiles dan como un hecho que los requerimientos van a cambiar durante el proceso de desarrollo.

Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante, al punto de requerir un representante de él durante el desarrollo. Estos proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio de requerimiento es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él, por lo que tanto el producto como el proceso son mejorados frecuentemente (Ghosh, 2015).

Este grupo de metodologías se basa en la suposición de que los requerimientos de los usuarios no se conocen desde el principio del desarrollo de un proyecto de software, y que la manera apropiada de entregar un producto de calidad y a tiempo, es proporcionar flexibilidad en las actividades de desarrollo (Cervone, 2011). Algunas de las metodologías ágiles más populares son: Scrum, Extreme Programming (XP), Dynamic Systems Development Method (DSMD), Adaptative Software Development (ASD), Crystal Clear, Feature Driven Development (FDD), Lean Software Development (LSD), Kanban, Open Up, entre otras.

2.1.2.1 Manifiesto Ágil

El año 2001 se creó el manifiesto por el desarrollo ágil de software. Este documento acuerda cuatro principios básicos del desarrollo de software para establecer prioridades y marcar diferencias de fondo frente a las metodologías tradicionales. Estos aspectos son (Beck, et al., 2001): “Se valora más...”

- A los individuos e interacciones, que los procesos y herramientas.
- El software funcionando, que una documentación extensiva.
- La colaboración con el cliente, que la negociación contractual.
- La respuesta ante el cambio, que seguir un plan.

Estos principios dan origen al manifiesto ágil e implican la satisfacción del cliente mediante; entregas tempranas y constantes de software que funcione, requerimientos cambiantes en cualquier etapa del proyecto, participación activa del cliente, simplicidad, equipos de desarrollo motivados y auto-organizados, comunicación efectiva, auto inspecciones y adaptación.

El manifiesto por el desarrollo ágil de software es el resultado del trabajo colaborativo de un grupo formado por diecisiete personas, entre desarrolladores de software, escritores y consultores, quienes lo construyeron y escribieron el 2001. La firma y publicación del manifiesto en ese año no implica que esa sea la fecha de origen de las metodologías ágiles o que antes de ese año no existieran, si no que representan un lineamiento común que deben seguir (Sommerville, 2010). La Figura 4 muestra la ilustración subida al sitio web www.agilemanifiesto.org al momento de levantar el sitio web. Esto se puede ver como un hito en el desarrollo de software, ya que a través de este sitio fueron declarados los principios y valores de la filosofía ágil.



Figura 4 - Imagen usada al levantar el sitio web www.agilemanifiesto.org (2001)

2.1.2.2 Scrum

La metodología Scrum es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas, artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento (Navarro, Fernández & Morales, 2013). Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas.

Los equipos Scrum cumplen con 3 características:

- **Son auto-gestionados**, lo que les permite elegir la mejor forma de hacer el trabajo, en vez de basarse en lineamientos de personas que no pertenecen al equipo y no están involucrados en el contexto.
- **Son multifuncionales**, puesto que los integrantes del equipo tienen todos los conocimientos necesarios para llevar a cabo el trabajo.
- **Trabajan en iteraciones**, en la cual van agregando nuevas funcionalidades o se modifican las cosas que el Product Owner requiere.

Scrum define tres roles, el Scrum Master, el Product Owner y el Equipo de Desarrollo (Navarro, Fernández & Morales, 2013). El Scrum Master tiene como función asegurar que el equipo está adoptando la metodología, sus prácticas, valores y normas; es el líder del equipo, pero no gestiona el desarrollo, es más bien un habilitador. Por su parte, el dueño del producto o Product Owner, es una sola persona y representa a los interesados como responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo. Entre sus funciones está gestionar la lista ordenada de funcionalidades requeridas, más conocida como Product Backlog. Finalmente, el Equipo de Desarrollo tiene como responsabilidad convertir lo que el cliente quiere, el Product Backlog, en iteraciones funcionales del producto. El equipo no tiene jerarquías, todos sus miembros tienen el mismo nivel y cargo. El tamaño recomendado del equipo de desarrollo es entre 3 y nueve personas. La Figura 5 muestra gráficamente los principales componentes de la metodología Scrum.

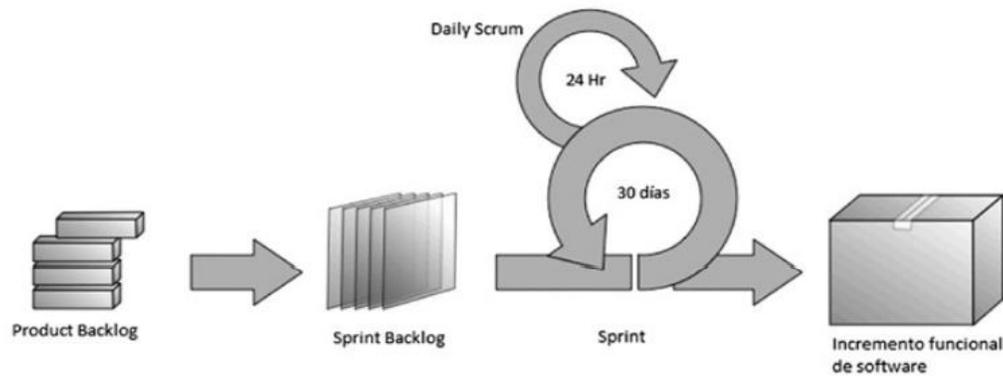


Figura 5 - Esquema Procedimiento Scrum (Navarro, Fernández, & Morales, 2013)

Scrum define un evento principal o Sprint, que corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada Sprint, es considerado como un proyecto independiente y su duración máxima es de un mes. Un Sprint se compone de los siguientes elementos; reunión de planeación del Sprint, Daily Scrum, trabajo de desarrollo, revisión del Sprint y retrospectiva del Sprint

La retrospectiva del Sprint es una reunión de tres horas del equipo Scrum en la que se analiza cómo fue la comunicación, el proceso y las herramientas; que estuvo bien, qué no, y se crea un plan de mejoras para el siguiente Sprint.

El Product Backlog es una lista, ordenada por valor, riesgo, prioridad y necesidad, de los requerimientos que el Product Owner define, actualiza y ordena. Esta lista tiene como característica particular que nunca está terminada, pues evoluciona durante el desarrollo del proyecto.

El Sprint Backlog es un subconjunto de ítems del Product Backlog y el plan para realizar en el incremento de producto. Debido a que el Product Backlog está organizado por prioridad, el Sprint Backlog es construido con los requerimientos más prioritarios del Product Backlog y con aquellos que quedaron por resolver en el Sprint anterior. El Sprint Backlog debe ser aceptado por el equipo. Si se detectan requerimientos adicionales deben ser incluidos en el Product Backlog y desarrollados en el siguiente Sprint, si su prioridad así lo indica.

El incremento es la suma de todos los ítems terminados en el Sprint Backlog. Si hay ítems incompletos deben ser devueltos al Product Backlog con una prioridad alta para ser incluidos en el siguiente Sprint. Se considera que un ítem está terminando si es funcional. La suma de ítems terminados es el producto a entregar.

El ciclo de vida de este marco de trabajo está compuesto de cuatro fases; planeación, puesta en escena, desarrollo y entrega (Larman, 2003). En la planeación se establece la visión, se fijan las expectativas y se asegura el financiamiento; en la puesta en escena se identifican más requerimientos y se priorizan para la primera iteración; en la implementación se desarrolla el sistema; y en la entrega se hace el despliegue operativo.

2.1.2.3 Programación Extrema (XP)

XP fue desarrollada por Kent Beck buscando guiar equipos de desarrollo de software pequeños o medianos, de entre dos y diez desarrolladores, en ambientes de requerimientos imprecisos o cambiantes (Beck, 1999). Esta metodología tiene como base cinco valores: simplicidad, comunicación, retroalimentación, respeto y coraje (Ronald, 2004). De los valores se desprenden los principios de XP, lo cuales son:

- Una rápida retroalimentación.
- Simplicidad asumida.
- Cambio incremental.
- Aceptación del cambio.
- Trabajo de calidad.

Las prácticas de esta metodología se derivan de sus valores y principios y están enfocadas en darle solución a las actividades básicas de un proceso de desarrollo, esto es: escribir código, realizar pruebas, escuchar (planear) y diseñar (Ronald, 2004). Las prácticas de XP incluyen: el Planning Game, pequeñas entregas, diseño simple, programación en pareja, pruebas, Refactoring, integración continua, propiedad común del código, paso sostenible, cliente en sitio, metáfora y estándares de código.

El Planning Game define el alcance y fecha de cumplimiento de una entrega funcional completa y divide las responsabilidades entre el cliente y los desarrolladores. El cliente define, utilizando historias de usuario, los requerimientos de manera general y precisan su importancia. En base a esto, los desarrolladores estiman el costo de implementarlas y definen las características de una entrega y el número de iteraciones que se necesitarán para terminarla. Para cada iteración, el cliente define cuales de las historias de usuario componen la entrega funcional que desean que se desarrolle. Se puede crear o modificar historias de usuario en cualquier momento, excepto cuando forman parte de una iteración en curso (Martin & Martin, 2006).

Las entregas pequeñas hacen referencia al uso de ciclos cortos de desarrollo que le muestran software terminado al cliente y obtienen retroalimentación de él. La definición de terminado está relacionada con las pruebas de aceptación. Un diseño simple indica que sistema debe ser tan simple como sea posible, lo cual implica que los desarrolladores deben preocuparse únicamente por las historias de usuario planeadas para la iteración actual (Ronald, 2004).

La programación en pareja indica que todo el código debe ser desarrollado por dos programadores. Las parejas deben cambiar con cierta frecuencia para que el conocimiento de todo el sistema quede en todo el equipo. Esta práctica fortalece los principios de diseño simple, calidad y propiedad colectiva del código.

Las pruebas son la guía del desarrollo del producto, ya que los detalles de las historias de usuario se obtienen en el desarrollo de la prueba de aceptación. Las pruebas son lo primero que se desarrolla; en base a ellas se desarrolla el código que las satisfaga. A este concepto se le denomina desarrollo orientado a las pruebas.

Por otro lado, el Refactoring consiste en realizar cambios que mejoren la estructura del sistema sin afectar su funcionamiento. Para garantizar la no afectación, después de cada cambio se corre la prueba unitaria, con el fin de corroborar sus beneficios. Esta práctica ayuda a mantener el código simple.

La integración continua establece que cada tarea que se completa, se integra al sistema. Este proceso puede darse incluso varias veces al día, en donde se corren las pruebas unitarias para validar si lo que se ha agregado no perjudica a las funcionalidades existentes.

La propiedad común del código implica que no hay una persona propietaria del código que se está desarrollando, o de una porción del mismo, por lo que cualquier desarrollador que esté participando en una iteración puede hacer cambios en el código, siempre y cuando le agregue valor al sistema.

El paso sostenible hace referencia al ritmo de trabajo del equipo. Se indica que debe ser adecuado para que sea factible terminar la iteración sin trabajar horas extras. La metodología establece además, que nunca se debe trabajar horas extras dos semanas consecutivas (Ronald, 2004).

Cliente en sitio se refiere a la necesidad de incluir un representante del cliente trabajando a tiempo completo con el equipo de desarrollo. Su función es resolver oportunamente dudas en la implementación de las historias de usuario.

Metáfora establece que todos los implicados en el proyecto deben tener la misma visión del sistema. Para lograrlo debe hacer abstracciones que establezcan un lenguaje común entre el cliente y los desarrolladores. La metáfora es la guía global del desarrollo.

Los estándares de código son las reglas que definen como se va a escribir la aplicación. La metodología define estos que estos estándares deben estar definidos de tal forma que el código en sí mismo sirva como un documento. Este tema es de altísima relevancia porque tanto la programación en pareja, como la propiedad de código común, son imposibles sin estos estándares (Schallmo, Williams & Lang, 2018).

El ciclo de vida de esta metodología, por su parte está compuesto por: exploración, planeación, iteraciones hacia la primera entrega, Productionizing y mantenimiento. En la fase de exploración se hace una estimación con base en las historias de usuario requeridas para la primera entrega; en la planeación, el cliente y los programadores definen las historias de usuario que se van a implementar y sus fechas; las iteraciones hacia la primera entrega, por su parte, se transforman en el calendario acordado con el cliente, donde cada una de ellas representa historias de usuario implementadas y probadas; en Productionizing se afina el funcionamiento del programa y se despliega; y en mantenimiento se continúan realizando mejoras y arreglos, e implementando nuevas funcionalidades.

2.1.2.4 Kanban

El término Japonés Kanban, fue empleado por Taiichi Onho (Toyota), para referirse al sistema de visualización empleado en los procesos de producción que coordinan en una cadena de montaje la entrega a tiempo de cada parte en el momento que se necesita, evitando sobreproducción y almacenamiento innecesario de productos. El término aplicado a la gestión ágil de proyectos se refiere a técnicas de representación visual de información para mejorar la eficiencia en la ejecución de las tareas de un proyecto (Gaete, Villarroel, & Figueroa, 2021). De acuerdo a Sugimori (Sugimori, Kusunoki, Cho & Uchikawa, 1977), Kanban cumple el rol tanto de enfoque ágil como herramienta y su objetivo principal es cumplir un conjunto de reglas definidas: Visualizar el flujo de trabajo. La Figura 6 esquematiza un tablero Kanban común.

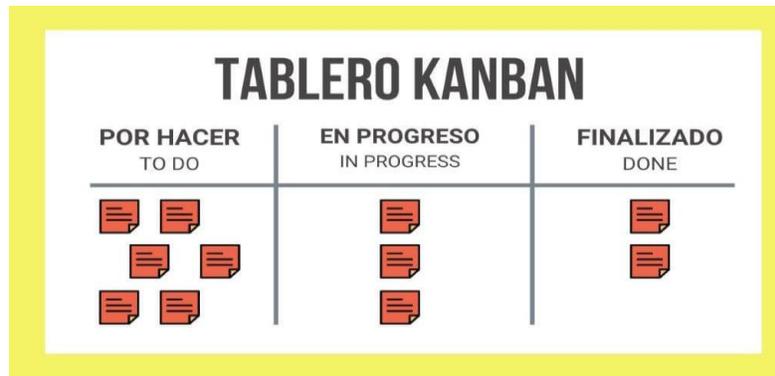


Figura 6 - Ejemplo de Tablero Kanban

Kanban se ha utilizado como un mecanismo de visualización del trabajo en el desarrollo de software ágil, donde los tableros y las reuniones diarias han sido los mecanismos principales para lograr el éxito. Los tres focos clave de Kanban son (Powell, 2018):

1. La materialización del flujo de trabajo.
2. La visualización del flujo de trabajo.
3. La restricción del trabajo en proceso (WIP).

Utilizando un tablero como el de la Figura 6 se visualiza y materializa el trabajo disponible, el trabajo en proceso y el trabajo completado. Todos los días, todos los miembros del equipo se reúnen alrededor del tablero Kanban en lo que se conoce como una reunión diaria, con una duración máxima de 15 minutos. Durante esta reunión, los participantes discuten lo que se logró el día anterior, el plan para el presente día y cualquier problema que pueda ser necesario abordar para lograr el plan. De esta manera, Kanban también fomenta la colaboración y la resolución de problemas en equipo para aumentar el rendimiento productivo. Ya que las tareas están definidas en el tablero, los esfuerzos del equipo se restringen a solo realizar el trabajo en proceso (WIP, por sus ciclas en inglés Work-In-Process).

El principal problema que presenta Kanban, es el de limitar el trabajo en curso, ya que al ser aplicado puede generar cuellos de botella en el desarrollo. Por ejemplo, al trabajar con un tablero de Kanban, en cada una de las columnas se asigna un número que representa la máxima cantidad de tareas que puede haber en dicha columna, y si en algún determinado momento la columna se llena, no se podrá recibir nuevas tareas hasta que se finalice alguna de las tareas en ejecución (Kirovska & Koceski, 2015). Lo anterior es un gran problema puesto que si alguna de las columnas de la tabla se llena, la columna anterior no podrá entregarle sus tareas finalizadas, lo que podría resultar en tareas inactivas, que son ineficiencia para el proyecto.

2.1.2.5 Lean

Lean o también conocido como LSD por su nombre en inglés Lean Software Development, es un conjunto de principios definidos por la industria de fabricación de automóviles de Japón de la década de los 80. El ingeniero de calidad de Toyota, John Krafcik, acuñó el término mientras observaba los procesos y herramientas utilizado para la eliminación de residuos en la producción de automóviles. No fue hasta el 2003 que María y Tom Poppendieck introdujeron Lean como un proceso de desarrollo de software en su libro “Lean Software Development: An Agile Toolkit” (Poppendieck & Poppendieck, 2003).

Los principios fundamentales de este enfoque son:

- Eliminación de desperdicios.
- Amplificar el aprendizaje.
- Decidir lo más tarde posible.
- Entregar lo más rápido posible.
- Potenciar al equipo.
- Establecer integridad.
- Visualizar siempre todo el proceso.

Lean ofrece una opción flexible en la cual se entrega una serie de recomendaciones altamente adaptables cuyo objetivo radica en la generación de valor al cliente a través de la eliminación de residuos y entregas rápidas (Poppendieck & Poppendieck, 2003).

Como se trata de recomendaciones en lugar de reglas, presentan un alto grado de subjetividad y cada equipo de trabajo deberá decidir cómo aplicarlas. Esto supone una mayor dificultad en la aplicación del enfoque, pero al mismo tiempo, una mayor adaptabilidad y versatilidad en los desarrollos. El mayor problema que presenta Lean está muy relacionado con su incesante afán por generar valor al cliente, ya que muchas veces para lograrlo será necesario acelerar ciertas entregas o prescindir de ciertos elementos o tareas importantes con tal de proporcionar valor al cliente lo antes posible. Esto no es del todo malo, ya que gracias a las entregas rápidas el cliente puede estar más involucrado en los procesos de desarrollo y el equipo de trabajo podrá recibir una mayor retroalimentación de él, pero a la vez, podría resultar en una disminución en la calidad el producto final.

2.1.3 Uso de las metodologías de desarrollo de software en Chile

Estudiantes y académicos de la Universidad del Biobío (Grandón, et al., 2018) realizaron una recolección de información respecto al uso de las metodologías de desarrollo de software en Chile. Enviaron encuestas a 170 ejecutivos de TI, de las que se obtuvieron respuestas de 35 ejecutivos quienes informaron 54 proyectos de desarrollo de software.

De esta investigación se obtuvo que, con respecto al total de proyectos informados, el 33% se ejecutó usando metodologías lineales, seguidos por las metodologías incrementales y evolutivas, con un 19% y 18%, respectivamente. El 11% de los proyectos declarados aplicaron metodologías ágiles, mientras que el 15% de los proyectos solo incorporó algunas prácticas ágiles aplicando una metodología tradicional. Finalmente, el 4% de los proyectos no utilizó metodología alguna. La Figura 7 representa lo antes mencionado en un gráfico circular.

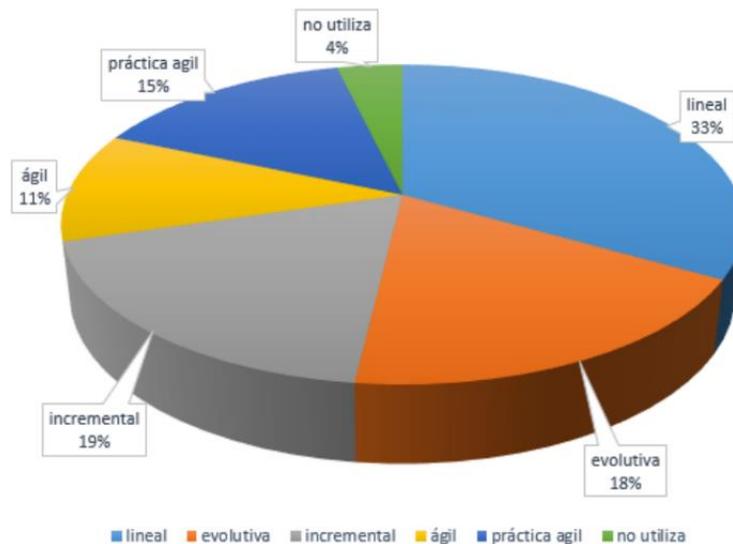


Figura 7 - Gráfico de uso de metodologías en Chile (Grandon, Campos, Fica, & Ávila, 2018).

En cuanto a la información demográfica de los encuestados, se representa en la Tabla 1.

	Características Demográficas de los Encuestados		
	<i>Atributos</i>	<i>Número</i>	<i>Porcentaje</i>
Género	Femenino	7	20%
	Masculino	28	80%
Edad	Menor de 30 años	8	23%
	Entre 31 y 40 años	9	26%
	Entre 41 y 50 años	13	37%
	Mas de 51 años	5	14%
Nivel Educativo	Técnico Profesional	7	20%
	Universitario	21	60%
	Maestría/MBA	6	17%
	Doctorado	1	3%
Cargo	Desarrollador	12	34%
	Jefe de Equipo de Desarrollo	3	9%
	Gerente de Informática	20	57%

Tabla 1 - Información demográfica encuestados sobre uso de metodologías de desarrollo (Grandón et. al., 2018).

2.1.4 Comparación entre metodologías tradicionales y ágiles

Navarro (Navarro, Fernández & Morales, 2013) conduce una revisión de literatura sobre metodologías ágiles; sus principios y fundamentos, establece criterios para definir la relevancia de las metodologías ágiles y también describe las más importantes. Además, realizó una comparación entre las principales características de las metodologías tradicionales versus las metodologías ágiles, las que se presentan en la Tabla 2.

Metodologías Tradicionales	Metodologías Ágiles
Predictivos	Adaptativos
Orientado a procesos	Orientado a personas

Proceso rígido	Proceso Flexible
Se concibe como un proyecto	Un proyecto es dividido en subproyectos
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación.

Tabla 2 - Comparación de aspectos claves entre metodologías tradicionales vs metodologías ágiles (Navarro, Fernández & Morales, 2013)

Se puede destacar que las metodologías ágiles apuestan por un procedimiento flexible y adaptativo, dividiendo el proyecto en subproyectos para realizar entregas constantes. La orientación a las personas y la constante comunicación con el cliente serán la clave para el éxito de estos proyectos.

Un estudio (Stoica, Mircea & Ghilic-micu, 2013) realizó revisiones de literaturas y generó un documento en donde se sintetizan las principales metodologías, tanto tradicionales como ágiles. Los autores plantearon una serie de diferencias entre las metodologías tradicionales y las metodologías ágiles, que son presentadas en la Tabla 3.

Diferencias	Desarrollo Tradicional	Desarrollo Ágil
Hipótesis fundamental	Los sistemas son completamente especificables, predecibles y se desarrollan a través de una planificación detallada y extensa.	El software adaptativo de alta calidad es desarrollado por pequeños equipos que utilizan el principio de mejora continua del diseño y las pruebas basadas en cambios y retroalimentación rápidos.
Estilo de la administración	Comando y control.	Liderazgo y colaboración.
Conocimiento administrativo	Explicito.	Tácito.
Comunicación	Formal.	Informal.
Modelo de desarrollo	Modelo de ciclo de vida (cascada, espiral o modelos modificados).	Modelo de entrega evolutiva.
Estructura Organizacional	Mecánica (burocrática, alta formalización), dirigida a grandes organizaciones.	Orgánico (flexible y participativo, fomenta la cooperación social), dirigido a pequeñas y medianas organizaciones.
Control de calidad	Dificultad de planificación y estricto control. Pruebas difíciles y tardías.	Control permanente de requerimientos, diseño y soluciones. Pruebas permanentes.

Requerimientos de usuario	Detallado y definido antes de la codificación.	Cambiantes.
Costo de reiniciar	Alto.	Bajo.
Pruebas	Después de completar el código.	En cada iteración.
Participación del cliente	Baja.	Alta.
Habilidades adicionales requeridas para ser desarrollador	Ninguna en particular.	Habilidades interpersonales y conocimiento básico del negocio.
Escala apropiada del proyecto	Proyecto de gran escala.	Proyecto de baja y mediana escala.
Desarrolladores	Orientado al plan, con habilidades adecuadas, acceso a conocimientos externos.	Ágil, con conocimiento avanzado y cooperativo.

Clientes	Con acceso al conocimiento, cooperativo, representativo y empoderado.	Dedicado, informado, cooperativo, representativo y empoderado.
Requisitos	Estables y conocidos de antemano.	Emergentes y con cambios rápidos.
Arquitectura	Diseño para requisitos actuales y predecibles.	Diseño para requisitos actuales.
Tamaños	Grandes equipos y proyectos.	Pequeños equipos y proyectos.
Objetivos primarios	Alta seguridad.	Valor rápido.

Tabla 3 - Diferencias entre metodologías de desarrollo tradicional y ágil (Stoica, Mircea, & Ghilic-micu, 2013)

2.2 Design Thinking

El Design Thinking, que literalmente significa “pensamiento de diseño” pero más bien representa “la forma en la que piensan los diseñadores”, es una metodología para generar ideas innovadoras, que centra su eficacia en entender y dar solución a las necesidades reales de los usuarios, por esta razón, la principal característica de esta metodología es centrarse en el humano. Otros conceptos claves relacionados al Design Thinking son la colaboración y pluralidad creativa. La pluralidad creativa se refiere a variedad de aspectos, tendencias, características, entre otras, con respecto a la generación de ideas en el proceso creativo.

En la literatura se pueden encontrar diversas definiciones para el término (Schallmo, Williams & Lang, 2018), algunas de las cuales se exponen en la Tabla 4:

Autor	Definición
Plattner et al., 2009	El enfoque Design Thinking es un enfoque sistemático y orientado al usuario para resolver problemas de la vida real. En lugar de centrarse en cómo se puede resolver técnicamente el problema, el enfoque principal es abordar las necesidades y requisitos del usuario.
Erbeldinger & Ramge, 2015	Design Thinking [...] es un pensamiento innovador con una orientación radical al usuario. Se basa en el principio interdisciplinario y conecta la actitud de apertura con la necesidad de resultados.
Mootee, 2013	“Design Thinking es la búsqueda de un equilibrio mágico entre negocios y arte, estructura y caos, intuición y lógica, concepto y ejecución, alegría y formalidad, y control y empoderamiento”.
Curedale, 2013	Design Thinking es una forma centrada en las personas de resolver problemas difíciles. Sigue un proceso interdisciplinario colaborativo y en equipo. Utiliza un conjunto de herramientas de métodos y puede ser aplicado por cualquier persona, desde los diseñadores y ejecutivos corporativos más experimentados hasta los escolares.

Ideo, 2012	<p>El pensamiento de diseño consiste en creer que podemos marcar la diferencia y tener un proceso intencional para obtener soluciones nuevas y relevantes que generen un impacto positivo.</p> <p>Design Thinking le da fe en sus habilidades creativas y en un proceso de transformación de desafíos difíciles en oportunidades para el diseño.</p>
------------	--

Tabla 4 - Definiciones de Design Thinking (Schallmo, Williams, & Lang, 2018)

De estas definiciones se destaca algunos aspectos relevantes:

- **Establecer metas:** Desarrollar una nueva solución para problemas existentes o futuros.
- **Orientación:** Está dirigido específicamente a los usuarios.
- **Proceso:** Se lleva a cabo sobre la base de procesos estructurados e iterativos.
- **Participantes:** Debe utilizar equipos multidisciplinarios.

Por lo tanto, del Design Thinking se sintetiza lo siguiente (Schallmo, Williams & Lang, 2018):

“El enfoque del Design Thinking persigue el objetivo de desarrollar nuevas soluciones para problemas existentes. Estas soluciones están constantemente orientadas a las necesidades de los usuarios y tienen una influencia positiva. El proceso de Design Thinking es estructurado e iterativo y desarrollado por un equipo multidisciplinario utilizando una serie de técnicas”.

2.2.1 Evolución

El término Design Thinking fue nombrado por primera vez por el economista, psicólogo, politólogo y teórico de las ciencias sociales Herbert Simon, quién el año 1969 escribió un libro llamado “la ciencia de lo artificial”, en el que basaba su argumentación en la descripción de los sistemas artificiales como un medio interno y uno externo para satisfacer un objetivo⁴. La premisa era simple: “Si el medio interno se adapta al externo, o viceversa, el objetivo del sistema artificial se cumple”.

Simon describe cuatro etapas básicas que posteriormente fueron útiles a la hora de definir los pasos del Design Thinking. Se empieza por definir los problemas, luego se investiga a través de breves entrevistas, después surgen las distintas ideas que podrían convertirse en solución y algunas de estas son prototipadas para elegir la mejor. Finalmente, la idea se implementa y evalúa. Si resulta que la solución falló, ha sido una experiencia de aprendizaje.

Otros personajes importantes en esta trama son David Kelley y Hasso Platner, profesor de ingeniería mecánica en la Universidad de Stanford y el jefe del consejo de administración y co-fundador de SAP, respectivamente. En conjunto crearon el d.School o escuela de diseño, tanto para la Universidad de Stanford en Estados Unidos, como para la Universidad de Potsdam en Alemania. Propusieron un proceso de Design Thinking de 6 etapas: comprender, observar, punto de vista, idear, prototipar y probar (d.School, 2010).

El año 2008, el diseñador inglés y también profesor de la Universidad de Stanford, Tim Brown, publicó un artículo en la Harvard Business Review, en el que mencionaba algunos principios del Design Thinking, un procedimiento de 3 etapas (inspiración, tormenta de ideas e implementación) y también, declaró las principales características que debe desarrollar un Design Thinker o pensador de diseño (Brown, 2008). Cabe destacar que Brown es el CEO y presidente de IDEO, una empresa dedicada a la innovación y diseño que tiene su sede en Palo Alto, California, y que es dirigida junto al profesor David Kelley.

En cuanto a SAP⁵, considera al Design Thinking como una de las últimas novedades en innovación empresarial. La definen como una metodología que se utiliza para desarrollar productos o poner a pruebas ideas utilizando el mismo enfoque con el que las mentes creativas encaran sus propios desafíos.

⁴ Herbert Simon, Design Thinking: <https://proyectoidis.org/las-ciencias-de-los-artificial/>

⁵ Definición de Design Thinking según SAP: <https://news.sap.com/latinamerica/2013/04/como-es-el-design-thinking/>

El portal de noticias de SAP, cita al diseñador Víctor Lombardi, quién creó una lista con las principales características de Design Thinking mencionada a continuación:

- **Es colaborativo:** Trabajar en equipo con personas que posean experiencias y habilidades complementarias enriquece la experiencia y genera consenso.
- **Es abductivo:** Fomenta la creatividad con ideas nuevas para encontrar características nuevas y mejores soluciones a los problemas.
- **Es experimental:** Se basa en construir prototipos y plantear diferentes hipótesis. Se analizan pruebas para definir qué es lo que funciona y qué es lo que no.
- **Es personal:** Considera el contexto de cada problema y persona implicada.
- **Es integrador:** Permite observar el problema desde una perspectiva global y considera todas las posibles consecuencias de una decisión-
- **Es interpretativo:** Realiza suposiciones para identificar los problemas y definir con claridad sus posibles soluciones.

El Design Thinking es definido como un enfoque integrador para crear nuevas soluciones basadas en las necesidades humanas y que aprovecha las tecnologías digitales para ofrecer valor comercial. Desde crear experiencias increíbles hasta acelerar la transformación, Design Thinking desempeña un papel vital en impulsar la innovación de los clientes.

2.2.2 Premisas del proceso creativo

Basado en los documentos del d.School: Bootcamp bootleg y del Design Process Mini-Guide (d.School, 2010). Algunas premisas planteadas en el documento “Mini guía: una introducción al Design Thinking” del d.School son:

- **Enfocarse en el valor humano:** Tener empatía por las personas para las cuales se están diseñando y valorar la retroalimentación de estos usuarios para lograr un buen diseño.
- **No lo digas, muéstralo:** Comunicar la visión de una manera significativa e impactante creando experiencias, usando visualizaciones ilustrativas y contando buenas historias.
- **Colaboración radical:** Juntar equipos de personas de variadas disciplinas y puntos de vista. La diversidad permite salir a la luz ideas radicales.

- **Estar consciente del proceso:** Se debe tener claro el proceso de diseño y saber que métodos se utilizan en cada fase.
- **Cultura de prototipos:** Hacer prototipos no es simplemente una manera de validar las ideas, es una parte integral del proceso de innovación.
- **Incitar a la acción:** El nombre Design Thinking no debe confundirse, no se trata solo de pensar si no que también de hacer.

2.2.3 Fases

Existen distintos modelos de procedimientos propuestos para el Design Thinking, cada uno de ellos con sus principios y etapas. Al revisar la literatura disponible encontramos 3 variaciones: el enfoque de Brown (Brown, 2008), el de Plattner (Plattner, 2009) y el del d.School (d.School 2010).

A. Design Thinking bajo el enfoque de Brown (Brown, 2008)

Tim Brown expone el enfoque estilo IDEO de Design Thinking, publicando un artículo en la Harvard Business Review. En este documento realiza algunas recomendaciones:

- **Integración de los pensadores del diseño:** los pensadores del diseño deben participar en el proceso desde el principio.
- **Centrado en el usuario:** la orientación hacia las necesidades, preferencias y comportamiento de los usuarios permite obtener nuevos conocimientos y alinear las soluciones con ellos.
- **Realización de experimentos:** los experimentos deben realizarse temprano y con frecuencia.
- **Integración del conocimiento externo:** los usuarios deben estar integrados en el desarrollo.
- **Seguimiento de proyectos de cualquier tamaño:** Se deben implementar tanto los proyectos incrementales a corto plazo como los conceptos revolucionarios a largo plazo.
- **Definición de presupuestos:** Design Thinking puede ser un resultado rápido y significativo disponible en poco tiempo que debería comercializarse. Debe disponerse de un presupuesto adecuado para este fin.
- **Construir equipos interdisciplinarios:** para ello, se debe buscar empleados en Universidades adecuadas.
- **Ejecutar el proceso por completo:** solo si se completa el proceso se pueden lograr los resultados esperados.

La Figura 8 muestra el procedimiento de Design Thinking propuesto por Tim Brown el año 2008:



Figura 8 - Procedimiento Design Thinking (Brown, 2008)

Inspiración: Se define el problema, se identifican oportunidades y cambios y se observa a los usuarios y su entorno.

Lluvia de ideas: Se realizan sesiones de lluvia de ideas y bocetos durante el proceso de generación de ideas. Se garantiza que sea un trabajo orientado al cliente permitiendo la participación de usuarios reales, mientras los prototipos son probados y desarrollados constantemente.

Implementación: Se implementa el prototipo definitivo. Para ello, se define una estrategia de comunicación y se crea un caso de negocio.

B. Design Thinking bajo el enfoque de Plattner (Plattner, Meinel, & Weinberg, 2009)

Este documento contiene un informe de experiencia sobre Design Thinking y es usado en el instituto Hasso Plattner de la Universidad de Potsdam y en la Universidad de Stanford en Palo Alto. Son definidos algunos principios del Design Thinking:

- Los equipos multidisciplinarios como una necesidad para el éxito del Design Thinking, ya que el espectro de competencias está más diversificado.
- No solo especialistas, porque ya conocen la solución y muchas veces llegan a sus límites.
- Consideración de diferentes estilos de aprendizaje para promover el proceso de Design Thinking.
- Espacio diáfano con mobiliario móvil y atractivo mobiliario para fomentar la creatividad y el trabajo en equipo.
- Repetición en forma de iteraciones para aprender de los errores y así aprovechar la oportunidad de generar mejoras. Una reiteración no se ve como una pérdida de tiempo, sino como una ganancia de conocimiento.

- Visualizar para hacer visibles las ideas en imágenes. Los textos y las notas están vinculados con fotos, videos, bocetos, dibujos y símbolos.
- Utilice la lluvia de ideas de acuerdo con ciertas reglas para desarrollar ideas y soluciones.
- Consideración del tiempo para desarrollar ideas rápidamente. La definición de plazos no impide la generación de ideas pero promueve la espontaneidad.

La Figura 9 muestra el procedimiento de Design Thinking propuesto por Hasso Plattner:



Figura 9 - Procedimiento Design Thinking (Plattner, Meinel, & Weinberg, 2009)

Entender: Esta fase describe la tarea y define el problema. Representa la fase más importante en el modelo de procedimiento ya que los errores que ocurren aquí afectan a todo el proceso. La atención se centra en la orientación del usuario y la cuestión del grupo destinatario. Dependiendo del Desafío de Diseño, se debe crear una comprensión combinada dentro de equipos multidisciplinarios.

Observar: Los miembros del equipo deben adquirir conocimientos expertos para la tarea en cuestión. Las soluciones existentes se analizan y cuestionan. Siguiendo la definición precisa de los usuarios, son observados y entrevistados personalmente. En el contexto de la observación, se debe analizar el uso de productos y comportamientos.

Punto de Vista: Se evalúan, interpretan y ponderan los conocimientos adquiridos. Se resumen las experiencias de todo el equipo para crear una base común. Se crea una persona típica, ficticia, que se describe de manera integral. Es importante separar los hechos relevantes de los no relevantes.

Idear: La lluvia de ideas genera ideas que contienen una solución. Tras la adquisición de numerosas ideas, se evalúan, seleccionan y comunican.

Prototipar: Al desarrollar prototipos, las ideas seleccionadas deben hacerse visibles y comunicarse para recibir comentarios de los usuarios. Los prototipos no tienen por qué ser extremadamente caros o perfectos.

Probar: Los prototipos desarrollados son probados por pensadores de diseño y usuarios para conocer las fortalezas y debilidades de una idea y aprender de los resultados.

C. Design Thinking bajo el enfoque del d.School (d.School, 2010)

Este procedimiento de Design Thinking será profundizado en la siguiente sección, ya que es el más popular actualmente y sentó las bases de la metodología Design-Led Development, con la cual se ejecutó este proyecto. El procedimiento propuesto por la d.School es esquematizado en la Figura 10:



Figura 10 – Procedimiento Design Thinking (d.School, 2010)

A continuación, se describen las etapas.

2.2.3.1 Empatía

Empatizar se define como el esfuerzo por comprender las cosas que se hacen y por qué, sus necesidades físicas y emocionales, cómo conciben el mundo y qué es significativo para ellos. Son las personas en acción las que inspiran a los diseñadores y direccionan una idea en particular. Esta fase es la base del Design Thinking.

Lo básico para empatizar es:

- Observar el comportamiento de los usuarios.
- Involucrarse con los usuarios a través de conversaciones, ya sea con preguntas de pasillo u otras más estructuradas.
- Mirar, escuchar y pedir algunas explicaciones.

Algunas de las actividades que generan artefactos son:

- Investigaciones de Mercado.
- Entrevistas.
- Creación de “Persona”.
- Storytelling.

2.2.3.2 Definir

Definir es una fase crítica en el proceso de diseño, ya que el objetivo es generar un “Point of View” (POV), que significa crear una declaración de problema viable y enfocado en el usuario. Definir es traer claridad y enfoque al espacio de diseño en que se definen y redefinen conceptos. Esta fase debe cumplir con ciertos criterios para que su funcionamiento sea óptimo:

- Enmarcar un problema con un enfoque directo.
- El proyecto debe inspirar al equipo.
- Generar criterios para evaluar ideas.
- Capturar las mentes y corazones de las personas estudiadas.

Algunas de las actividades que generan artefactos son:

- Planteamiento del Problema.
- Point of View o POV.

2.2.3.3 Idear

En esta fase se conciben una gran cantidad de ideas que dan muchas alternativas de donde elegir posibles soluciones, en vez de encontrar una única mejor solución. La creación de múltiples ideas permite abordar distintos focos:

- Aprovechar las distintas visiones de cada equipo de trabajo, fomentando el trabajo colectivo.
- Descubrir áreas inesperadas de exploración.

Algunas de las actividades que generar artefactos son:

- Lluvia de ideas.
- Análisis de POV.
- S.C.A.M.P.E.R.

2.2.3.4 Prototipar

El objetivo de esta fase es la generación de elementos informativos, como dibujos u objetos, con la intención de responder a preguntas que nacen acerca de la solución final. El prototipado debe ser rápido y barato, pero a la vez debe entregar puntos para debatir y retroalimentar a los usuarios y el equipo de diseño. Este proceso se va refinando mientras el proyecto avanza y los prototipos van mostrando más características.

¿Para qué hacer prototipos?

- Para inventar y construir pensando en resolver el problema.
- Para comunicar, porque si una imagen vale más que mil palabras, un prototipo vale más que mil imágenes.
- Para comenzar conversaciones más eficientes con los usuarios, al estar concentrados sobre el prototipo.
- Para cometer los errores antes.
- Para evaluar las alternativas al desarrollarlas sin comprometerse con una sola en específico.

¿Cómo hacer prototipos?

Se comienza construyendo, aun cuando no se sepa lo que se construye, el acto de agrupar los materiales es suficiente para comenzar. También, se debe trabajar los prototipos con el usuario en mente, pensando en: ¿Qué se espera evaluar con el usuario? ¿Cuáles son los comportamientos esperados?

Algunas de las actividades que generan artefactos son:

- Storyboards.
- Prototipos de baja y alta fidelidad.

2.2.3.5 Evaluar

Esta fase consiste en solicitar retroalimentaciones y opiniones sobre los prototipos que han sido creados. Una buena regla es siempre hacer un prototipo creyendo que estamos en lo correcto pero se debe evaluar pensando en estar equivocado. Es importante evaluar para refinar los prototipos y las soluciones, aprender más sobre el usuario y refinar el POV.

¿Cómo evaluar? D.School (2010) entrega algunos consejos para evaluar:

- No lo digas, muéstralo: Dale a los usuarios los prototipos sin explicar nada para dejar que interprete el objeto.
- Crear experiencias: No es suficiente entregar el prototipo, lo ideal es crear el ambiente y recrear la experiencia.
- Pídele a un usuario que compare.

2.2.4 Design Thinkers

Los Design Thinkers o pensadores de diseño no son creados por escuelas de diseño, a pesar de que la mayoría de los profesionales han tenido algún tipo de formación en diseño. Según Tim Brown (Brown, 2008) muchas personas ajenas al diseño profesional tienen una aptitud natural para el Design Thinking, que el desarrollo y las experiencias correctas pueden desbloquear y potenciar. A continuación, son presentadas las características buscadas por los Design Thinkers:

- **Empatía:** Al adoptar un enfoque centrado en las personas, los Design Thinkers pueden imaginar soluciones que son inherentemente deseables y satisfacen necesidades explícitas o latentes. Además, son capaces de darse cuenta de cosas que otros no ven y utilizan sus conocimientos para inspirar la innovación.
- **Pensamiento integrador:** El Design Thinking no solo se basa en procesos analíticos, sino que también muestra la capacidad de ver todos los aspectos sobresalientes, y a veces contradictorios de un problema de confusión y crear soluciones novedosas que van más allá y mejoran drásticamente las alternativas existentes.
- **Optimismo:** Asumen que no importa cuán desafiantes sean las limitaciones de un problema dado, al menos una solución potencial es mejor que las alternativas existentes.
- **Experimentalismo:** Las innovaciones significativas no provienen de ajustes incrementales. Los Design Thinkers plantean preguntas y exploran las limitaciones de forma creativas, que avanzan en una dirección completamente nuevas.
- **Colaboración:** La creciente complejidad de productos, servicios y experiencias, ha reemplazado el mito del genio creativo solitario por la realidad del colaborador interdisciplinario entusiasta.

2.3 Design-Led Development

Antes de comenzar a detallar el procedimiento propuesto por esta metodología, se aclara que, dada la falta de estudios asociados a este procedimiento, la principal fuente es el sitio oficial de la herramienta SAP Build.

Design-Led Development o DLD es otra metodología basada en diseño y que sigue los lineamientos del Design Thinking para el proceso de generar una solución. La Figura 11 contiene el procedimiento propuesto en el sitio oficial de Build.

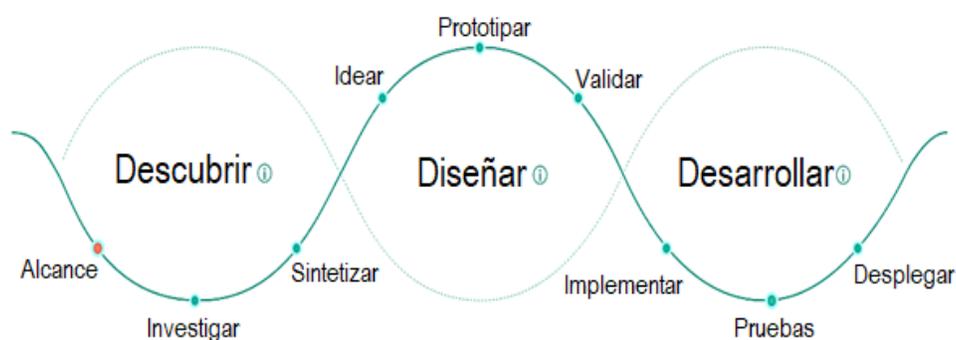


Figura 11 - Esquema del DLD: Etapas y sub-etapas. Fuente: <https://www.build.me/learning> (Adaptado)

A continuación, será descrita cada etapa y sub-etapa del DLD. Para las sub-etapas se describirán sus objetivos, participante, tiempo recomendado, materiales necesarios, actividades y recomendaciones. Cabe destacar que para la mayoría de las etapas la metodología recomienda una o más actividades. El detalle de las actividades puede encontrarse en los Anexos del documento.

Entes de pasar a la descripción de las etapas, es importante destacar que el equipo DLD debe estar compuesto por un Project Manager, Product Owner, DT Coach y el resto del equipo.

- El Project Manager es quien gestiona el proyecto, iteraciones y habilita al equipo para maximizar su productividad.
- El Product Owner es el representante de los interesados y constantemente debe estar en comunicación con el equipo del proyecto.
- El DT Coach es una persona capacitada en la metodología Design Thinking, es decir, debe tener el conocimiento teórico y además debe ser un Design Thinker.

- El equipo del proyecto es multidisciplinario, compuesto por: diseñadores, personas del área de la tecnología y personas del área de dominio del proyecto en general. El equipo del proyecto es la pieza fundamental, estos pueden jugar el papel de moderadores, tomadores de nota, diseñadores, programadores, etc., según se requiera.

2.3.1 Descubrir

Descubrir es la primera etapa y donde se exploran oportunidades para agregar valor a los usuarios e interesados, se identifica el problema a resolver y para quién se trabajará. Está compuesta por tres sub-etapas: Alcance, Investigar y Sintetizar.

2.3.1.1 Alcance

Alcance es la primera sub-etapa asociada a descubrir. Los objetivos son comunicar una comprensión clara del problema, involucrar a las personas adecuadas al proyecto y realizar una planificación orientada a maximizar tanto el trabajo como el tiempo del proyecto.

A. Acordar el alcance del proyecto

Tiempo: 2 horas.

Personas: Project Manager.

Se necesita: Planilla de documento de alcance del proyecto.

El documento de alcance del proyecto es un elemento clave para comunicar de qué se trata el proyecto y describir el proyecto a las demás personas. Ayudará a acordar lo que se entregará como parte del proyecto, quién está haciendo qué y qué plazos tiene para realizar estas entregas. Además, este documento ayuda a declarar los objetivos, identificar entregables clave, proporcionar un cronograma aproximado, administrar las expectativas de las partes interesadas y a obtener una aceptación de los patrocinadores.

Los pasos para crear un documento de alcance son:

- Definir el objetivo general del proyecto.
- Describir el alcance (lo que se considera dentro del alcance y lo que no).
- Enumerar las partes interesadas del proyecto.
- Describir el equipo del proyecto.
- Enumerar entregables junto a los criterios de éxito.
- Enumerar los hitos del proyecto.
- Validar el documento de alcance con el equipo y las partes interesadas.

2.3.1.2 Investigar

Investigar es la segunda sub-etapa asociada a descubrir. Los objetivos son conocer las necesidades reales de los usuarios y descubrir oportunidades para mejorar su realidad.

A. Entrevistas a los usuarios

Tiempo: 2 horas de preparación, 45-60 minutos por entrevista.

Personas: 5 usuarios (1 por sesión), 1 moderador y 1-2 tomadores de notas.

Se necesita: Guía de entrevista (Script), bloc de notas, lápices y grabadoras.

Observar y entrevistas a los usuarios proporciona muchos conocimientos que ayudan a tomar decisiones informadas mientras se crea un producto de software. Este proporciona la oportunidad de ponerse en el lugar del usuario, comprender sus objetivos, motivaciones y dolores.

Los pasos para realizar una entrevista son:

- Decidir los roles y números de usuario a entrevistas. Establecer un plan y cronograma para las entrevistas.
- Preparar la guía de entrevista. Se recomienda comenzar por lo básico, preguntando por algunos datos del trabajador, con quién trabaja, prioridades de trabajo principales y cómo es su entorno de trabajo.
- Obtener ayuda para tomar notas, lo que también ayuda para incluir a miembros del equipo que no acostumbran a interactuar con los usuarios.

- Capturar hechos y no interpretaciones. Por lo tanto, se recomienda registrar exactamente lo que dice el entrevistado. Se recomienda realizar preguntas aclaratorias si no se comprende completamente lo que se explicó y reproduzca lo entendido.
- Agradecer a los usuarios por el tiempo utilizado para realizar la entrevista.

2.3.1.3 Sintetizar

Sintetizar es la tercera y última sub-etapa asociada a descubrir. El objetivo es concentrarse en los problemas concretos, comprender mejor a los usuarios y crear PERSONAS (o perfiles de personalidad) para los usuarios objetivos con puntos de vista detallados.

A. Destilar Hallazgos

Tiempo: 1 hora de preparación, 60 minutos destilando hallazgos por cada entrevistado.

Personas: Equipo del proyecto.

Se necesita: Una sala de reuniones con una pared disponible para usar pizarras y notas adhesivas.

Se reúne al equipo y se comparten las historias de usuario recopiladas durante las entrevistas. Se compartir esta información crea un entendimiento común dentro del equipo sobre: quién es el usuario, qué hace y exactamente lo que necesita.

Los pasos para destilar hallazgos son:

- Para preparar la sesión, todos los participantes de las entrevistas deben revisar sus notas y resaltar los puntos clave. Si se tomaron imágenes o se han creado bocetos con apuntes, deben ser colgados en la pared para que todo el equipo pueda verlos.
- Los tomadores de notas deben compartir los aspectos más destacados de la investigación.
- En notas adhesivas describir aspectos destacados de: las características los usuarios, necesidades, tareas, puntos débiles, herramientas utilizadas, entorno de trabajo, características claves de la solución deseada, etc.
- Buscar puntos de datos comunes como: objetivos similares, herramientas utilizadas, etc. Agrupar las notas puestas con anterioridad y agruparlas en temas comunes.

B. Reformular el enunciado del problema

Tiempo: 45 minutos.

Personas: Equipo del proyecto.

Se necesita: Una pizarra.

Al comenzar con el proyecto, el enunciado del problema a resolver puede ser un poco confuso, ya sea por un nivel demasiado alto o por no ser lo suficientemente preciso, utilizando palabras que tal vez no sean claras para todo el equipo. Esta declaración debe discutirse con el equipo para aportar claridad y reformular el enunciado del problema según acuerdo común.

Los pasos para reformular el enunciado del problema son:

- Escribir el enunciado del problema en una pizarra.
- Subrayar las palabras clave de la declaración.
- Definir significado de cada palabra clave y seleccionar el mejor sinónimo.
- Realizar preguntas sobre el problema.
- Reformular la oración completa del enunciado del problema.

C. Crea una PERSONA

Tiempo: 2 horas por persona.

Personas: Moderador de la entrevista y tomadores de notas.

PERSONA sintetiza los hallazgos comunes de la investigación y se crea analizando las características, contextos, necesidades, objetivos y puntos débiles derivados de las etapas anteriores. PERSONA es un artefacto importante, que toma valor más adelante al tomar decisiones de diseño y desarrollo, ayudando a mantener el enfoque al abordar las necesidades correctas.

Los pasos para crear PERSONA son:

- Dar un nombre a la persona y agregar una foto o un boceto que ilustre las características de la persona.
- Crear una cita breve, real o compilada a partir de varias citas reales, que ilustre los objetivos o motivaciones del usuario.
- Describir necesidades, objetivos, características y herramientas que puedan ser importantes para el diseño de su solución.
- Validar PERSONA con los usuarios finales.

D. Definir Puntos de Vista

Tiempo: 30 minutos por punto de vista.

Personas: Equipo del proyecto.

Después de haber creado al artefacto PERSONA, lo más posible es que se haya creado una larga lista de necesidades. La metodología recomienda enfocarse y abordar por separado cada necesidad creando un punto de vista o POV. Definir un punto de vista ayuda a concentrarse en qué problema se está tratando de resolver y para quién.

Los pasos para definir un punto de vista son:

- Mirar los hallazgos de PERSONA y la investigación que se ha realizado.
- Describir al usuario según las características descubiertas.
- Elegir una de las necesidades que haya priorizado.
- Capturar la información asociada a la necesidad seleccionada. Se recomienda profundizar para comprender las emociones del usuario.

2.3.2 Diseñar

Diseñar es la segunda etapa del procedimiento propuesto por el DLD. Aquí se comprende el problema y las necesidades, para comenzar a crear la mejor solución. Son realizados los prototipos y las pruebas, para comenzar a iterar el diseño con los usuarios.

2.3.2.1 Idear

Idear es la primera sub-etapa asociada a diseñar. Los objetivos son pasar del problema a la solución, converger ideas deseables para seleccionar las mejores, evaluar la fiabilidad y factibilidad de las ideas, y finalmente visualizar la experiencia de usuario a través de un Storyboard (guión gráfico).

A. Crear casos de uso

Tiempo: 30 minutos por caso de uso (los casos de uso más complejos tardan más en describirse).

Personas: Equipo del proyecto.

Se necesita: Los artefactos que se han producido hasta el momento, según las actividades pasadas.

Los casos de uso describen secuencias de interacciones entre los usuarios y el sistema, definen el producto desde la perspectiva del usuario y se especifica qué objetivos o tareas debe admitir y habilitar la interfaz de usuario. Al expresar los requisitos en el contexto de pasos o acciones, los casos de uso ayudan a identificar cuando y bajo qué condiciones ocurren ciertos comportamientos. Esto sirve como una herramienta de comunicación que los miembros del equipo pueden utilizar para comprender mejor la experiencia del producto y como guía para el diseño de la interfaz de usuario.

Los pasos para crear los casos de uso son:

- Reunir toda la información de la investigación: la personalidad (PERSONA), los puntos de vista y los Storyboards.
- Identificar los casos de uso en los Storyboards. Primero se comienza con una imagen general, para luego pasar a los casos de alta prioridad, seguido de los de menor prioridad.

- Concentrarse en un usuario/persona a la vez.
- Describir los objetivos de la PERSONA o POV.
- Crear un caso de uso por objetivo.
- Anotar todos los pasos de interacción del usuario cuando proporcione detalles y enumere los datos de interacción necesaria.
- Crear diagramas de flujo para visualizar como se organizan los casos de uso y como se relacionan entre sí.
- Validar los casos de uso con los usuarios finales.

B. Lluvia de ideas con el equipo

Tiempo: 2 horas.

Personas: Equipo del proyecto y personas no relacionadas al proyecto.

Se necesita: Notas adhesivas, rotuladores, lápices, borradores y espacio en la pared.

Cada problema es una oportunidad para la innovación. Usando la técnica “¿Cómo podríamos?” del inglés “HOW MIGHT WE?”, se prepara para el siguiente paso, la lluvia de ideas. Esta es realizada sobre posibles conceptos y soluciones, dando prioridad a las necesidades del usuario. Las perspectivas y los antecedentes diversos conducen a ideas innovadoras.

Para realizar una lluvia de ideas con el equipo son:

- Conseguir un grupo variado de 5 a 8 participantes.
- Preparar un espacio de trabajo con hojas cortadas a la mitad, lápices, borradores, marcadores, notas adhesivas y espacio en la pared.
- Repasar las reglas de la lluvia de ideas.
- Tomar 5 a 10 minutos para escribir ideas en las medias hojas alrededor de su “¿Cómo podríamos?” o área temática.
- Presentar y discutir todas las ideas en las hojas.
- Capturar los puntos de discusión en notas adhesivas y agrupar ideas para encontrar temas en común, priorizando según las necesidades del usuario.

C. Crear un Storyboard

Tiempo: 4 horas.

Personas: Equipo del proyecto.

Se necesita: Las hojas de la lluvia de ideas y nota adhesivas.

Un Storyboard es un viaje de alto nivel que describe a un usuario que realiza una tarea utilizando un producto o servicio recién imaginado. El Storyboard ilustra el contexto físico en el que se encuentra el usuario, las herramientas que utilizará y las personas con las que interactúa. Todavía no detalla las soluciones con pantallas reales, si no que incorpora las ideas que proviene de la lluvia de ideas. Un Storyboard o guión gráfico ayuda al equipo y a las partes interesadas a empatizar con el usuario y entusiasmarse con la solución de principio a fin.

Los pasos para crear un Storyboard son:

- Reunirse junto a uno o dos compañeros del equipo, buscar un espacio de trabajo y reproducir las declaraciones “¿Cómo podríamos?” y las ideas generadas durante la sesión de lluvia de ideas.
- Construir la historia entorno a su usuario objetivo.
- Compartir las historias con los clientes, usuarios finales, partes interesadas, etc. Y usar sus aportes para mejorarla.
- Identificar oportunidades para soluciones software en el Storyboard.

D. Preparar un espacio de trabajo creativo

Tiempo: 1 hora por preparación.

Personas: Equipo del proyecto.

Se necesita: Una lista de verificación de los materiales creativos.

El propósito de ordenar el espacio de trabajo es fomentar una mentalidad de pensamiento y colaboración multidisciplinaria, donde los muros y las barreras de la comunicación se eliminan y el equipo adquiere un sentido de identidad propia y progreso. En teoría, cualquier habitación con cualquier presupuesto puede convertirse en un espacio de trabajo creativo, siempre que el diseño sea configurable, los materiales sean flexibles y el espacio sea abierto y acogedor.

Para preparar un espacio de trabajo creativo los pasos son:

- Revisar el espacio disponible y el tamaño del equipo. Mantener el espacio abierto y flexible para la colaboración.
- Paredes y ventanas despejadas para maximizar las áreas de superficie para la creación de prototipos y la instalación de pizarras blancas móviles o caballetes.
- Proporcionar un lugar de almacenamiento (un carro rodante es ideal) para los materiales y suministros creativos.
- Agregar elementos adicionales como proyectores y parlantes según sea necesario para aumentar la utilidad de la habitación.

2.3.2.2 Prototipar

Prototipar es la segunda sub-etapa asociada a diseñar. Sus objetivos son proporcionar una versión de prototipo de baja fidelidad, mostrar el diseño, interfaz y contenido, demostrar acciones y comportamientos. Esto proporciona una validación temprana.

A. Artefactos de Design Thinking

Tiempo: 1 hora.

Personas: Product Owner, DT Coach.

Se necesita: Los artefactos generados en el Design Thinking (más imágenes, dibujos, etc.).

La metodología recomienda generar un espacio de trabajo en la herramienta SAP Build para recopilar todos los artefactos recolectados a lo largo del proyecto. Este espacio de trabajo permite compartir el entorno con otros participantes, el único requerimiento es que tengan una cuenta para acceder.

La principal característica de Build, es que permite crear prototipos procesables para obtener comentarios de otros usuarios de Build, el cual es el paso clave de esta actividad.

B. Dibujar prototipos

Tiempo: 1 día.

Personas: el ejecutor del proyecto.

Se necesita: Imprimir plantillas y lápices o pizarra y marcadores.

Se crean prototipos dibujados a mano que muestren los elementos principales de la interfaz de usuario y el flujo general de su solución. Hacer este prototipo de bajo costo desde el principio permite tener una idea sobre qué conceptos funcionan mejor para el usuario.

Los pasos para dibujar un prototipo son:

- Para comenzar es necesario tener el Storyboard e identificar los pasos de la historia que involucran la solución de software. Esto brinda una idea de las pantallas principales que podría tener la solución futura.
- Dibujar el diseño de cada pantalla con los elementos principales. Por ejemplo; el texto, campos de entrada, botones de acción, etc.
- Ahora se cargan los bocetos en Build para darles dinamismo y presentarlo a los usuarios finales para obtener comentarios.

C. Crear Wireframes**Tiempo: 2-4 horas.****Personas: Equipo del proyecto.****Se necesita: Una herramienta de prototipado digital**

Los Wireframes son representaciones visuales de una interfaz. Son fácilmente modificables, de bajo o alta fidelidad, y son una herramienta importante utilizada para el diseño y desarrollo de productos. Comunican las características e interacciones del producto, lo que ayuda a los miembros del equipo y a las partes interesadas a comprender, alinear e iterar mejor los diseños finales. El propósito de un Wireframe es apoyar la documentación y explicar el éxito de su diseño a los demás participantes del proyecto para seguir recolectando comentarios.

Los pasos para crear un Wireframe son:

- Pensar en la estructura general del diseño de la solución propuesta.
- Crear un mapa o secuencia que muestre las pantallas de la aplicación y la ruta que los usuarios deben tomar para navegar por ellas. Crear un Wireframe por pantalla. Se recomienda evitar hacer una apariencia demasiado pulida, ya que el proceso de exploración del diseño conlleva realizar muchos cambios.
- Agregar contenido adicional como textos, imágenes, elementos de la interfaz de usuario, etc.
- Describir las interacciones, como clics, toques, entradas de datos y otros posibles comportamientos. Además, se recomienda colocar notas junto al Wireframe proporcionando más detalles.

2.3.2.3 Validar

Validar es la tercera sub-etapa de Diseñar. Sus objetivos son verificar el enfoque de la investigación, evaluar la efectividad del diseño y observar y escuchar a los usuarios

A. Probar los bocetos con los usuarios

Tiempo: 2 horas de preparación, 45-60 minutos por test con cada usuario, 60 minutos de análisis.

Personas: 3-5 usuarios (1 por sesión), un moderador y un tomador de notas.

Se necesita: Los bocetos, notas adhesivas, libretas y lápices.

Antes de llevar el diseño al desarrollo es importante asegurarse de que realmente funcione para los usuarios. Probar con los usuarios puede ahorrar tiempo y recursos. Preguntar a los usuarios como complementarían una tarea mientras se observa y escuchar da la oportunidad de identificar nuevos problemas y mejorar los diseños para la próxima iteración.

Los pasos para probar los bocetos con los usuarios son:

- Antes de realizar la prueba es importante declarar los objetivos ¿Qué se quiere averiguar? Se recomienda dar una lectura a las tareas que el usuario espera realizar en la solución, los diseños y crear un escenario que los involucre a todos.
- Reclutar participantes de prueba.
- Decidir donde se ejecutará la prueba. Se puede hacer en persona, usando los bocetos a mano, pero también digitalmente usando Build.
- Ejecutar la prueba. Durante la prueba es importante captar comentarios honestos y alentar a los participantes a pensar en voz alta.
- Después de la prueba se debe analizar hallazgos y compartirlos con el equipo. Se deben realizar mejoras para la próxima iteración de diseño en función de los hallazgos.

2.3.3 Desarrollar

Desarrollar es la última etapa del DLD. Una vez probado el diseño de software con los usuarios, ahora se debe convertir la solución en realidad. Para esto se valida, ajusta y publica el producto.

2.3.3.1 Implementar

Implementar es la primera sub-etapa de desarrollar. Sus objetivos son dar vida a la solución real, cubrir el desarrollo Front-End y back-End, además de generar documentos que pueden ser de utilidad.

A. Mapeo de historias de usuario

Tiempo: Un día.

Personas: Equipo del proyecto.

Se necesita: Una gran pared, notas adhesivas de diferentes colores y rotuladores.

El mapeo de historias de usuario o User Story Mapping es un método ágil que ayuda a comprender mejor el panorama general del producto y a generar una comprensión común de las prioridades del usuario. La estructura del mapeo de historias de usuario permite identificar el MVP (productivo mínimo viable) para satisfacer a los primeros usuarios de acuerdo a las prioridades que se hayan fijado.

Los pasos para hacer un mapeo de historias de historias son:

- Reunir todos los artefactos: Persona, Storyboard y prototipos.
- Crear un gráfico de mapeo de historias de usuario en una pared grande.
- Describir la secuencia del uso del producto.
- Pensar en cuales son los principales pasos de uso que hará un usuario con el producto.
- Luego, dividir los pasos de uso en actividades de los diferentes personajes que interactúan en cada paso.
- Crear historias de usuario: Para cada actividad, describir que tareas debe realizar el usuario para completar la actividad.
- Priorizar historias de usuario y de acuerdo a esto, obtener una acumulación para el primer lanzamiento.

2.3.3.2 Probar

Probar es la segunda sub-etapa de Desarrollar. Sus objetivos son asegurar que el diseño funcione según lo planeado y cumpla con los objetivos de rendimientos. Además se valida la conveniencia y viabilidad del producto.

A. Realizar prueba de usabilidad

Tiempo: 30-60 minutos por participante.

Personas: Usuarios que coincidan con el perfil esperado.

Se necesita: Un conjunto de tareas y preguntas, un prototipo, bocetos, etc.

Es importante probar cómo los usuarios interactúan con el diseño y modificarlo si es necesario, al principio de la fase de diseño o desarrollo, antes que hacer cambios se vuelva más difícil. La usabilidad es un término que describe como los usuarios perciben el diseño de efectividad general y su nivel de satisfacción al trabajar con él. Obtener comentarios de los usuarios ayuda a comprender las áreas problemáticas y saber cuándo y dónde realizar mejoras.

Los pasos para realizar una prueba de usabilidad son:

- Pensar por qué se requiere probar el diseño y que preguntas necesita que respondan los usuarios.
- Determinar quién debe realizar la prueba (además, se necesita de un moderador).
- Escribir una lista de tareas para que realice el usuario.
- Preparar un prototipo del diseño.
- Encontrar personas que coincidan con los usuarios esperados y que estén dispuestas a realizar la prueba.
- Analizar los resultados y compartir hallazgos con el equipo.

2.3.3.4 Desplegar

Desplegar es la tercera sub-etapa de desarrollar. Su objetivo es liberar la solución para la producción final, entregar el producto terminado a los clientes e involucrar a los primeros usuarios, de los cuales también se espera recopilar comentarios.

A. Importar proyecto de compilación de SAP Build a Web IDE.

Tiempo: 15 minutos.

Personas: Encargado de la cuenta de SAP Build.

Se necesita: Cuenta en SAP Cloud Platform.

Se importa un prototipo completo y publicado en SAP BUILD a SAP Web IDE. Esto permite convertir el prototipo en una aplicación en vivo. Este prototipo se convertirá en plantilla perfecta para crear la versión real usando UI5/Fiori.

Los pasos para importar un proyecto de compilación de SAP Build al Web IDE son:

- Habilitar los servicios de SAP Build.
- Importar los destinos.
- Abrir SAP Web IDE.
- Habilitar el complemento de compilación de SAP.
- Crear un nuevo proyecto.
- Seleccionar un proyecto para importar y ejecutar la aplicación.

2.4 Resumen de la Metodología

La Figura 12 presenta un diagrama de procedimiento de la metodología Design-Led Development con las actividades y tiempos asociados. Además, la metodología propone la participación de; un Project Manager como líder y gestor del proyecto, iteraciones y habilitador; un Product Owner como representante de los interesados en el equipo del proyecto; un DT Coach experto en Design Thinking; y un equipo ejecutor del proyecto. El tamaño del equipo dependerá del contexto de la situación pero no debería superar las 6 personas, por lo que en total se requiere una cantidad de 9 participantes activos del proyecto. Esporádicamente se puede incluir algún experto en el tema o a los interesados del proyecto.

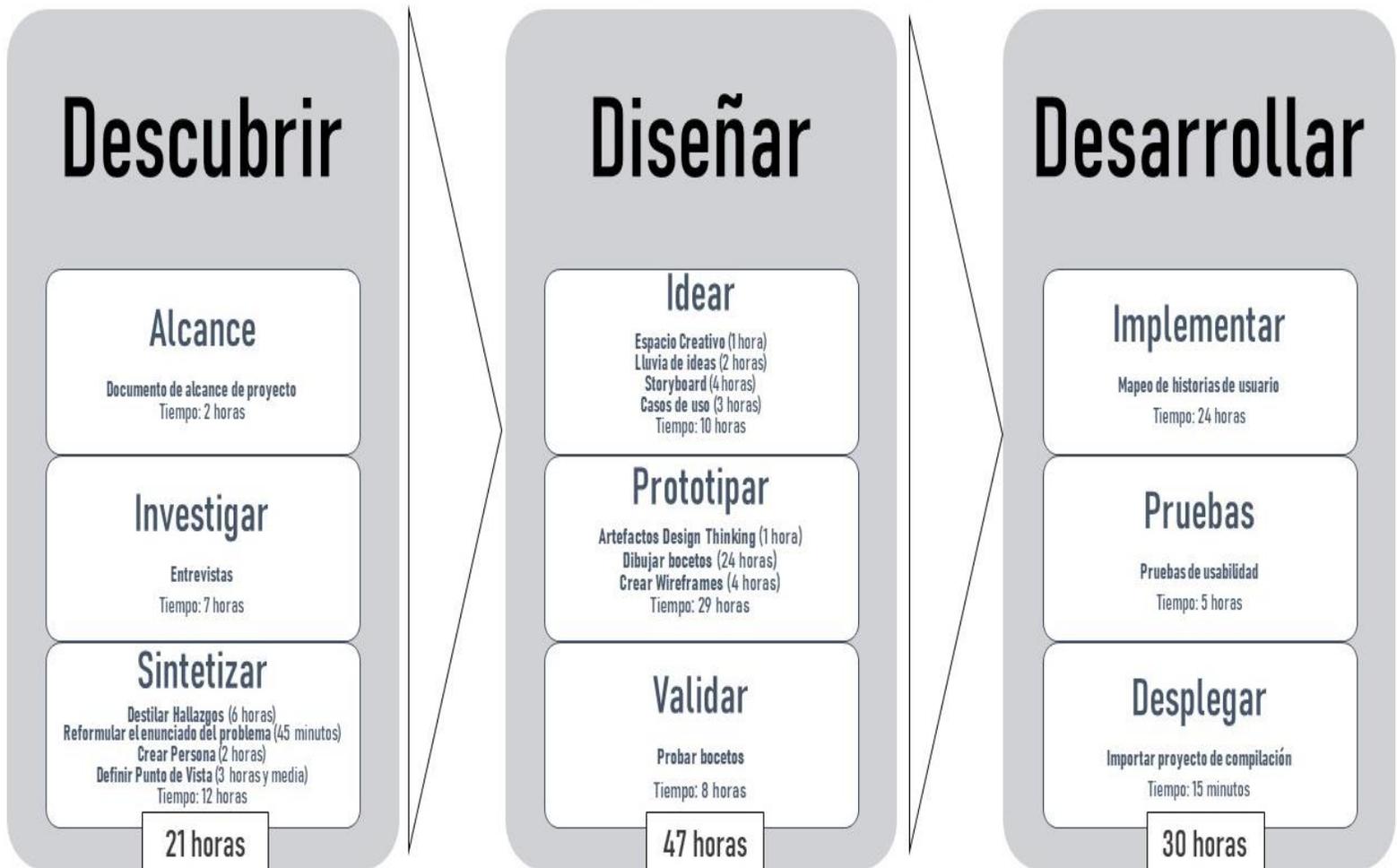


Figura 12 - Diagrama de procedimientos Design-Led Development DLD (Elaboración propia)

En consideración a los tiempos propuestos por cada etapa y un trabajo efectivo de 5 horas al día por parte del equipo, se recomiendan las siguientes duraciones aproximadas: 4 días para descubrir, 10 días para diseñar y 6 días para desarrollar. En conclusión, teniendo en cuenta trabajar 5 días a la semana, para un equipo conformado por las personas indicadas, debería tomar un mes cada iteración.

2.5 Trabajos e Investigaciones Similares

Al consultar la literatura con respecto a la metodología seleccionada, no fueron encontrados casos de estudio. En el blog de SAP, en las entradas relacionadas a Build y Design-Led Development, personas declaran usarlo en sus trabajos, sin embargo, al parecer, no se han escrito artículos relacionados al tema, por lo que es una gran motivación seleccionar esta metodología para el estudio.

Con respecto a Design Thinking existe más material. En el contexto chileno se destaca una investigación (Castillo, Alvarez & Cabana, 2014) realizada en la Universidad de la Serena en donde se muestra el uso del Design Thinking en la resolución de problemas, elaboración de modelo de negocios, planificación estratégica y el desarrollo de ideas, por lo que señalan que las empresas requieren cada vez más del uso de esta metodología. Este estudio describe la metodología y herramientas disponibles para su aplicación, y propone la metodología en cuestión para el ámbito del aprendizaje de estudiantes, empresarios y emprendedores.

Entre otra literatura consultada, se destaca un estudio sobre el uso del Design Thinking como enfoque metodológico para la instrucción de ingeniería de software a nivel de pregrado, en cursos que tienen como objetivo particular crear productos de software innovadores desde cero (Corral & Fronza, 2018). El artículo muestra dos casos de estudio utilizando Design Thinking y Agile por separado. Los dos cursos se completaron según lo planeado y programado dentro del período de tiempo asignado. No se identificaron incidencias importantes (por ejemplo, abandono de cursos). En ambos entornos de trabajo, cada equipo pudo ofrecer una solución de software funcional. Sin embargo, se identificaron varios puntos en común y diferencias entre los dos entornos de trabajo descritos a continuación:

- Ambos grupos dirigieron sus esfuerzos al desarrollo de software centrando en el humano.
- El grupo que implementó Agile fue más efectivo aprovechando las iteraciones para mejorar la calidad del producto final, mientras que el grupo que usó Design Thinking se concentró en la construcción de la solución, reduciendo el alcance del producto final a un producto mínimo viable.

- El grupo Agile implementó un ritmo de entrega impulsado por software que incluía una serie de productos intermedios que se utilizaron para crear documentación. El grupo que usó Design Thinking no entró documentación.
- El grupo Agile desarrolló una serie de soluciones cuyo valor puede ser desafiado por múltiples segmentos de usuario. Los grupos de Design Thinking convergieron de forma natural para implementar soluciones más relevantes para las necesidades de su propio entorno.
- Independiente de la metodología utilizada, los equipos pudieron gestionar su trabajo de forma independiente con una mínima intervención del profesorado.
- Ambos grupos pudieron ofrecer un discurso profesional a un panel independiente de expertos para discutir y proporcionar argumentos claros para sustentar sus ideas.

Por último, se destaca una revisión de literatura (Pereira & Russo, 2018) que recopiló, categorizó y revisó 29 artículos relacionados al tema. El autor propone al Design Thinking como un enfoque integrado con las metodologías de desarrollo de software ágil, que promueve la comprensión de las necesidades del cliente. Entre los hallazgos de esta revisión se encuentra:

- El desarrollo de software ágil promueve la colaboración de los interesados del proyecto, con transparencia y mayor valor percibido por parte del cliente. Esto mejora la calidad del producto y reduce el tiempo para abordar posibles errores.
- El 84% de las organizaciones aún se encuentra en un nivel ágil de baja madurez y el 40% de los practicantes ágiles señalan este método como una forma de gestionar equipos distribuidos, mientras que el 22% lo indica como un método para reducir costo en proyectos de desarrollo de software.
- El enfoque Design Thinking o HCD (Human Centered Design) actúa a partir de las aplicaciones del pensamiento de diseño del diseñador (Design Thinker) para identificar problemas y construir soluciones deseables para los clientes, que sean tecnológicamente factibles y viables para el negocio de la organización involucrada.

3. Contexto del Desafío

En particular, el proyecto se desarrolla en la empresa Essbio, la cual se dedica al rubro de los servicios sanitarios. En este capítulo se describen los datos generales de la organización, se presentan los organigramas estructurales hasta específicamente la subgerencia de informática y el departamento de sistemas operacionales y BI, el cual es el principal impulsor del proyecto. Finalmente, el capítulo presenta el problema, el entorno tecnológico con el que se cuenta y el objetivo empresarial del proyecto.

3.1 Datos Generales de la Empresa

A continuación son descritos los datos organizacionales correspondientes a Essbio, disponibles en su sitio web oficial, a los que se accedió durante el mes de julio del 2021.

3.1.1 Misión

Proveer soluciones y servicios de alta calidad en el ciclo integral del agua (agua potable, alcantarillado y descontaminación de aguas servidas) y en la gestión integral de residuos. Entregar un servicio de excelencia, basado en un profundo conocimiento de nuestros clientes, altos niveles de eficiencia e innovación, y un verdadero compromiso de responsabilidad social con la comunidad y el medioambiente. En el contexto de la permanente generación de valor para nuestros accionistas. Todo esto gracias al trabajo de un equipo de personas satisfechas y orgullosas con su vida.

3.1.2 Visión

Ser un operador de clase mundial en servicios sanitarios al que sus clientes elegirían por su eficiencia y alta calidad de servicio.

3.1.3 Valores Corporativos

- ✓ **Transparencia:** Valor orientado del trabajo, que buscar realizar acciones y tener conductas que no oculten o silencien la realidad de los hechos.
- ✓ **Confianza:** Enfocado a que las personas realicen acciones coherentes y consistentes, de manera que los otros lo perciban creíble y predecible, afianzando la relación.
- ✓ **Respeto:** Orientado a relacionarse con los otros, reconociendo, aceptado y apreciando sus cualidades y derechos hacia todos los sectores.
- ✓ **Excelencia:** Vista como una práctica común de cada miembro de la compañía, sustentable la confianza y permanente el respeto.

3.1.4 Organigrama General

La Figura 13 contiene el organigrama general de la organización. Todos los organigramas presentados en las siguientes Figuras fueron obtenidos de la intranet institucional dispuesta por la empresa.

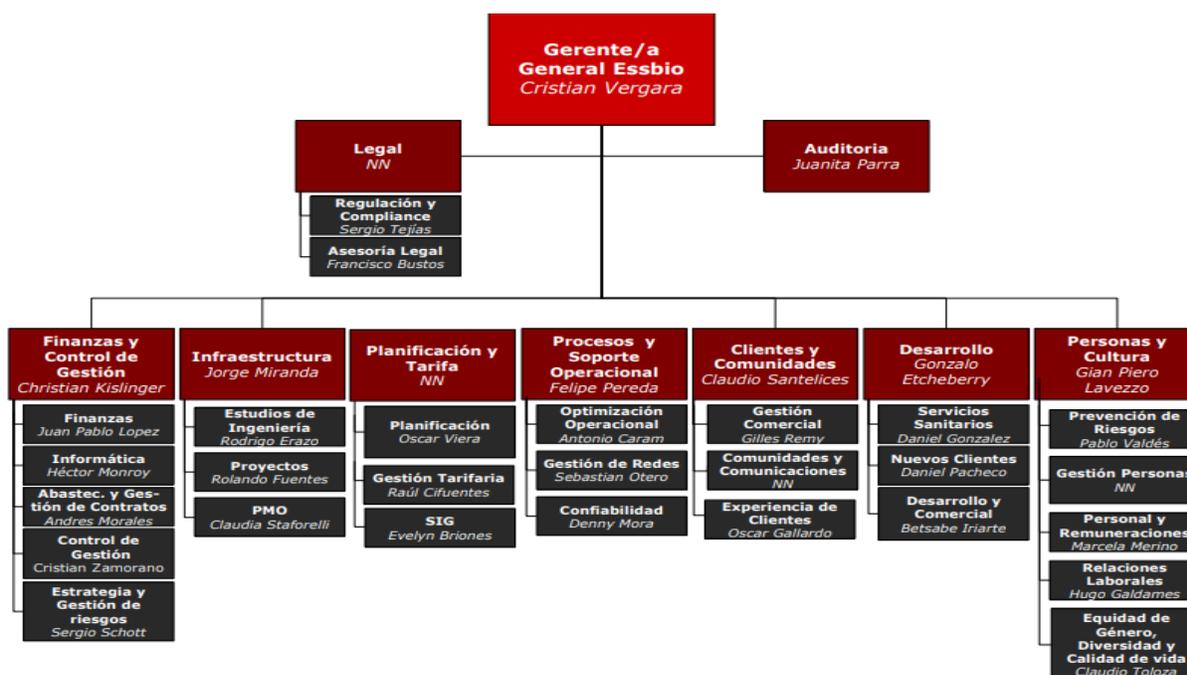


Figura 13 - Organigrama General de Essbio.

Se puede observar que Gerencia General trabaja en conjunto con Gerencia Legal y Gerencia de Auditoría.

Las demás gerencias son:

- **Gerencia de Finanzas y Control de Gestión.**
- Gerencia de Infraestructura.
- Gerencia de Planificación y Tarifa.
- Gerencia de Procesos y Soporte Operacional.
- Gerencia de Clientes y Comunidades.
- Gerencia de Desarrollo.
- Gerencia de Personas y Cultura.

La Figura 14 muestra las definiciones de éxito de la estrategia de la empresa. El éxito ha sido dividido en seis focos, para los cuales existe un líder responsable y una definición de éxito al año 2023.



Figura 14 - Definiciones de éxito de la estrategia Essbio (2021).

La Figura 15 muestra el mapa estratégico general de Essbio definido este año.

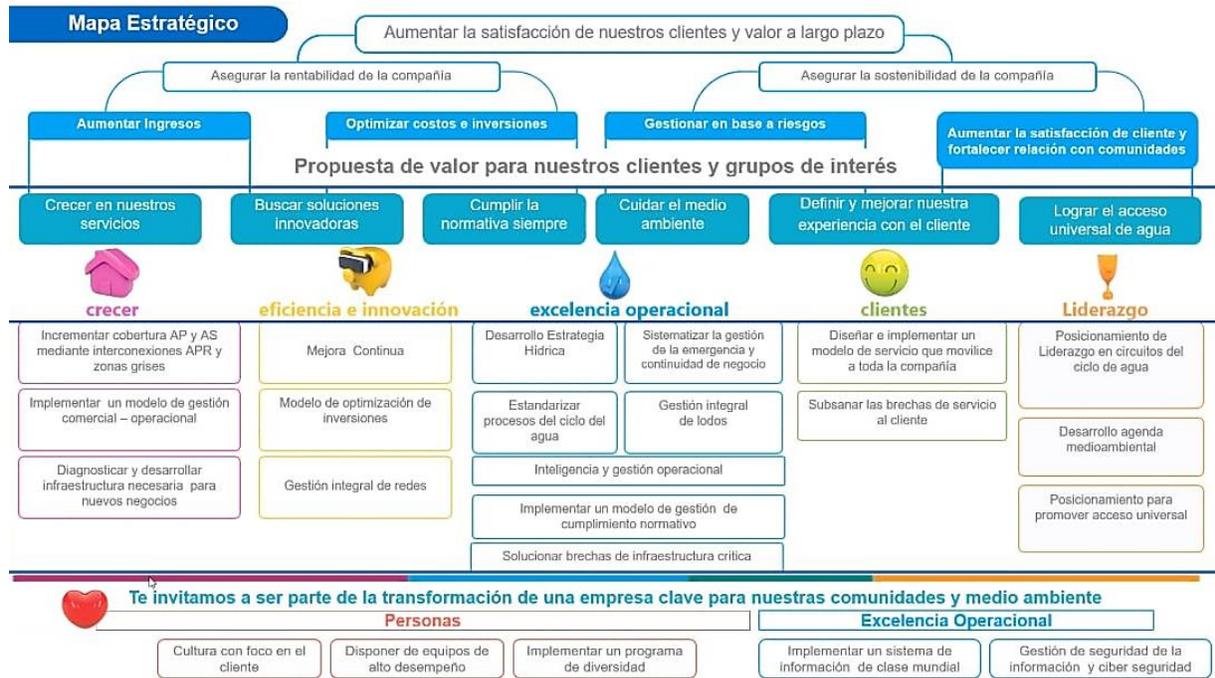


Figura 15 - Mapa Estratégico Essbio (2021).

3.2 Gerencia de Finanzas y Control de Gestión

El desarrollo del proyecto es realizado en específico bajo el alero de la Subgerencia de Informática. Esta unidad pertenece a la Gerencia de Finanzas y Control de Gestión. La Figura 16 presenta el organigrama de la Gerencia de Finanzas y Control de Gestión:

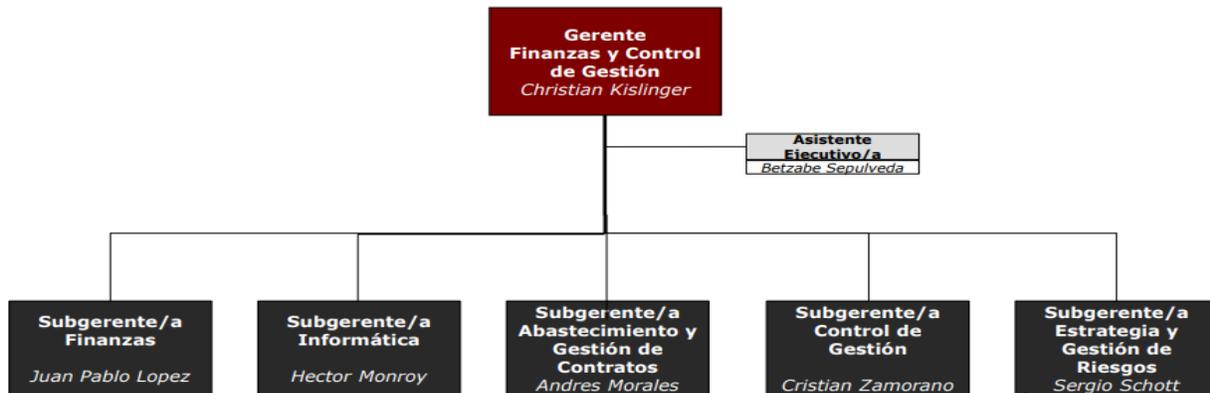


Figura 16 – Organigrama Gerencia de Finanzas y Control de Gestión Essbio.

El objetivo de la Gerencia de Finanzas y Control de Gestión es contribuir a la formulación y consecución de los desafíos estratégicos de la compañía. En este sentido, a través del área de Informática, la gerencia promueve el uso de las tecnologías disponibles, como también el desarrollo de nuevos proyectos que permiten lograr la optimización y eficacia de los procesos de negocio.

En términos de Abastecimiento y Gestión de Contratos, se esfuerza en lograr negociaciones con los contratistas que sean económicamente sustentables y operacionalmente eficientes, como también abastecer oportunamente los insumos requeridos por las distintas áreas. Junto con lo anterior, el resguardo de los activos fijos de la compañía, como también una adecuada cobertura de seguros de estos, son desafíos permanentes del área de Abastecimiento.

Control de Gestión juega un papel clave en el análisis y control de las actividades que desarrollan las distintas áreas de la compañía, de manera de contribuir a mejorar el desempeño de éstas para alcanzar las metas establecidas en términos financieros, operacionales y de inversiones. Por su parte, el área de Finanzas permite un adecuado cumplimiento de los requerimientos normativos, tanto tributarios como también financieros que rigen a las sociedades anónimas. Además, la administración del capital de trabajo y flujo de caja son un imperativo financiero permanente.

Finalmente, a través del área de Riesgos, la gerencia asegura el control y seguimiento de los riesgos de negocio que enfrenta constantemente la compañía en el desarrollo de sus actividades, especialmente los riesgos asociados a la industria sanitaria.

La Gerencia de Finanzas y Control de Gestión está compuesto por 5 subgerencias:

- Subgerencia de Finanzas
- **Subgerencia de Informática**
- Subgerencia de Abastecimiento y Gestión de Contratos
- Subgerencia de Control de Gestión
- Subgerencia de Estrategia y Gestión de Riesgos

3.2.1 Datos Generales Subgerencia de Informática

La Figura 17 presenta el organigrama de la Subgerencia de Informática:

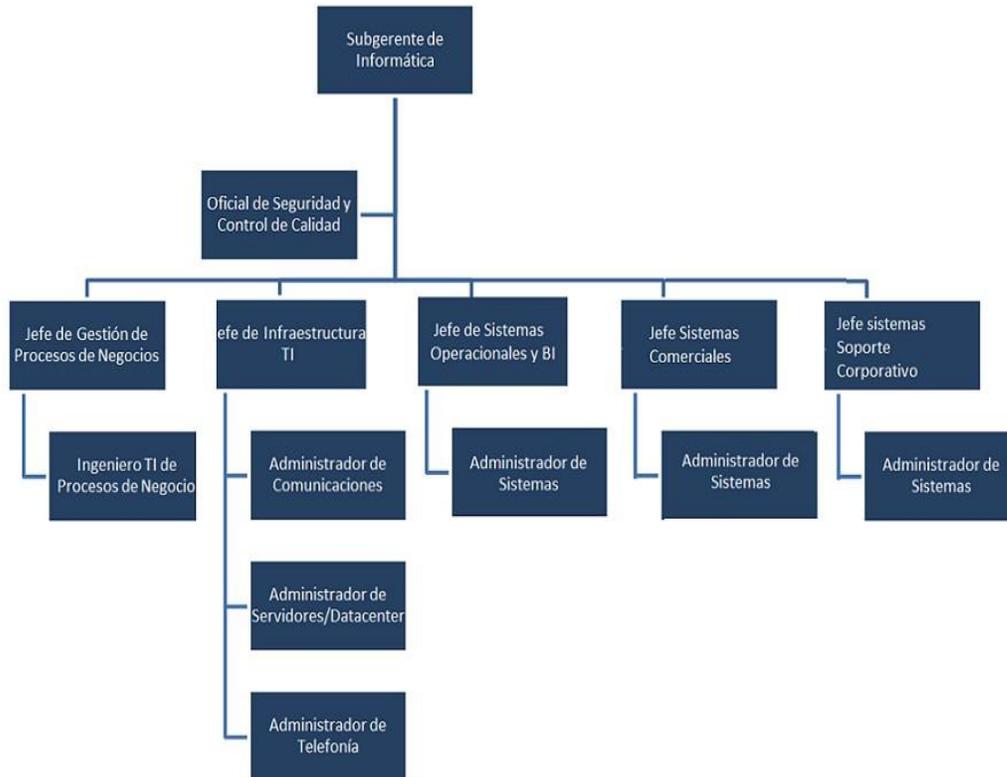


Figura 17 - Organigrama Subgerencia de Informática Essbio

Las divisiones que componen la Subgerencia de Informática son:

- Gestión de Procesos de Negocio.
- Infraestructura TI.
- **Sistemas Operacionales y BI.**
- Sistemas Comerciales.
- Sistema Soporte Corporativo.

Transversal a estas divisiones se encuentra el puesto del Oficial de Seguridad y Control de Calidad. A continuación, es descrita la misión y visión de la subgerencia de informática y, además, se presenta la Figura 18 que plantea el alineamiento estratégico de la unidad.

Visión

Llegar a ser en el corto plazo un actor relevante en la formulación y materialización de la estrategia de la Empresa.

Misión

Contribuir al desarrollo de ventajas competitivas sustentables desde el ámbito de las tecnologías de la información y la gestión de los procesos:

- A. Promoviendo el uso de las tecnologías de la información en la empresa.
- B. Desarrollando proyectos que contribuyan a materializar la estrategia de la empresa.
- C. Desarrollando un equipo humano comprometido con la estrategia y la cultura de la empresa.



Figura 18 - Visión Transversal: Alineamiento Estratégico. Subgerencia de informática Essbio.

3.3 Problemática

Específicamente, en esta sección es descrita la problemática de manera general, se menciona el entorno tecnológico y herramientas utilizadas, para finalmente se exponer el objetivo empresarial que busca cumplir el proyecto.

3.3.1 Descripción General

El proceso que se necesita implementar, es la lectura en piso de planta de parámetros de producción de agua potable y tratamiento de aguas servidas en equipos, sobre 261 plantas en las regiones VI, VII y VIII de Chile. Actualmente, el 25% de los equipos de plantas son leídos a través de dispositivos SCADA⁶, mientras que el resto son leídos manualmente por los operadores. La mayoría de los operadores suben datos desde las centrales administrativas a través de computadoras de escritorio o portátiles.

Por lo que el objetivo es disminuir el tiempo que les requiere a los operadores subir los datos correspondientes a las mediciones de parámetros de los equipos de planta, con la finalidad de controlarlos e integrarlos con respecto a las variables operacionales de la empresa, para así obtener una mejor calidad de la información y apoyar en la toma de decisiones.

La Tabla 5 identifica las áreas interesadas en el valor que puede aportar el desarrollo de esta solución tecnológica:

Área Ejecutora	Área de Negocio
Gerencia de Administración y Finanzas	Gerencia de Procesos y Soporte Operacional
Subgerencia de Informática	Subgerencia de Confiabilidad
Departamento de Gestión de Procesos Operacionales e Inteligencia de Negocios.	Gerencias Regionales VI, VII y VIII.

Tabla 5 - Áreas ejecutoras y área de negocios

⁶ SCADA: Sistema informático para la supervisión y control de procesos que recogen y analizan datos en tiempo real.

3.3.2 Entorno tecnológico

Las herramientas utilizadas para llevar a cabo este proyecto se pueden conseguir gratuitamente. Estas son: SAP Build, SAP Business Application Studio y SAP Fiori. A continuación se describe cada una de ellas.

SAP Build⁷

La experiencia de usuario (UX) es el término de moda en el universo TI y hace referencia a las aplicaciones que están bien diseñadas, son intuitivas y fáciles de usar. Las apps móviles a las que todas las personas acceden desde sus teléfonos o sus tablets significaron en este sentido un punto sin retorno: el software corporativo críptico, difícil de usar y con más pasos de los necesarios para resolver una tarea sencilla ya no puede existir. Históricamente, los usuarios corporativos mostraban su mejor sonrisa durante las etapas de relevamiento, cuando contaban qué querían y cómo lo querían, y luego exhibían “sus dientes” en el momento de la prueba, ya que consideraban que eso que estaban recibiendo no se parecía en casi nada con lo que habían solicitado.

Para cerrar esta brecha existe SAP Build: un conjunto de herramientas basadas en servicios que permiten aprender, generar prototipos, sumar el Feedback de los usuarios e inspirarse en el trabajo ya recorrido por otros desarrolladores. En definitiva, humaniza el proceso de desarrollo de aplicaciones. La Figura 19 muestra la interfaz principal del sitio oficial de Build.

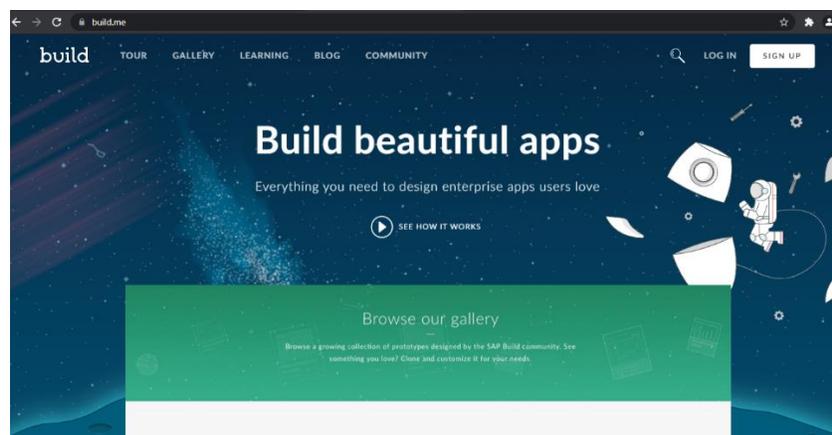


Figura 19 - Portal del sistema Web SAP Build

⁷ SAP Build, definición obtenida desde <https://news.sap.com/latinamerica/2017/09/como-es-sap-build-la-herramienta-gratuita-de-sap-para-desarrollar-aplicaciones-que-funcionan/>

En el aspecto de aprendizaje, se apoya en Design-Led Development, una técnica de desarrollo que consiste en enfocarse en los usuarios para los cuales se creará la aplicación. Se obtiene información sobre lo que estos dicen, piensan, sienten y hacen. El proceso es complejo, pero con el tiempo y la experiencia se vuelve intuitivo. SAP Build incorpora herramientas e instrucciones paso a paso para que el equipo de desarrollo avance siempre en la dirección correcta.

Respecto del prototipado, permite generar maquetas interactivas en base a los bocetos. Pero no se trata de una muestra estática y aburrida de la solución: es posible convertir los bocetos en prototipos refinados y transformarlos a su vez en aplicaciones terminadas sólo arrastrando y soltando elementos incorporados a la familia de herramientas.

La recolección del Feedback no se limita a escuchar las quejas (o las felicitaciones) de los usuarios: SAP Build incorpora propuestas para crear de manera sencilla estudios de usabilidad remotos, de forma tal que cada interacción que un usuario tenga con la aplicación es una fuente de información para mejorarla. También existen estrategias para recolectar y aprovechar los comentarios informales sobre el producto.

Por último, la comunidad de desarrolladores Build está construyendo nuevos prototipos todo el tiempo, que los demás pueden personalizar de acuerdo a su aplicación y a sus necesidades. La Figura 20 presenta el espacio de trabajo de Build. La versión gratuita de esta herramienta solo permite colaborar en un máximo de tres proyectos en paralelo. Los proyectos propios pueden ser archivados en caso de querer participar en un cuarto proyecto. Además, el sistema notifica las invitaciones para participar en proyectos gestionados por otro usuario de Build.

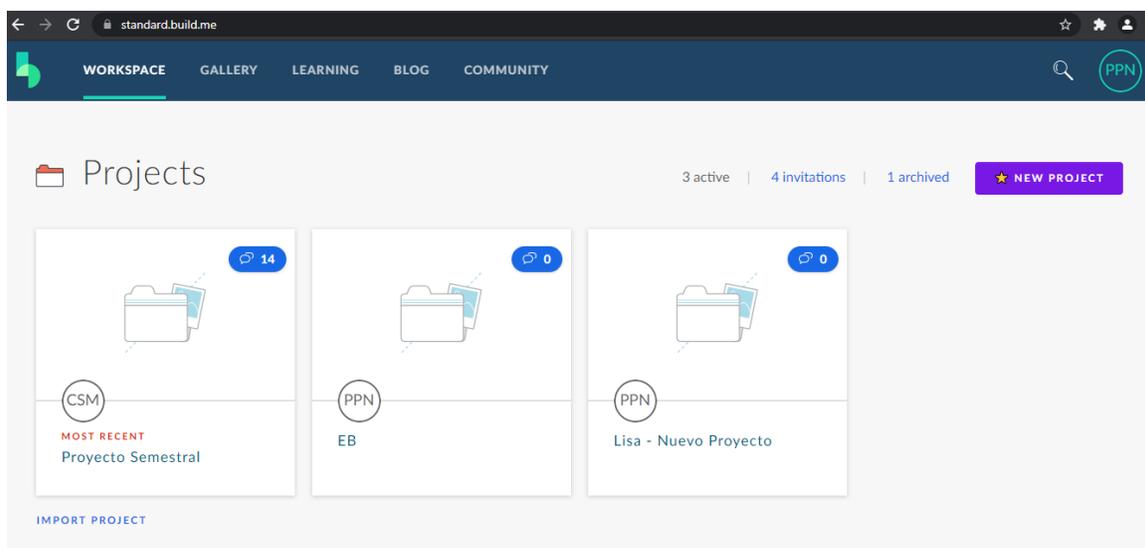


Figura 20 - Espacio de trabajo de SAP Build

SAP Business Application Studio⁸

SAP Business Application Studio es un nuevo servicio de SAP Business Technology Platform (BTP) que ofrece un entorno de desarrollo moderno adaptado para el desarrollo eficiente de aplicaciones comerciales para SAP Intelligent Enterprise.

SAP Business Application Studio se basa en un IDE de código abierto, Eclipse Theia, que adopta la experiencia de Microsoft Visual Studio Code. Disponible como un servicio en la nube, SAP Business Application Studio proporciona una experiencia similar a la de un escritorio similar a los IDE líderes, con línea de comandos y editores optimizados. En el corazón de SAP Business Application Studio se encuentran los espacios de desarrollo. Estos espacios son comparables a las máquinas virtuales aisladas en la nube que contienen herramientas personalizadas y tiempos de ejecución preinstalados por escenario empresarial, como SAP Fiori, extensiones de SAP S / 4HANA, Workflow, Mobile y más. La Figura 21 es una captura del directorio principal de SAP Business Application Studio.

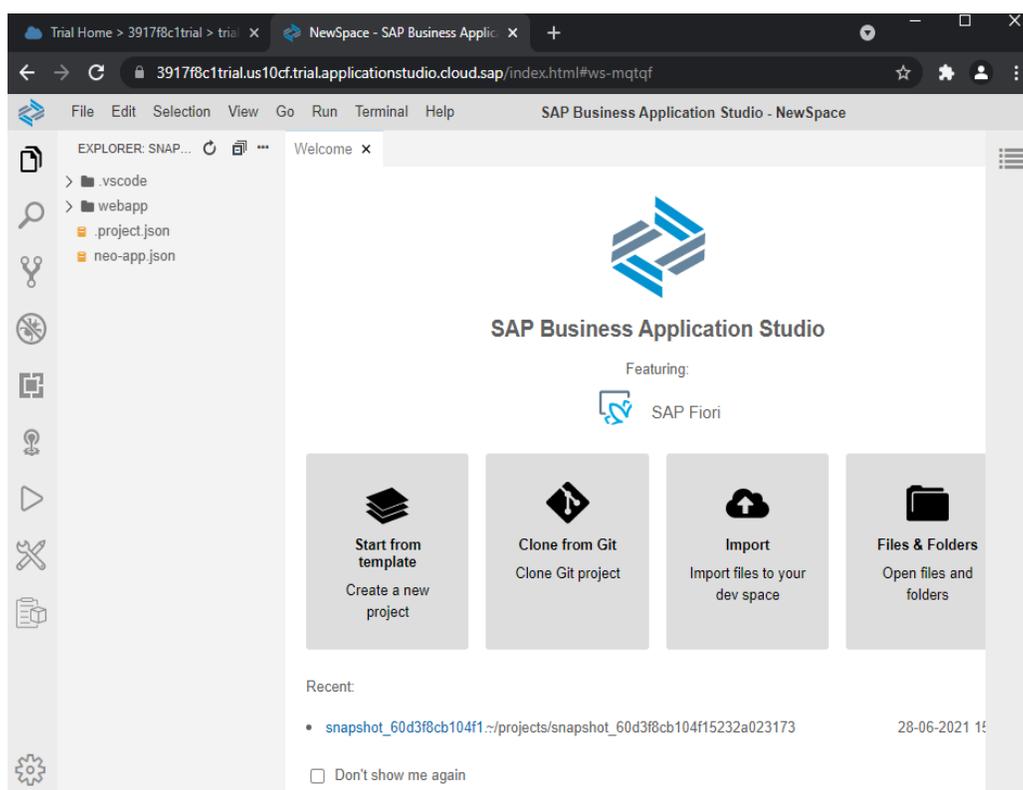


Figura 21 - Captura de un directorio en SAP Business Application Studio

⁸ SAP Business Application Studio, definición obtenida desde <https://help.sap.com/>

SAP Fiori⁹

SAP Fiori es la experiencia de usuario de la empresa inteligente que cambia su forma de trabajar. Equipa a diseñadores y desarrolladores con un conjunto de herramientas y directrices para crear apps para cualquier plataforma más rápido que nunca, brindando una experiencia coherente e innovadora tanto para los creadores como para los usuarios.

Las soluciones de SAP Fiori fueron creadas no solo bajo los conceptos de la simplicidad, interactividad y la eficacia, sino también bajo un esquema de portabilidad muy amplio. El motivo de ello es que gracias a la tecnología desarrollada por esta compañía, este software funciona como si fuera una aplicación sin necesidad de serlo. La instalación se suprime al lograr que las aplicaciones del entorno SAP Fiori funcionen directamente en el navegador de Internet de preferencia de cada profesional. Es posible sacar provecho de este sistema tanto desde el ordenador como desde el teléfono móvil, lo que hace que su acceso sea fácil y rápido, evitando las limitaciones que trae consigo una instalación que se realice localmente en un almacenamiento de memoria.

Las ventajas de SAP Fiori pueden resumirse en cinco puntos:

1. **Simplicidad:** Fiori se basa en la simplicidad, por lo que su sistema es capaz de desmenuzar tareas y aplicaciones complejas y convertirlas en otras más simples, consiguiendo que esos procesos no sean tan tediosos.
2. **Adaptabilidad:** Se adapta a cualquier navegador o sistema, ayudando a que llegue a una gran cantidad de usuarios sin ningún tipo de problema.
3. **Integración:** SAP Fiori permite integrar distintas aplicaciones en un plano siguiendo un sistema conocido como 1-1-3, el cual consta de 1 escenario, 1 usuario y 3 pantallas para integrar distintas funciones y hacer más simple el funcionamiento.
4. **Coherencia:** con Fiori se consigue coherencia, dado que **todo encaja en un concepto único** que extiende SAP en cada uno de sus servicios. Las funciones y las aplicaciones se desarrollan en paralelo a la política de SAP y esto garantiza que los profesionales siempre tendrán la seguridad de que caminan por terreno seguro.
5. **Valor añadido:** Con esta tecnología hay que contar todos los puntos positivos anteriores y añadir que también destaca en su interfaz y su presentación, por no decir que el soporte que se proporciona a los profesionales es de primera línea, resultando muy útil en todo tipo de casos.

⁹ SAP Fiori, definición obtenida desde: <https://www.sap.com/latinamerica/products/fiori.html>

3.3.3 Objetivo empresarial del proyecto

Utilizando las herramientas descritas con anterioridad: SAP Build y SAP Business Application Studio; se desarrollará una solución tecnológica con el objetivo de minimizar el tiempo requerido por los operadores para tomar la lectura de los datos asociados a los parámetros de los equipos, mientras se recorre el piso de planta.

4. Desarrollo de una Solución

Antes de comenzar a detallar la ejecución del proyecto es importante considerar los riesgos y las mitigaciones correspondientes, por lo que desde la empresa se solicitó realizar este análisis. El resultado del análisis es sintetizado en la Tabla 6.

Riesgo	Descripción	Mitigación	Contingencia
Falta de usuarios.	El grupo de usuarios completo o parcialmente no pueden asistir a una reunión.	Planificar las reuniones con anterioridad.	Buscar usuario de emergencia o alguien del equipo simulará la participación del usuario ausente.
Problemas de conexión en la reunión.	Si el grupo de participantes completo o parcialmente presenta problemas de conexión durante una reunión.	Avisar problemas de conectividad previos a la reunión.	La reunión será reprogramada a convenir.
BUILD offline.	La web de la herramienta BUILD es bajada.	Documentar todo el contenido de la web que pueda recopilarse y mantener una copia en otra plataforma.	Usar el respaldo del contenido. En caso de ser un prototipo de alto nivel, comunicarse con algún operador de SAP.

Código ilegible o complicado.	El código del proyecto es difícil de entender o está desordenado.	Comentar código para facilitar su comprensión y refactorización.	Acceder al documento oficial del proyecto de título para analizar los avances del proyecto.
-------------------------------	---	--	---

Tabla 6 – Riesgos, mitigaciones y contingencias

Además, se realizó una propuesta de planificación inicial, presentada en la Tabla 7. Entre la primera fecha de inicio hasta la última fecha de fin hay 54 días.

Etapa	Subetapa	Inicio	Fin	Duración
Descubrimiento	Alcance	21-06-2021	22-06-2021	2
	Investigación	23-06-2021	25-06-2021	3
	Sintetizar	25-06-2021	09-07-2021	15
Diseño	Idear	12-07-2021	15-07-2021	4
	Prototipo	19-07-2021	23-07-2021	5

	Validar	26-07-2021	27-07-2021	2
Desarrollo	Implementación	02-08-2021	05-08-2021	4
	Test	06-08-2021	10-08-2021	5
	Desplegar	11-08-2021	13-08-2021	3

Tabla 7 – Planificación inicial del proyecto

4.1 Descubrir

Descubrir es la primera etapa del DLD. En específico, esta etapa está dividida en 3 subetapas: alcance, investigación y sintetizar. El objetivo de esta etapa es identificar el problema a resolver y para quién será resuelto.

4.1.1 Alcance

Alcance es la primera subetapa de descubrir y el objetivo es comunicar una comprensión clara del problema, involucrar a las personas adecuadas y planificar orientando a maximizar tanto el trabajo como el tiempo del proyecto.

Todo este análisis es plasmado en un documento de alcance, este documento cuenta con ciertas características:

- Es un documento dinámico que itera a lo largo del proyecto.
- Ayuda a comprender y reducir el alcance del proyecto.
- Permite evaluar la necesidad de cambios en el alcance y la gestión de riesgos.
- Mejora la comunicación de la comprensión del problema.

Este documento está compuesto por los siguientes elementos:

1. **Objetivo general:** Define para quién se está diseñando una solución, que necesidad comercial se está abordado en el proyecto y cuál es la propuesta de valor única. Esta parte se puede ver como una especie de discurso de marketing del proyecto y puede ir cambiando en el tiempo.
2. **Alcance del proyecto:** Describe lo que está en el alcance del proyecto en la iteración correspondiente. Por ejemplo; una prueba de concepto, o bien, una introducción del producto o servicio en el mercado.
3. **Partes interesadas:** Se enumeran los patrocinadores y las partes interesadas del proyecto.
4. **Organización del proyecto:** Describe el equipo del proyecto y sus funciones principales.
5. **Entregables del proyecto:** Se enumeran los resultados esperados del proyecto. Además, se pueden incluir criterios de aceptación (condiciones para aceptar los distintos entregables).
6. **Hitos:** Describe un cronograma aproximado del proyecto, con los principales hitos y fechas.

A continuación, se presentan las partes del documento de alcance para este proyecto. El documento completo está en el Anexo 1.

Nombre del documento: Proyecto toma de mediciones en terreno (Fiori & Build).

1. **Objetivo general:** Disminuir el tiempo que les requiere a los operarios subir los datos correspondientes a las mediciones de parámetros al sistema, con la finalidad de controlar estos datos con respecto a las variables operacionales de la empresa, para obtener una mejor calidad de información y apoyar en la toma de decisiones.
2. **Alcance del proyecto:** Implementar un prototipo del sistema de toma de mediciones en terreno utilizando SAP Fiori UI5 y la herramienta SAP Build, y subirlo al entorno de desarrollo integrado web proporcionado el Business Technology Platform.
3. **Partes interesadas:**
 - a. Gerencia Procesos y Soporte Operacional.
 - b. Subgerencia de Confiabilidad.
 - c. Gerencias Regionales VI, VII y VIII.
 - d. Gerencia Administración y Finanzas / Subgerencia de Informática / Departamento de gestión de procesos operacionales e inteligencia de negocios (Área ejecutora).

4. **Organización del proyecto:**

- a. **Coordinadores:** Patricio Pardo, Elizabeth Grandón, Jefe de departamento Sistemas Operacionales y BI.
- b. **Contribuyentes Directos: Operadores.**
- c. **Contribuyentes Apoyo:** Subgerente de Confiabilidad.

5. **Entregables del proyecto:**

- a. Artefactos Persona, Replanteamiento del Problema y Punto de Vista.
- b. Prototipo de alta fidelidad.
- c. Código del prototipo del proyecto.

6. **Hitos:**

- a. Entregables etapa DESCUBRIR (Alcance, Investigación y Sintetizar) 12-07-2021.
- b. Entregables etapa DISEÑO (Idear, Prototipo y Validación) 28-07-2021.
- c. Entregables etapa DESARROLLO (Implementación, Test y Desplegar) 13-08-2021.

4.1.2 Investigar

Investigar es la segunda subetapa de descubrir y el objetivo es conocer las necesidades reales de los usuarios y descubrir oportunidades para mejorar su estilo de vida. Para lograr este objetivo, la metodología recomienda realizar una serie de entrevistas pero antes, hay que realizar un guion.

El guion de la entrevista consta de preguntas y notas para guiar la entrevista, ayuda a garantizar que la sesión de investigación cubra todos los temas en un orden que proporcione al equipo la información necesaria. El guion debe incluir; una introducción y configuración de contexto, preguntas de antecedentes, preguntas de alto nivel, preguntas detalladas y un cierre. Además, se recomienda definir algunas reglas antes de comenzar una entrevista, para tener definido los pasos a realizar en caso de algún inconveniente. El Script generado para realizar las entrevistas en este proyecto está en el Anexo 2. La Tabla 8 contiene el nombre de los usuario, cargo y hora en que se realizó la entrevista.

Nombre del Usuario	Fecha y Hora	Cargo
OPERADOR AP 1	23-06-2021 10:00	Operador producción agua potable Biobío.
OPERADOR AP 2	23-06-2021 11:30	Operador producción agua potable O'Higgins.
OPERADOR AP 3	24-06-2021 10:30	Operador producción agua potable O'Higgins.
OPERADOR AS 1	25-06-2021 13:30	Operador tratamiento aguas servidas Biobío.
OPERADOR AS 2	24-05-2021 14:00	Operador tratamiento aguas servidas Biobío.

Tabla 8 - Registro de entrevistas

Las entrevistas fueron grabadas y posteriormente transcritas para su análisis. Se destaca que el tiempo promedio requerido para realizar las entrevistas es de 20 minutos por usuario, aproximadamente. Estas grabaciones fueron de gran utilidad debido a que las entrevistas fueron realizadas a través de llamadas telefónicas, ajustándose al contexto pandemia además de que los operadores trabajan en plantas muy dispersas geográficamente a lo largo de 3 regiones del país. Cabe destacar que por temas de privacidad los nombres reales de los operadores han sido omitidos.

4.1.3 Sintetizar

Sintetizar es la tercera subetapa de descubrir y los objetivos son; concentrarse en los problemas correctos, comprender mejor a los usuarios y crear el artefacto Persona, que es un perfil de personalidad estándar para los usuarios objetivos. Para sintetizar son necesarias cuatro actividades: destilar hallazgos, replantear el enunciado del problema, crear persona y definir un punto de vista.

Destilar hallazgos

Para destilar hallazgos correctamente la metodología propone los siguientes pasos:

1. Procesar datos previos para facilitar su uso y resaltar puntos principales de las entrevistas.
2. Reunir al equipo, repetir la entrevista, escuchar las características, necesidades, tareas, puntos débiles, herramientas y entorno de trabajo de los usuarios. Con adhesivos de color, colocar estas descripciones en una pared.
3. Agrupar los hallazgos buscando puntos de datos comunes para estandarizar al usuario.

El análisis individual de cada usuario y el general agrupado es presentado en los Anexos 3 y 4, respectivamente. La Tabla 9 contiene la agrupación general de características de los usuarios.

Punto	Hallazgos
<p>Características del usuario</p>	<p>Operador de planta de producción de agua potable o tratamiento de aguas servidas.</p> <p>Más de diez años de experiencia en el mismo cargo.</p> <p>Trabaja en más de una planta.</p>
<p>Objetivos y motivaciones</p>	<p>Mantener las plantas 100% operativas.</p> <p>Mantener los datos actualizados en el sistema diariamente.</p> <p>Motivación ante el aporte tecnológico en los procesos de la empresa.</p> <p>Resolver incidentes lo antes posible.</p> <p>Mantener la calidad y continuidad del servicio.</p>
<p>Tareas y actividades</p>	<p>Recorrer las plantas bajo su responsabilidad.</p> <p>Tomar lecturas diarias a parámetros de distinta naturaleza (tomada de un visor, planilla de cálculos, toma de muestras, etc.)</p> <p>Verificación cotidiana de maquinarias.</p> <p>Subir los datos recolectados al sistema.</p> <p>Resolver incidentes para continuar con la rutina normal.</p>

<p>Necesidades</p>	<p>Necesita un sistema que valide los datos más obvios, como por ejemplo: no subir datos los días domingos.</p> <p>Necesita que el sistema muestre un error apenas el dato es ingresado y no al final del periodo.</p> <p>Necesita un sistema acotado, preciso y con una curva de aprendizaje rápida.</p> <p>Necesita que el sistema calcule por si solo los datos derivados.</p>
<p>Perspectivas</p>	<p>Buena disposición al cambio tecnológico.</p> <p>Usar la constancia y rutina como fortaleza principal.</p>
<p>Puntos de dolor y deseos</p>	<p>Las zonas más rurales dificultan tener una buena conexión a internet, por lo que los sistemas web tienden a tener fallos por conectividad.</p> <p>El sistema actual no es lo suficientemente inteligente y falla con algunas validaciones obvias. No aporta a mitigar los errores humanos.</p> <p>Los incidentes y visitas atrasan sus procesos cotidianos.</p> <p>Desea manuales y capacitaciones para comenzar con una rápida adaptación a los sistemas.</p> <p>Desea notificaciones de alerta, considerando datos en tiempo real.</p>

Tabla 9 - Agrupación general de hallazgos

Replantear enunciado del problema

Un enunciado de problema reformulado es una versión mejor definida y más clara de su enunciado inicial. Al discutir el enunciado con el equipo, se llega a un consenso sobre su significado y también se desbloquean algunas oportunidades de innovación al adaptar diferentes perspectivas sobre el problema. Es clave para el proyecto asegurar que se está resolviendo el problema correcto.

Para reformular o replantear el enunciado del problema, la metodología recomienda los siguientes pasos:

1. Escribir en letras grandes el enunciado del problema en una pizarra.
2. Decidir las palabras claves.
3. Tomar las palabras claves y ver los sinónimos disponibles para ver con cual darle una mayor claridad al problema y evitar dobles lecturas.
4. Reflexionar las siguientes preguntas: ¿Cuál es el objetivo del proyecto? ¿Qué problema está tratando de resolver? ¿Qué preguntas se está tratando de responder? ¿Cuáles son sus suposiciones? ¿Cuál es el contexto?
5. Reformular el enunciado teniendo en cuenta lo anterior. Se recomienda usar la técnica HMW (How Might We?, en español “¿Cómo podríamos?).

Para este proyecto se tiene el siguiente enunciado y son destacadas las palabras claves:

*“Se busca implementar una aplicación que permita realizar una **lectura en piso de planta** de parámetros asociados a la **producción de agua potable y tratamiento de aguas servidas** en los equipos de cada planta, según corresponda. Actualmente, el 25% de los parámetros son leídos con dispositivos SCADA, mientras que el resto son **leídos manualmente** por los operadores y subidos en las oficinas. Por lo que se busca optimizar los tiempos de los operadores, usando una **aplicación móvil** para tomar la lectura de los datos mientras se recorre la planta”.*

Al reformular este enunciado utilizando la técnica HMW recomendada por la metodología, se obtuvo lo siguiente:

“¿Cómo podemos optimizar el tiempo que les requiere a los operadores registrar los datos de los parámetros leídos al recorrer el piso de planta?”.

Crear Persona

El artefacto Persona se crea para representar tipos específicos de usuarios que surgen durante la investigación. Persona es una persona imaginaria que ayuda a desarrollar empatía mientras se responden preguntas como: ¿Esta función resuelve el problema de nuestra persona? Para crear Persona, la metodología recomienda revisar las características comunes de los usuarios identificados y crear áreas de categoría de información básica en una pared, y con notas adhesivas, agrupar las características. El artefacto persona se puede encontrar en el Anexo 5. A continuación, se presenta el contenido del artefacto persona por partes en la Tabla 10.

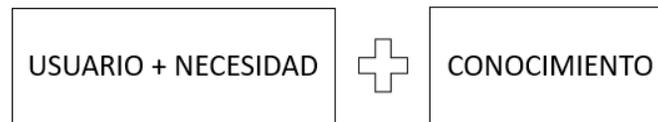
Ítem	Contenido
Nombre	Claudio , Operador Producción y Tratamiento
Quote	“No pretendo quedarme con un Sistema arcaico que va a ir pasando el tiempo y vamos a ir quedando obsoletos, tenemos que irnos movilizandoo, de eso se trata la mejora continua ”
Resumen	Operador de planta con más de 10 años de experiencia en el mismo cargo. Trabaja en más de una planta, las que normalmente están separadas por varios kilómetros de distancia.
Trabaja con	Desarrolla sus laborales normales en solitario. Un supervisor zonal coordina y le habilita para realizar sus labores. El soporte de TI le apoya con los problemas relacionados al sistema actual.

<p>Responsabilidades en el trabajo</p>	<p>Recorrer diariamente sus plantas y verificar el correcto funcionamiento de las maquinarias.</p> <p>Acudir oportunamente a resolver los incidentes ocurridos en sus plantas y apoyar a otros operadores en caso de ser necesario.</p> <p>Tomar lecturas de datos diarias a los distintos parámetros asociados a sus plantas, para posteriormente subirlos al sistema actual.</p>
<p>Objetivos principales</p>	<p>Mantener sus plantas operativas y funcionando al 100% para conseguir una calidad y continuidad del servicio óptima.</p> <p>Mantener una actualización diaria de los datos recolectados en cada planta.</p>
<p>Necesidades</p>	<p>Necesita un sistema acotado, preciso y con una rápida curva de aprendizaje, que no solicite datos redundantes o innecesarios.</p> <p>Necesita un sistema inteligente que calcule por sí solo los datos derivados, que valide eficientemente el ingreso de datos y que pida inmediatamente las justificaciones de los datos que no están dentro del rango.</p>
<p>Puntos de dolor</p>	<p>Los incidentes y visitas largas atrasan sus procesos cotidianos.</p> <p>El sistema actual falla en validaciones obvias, no aportando a amortiguar los errores humanos.</p> <p>No tener una buena conexión a internet entorpece el proceso de subida de datos al sistema actual.</p>

Tabla 10 - Partes del artefacto Persona

Definir un Punto de Vista

Se define un punto de vista para concentrar los esfuerzos en cada necesidad importante de Persona por separado. La metodología recomienda anotar al usuario y una necesidad y anotar el conocimiento asociado a esta relación.



Para el caso de este proyecto se puede definir el siguiente punto de vista:

Claudio, operador de producción y tratamiento, necesita una forma rápida y confiable de ingresar los datos de los parámetros tomados en el piso de planta, de modo que pueda usar su tiempo para enfocarse en las demás labores de importancia que realiza cotidianamente.

Después de generar todo este conocimiento, se tiene un amplio entendimiento del problema, los usuarios y el proyecto.

4.2 Diseñar

Diseñar es la segunda etapa del procedimiento DLD. Aquí se comprende el problema y las necesidades, para comenzar a crear la mejor solución. Son realizados los prototipos y las pruebas, para iterar el diseño con los usuarios.

4.2.1 Idear

Idear es la primera sub-etapa asociada a diseñar. Los objetivos son pasar del problema a la solución, converger ideas deseables para seleccionar las mejores, evaluar la fiabilidad y factibilidad de las ideas, y finalmente visualizar la experiencia de usuario a través de un Storyboard. A continuación, se presentan las actividades y sus artefactos asociados a esta sub-etapa.

Preparar un espacio de trabajo

La metodología da una serie de recomendaciones para adaptar un entorno de trabajo, sin embargo, y teniendo en cuenta el contexto de pandemia, las sesiones y conversaciones realizadas con el personal de la empresa fue, o bien por llamada telefónica, correo electrónico o sesiones de video llamadas.

Lluvia de ideas

Teniendo en cuenta toda la información recopilada en la investigación, se utiliza el Punto de Vista y la técnica HMW para convertirlo en una serie de preguntas que resolver, para abordar en subtemas. Las preguntas y respuestas generadas son:

- ¿Cómo podríamos minimizar el tiempo que requiere la subida de datos al operador?
 - Subir los datos a través de una aplicación móvil.
 - Personalizar el orden de la subida de datos para ajustarlo al recorrido que más le acomoda al operador.
- ¿Cómo podríamos asegurar la confiabilidad del proceso de subida de datos?
 - El sistema podría evaluar el nivel de conectividad antes de enviar los datos.
 - El sistema podría contar con una base de datos embebida.

- ¿Cómo podríamos entregar una aplicación segura?
 - El sistema podría contar con cuentas de usuario antes de acceder a sus funcionalidades.
 - Cuando el usuario deje de usar el sistema, debería cerrar su sesión para evitar que otra persona manipule o mal use la plataforma.

Storyboard

El Storyboard ilustra el contexto físico en el que se encuentra el usuario, las herramientas que utilizará y las personas con las que interactúa. Todavía no detalla las soluciones con pantallas reales, pero es una forma eficiente de comunicar una experiencia de usuario propuesta,

Los dos Storyboard generados en esta actividad están en el Anexo 6. El primer Storyboard describe una situación común de un operador asistiendo a una planta, anotando los datos en una libreta para luego ir a la oficina a subirlos al sistema actual. El segundo Storyboard describe la nueva situación que generaría utilizar una aplicación móvil para registrar los datos.

Casos de Uso

Los casos de uso describen secuencias de interacciones entre los usuarios y el sistema, definen el producto desde la perspectiva del usuario y se especifica que objetivos o tareas debe admitir y habilitar la interfaz de usuario. La Tabla 11 contiene una lista de casos de uso desarrollados en el proyecto, asociadas a una serie de necesidades del usuario. La especificación de casos de uso está en el Anexo 7 utilizando la planilla propuesta por la metodología.

Código	Nombre	Necesidad Asociada
CU 01	Acceso al Sistema	El operador necesita acceder al sistema para subir los datos correspondientes a los parámetros.

CU 02	Configurar Ruta	El operador necesita Configurar la ruta con la que recorrerá una planta, para ir subiendo los datos a medida que los va tomando.
CU 03	Modificar Datos	El operador necesita modificar los registros de los parámetros de las plantas en caso de detectar algún error o confusión.
CU 04	Generar Reportes	El operador necesita unos reportes del detalle de los parámetros asociado por planta.
CU 05	Cerrar Sesión	El operador necesita cerrar la sesión de la cuenta.
CU 06	Comenzar Recorrido	El operador necesita registrar los datos de los parámetros del día.

Tabla 11 - Lista de Casos de Uso

4.2.2 Prototipar

Prototipar es la segunda sub-etapa asociada a diseñar y sus objetivos son proporcionar una versión de prototipo de baja fidelidad, mostrar el diseño, interfaz y contenido, demostrar acciones y comportamientos.

Artefactos Design Thinking

A estas alturas del proyecto se ha generado mucho conocimiento traducido en los artefactos. La metodología propone subir todos los artefactos al espacio de trabajo otorgado por la herramienta Build. Para acceder primero se necesita crear una cuenta en el portal principal de SAP Build, que se pudo ver en la Figura 19. La Figura 22 contiene los datos que son solicitados para la creación de la cuenta. Luego de ingresar los datos, antes de finalizar el proceso de creación de cuenta, es importante verificar la dirección de correo electrónico ingresada, como se muestra en la Figura 23.

The image shows two parts of the SAP Build registration process. On the left is the 'Create a BUILD account' page, which features the 'build' logo and a form with an 'EMAIL' field containing the text 'Enter your Email to Proceed' and a 'PROCEED' button. Below the button is a link: 'Already have an account? [Log on here!](#)'. On the right is the 'Registro' (Registration) form, which includes the following sections:

- Datos personales**: Fields for 'Nombre *', 'Apellidos *', and 'Correo electrónico *' (containing 'patpardo@alumnos.ubiobio.cl').
- Establecer contraseña**: Fields for 'Contraseña *' and 'Vuelva a introducir la contraseña *'.
- Términos y condiciones**: Two checkboxes: 'He leído la [Declaración de protección de datos](#) y estoy de acuerdo con ella. *' and 'He leído y entendido los Términos y Condiciones de SAP BUILD. *'.
- A note: '*Obligatorio'.
- A blue 'Registrarse' button.
- The SAP logo at the bottom.

Figura 22 - Creación de cuenta en SAP Build

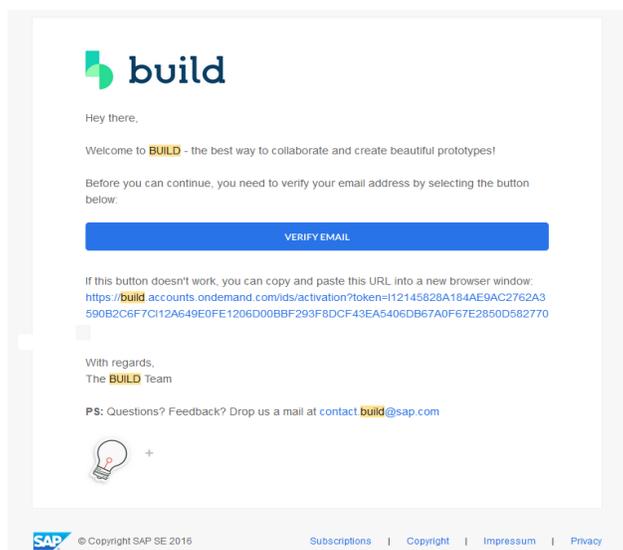


Figura 23 - Verificación de correo SAP Build

Una vez verificada la cuenta, ya se puede iniciar sesión desde el portal principal de SAP Build. La Figura 20 muestra el espacio de trabajo en donde se almacenan los proyectos en SAP Build. La cuenta gratuita permite participar en 3 proyectos en paralelo como máximo, en los cuales se puede ser el Owner (dueño) o participante. En caso de requerir participar en un cuarto proyecto, el sistema nos permite archivar los proyectos de los cuales se es el dueño, para crear un nuevo o participar en uno extra.

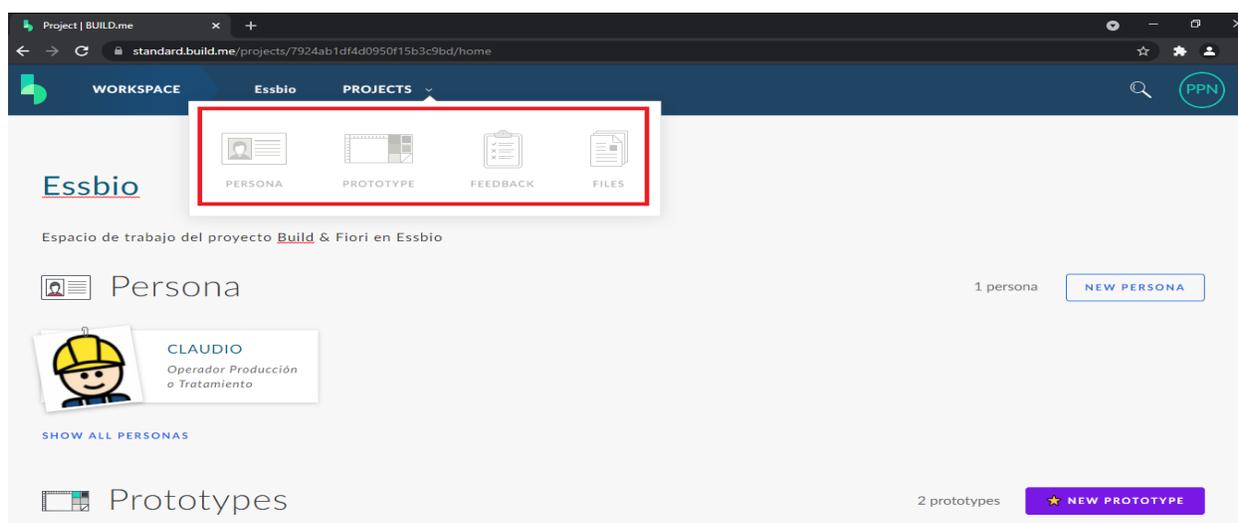


Figura 24 - Espacio de Trabajo Proyecto Essbio es Build

La Figura 24 es una captura de pantalla del espacio de trabajo del proyecto en Essbio. En el recuadro rojo se muestran las principales características de SAP Build.

- **Persona:** Ya que DLD es una metodología centrada en el humano, Persona es una de los principales artefactos. La herramienta permite subir la información sintetizada en el artefacto Persona para que los hallazgos sean compartidos por el equipo. El artefacto Persona de este proyecto está en el Anexo 5.
- **Prototype:** Esta sección es donde se desarrollan los prototipos de baja fidelidad y de alta fidelidad. Para los prototipos de baja fidelidad, el sistema permite importar capturas de los bocetos desarrollados por el equipo y enlazar estas capturas o vistas, para recrear la interactividad de lo propuesto.
- **Feedback:** Esta sección permite generar preguntas en base a los prototipos, las que son respondidas por los participantes del proyecto en Build. Esto provee la capacidad de recopilar información importante para mejorar los prototipos, la experiencia de usuario y refrescar los conocimientos actuales.
- **Files:** Esta sección permite importar todos los entregables para que esta información sea transversal en el equipo. Los formatos png, jpge y pdf pueden ser abiertos en el mismo SAP Build para ser revisados, a diferencia de otros formatos más específicos como los asociados a la SUITE de Microsoft (No se recomienda subir los artefactos en estos formatos).

La Figura 25 es una captura de pantalla tomada desde la sección de archivos de SAP Build, en el proyecto colaborativo con Essbio.

The screenshot shows the 'Files' section of a workspace. At the top, there is a navigation bar with 'WORKSPACE', 'Essbio', and 'FILES' tabs. A search icon and a 'PPN' logo are also visible. Below the navigation bar, the 'Files' section is titled with a folder icon and the word 'Files'. To the right of the title, there is an upload icon and a 'Sort By' dropdown menu set to 'Name A→Z'. The main area contains a table with the following columns: Name, Uploaded By, Size, and Modified. The table lists 28 files, including PNG images, PDF documents, and a script. All files were uploaded by 'Patricio Pardo Novoa'.

Name	Uploaded By	Size	Modified
1 Portal.png	Patricio Pardo Novoa	422 kB	24 Aug 2021 02:11 AM
10 Recorrido 1.png	Patricio Pardo Novoa	393 kB	24 Aug 2021 02:11 AM
11 Recorrido 2.png	Patricio Pardo Novoa	359 kB	24 Aug 2021 02:11 AM
12 Recorrido 3.png	Patricio Pardo Novoa	385 kB	24 Aug 2021 02:11 AM
2 Menu Principal.png	Patricio Pardo Novoa	426 kB	24 Aug 2021 02:11 AM
3 Rutas 1.png	Patricio Pardo Novoa	383 kB	24 Aug 2021 02:11 AM
4 Rutas 2.png	Patricio Pardo Novoa	394 kB	24 Aug 2021 02:11 AM
5 Datos 1.png	Patricio Pardo Novoa	394 kB	24 Aug 2021 02:11 AM
6 Datos 2.png	Patricio Pardo Novoa	459 kB	24 Aug 2021 02:11 AM
7 Datos 3.png	Patricio Pardo Novoa	411 kB	24 Aug 2021 02:11 AM
8 Reportes.png	Patricio Pardo Novoa	410 kB	24 Aug 2021 02:11 AM
9 Cerrar sesion.png	Patricio Pardo Novoa	386 kB	24 Aug 2021 02:11 AM
Alcance de Proyecto - Proje Scope.pdf	Patricio Pardo Novoa	663 kB	16 Jul 2021 12:00 PM
Brainstorming.pdf	Patricio Pardo Novoa	609 kB	24 Aug 2021 01:35 AM
caricatura.png	Patricio Pardo Novoa	14 kB	16 Jul 2021 11:20 AM
caricatura_cropped.png	Patricio Pardo Novoa	35 kB	16 Jul 2021 11:20 AM
CU 01 - Acceso al sistema.pdf	Patricio Pardo Novoa	255 kB	24 Aug 2021 01:32 AM
CU 02 - Configurar Ruta.pdf	Patricio Pardo Novoa	217 kB	24 Aug 2021 01:32 AM
CU 03 - Modificar Datos.pdf	Patricio Pardo Novoa	216 kB	24 Aug 2021 01:32 AM
CU 04 - Generar Reportes.pdf	Patricio Pardo Novoa	215 kB	24 Aug 2021 01:32 AM
CU 05 - Cerrar Sesión.pdf	Patricio Pardo Novoa	214 kB	24 Aug 2021 01:32 AM
CU 06 - Comenzar Recorrido.pdf	Patricio Pardo Novoa	215 kB	24 Aug 2021 01:32 AM
Destilar Hallazgos - Jamboard General.pdf	Patricio Pardo Novoa	183 kB	16 Jul 2021 12:00 PM
gQRsEgnY_400x400-removebg-preview.png	Patricio Pardo Novoa	11 kB	15 Jul 2021 11:12 PM
PERSONA.pdf	Patricio Pardo Novoa	818 kB	16 Jul 2021 12:01 PM
POV Artefacto.pdf	Patricio Pardo Novoa	597 kB	16 Jul 2021 12:05 PM
Replanteamiento de la declaración del Pr....pdf	Patricio Pardo Novoa	319 kB	16 Jul 2021 12:01 PM
Script Entrevista.pdf	Patricio Pardo Novoa	424 kB	16 Jul 2021 12:00 PM
Storyboard.pdf	Patricio Pardo Novoa	2 MB	24 Aug 2021 01:34 AM

Figura 25 - Artefactos en el espacio de trabajo SAP Build

Dibujar Prototipos

Se crean prototipos dibujados a mano que muestren los elementos principales de la interfaz de usuario y el flujo general de su solución. Hacer este prototipo de bajo costo desde el principio permite tener una idea sobre qué conceptos funcionan mejor para el usuario.

La Figura 26 es una captura de los dos primeros bocetos, correspondiente al portal de acceso y menú principal. Las demás capturas están en el Anexo 8.

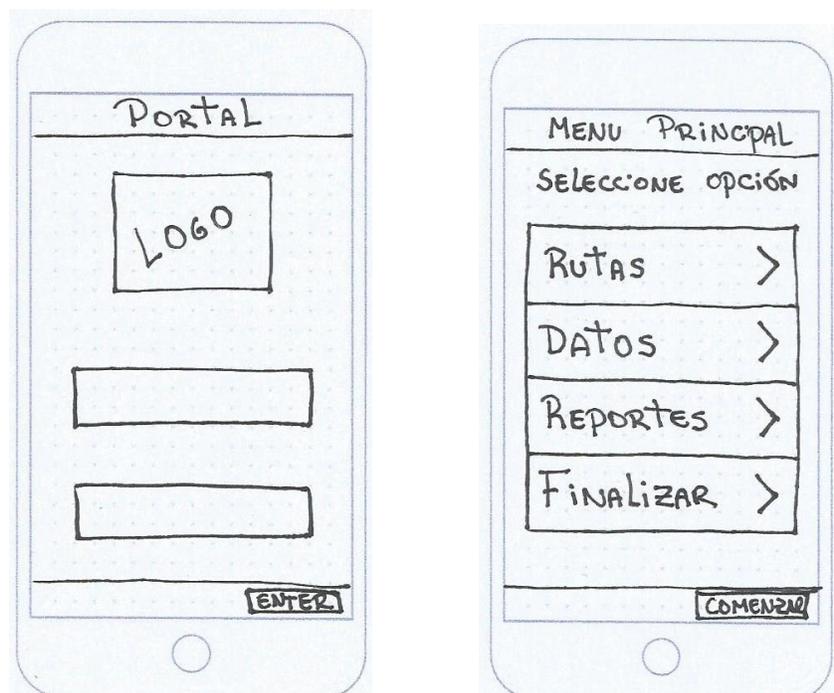


Figura 26 - Bocetos Portal de Acceso y Menú principal

Crear Wireframes

Los Wireframes son representaciones visuales de una interfaz. Son fácilmente modificables, de bajo o alta fidelidad, y son una herramienta importante utilizada para el diseño y desarrollo de productos. Utilizando Build, fueron integrados los bocetos, se enlazaron y se hizo un Mapa de navegación, presentado en la Figura 27 y el Anexo 9. El uso de estos bocetos enlazados como se presenta en el mapa de navegación compone el prototipo de baja fidelidad. Las flechas indican el flujo de navegación y el sistema permite generar zonas clickeables para activar esta navegación.

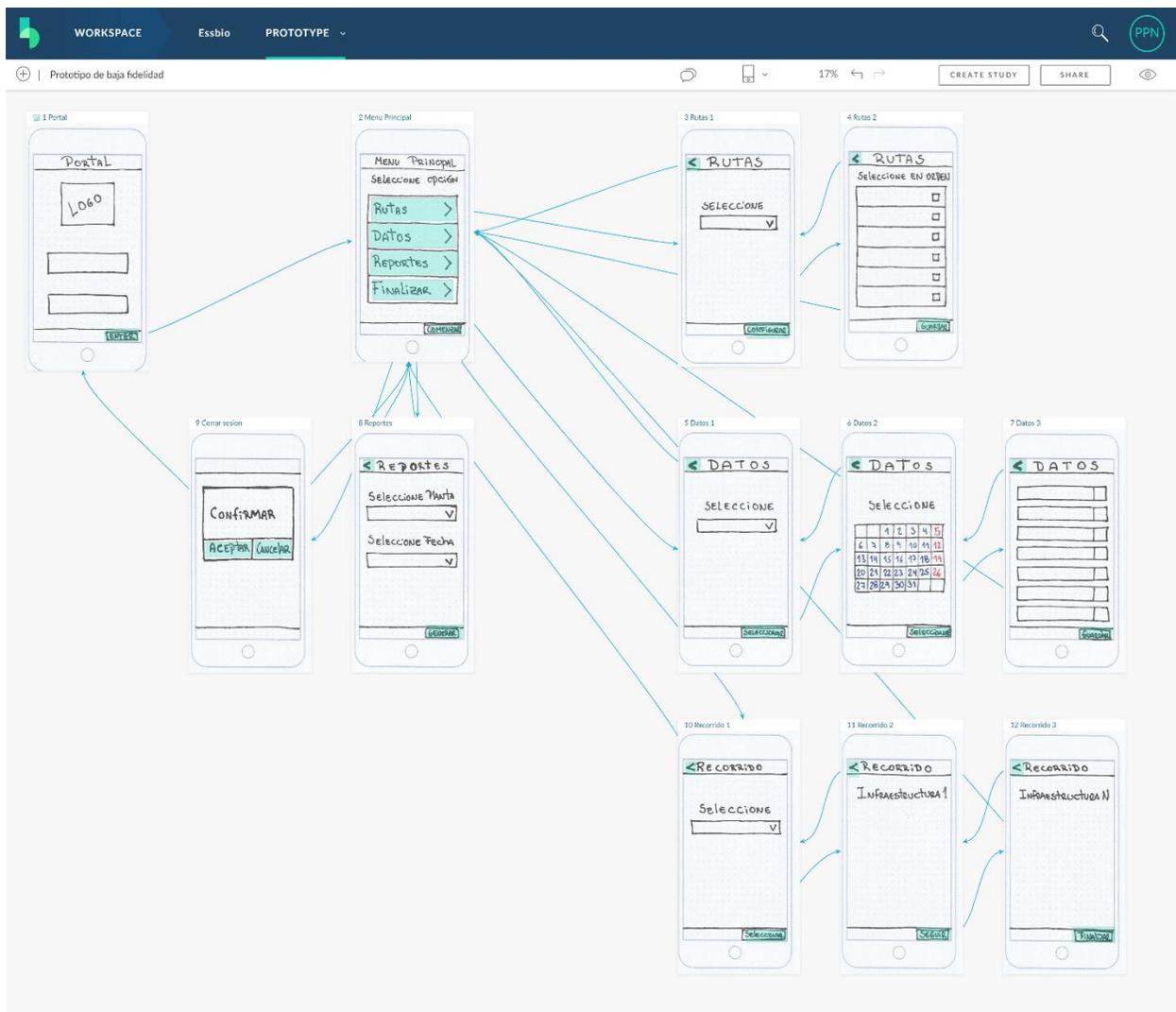


Figura 27 - Mapa de navegación del prototipo de baja fidelidad en Build

Para crear el prototipo de alta fidelidad se utilizó un marco configurado responsivo como lo muestra la Figura 28. También existe la posibilidad de utilizar prototipos hechos desde una galería, pero esta opción no es tomada en cuenta para este proyecto.

Prototipo alta fidelidad

Prototipo responsivo pensado para celular

IMAGES **TEMPLATES**

Customize Fiori/UI5 templates to build out your prototypes

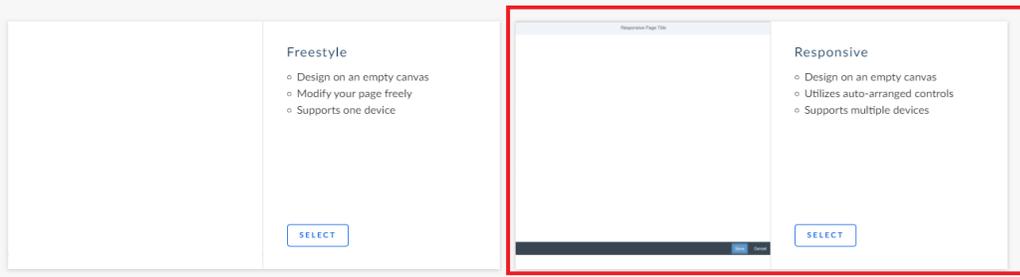


Figura 28 – Selección de marco responsivo

Luego de crear la primera página se accede a la edición. La Figura 29 es una captura del entorno ofrecido por SAP Build para realizar los prototipos de alta fidelidad.

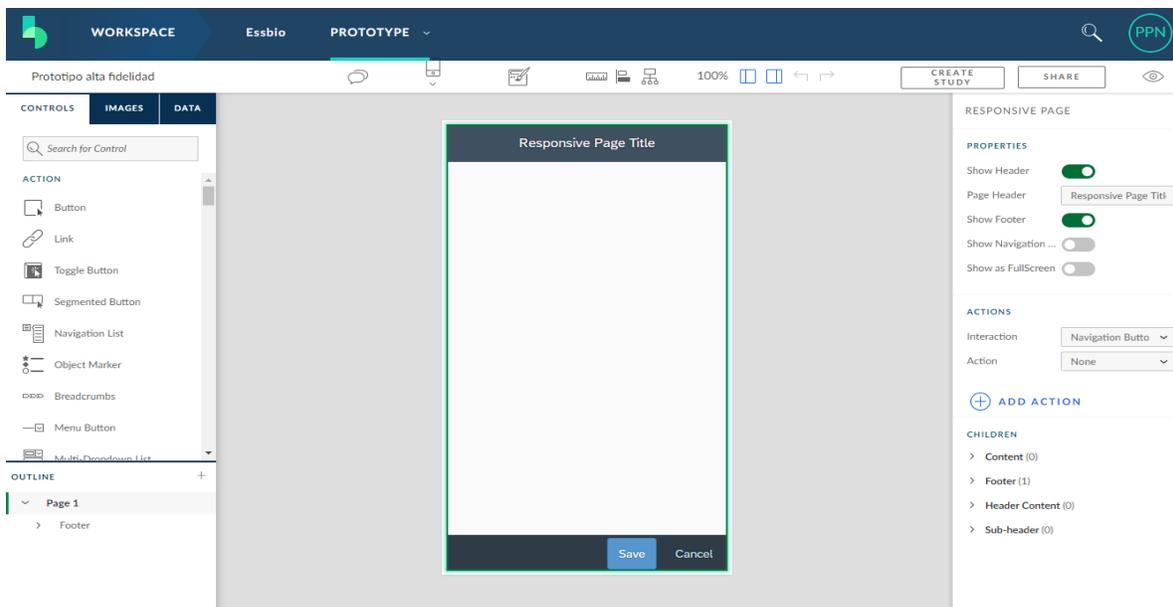


Figura 29 - Creación Prototipo de Alta Fidelidad en Build

En la parte izquierda se encuentran los controles y el Outline. Los controles tienen los componentes que predispone Build para usarse en la creación de prototipos de alta fidelidad y están agrupados por: acción, títulos, gráficos, containers, presentación, marcos, listas, entradas de usuario y mapas. El Outline es un árbol en donde las primeras hojas son las vistas y de las segundas en adelante son los componentes que forman una página.

En la parte derecha se encuentran las propiedades y las acciones. Las propiedades se relacionan al aspecto de las vistas y las acciones a los comportamientos de los componentes de la página. Esto se puede entender como el CSS y el JavaScript de cada página, respectivamente.

En la parte superior derecha hay 3 botones importantes:

- Crear un estudio: Permite generar una pregunta entorno a la vista para recopilar información de los participantes del proyecto en el entorno Build.
- Compartir: Permite obtener un enlace con el que personas ajenas al proyecto pero con cuenta de Build pueden interactuar con el prototipo e incluso comentar.
- View: Permite visualizar el prototipo como si fuera una aplicación real.

Al replicar el prototipo de baja fidelidad en alta fidelidad se obtuvieron un total de 12 vistas, entre las que se destacan:

- 1 vista de portal.
- 1 vista de menú principal.
- 2 vista para Configurar ruta.
- 4 vistas para editar datos.
- 1 vista para generar reportes.
- 3 vistas para recrear un recorrido.

La Figura 30 contiene el mapa de navegación del prototipo de alta fidelidad. Nuevamente las flechas azules recrean el flujo de navegación entre la vistas. El Anexo 10 contiene las capturas de todas las vistas generadas para el prototipo de alta fidelidad y el Anexo 11 contiene el mapa de navegación de este prototipo.

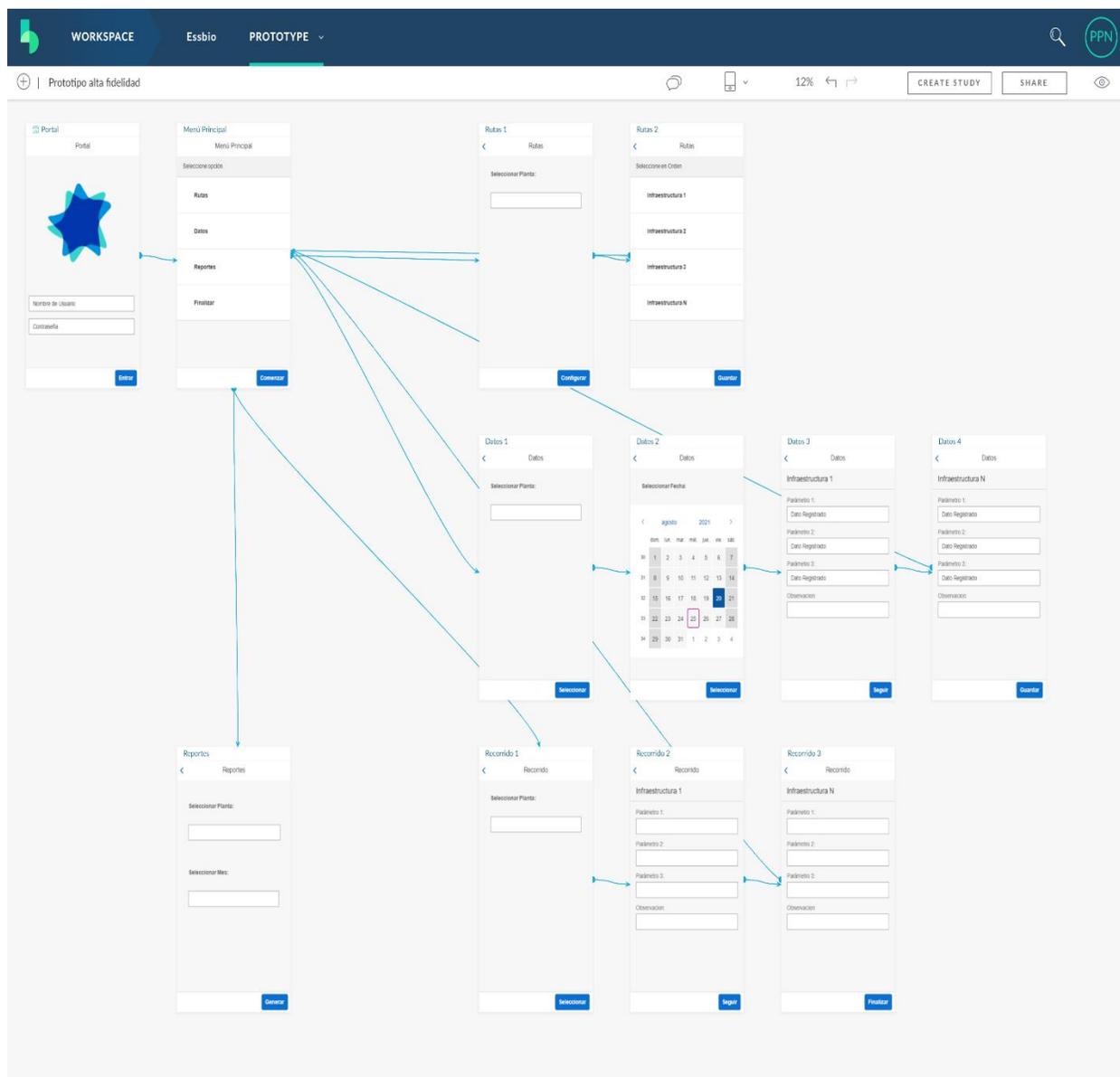


Figura 30 - Mapa de navegación del prototipo de alta fidelidad en Build

El flujo presentado en la Figura 30 es simple, existe un portal de acceso en el que se ingresan los datos de la cuenta. Una vez se validan los datos el usuario puede acceder al menú principal el cual se enlaza con las funcionalidades: configuración de la ruta, edición de datos, generación de reportes, cerrar sesión de usuario y comenzar el recorrido por el piso de planta.

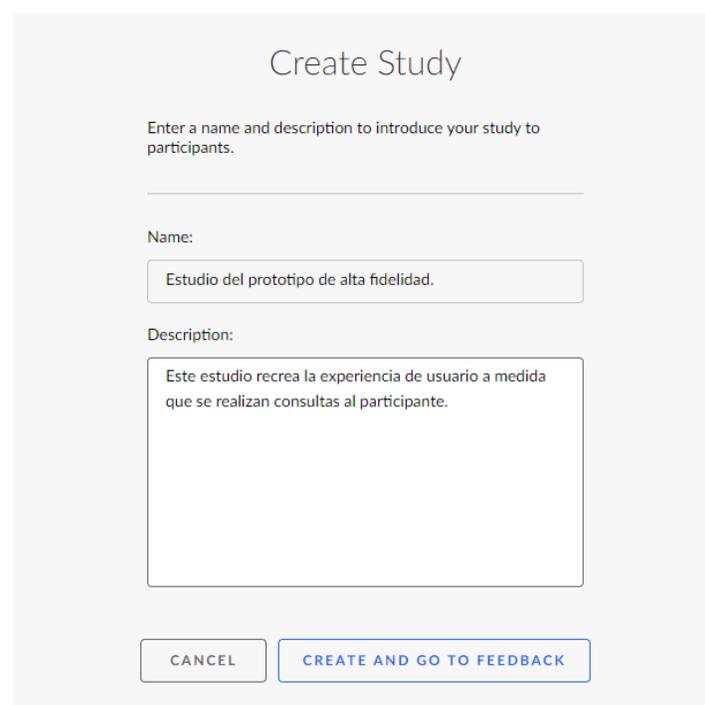
4.2.3 Validar

Validar es la tercera subetapa de Diseñar y sus objetivos son verificar el enfoque de la investigación, evaluar la efectividad del diseño y observar y escuchar a los usuarios. Para esto se pondrán a prueba los prototipos y se recibirán comentarios de parte de algunas personas involucradas en el proyecto, principalmente de TI.

Probar Bocetos

Antes de llevar el diseño al desarrollo es importante asegurarse de que realmente funcione para los usuarios. Para esto, la herramienta Build permite recopilar comentarios. Para facilitar e incentivar la participación, se creó una cuenta genérica con la cual serán agregados los comentarios.

Es importante ingresar a uno de los prototipos, en este caso el de alta fidelidad, para generar un estudio utilizando la cuenta Owner del proyecto al dar clic en el botón “Create Study”. La Figura 31 es una captura de la creación del estudio, en donde Build requiere de un nombre y una descripción.



Create Study

Enter a name and description to introduce your study to participants.

Name:

Estudio del prototipo de alta fidelidad.

Description:

Este estudio recrea la experiencia de usuario a medida que se realizan consultas al participante.

CANCEL CREATE AND GO TO FEEDBACK

Figura 31 - Inicio de la creación de un estudio en Build

La Figura 32 es una captura de cómo se configura el estudio en la herramienta Build. En la parte superior se encuentra el título y la descripción dada, y también un botón para realizar una visualización del estudio y otro para hacerlo público. En medio está la sección de preguntas, para crear una hay que dar clic al botón nueva pregunta.

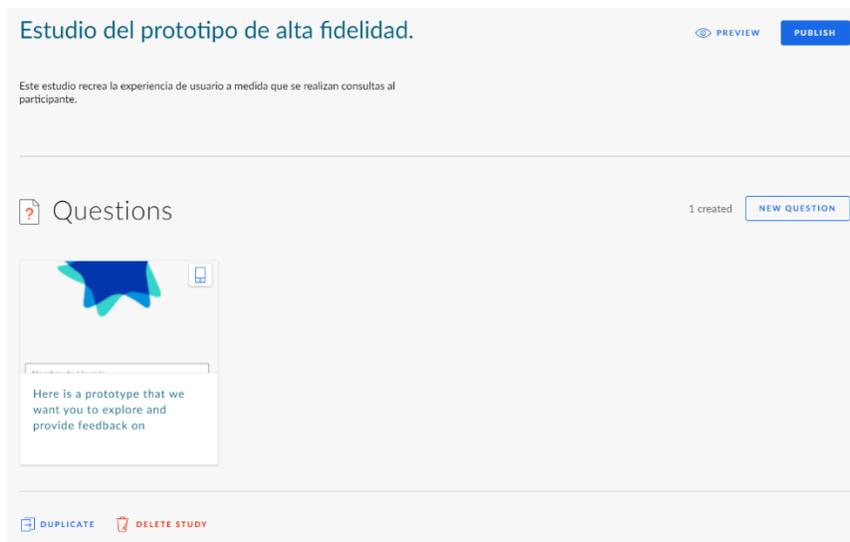


Figura 32- Generación de estudio en Build.

Al hacer una pregunta, la herramienta pide seleccionar sobre qué imagen o prototipo, para este caso se seleccionará el prototipo de alta fidelidad. La Figura 34 es una captura de la sección para Configurar las preguntas. La lógica es seleccionar cada vista y agregar una o más preguntas que serán respondidas posteriormente por los participantes del estudio. Las preguntas pueden tener un límite de anotaciones y respuestas, o bien ser respondidas mediante una acción. Lo último que se puede destacar, es que permite seleccionar el dispositivo y orientación usada, tal como lo muestra la captura en la Figura 33.



Figura 33 - Selección de formato para el estudio.

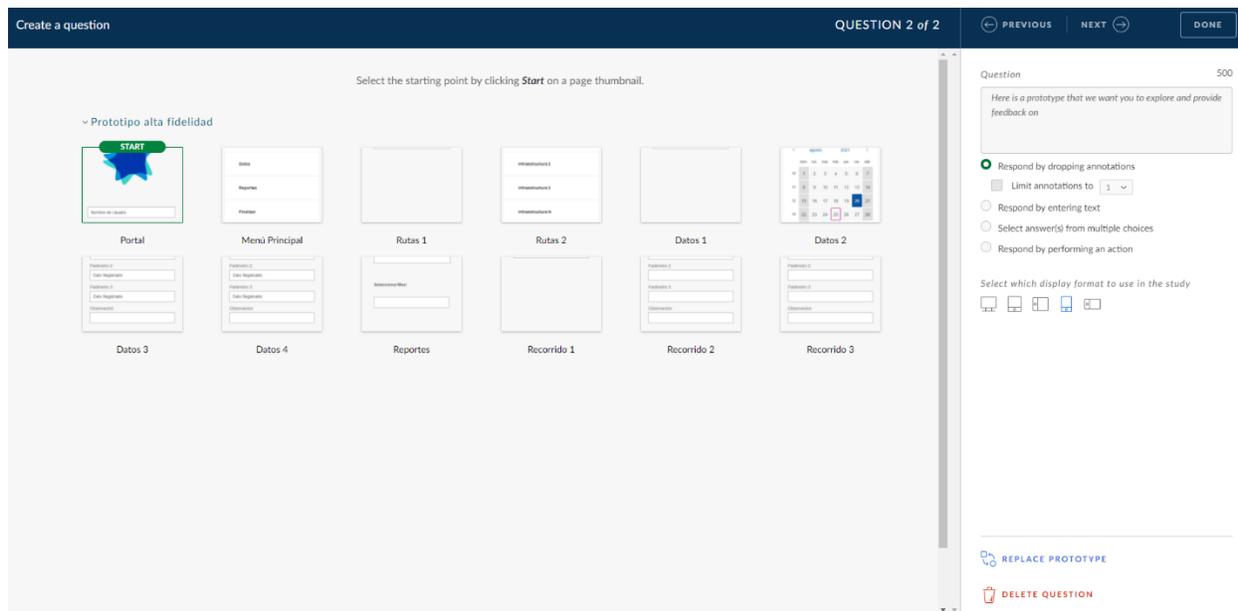


Figura 34 - Configuración de pregunta para estudio en Build

Una vez publicado el estudio, personas ajenas al proyecto pueden realizar el recorrido usando una cuenta de Build. Se recrea esta acción utilizando la cuenta de la profesora guía. La Figura 35 muestra el acceso al estudio en donde se puede identificar el título, el nombre de la cuenta que creó el estudio, la descripción y la cantidad de preguntas que contiene el estudio. El estudio inicia al darle clic en el botón “Start Study”. Cabe destacar que se puede optar por comentar anónimamente.

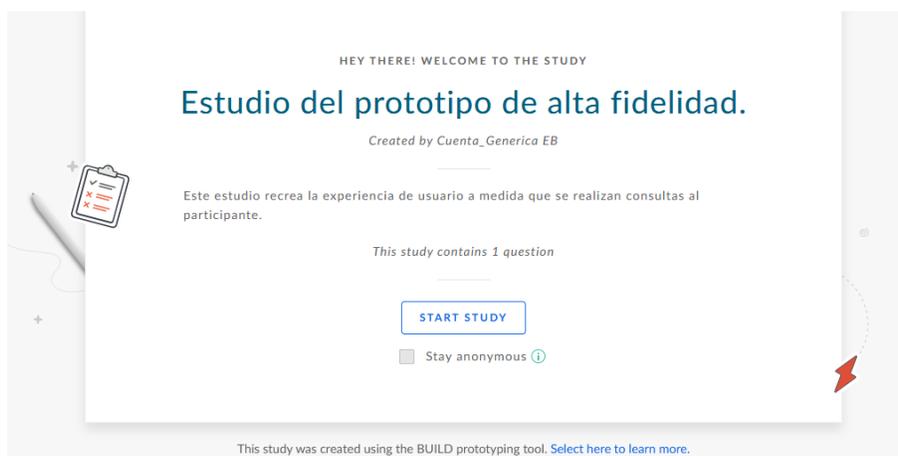


Figura 35 - Acceso al estudio de prueba

La Figura 36 es una captura de la pregunta realizada por el estudio. Aquí el participante puede recrear la navegabilidad del prototipo e ir comentando. Para comentar debe dar clic en el botón “Drop Comments”. La Figura 37 es una captura de cómo se registra un comentario, está compuesto por un texto y la selección de una emoción.

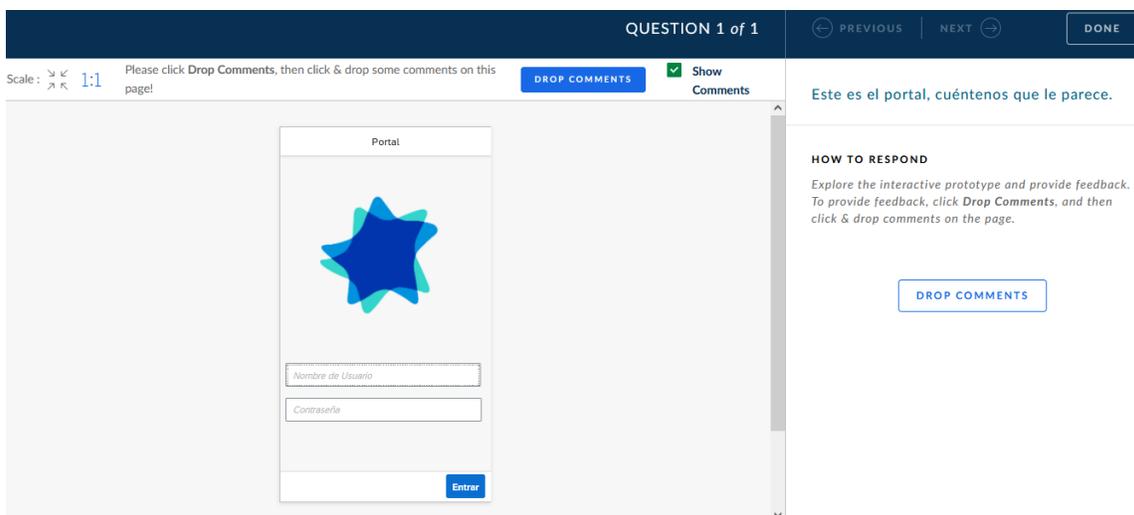


Figura 36 - Participación en el estudio configurado en Build

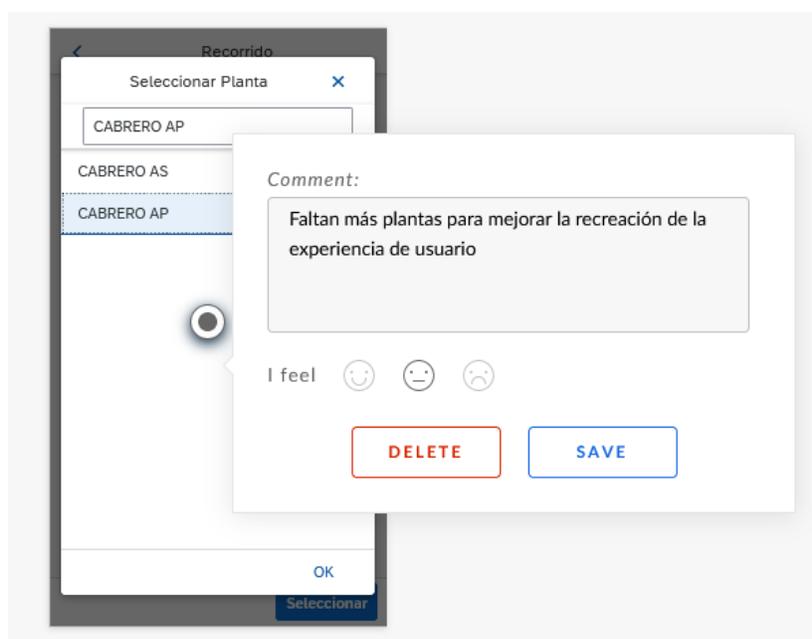


Figura 37 - Ejemplo de un comentario en un estudio

Una vez recorrido el prototipo y comentado todo lo necesario, el participante puede finalizar la sesión. La Figura 38 es una captura del mensaje que entrega Build al finalizar la participación en un estudio.

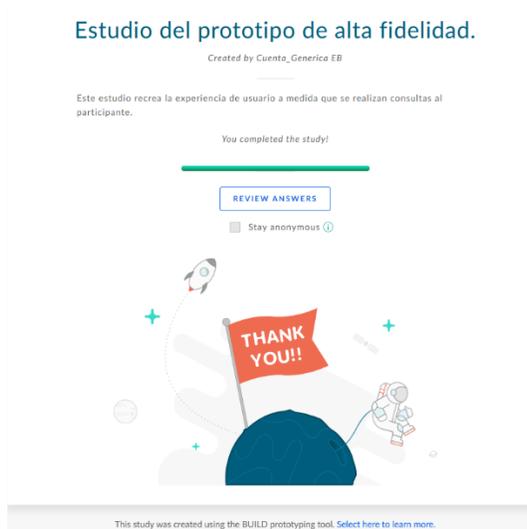


Figura 38 - Finalización del estudio en Build

Luego de recibir a algunos cuantos participantes el sistema genera información gráfica acerca de las interacciones como se muestra en la Figura 39. Se resume la cantidad de participantes, preguntas contestadas, tiempo promedio por pregunta, cantidad de comentarios, reacciones, etc. Esta sección también permite pre visualizar el estudio y cuenta con el enlace para seguir su difusión. Otro aspecto clave a destacar, es que se puede obtener información por preguntas y por participantes. También se puede exportar una planilla Excel con los datos para su explotación.

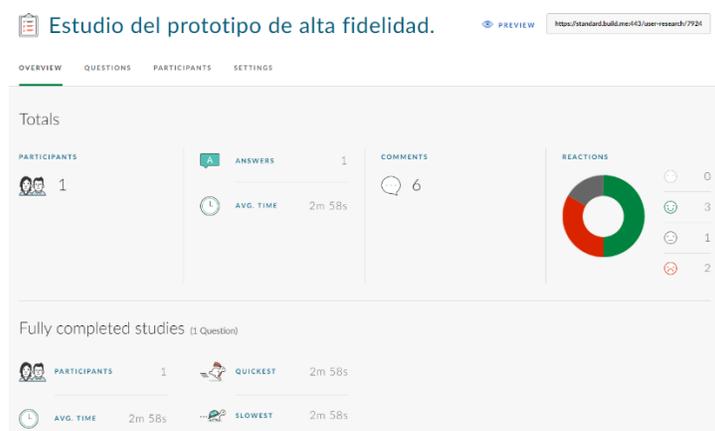


Figura 39 - Resultados del estudio en Build

Además, tanto en los estudios, como en el mapa de navegación y vistas individuales, Build permite a los participantes del proyecto dejar notas para aclarar funcionalidades o bien sugerir modificaciones. La Figura 40 muestra un ejemplo de un comentario realizado con la cuenta genérica.

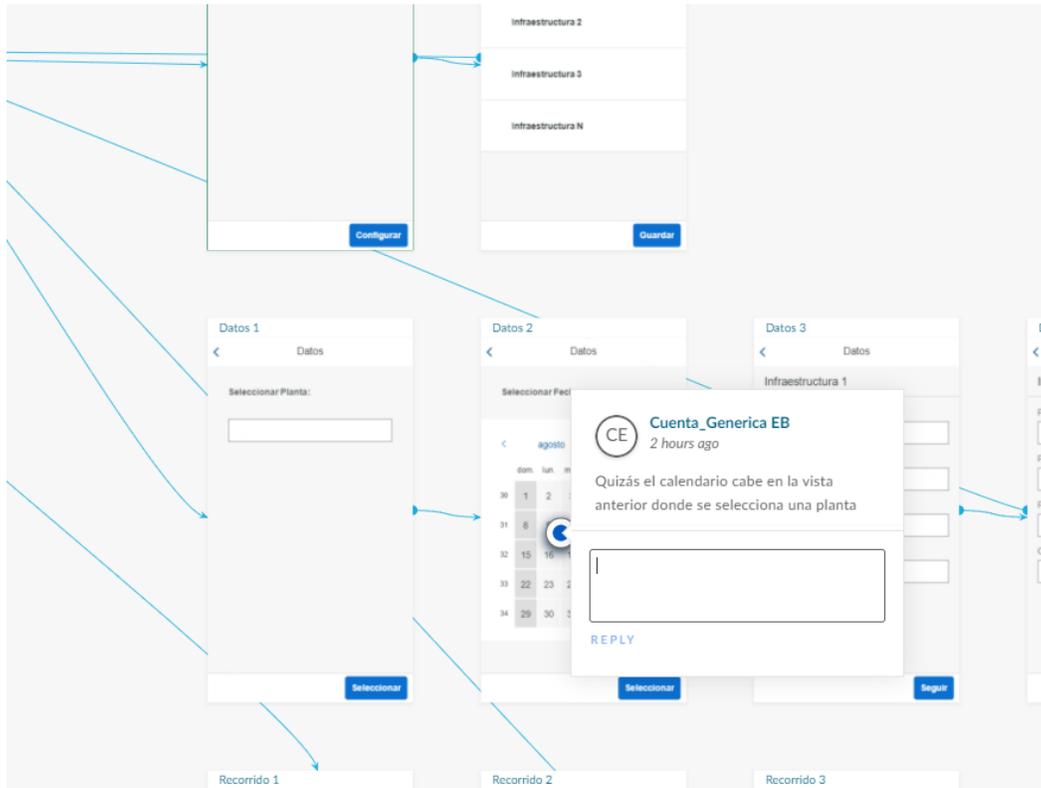


Figura 40 - Comentario realizado en el mapa de navegación del prototipo de alta fidelidad en Build

La recopilación de sugerencias y retroalimentación de otros usuarios de Build permite generar información importante acerca de la aceptación del producto que se está prototipando. Esta es una de las principales fortalezas de esta herramienta, robustecer el proceso de generación de experiencia de usuario.

4.3 Desarrollar

Desarrollar es la última etapa del DLD. Una vez probado el diseño con los usuarios, se debe convertir en una solución útil. Para esto se valida, ajusta y publica el producto.

4.3.1 Implementar

Implementar es la primera subetapa de Desarrollar y su principal objetivo es comenzar a dar vida a la solución. La metodología propone realizar un mapeo de historias de usuario con el cual respaldar un plan de desarrollo.

Mapeo de historias de usuario

El mapeo de historias de usuario es un método ágil que ayuda a comprender mejor el panorama general del producto y a generar una comprensión común de las prioridades del usuario. La estructura del mapeo permite identificar el mínimo producto viable o MVP, por si siglas en inglés. Generalmente este artefacto es más grande pero en este proyecto, debido a la definición de alcances, solo se trabajó la solución para los operadores, los cuales representan a la Persona 1. La Tabla 12 contiene artefacto mapeo de historias de usuario, generado en este proyecto. Cada actividad está asociada a los casos de uso definidos con anterioridad.

Objetivo del proyecto	Paso 1	Paso 2	Paso 3
PERSONA 1	Ingresar	Configurar Ruta	Realizar Recorrido
	Cerrar Sesión	Generar Reportes	Modificar Datos
PRIORIDADES	MVP: Producto Mínimo Viable		
	El usuario quiere ingresar con los datos de su cuenta	El usuario quiere personalizar la ruta con la que recorrerá la planta	El usuario quiere recorrer la planta y tomar las lecturas en el mismo orden que configuró la ruta
	El usuario quiere cerrar su sesión después de utilizarla	El usuario quiere generar reportes de parametrización	El usuario quiere modificar los datos que ya se han subido

Tabla 12 - Mapeo de Historias de Usuario

Se puede resumir que las actividades importantes, que permiten generar el producto mínimo viable son 3: Ingresar al sistema, Configurar la ruta y realizar el recorrido. Las demás funcionalidades declaradas en los casos de uso pasan a segundo plano, por lo que tienen una prioridad más baja a la hora del desarrollo.

4.3.2 Probar

Probar es la segunda subetapa de Desarrollar y sus objetivos son asegurar que el diseño funcione según lo planeado y cumpla con los objetivos de rendimientos. Además se valida la conveniencia y viabilidad del producto. La metodología recomienda realizar una prueba de usabilidad.

Prueba de usabilidad

La usabilidad es un término que describe cómo los usuarios perciben el diseño de efectividad general y su nivel de satisfacción al trabajar con él. Obtener comentarios de los usuarios ayuda a comprender las áreas problemáticas y saber cuándo y dónde realizar mejoras.

Debido que el alcance de este proyecto es obtener los prototipos y el código, las pruebas de usabilidad quedan afuera de la ejecución, debido a que están enfocadas en la percepción de las funcionalidades por parte de los usuarios que usarán el sistema. Para esto, se necesita tener el software en un entorno de producción o pruebas e ir a probarlo con el usuario en el mismo lugar en donde realiza sus labores diarias. Para este caso, sin embargo, se pueden homologar al paso 2.3 Validar del DLD, en donde se obtiene comentarios valorables para un prototipo.

4.3.3 Desplegar

Desplegar es la tercera sub-etapa de desarrollar y su objetivo es liberar la solución para la producción final, entregar el producto terminado a los clientes e involucrar a los primeros usuarios, de los cuales también se espera recopilar comentarios. Acá la metodología recomienda pasar el código al Web IDE de SAP, para su posterior finalización.

Importar proyecto de compilación de SAP Build a Web IDE

Se importa un prototipo completo y publicado en SAP BUILD a SAP Web IDE. Esto permite convertir el prototipo en una aplicación en vivo. Este prototipo se convertirá en plantilla perfecta para crear la versión real usando UI5/Fiori.

Para obtener el código es necesario ir a la ubicación del prototipo de alta fidelidad y dar clic en el botón “Share”, esto abrirá una ventana en la que se puede obtener el enlace del proyecto, lo importante es que existe un pestaña para realizar la descarga. La Figura 41 es una captura de la sección en donde se descarga el directorio con los códigos correspondientes al prototipo de alta fidelidad. Cabe destacar que la descarga corresponde a un archivo comprimido .zip, el que se conserva para posteriormente ser importado.

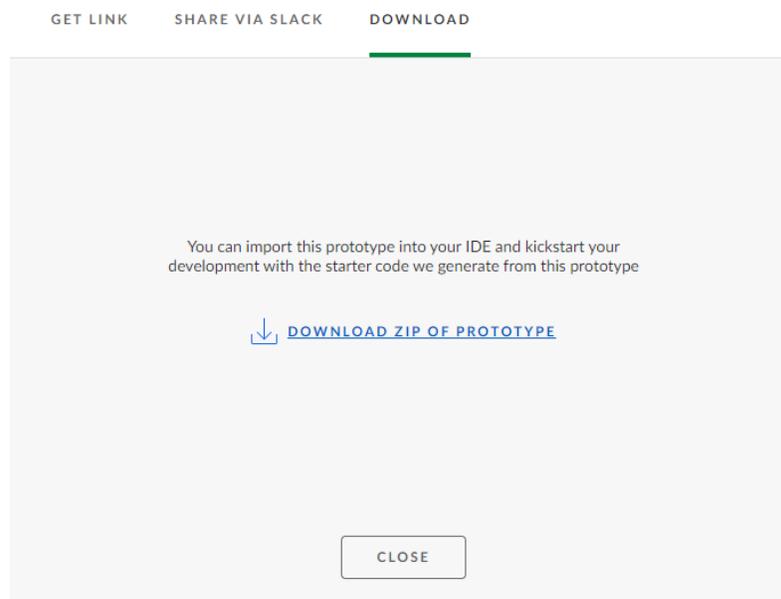


Figura 41 - Descarga del código del prototipo de alta fidelidad de Build

El siguiente paso es crear una cuenta en la plataforma SAP Business Technology Platform, que es donde se accede al entorno de desarrollo integrado web de SAP. La Figura 42 es una captura de la interfaz principal de esta herramienta. Actualmente, la nueva versión del IDE Web de SAP es el Business Application Studio.

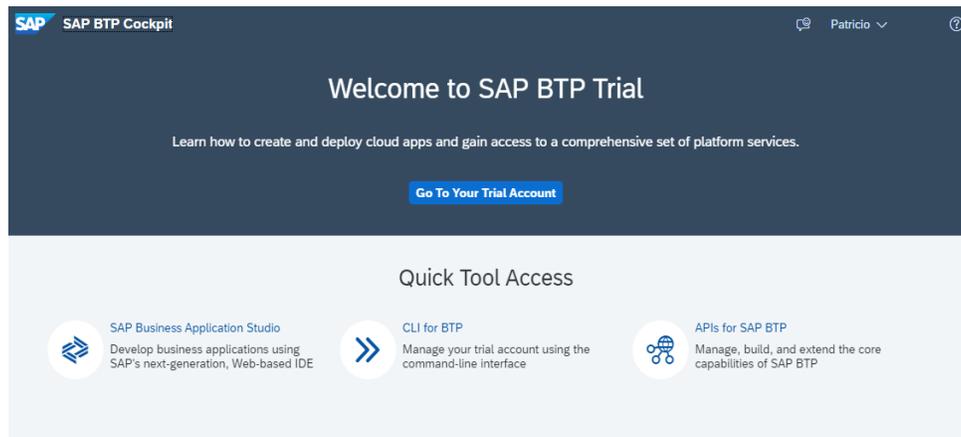


Figura 42 - SAP Business Technology Platform.

El SAP Business Application Studio permite crear y gestionar un entorno de desarrollo según el tipo de aplicación que se desarrolla. Actualmente permite instalar algunas extensiones y herramientas extras para ampliar las funcionalidades del entorno. La Figura 43 muestra el espacio creado llamado “nuevo”, permite desarrollar una aplicación SAP Fiori. Luego de acceder al espacio de trabajo creado se llega a lo mostrado en la Figura 21.

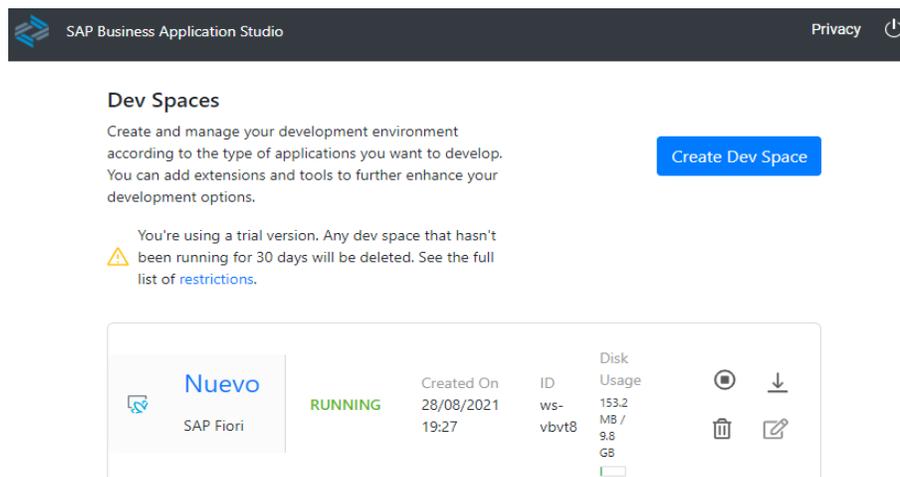


Figura 43 - Creación de espacio de desarrollo en SAP Business Application Studio

El siguiente paso es importar el proyecto .zip obtenido con anterioridad. Para lograrlo hay que dar clic en el botón de importar archivos, seleccionarlo y abrirlo, como lo muestra la Figura 44. El sistema cargará la importación, habilitará algunas extensiones para luego permitir la compilación de proyecto.

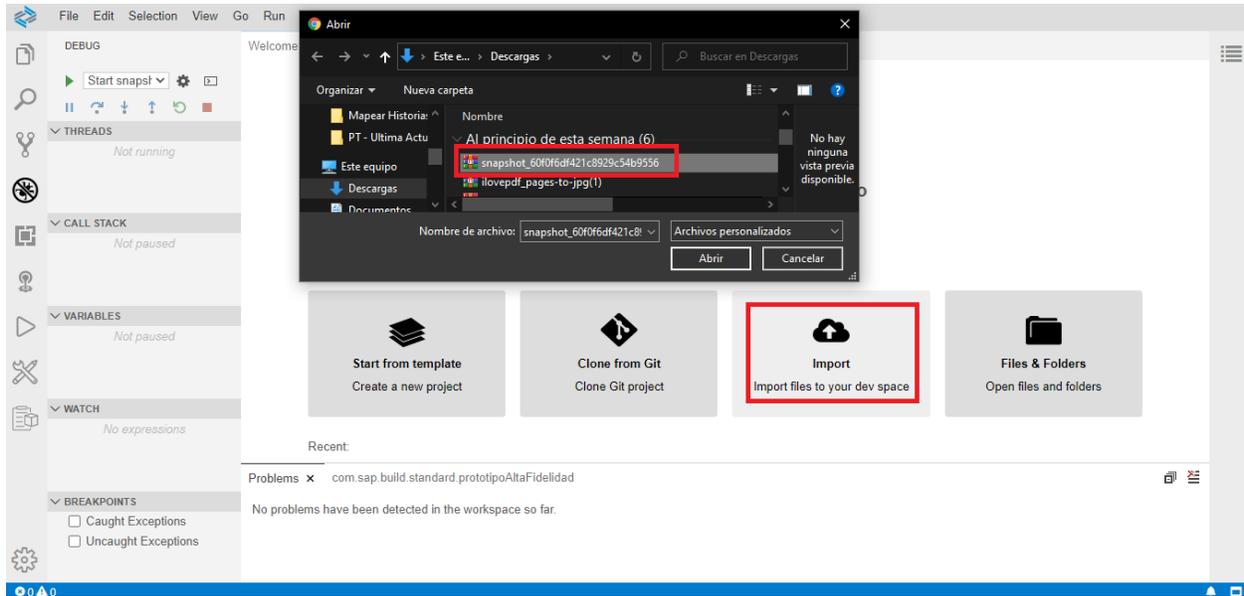


Figura 44 - Importación del proyecto Build al Business Application Studio

La Figura 45 es una captura del directorio del proyecto obtenido desde Build ya importado en el entorno de programación integrado web. A simple vista se pueden apreciar diferentes carpetas que constituyen el directorio del proyecto, entre ellas se destaca: controller, model y view, ya que se corresponden con la arquitectura de software modelo-vista-controlador, separando los códigos por el lado del servidor y por el lado del cliente.

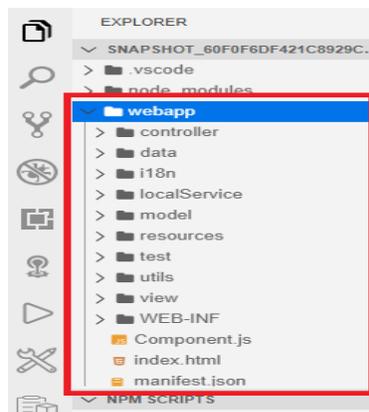


Figura 45 - Directorio proyecto Build

Presionando el botón F5, el IDE realiza la compilación y entrega una URL para acceder en vivo. La Figura 46 contiene una captura del portal de la compilación del prototipo de alta fidelidad, las demás capturas están disponibles en el Anexo 11.

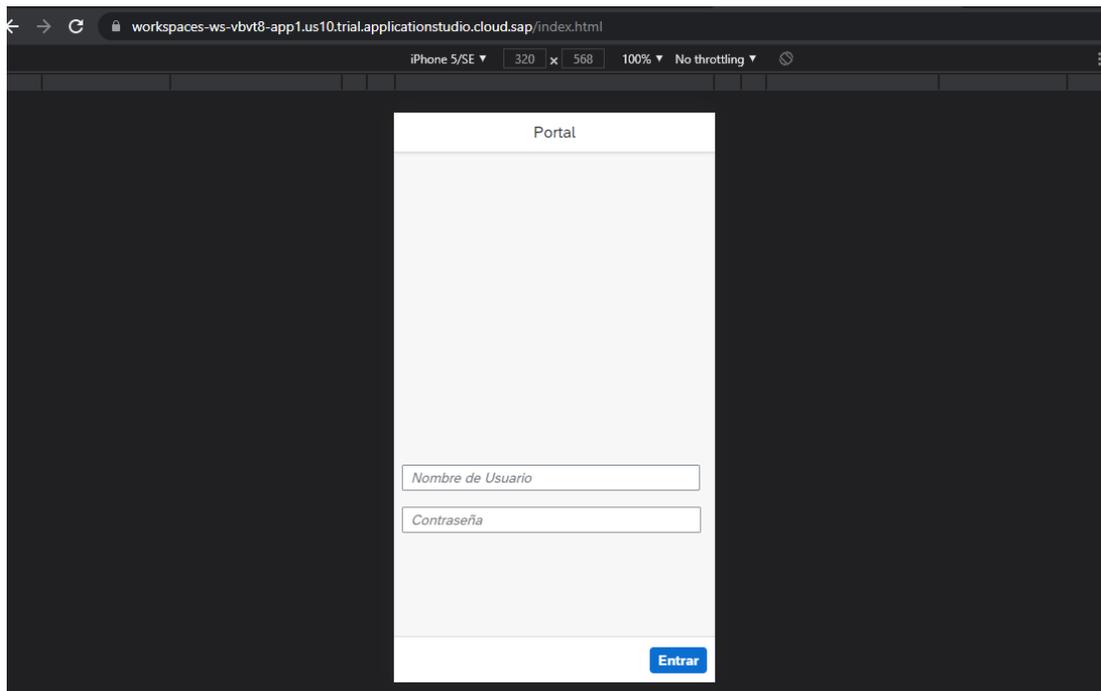


Figura 46 - Portal de la aplicación en vivo

Este es el último paso dado en el proyecto y los artefactos obtenidos son:

- Los documentos del Design Thinking.
- Prototipos de baja y alta fidelidad.
- Código del prototipo de alta fidelidad compilable.

5. Discusión y Conclusiones

El último capítulo de este documento plantea las discusiones y conclusiones obtenidas del proyecto; se realiza un análisis de objetivos propuestos versus logrados, se evidencian los principales aportes del proyecto, se contrastan los resultados, se propone una senda para futuros trabajos relacionados y finalmente, se termina comentando la experiencia general otorgada por este proyecto como totalidad.

5.1 Análisis de Objetivos

El objetivo general inicial del proyecto es entregar una solución tecnológica para la empresa Essbio, la cual consiste en el desarrollo de una aplicación con interfaz Fiori, a través de la metodología DLD (Design-Led Development) que cuenta con lineamiento de DT (Design Thinking), usando las herramientas de SAP: Build y Business Technology Platform (BTP). De esto se puede comentar que el proyecto permitió sentar las bases para concretar el objetivo general; se usó el DLD, Build y BTP para realizar prototipo validado por interesados y usuarios finales, para aportar con los objetivos de la empresa. La Tabla 13 contiene un análisis más fino de los objetivos específicos planteados al inicio del proyecto.

Objetivo Específico	Cumplimiento
Investigar los fundamentos de las metodologías DLD y DT; sus pasos, el valor de su uso y sus puntos de convergencia.	Este objetivo fue completado, el resultado se encuentra en el marco teórico del documento.
Aplicar la metodología DLD, para el desarrollo de una solución tecnológica de Software, generando prototipos de baja y alta fidelidad, por medio de la herramienta de SAP: Build, los que serán desafiados por sus futuros usuarios antes de su aceptación.	Este objetivo fue completado, se siguieron los pasos recomendados por el DLD y se generaron un conjunto de artefactos, incluidos los prototipos de baja y alta fidelidad desarrollados con la herramienta SAP Build.

<p>Implementar funcionalidades sobre prototipos aceptados, exportados desde la herramienta Build, usando el entorno de desarrollo integrado (IDE) Proporcionado por BTP.</p>	<p>De acuerdo al alcance definido con la empresa, que fue generar prototipos aceptados y su código, este objetivo se cumplió. Esta definición se puede encontrar en el documento de alcance de proyecto.</p>
<p>Desplegar el producto de software a producción en un entorno SAP.</p>	<p>Se debe considerar que un paso a productivo en la empresa Essbio conlleva un proceso con muchas etapas, en donde se debe realizar un análisis técnico completo, incluyendo aspectos legales de derechos de uso, ciberseguridad, pruebas de todo tipo (unitarias, de sistema, de estrés), entre otras. Sin embargo, este objetivo puede darse por cumplido, ya que el prototipo logró ser subido al entorno web proporcionado por el Business Technology Platform, con su herramienta SAP Business Application Studio.</p>

Tabla 13 - Objetivos iniciales vs Cumplimiento

De acuerdo a lo anterior, se resume que la experiencia de aprendizaje fue significativa. Los tres primeros objetivos fueron completados y el cuarto parcialmente, debido principalmente al cambio generado por SAP en sus herramientas de desarrollo (especialmente el IDE Web) y la falta de experiencia del ejecutor del proyecto, considerando que el tema desarrollado es innovador y con escasa documentación de apoyo.

5.2 Principales Aportes

El principal aporte de este proyecto es tener un primer acercamiento a una metodología basada en diseño, ya sea el Design Thinking o DLD. Esta afirmación va más allá de la experiencia personal del alumno y se ajusta a la realidad de la carrera, ya que al consultar en el sistema Werken de la red de bibliotecas de la Universidad del Bío-Bío, no se encontró ningún trabajo similar. Actualmente este tipo de metodologías es cada vez más usada en entornos colaborativos y multidisciplinarios, por lo que se debería incorporar en las mallas curriculares de las carreras de la Universidad. Esta es una gran oportunidad para potenciar la creatividad y capacidad de trabajo en equipo de los alumnos, y además, generar proyectos innovadores y significativos, sobre todo para aportar en vinculación con la comunidad y el mundo empresarial.

En el proceso investigativo, buscando el origen del impulso del Design Thinking, se averiguó que tanto la Universidad de Stanford (Estados Unidos) como la Universidad de Potsdam (Alemania), desde el año 2005 y 2006, respectivamente, agregaron el Design Thinking como asignatura obligatoria en las mallas curriculares de todas sus carreras de ingeniería. Es decir, para ser ingeniero de esas Universidades, se debe cursar el taller de Design Thinking, en donde los estudiantes conocen compañeros de distintas carreras, colaboran en proyectos multidisciplinarios y generan soluciones innovadoras y significativas.

Actualmente en Latinoamérica, solo la Universidad Mayor ha tomado la iniciativa y creó su propia D-School tras realizar una alianza con el instituto Hasso Plattner, correspondiente a la Universidad de Potsdam¹⁰. La Universidad Mayor declara: “El D-School U. Mayor se constituye como un elemento central en la formación de todos nuestros estudiantes, ya que a través de asignaturas y talleres tendrán la oportunidad de interiorizarse en esta metodología, cuyo proceso será llevado a cabo por la recién creada **Facultad de Estudios Interdisciplinarios**. Este programa es de amplio alcance e incluye el trabajo con estudiantes, académicos y colaboradores, con el objetivo de introducir el modelo como parte de nuestra cultura organizacional, dando un nuevo impulso a nuestro espíritu emprendedor. Todos los estudiantes de la Universidad Mayor experimentarán en el D-School, en las distintas etapas de su vida académica, a través de talleres y asignaturas. De esta manera, vivenciarán esta nueva forma de pensamiento que les permitirá desarrollar habilidades que los prepararán para poder interactuar en un mundo complejo, diverso, transdisciplinario, colaborativo y cambiante.”

La Figura 47 presenta un mapa global de las D-School alrededor del mundo. Solo existen 9 D-School, de los cuales cuatro están en el hemisferio sur y solo una en Latinoamérica.

¹⁰ D-School Universidad Mayor: <https://www.umayor.cl/um/Dschool/Dschool/10000>



Figura 47 - Mapa Global de D.School (obtenido de la web oficial de la Universidad Mayor)

En cuanto a la empresa Essbio, el aporte de este trabajo es poseer una revisión de literatura y un caso de estudio, lo cual les servirá al momento de querer replicar la metodología en otros proyectos. En particular, este caso de estudio sirvió para tener una visión diferente del proyecto de sistema de toma de mediciones en piso de planta, generando información relevante sobre el entendimiento de los operadores en el contexto de su trabajo. Además, se declaró el objetivo empresarial del proyecto, se obtuvo un prototipo validado y el directorio del proyecto con el código.

5.3 Trabajos Futuros

Mientras se ejecutaba este proyecto, se encontró una entrada en un blog oficial de SAP, en donde se informa que SAP Build se reemplazará por una herramienta similar, pero con nuevas características, la que se espera ser lanzada aproximadamente el año 2025.

Existen otras herramientas que soportan el proceso de creación de experiencia de usuario como:

- **Pencil Project:** Este programa para UX Design permite diseñar Wireframes, prototipos y diagramas para visualizar las rutas y procesos que componen el UX de los productos.
- **InVision:** Esta aplicación UX permite visualizar Wireframes, prototipar mockups y cargar distintos tipos de material audiovisual para apoyar el proceso.
- **Caravel:** Permite gestionar documentos, Wireframes, modelados y diagramas con un número ilimitado de participantes.
- **Balsamiq Mockups:** Esta herramienta permite recrear maqueta de papel pero en formato digital.
- **Adobe Experience Design:** Permite dibujar, reutilizar, mezclar vectores, crear ilustraciones de Wireframes, diseñar pantallas, prototipos interactivos y activos para producción.
- **Figma:** Permite crear prototipos y maquetas dinámicas, probar usabilidad y sincronizar todo el progreso. Pueden trabajar varias personas a la vez.

De estas herramientas las más conocidas son las últimas tres y pueden potenciar la aplicación de DLD o Design Thinking, respaldando el proceso creativo. Se propone utilizar algunas de estas herramientas para continuar o comenzar de cero una investigación aplicada con características similares a la documentada en este proyecto.

En cuanto a las metodologías basadas en diseño como tal, en el punto 5.2 se recomendó su uso para las mallas curriculares de las distintas carreras de la Universidad, para entregar mayor valor en la formación de profesionales, entregando técnicas y herramientas que son tendencia mundial.

Finalmente, trabajos futuros podrían aplicar metodologías basadas en diseño en otras empresas o instituciones utilizando las directrices generadas en este proyecto.

5.5 Experiencia General

La experiencia general obtenida del desarrollo de este proyecto se nutrió de dos actividades importantes realizadas por el alumno: a) una experiencia laboral en la empresa Essbio y b) una experiencia académica como alumno ayudante en las materias desarrolladas en este proyecto. a) Actualmente, debido a la pandemia Essbio no permite la colaboración con alumnos tesisistas o practicantes, por lo que para realizar este proyecto, y después de varios intentos, se optó por postular a un puesto de trabajo. Precisamente, había una vacante para realizar un reemplazo del administrador del sistema que actualmente se usa para realizar la subida de datos diaria de los operadores de producción de agua potable y tratamiento de aguas servidas. Este sistema está en proceso de actualización por una nueva versión, por lo que el trabajo principal fue realizar migración de cuentas y capacitar operadores. La empresa cuenta con operación en 3 regiones del país, en casi cada localidad de estas regiones, por lo que el trabajo de capacitación se realizó individualmente por operadores, debido a la distribución geográfica, mala conectividad o dificultades personales. Esto ayudo a comprender las dificultades y deseos de los usuarios finales (primeras etapas de la metodología DT y DLD), ya que las capacitaciones eran muy personalizadas, adaptándose a cada operador. Otra responsabilidad principal fue asegurar el correcto funcionamiento de la nueva versión del sistema que se estaba migrando, por lo que muchas veces se apoyó el proceso de subida de datos de los operadores a través de llamadas telefónicas o usando la herramienta TeamViewer, de conexión remota.

b) En paralelo, en conjunto con la profesora guía, se realizó la ayudantía de la asignatura Sistemas de Información 1-2021 de la carrera Ingeniería de Ejecución en Computación e Informática. Parte de las horas prácticas fueron usadas para realizar un taller de Design Thinking a los estudiantes de esta asignatura basada en un curso de la plataforma OpenSap.com mostrado la Figura 48.

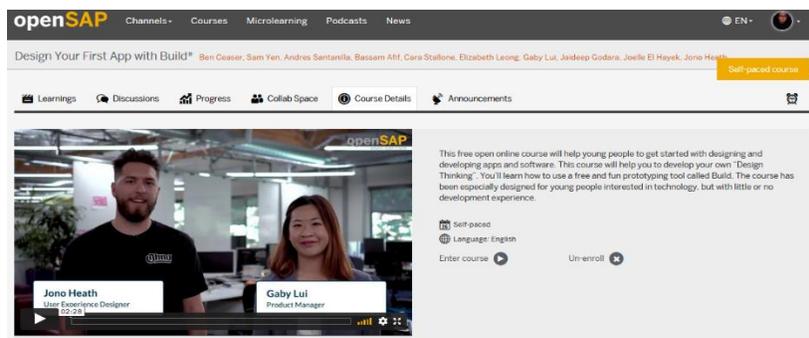


Figura 48 - Curso de Design Thinking y Build en OpenSap.Com

La Figura 49 muestra el avance del curso que tomo el alumno tesista para ampliar sus conocimientos del tema y poder aplicarlos tanto en la empresa Essbio como en la ayudantía de la clase de Sistemas de Información. El curso de OpenSAP consistió en aprender los conceptos básicos, etapas y artefactos asociados a la metodología basada en diseño Design Thinking, incluyendo el enfoque SAP. El Anexo 13 contiene una captura del documento de participación generado por la plataforma al terminar las semanas. Es un curso de 4 semanas el cual fue adaptado al contexto universitario y se replicó obteniendo excelentes resultados en el rendimiento de la mayoría de los alumnos. De este modo, los estudiantes del curso Sistemas de Información trabajaron en desarrollar soluciones innovadoras para una problemática definida en un caso de estudio.

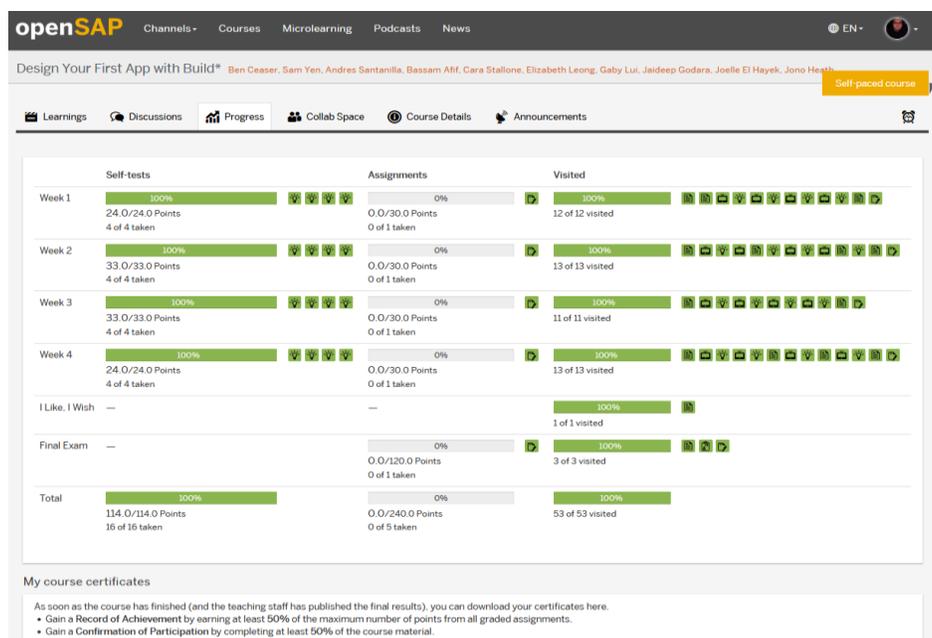


Figura 49 - Avance del curso en OpenSap.com

Las dos labores mencionadas fueron el impulso perfecto para alinear los objetivos de aprendizaje general del término de ciclo como estudiante de pregrado de la carrera de Ingeniería Civil en Informática. Por una parte fue la inserción al mundo laboral, el aprendizaje a través de la administración del sistema actual de toma de mediciones de Essbio, la generación de la empatía necesaria para entender a los operadores y la adquisición de experiencia en un equipo de TI consolidado; y por otro lado fue la interiorización en el Design Thinking y la experiencia como profesor ayudante en un proyecto curso. Personalmente fue una experiencia enriquecedora y permitió tener el primer acercamiento con una gran empresa en el contexto de trabajo real, a modo de cierre de la formación profesional.

6. Bibliografía

- Abrahamsson, P., Warsta, J., Siponen, M., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *Proceedings 25th International Conference on Software Engineering IEEE*.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Stoughton: Addison Wesley.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & M., D. (2001). <http://agilemanifesto.org>. Obtenido de <http://agilemanifesto.org>: <http://agilemanifesto.org/iso/es/manifesto.html>
- Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT Software*, 14-24.
- Brown, T. (2008). Design Thinking. *Harvard Business Review*.
- Castillo, M., Alvarez, A., & Cabana, R. (2014). Design Thinking: como guiar a estudiantes, emprendedores y empresarios en su aplicación. *Ingenieria Industrial ISSN 1815-5936*.
- Cervone, H. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives*, 18-22.
- Corral, L., & Fronza, I. (2018). Design Thinking and Agile Practices for Software Engineering. *SIGITE'18*. New York: Association for Computing Machinery.
- d.School. (2010). Bootcamp bootleg y del Design Process Mini-Guide.
- Davis, A., Bersoff, E., & Comer, E. (1988). A Strategy for Comparing Alternative Software Development Life Cycle Models. *IEEE Transactions on Software Engineerring*, 1453-1461.
- Fowler, M. (13 de Diciembre de 2005). *martinFowler.com*. Obtenido de [martinFowler.com](http://martinfowler.com): <https://martinfowler.com/articles/newMethodology.html>
- Gaete, J., Villarroel, R., & Figueroa, I. (2021). Enfoque de aplicación ágil con Scrum, Lean y Kanban. *Revista chilena de ingeniería, vol. 29*, 141-157.
- Ghosh, S. (2015). Systemic Comparison of the Application of EVM in Traditional and Agile Software Project. *PM World Journal*.

- Grandon, E., Campos, P., Fica, R., & Ávila, F. (2018). A Development Methodologies Recommender System Based on Knowledge from the Software Industry. *IEEE*.
- Khurana, H., & Sohal, J. (2011). Agile: The necessitate of contemporary software developers. *International Journal of Engineering Science & Technology*, 1031-1039.
- Kirovska, N., & Koceski, S. (2015). Usage of Kanban methodology at software development teams. *Journal of Applied Economics and Business*, 25-34.
- Larman, C. (2003). *Agile and Iterative Development A Manager's Guide*. Boston: Addison Wesley.
- Larman, C., & Basili, V. (2003). Iterative and Incremental Development: A Brief History. *IEEE Computer Society*, 47-56.
- Martin, R., & Martin, M. (2006). *Agile principles, patterns, and practices in c#*. Westford, Massachusetts: Pretince Hall.
- Navarro, A., Fernández, J., & Morales, M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA Universidad Autónoma del Caribe*, 30-39.
- Pereira, J., & Russo, R. (2018). Design Thinking Integrated in Agile Software Development: A Systematic Literature Review. *Procedia Computer Science* (págs. 775-782). Sao Paulo, Brazil: ScienceDirect.
- Plattner, H., Meinel, C., & Weinberg, U. (2009). Design Thinking. *Innovation lernen*.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley.
- Powell, D. (2018). Kanban for Lean Production in High Mix, Low Volume Environments. *Kanban for Lean Production in High Mix, Low Volume Environments*.
- Ronald, J. (2004). *Extreme Programming Explained*. Boston: Addison-Wesley.
- Royce, W. (1970). Managing the development of large software systems: Concepts and techniques. *IEEE Wescon*, 1-9.
- Schallmo, D., Williams, C., & Lang, K. (2018). An Integrated Design Thinking Approach – Literature: An Integrated Design Thinking Approach – Literature Thinking. *The ISPIM Innovation Conference*.
- Sommerville, I. (2010). *Software engineering [9ª ed.]*. Boston: Addison Wesley.

Stoica, M., Mircea, M., & Ghilic-micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economică vol. 17, no. 4*, 71-72.

Sugimori, Y., Kusunoki, K., Cho, F., & Uchikawa, S. (1977). *International Journal of Production Research*, 553-564.

7. Anexos

7.1 Anexo 1 – Documento de alcance de proyecto

Proyecto toma de mediciones en terreno (Fiori & Build)

OBJETIVO GENERAL

Disminuir el tiempo que les requiere a los operadores subir los datos correspondientes a las mediciones de parámetros al sistema, con la finalidad de controlar estos datos con respecto a las variables operacionales de la empresa, para obtener una mejor calidad de información y apoyar en la toma de decisiones.

ALCANCE DEL PROYECTO

Implementar un prototipo del sistema de toma de mediciones en terreno utilizando SAP Fiori UIS y la herramienta SAP Build, y subirlo al entorno de desarrollo integrado web proporcionado el Business Technology Platform.

PARTES INTERESADAS

- Gerencia Procesos y Soporte Operacional.
- Subgerencia de Confiabilidad.
- Gerencias Regionales VI, VII y VIII.
- Gerencia Administración y Finanzas / Subgerencia de Informática / Departamento de gestión de procesos operacionales e inteligencia de negocios (Área ejecutora).

ORGANIZACIÓN DEL PROYECTO

- **Coordinadores:** Patricio Pardo, Carlos Ceballos, Elizabeth Grandón.
- **Contribuyentes Directos:** Claudio Melo, Víctor Galaz, Alex Vilches, Marcos Medina, Camilo Altamirano.
- **Contribuyentes Apoyo:** Denny Mora, Paulina Sepúlveda.

ENTREGABLES DEL PROYECTO

- Artefactos Persona, Replanteamiento del Problema y Punto de Vista.
- Prototipo de alta fidelidad.
- Código del prototipo del proyecto.

HITOS

Hitos	Fecha
• Entregables etapa DESCUBRIR (Alcance, Investigación y Sintetizar)	12-07-2021
• Entregables etapa DISEÑO (Idear, Prototipo y Validación)	28-07-2021
• Entregables etapa DESARROLLO (Implementación, Test y Desplegar)	13-08-2021

7.2 Anexo 2 – SCRIPT Entrevista

INTRODUCCIÓN Y CONTEXTO

Hola, mi nombre es Patricio Pardo, de profesión ingeniero civil informático, actualmente me desempeño como ingeniero de procesos y mejora continua en Essbio. En conjunto con la Universidad del Bío-Bío, se me ha encargado coordinar un proyecto colaborativo de co-innovación. Por parte de la Universidad nos acompaña la profesora Elizabeth Grandón y será la encargada de la vinculación de la empresa con la Universidad.

Este proyecto tiene como finalidad generar una solución tecnológica que disminuya el tiempo que les requiere a los operarios subir los datos correspondientes a las mediciones de parámetros al sistema actual, con el objetivo de controlar estos datos con respecto a las variables operacionales, para así obtener una mejor calidad final de información y apoyar el proceso de toma de decisiones de la empresa.

Lo invito a comentarnos sus apreciaciones y experiencias de la manera más franca posible, ya que la información que extraeremos de esta serie de entrevistas es vital para el proceso de desarrollo. Con una correcta comprensión del problema, y aplicando la empatía necesaria con los futuros usuarios, es como podremos generar una solución mucho más significativa. Esta serie de entrevistas se basa en sesiones de una hora como máximo, a 6 operarios (3 operadores AP y 2 operadores AS).

Le recuerdo que sus datos son confidenciales y en caso de que necesite detener la entrevista por el motivo que sea, siéntase en la libertad de comunicármelo.

Antes de comenzar, es necesario pedir su consentimiento para grabar esta sesión, ya que es el mejor respaldo que podemos crear para recurrir a esta fuente de conocimiento cuando lo necesitemos más adelante. ¿Está de acuerdo con que la sesión sea grabada?

REGLAS Y CONSIDERACIONES

- Bueno, antes de pasar a las preguntas, es necesario definir algunas cuantas reglas y consideraciones.
- Si usted [el usuario] necesita interrumpir la reunión por el motivo que sea me da aviso. De todos modos no debería superar la hora de duración.
- La información recopilada durante esta entrevista ayudará a la toma de decisiones de diseño y características del software.
- No hay respuestas correctas o incorrectas, cualquier aporte es importante.
- Debe tener en cuenta que usted representa a otros operadores con el mismo rol.
- Ante cualquier duda o comentario no dude en pedir la palabra.

PREGUNTAS DE ANTECEDENTES

1. ¿Cuál es su nombre completo?
2. ¿Cuál es el nombre de su cargo en la empresa?
3. ¿Cuánto tiempo lleva desempeñando este cargo?
4. Describa brevemente su función.
5. Mencione sus principales responsabilidades.
6. Con respecto al uso de sistemas informáticos ¿Considera que se le complica adaptarse? Y en el caso de otros compañeros ¿Se repite esa tónica?

PREGUNTAS DE ALTO NIVEL

1. Actualmente, ¿Cómo realiza usted el proceso de toma de mediciones? Incluyendo del inicio de la actividad, obtención del dato (desde un visor, procesamiento de laboratorio o cálculos), la subida al sistema, hasta el cierre del periodo.
2. Cuando realiza este proceso, ¿Cuáles son las principales dificultades que enfrenta? (dificultad para la tecnología, exceso de tiempo, falta de información adicional, problemas con los equipos y la conectividad, etc.).

3. ¿Cómo soluciona o amortigua las dificultades que enfrenta?
4. En su opinión, ¿Cuáles son los puntos débiles del proceso actual de toma de mediciones?
5. ¿Cómo cree usted que se podrían minimizar los puntos débiles del proceso?
6. ¿Usted trabaja en más de una planta? Si es así ¿Repite los mismos procesos en una y otra? ¿Qué diferencias, con respecto al proceso de tomar de mediciones, puede contarme entre una planta y otra?

CIERRE

Ya hemos cubierto los puntos de la entrevista.

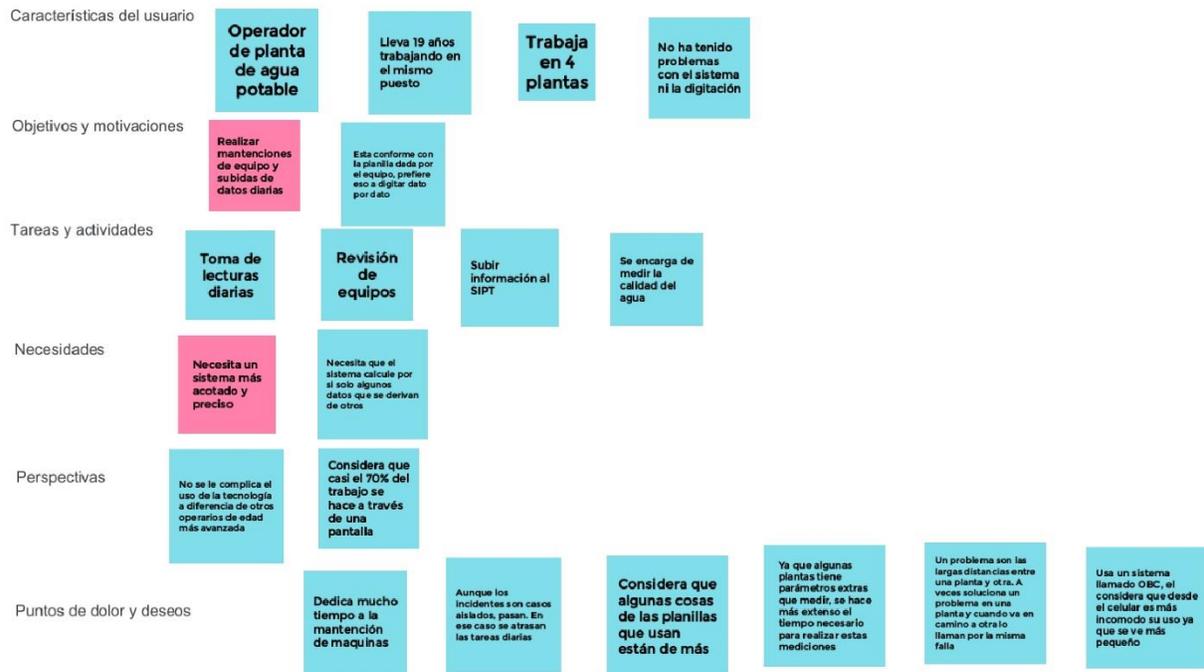
Recapitulando:

- Hoy le hemos informado a grandes rasgos en que consiste el proyecto.
- Hemos recopilado información con respecto a su cargo y rol en la organización.
- Hemos obtenido una útil apreciación del proceso de toma de mediciones, destacando el flujo del proceso, algunas dificultades y puntos débiles del proceso y el sistema.
- Además, he tenido la oportunidad de presentarme y conocerlo a usted.

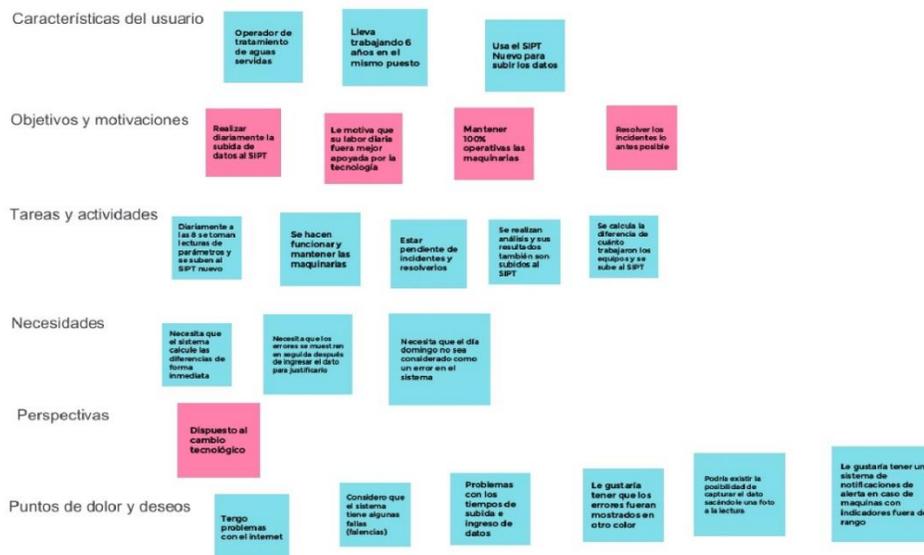
Espero que esta sesión de entrevista haya sido de su agrado, le recuerdo que próximamente necesitaremos nuevamente de su colaboración, para ayudarnos a realizar las pruebas de usabilidad sobre el prototipo. Estoy agradecido de su buena disposición y tiempo sacrificado. Le recuerdo que la información obtenida en esta y las demás entrevistas es vital para lograr una buena solución y aportar al trabajo de todos en la organización.

7.3 Anexo 3 – Hallazgos por usuario

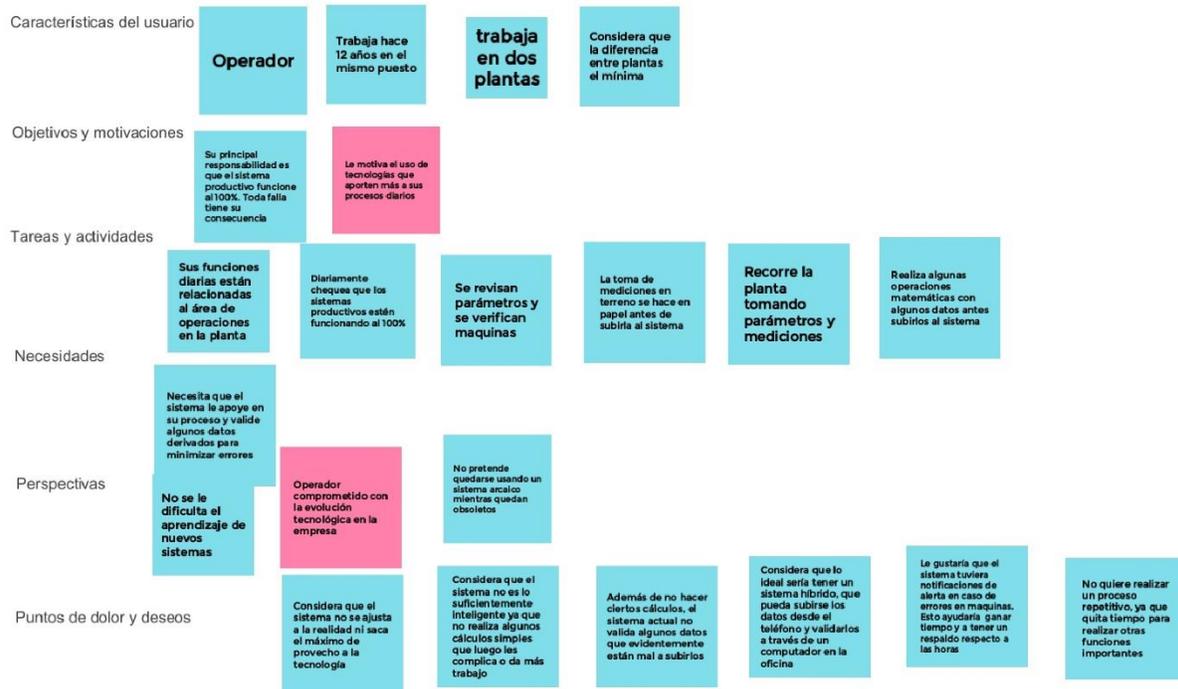
OPERADOR AP 1



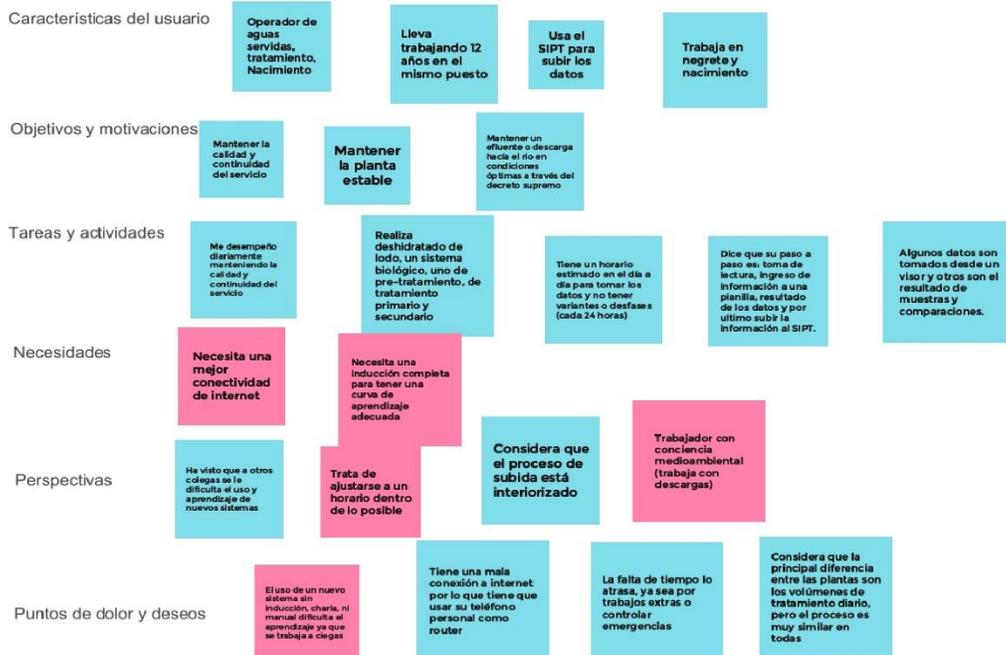
OPERADOR AS 1



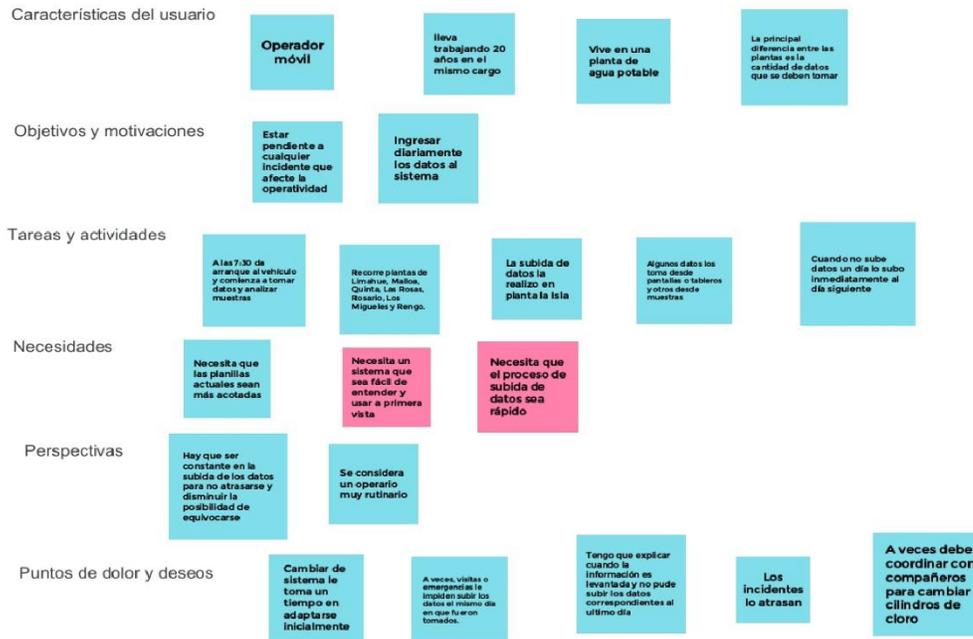
Operador AP 2



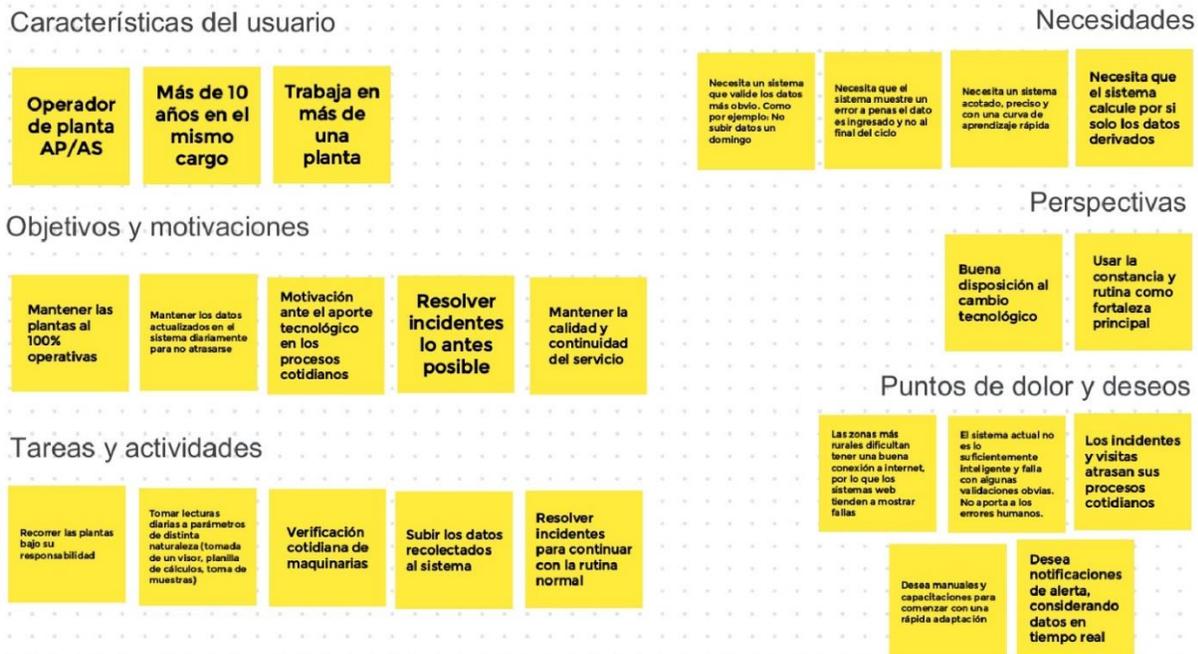
Operador AS 2



Operador AP 3



7.2 Anexo 4 – Hallazgos generales



7.5 Anexo 5 – Persona


Persona Template



Claudio

Operador Producción y Tratamiento

“No pretendo quedarme con un Sistema arcaico que va a ir pasando el tiempo y vamos a ir quedando obsoletos, tenemos que irnos movilizandolo, de eso se trata la mejora continua ”

ABOUT

Operador de planta con más de 10 años de experiencia en el mismo cargo. Trabaja en más de una planta, las que normalmente están separadas por varios kilómetros de distancia.

WORKS WITH

Desarrolla sus laborales normales en solitario.
Un supervisor zonal coordina y le habilita para realizar sus labores.
El soporte de TI le apoya con los problemas relacionados al sistema actual.

JOB RESPONSIBILITIES

- Recorrer diariamente sus plantas y verificar el correcto funcionamiento de las maquinarias.
- Acudir oportunamente a resolver los incidentes ocurridos en sus plantas y apoyar a otros operadores en caso de ser necesario.
- Tomar lecturas de datos diarias a los distintos parámetros asociados a sus plantas, para posteriormente subirlos al sistema actual.

NEEDS

- Necesita un sistema acotado, preciso y con una rápida curva de aprendizaje, que no solicite datos redundantes o innecesarios.
- Necesita un sistema inteligente que calcule por si solo los datos derivados, que valide eficientemente el ingreso de datos y que pida inmediatamente las justificaciones de los datos fuera de rango.

MAIN GOALS

- Mantener sus plantas operativas y funcionando al 100% para conseguir una calidad y continuidad del servicio óptima.
- Mantener una actualización diaria de los datos recolectados en cada planta.

PAIN POINTS

- Los incidentes y visitas largas atrasan sus procesos cotidianos.
- El sistema actual falla en validaciones obvias, no aportando a amortiguar los errores humanos.
- No tener una buena conexión a internet entorpece el proceso de subida de datos al sistema actual.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

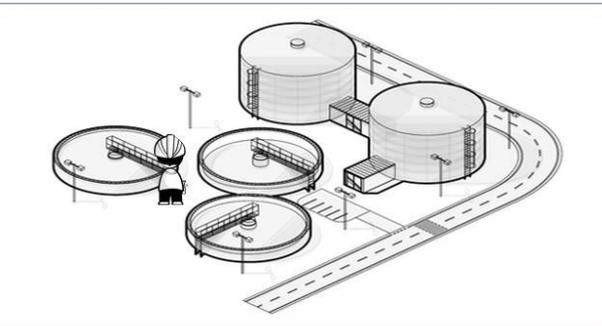
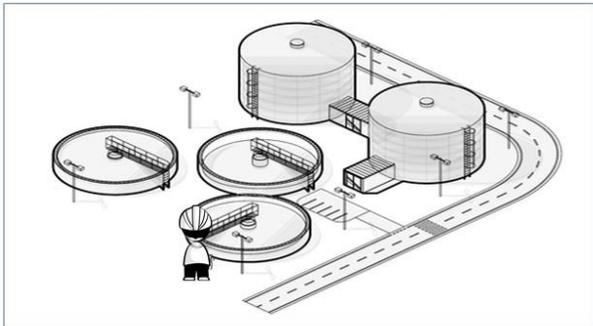
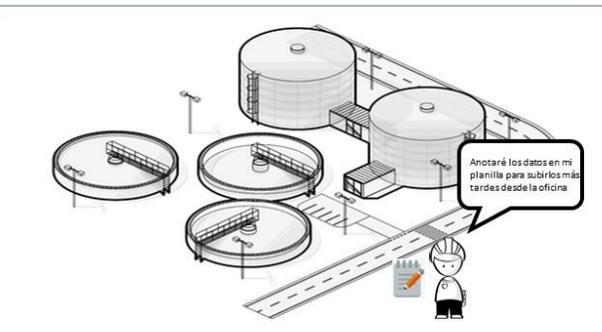
7.6 Anexo 6 – Storyboards

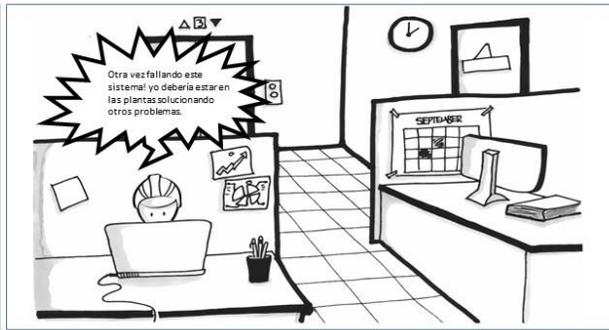
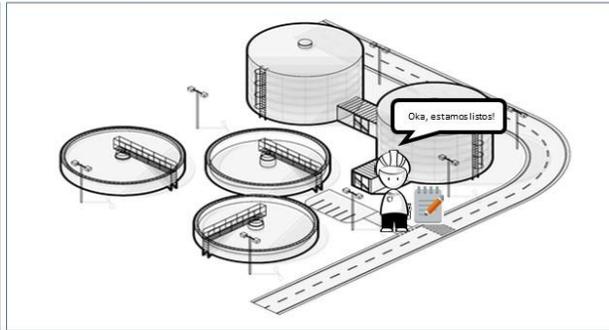
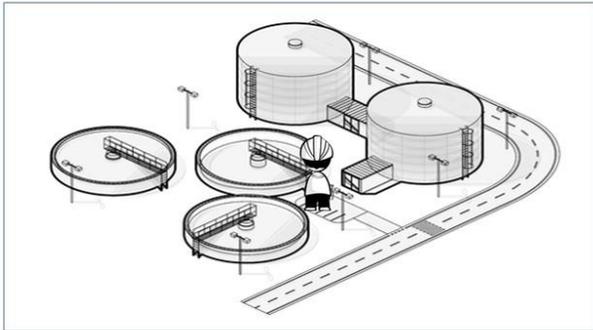
Escena 1: Sin aplicación móvil



Operador de Producción de Agua potable y Tratamiento de Aguas Servidas.

Más de 10 años de experiencia en el mismo cargo. Trabaja en más de una planta, las que normalmente están separadas por varios kilómetros de distancia.





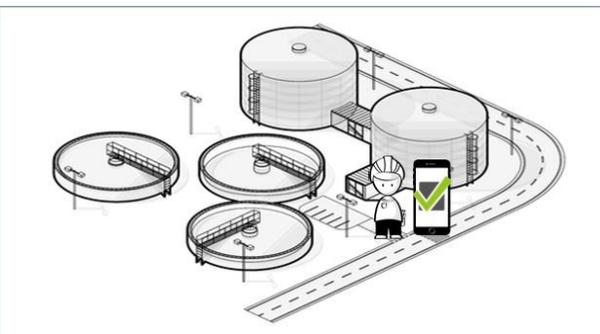
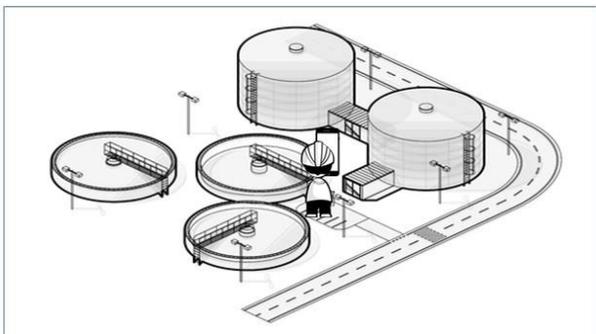
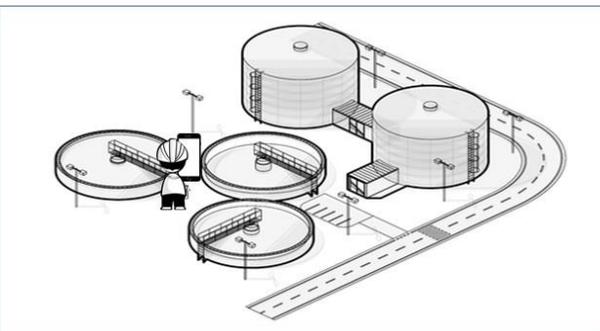
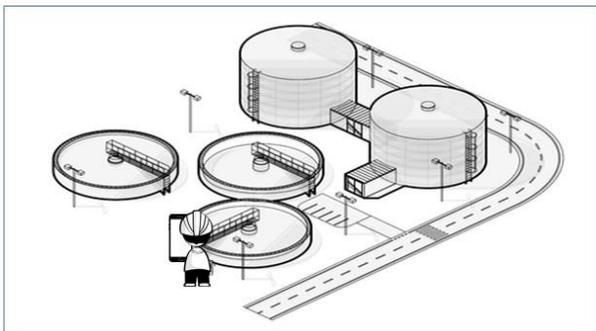
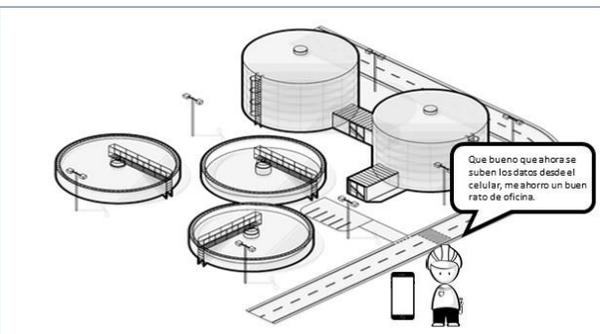
Escena 2: Con aplicación móvil



Operador de Producción de Agua potable y Tratamiento de Aguas Servidas.

Más de 10 años de experiencia en el mismo cargo. Trabaja en más de una planta, las que normalmente están separadas por varios kilómetros de distancia.





7.7 Anexo 7 – Casos de Uso

Diagrama de Casos de Uso

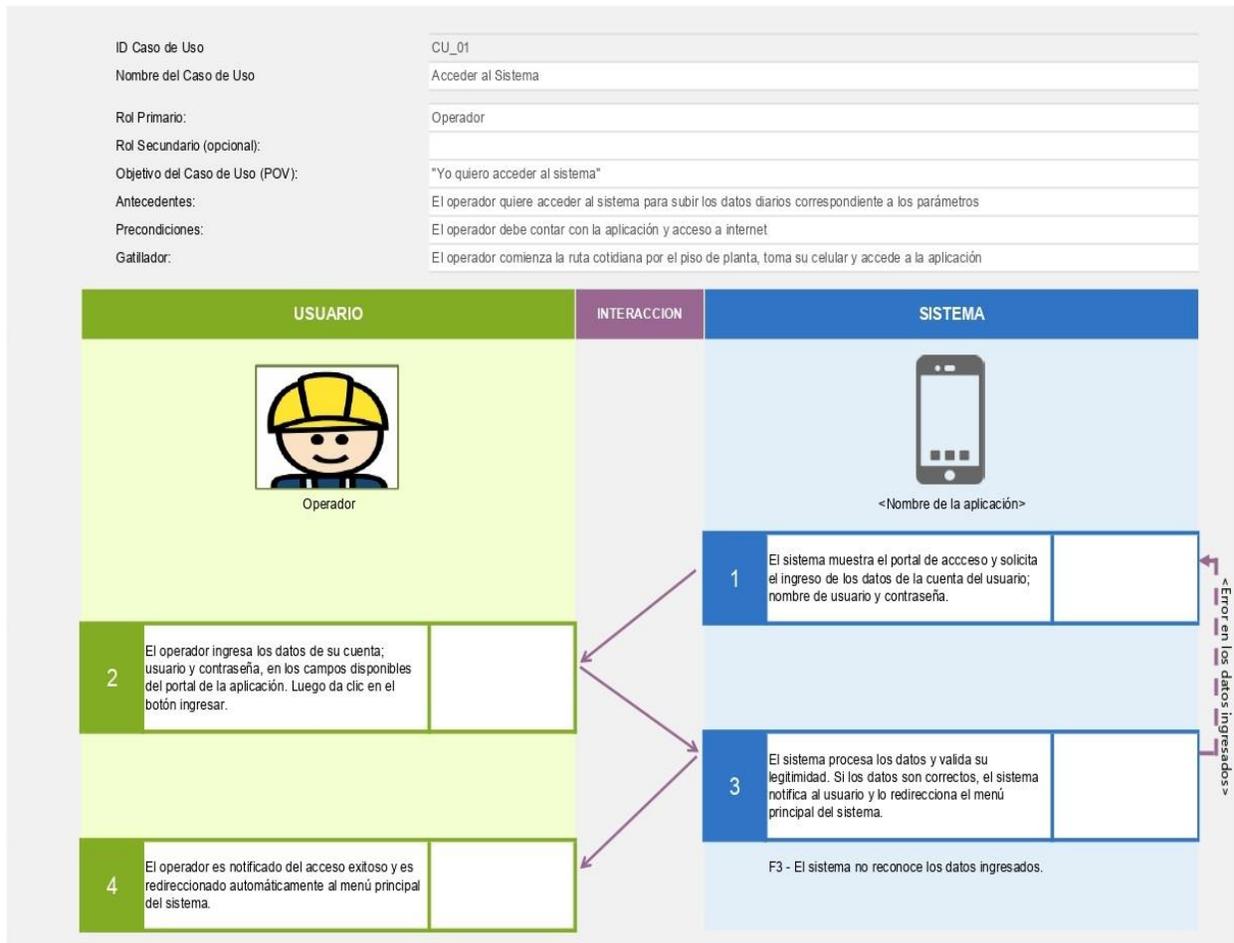


Diagrama de Casos de Uso

ID Caso de Uso	CU_02
Nombre del Caso de Uso	Configurar Ruta
Rol Primario:	Operador
Rol Secundario (opcional):	
Objetivo del Caso de Uso (POV):	"Yo quiero configurar la ruta a recorrer en la toma de mediciones en piso de planta"
Antecedentes:	El operador necesita configurar la ruta con la que recorrerá una planta, para ir subiendo los datos a medida que los va tomando.
Precondiciones:	El operador debe haber iniciado la aplicación y accedido al sistema con los datos de su cuenta
Gatillador:	El operador siente que la ruta preconfigurada no le acomoda y necesita cambiarla

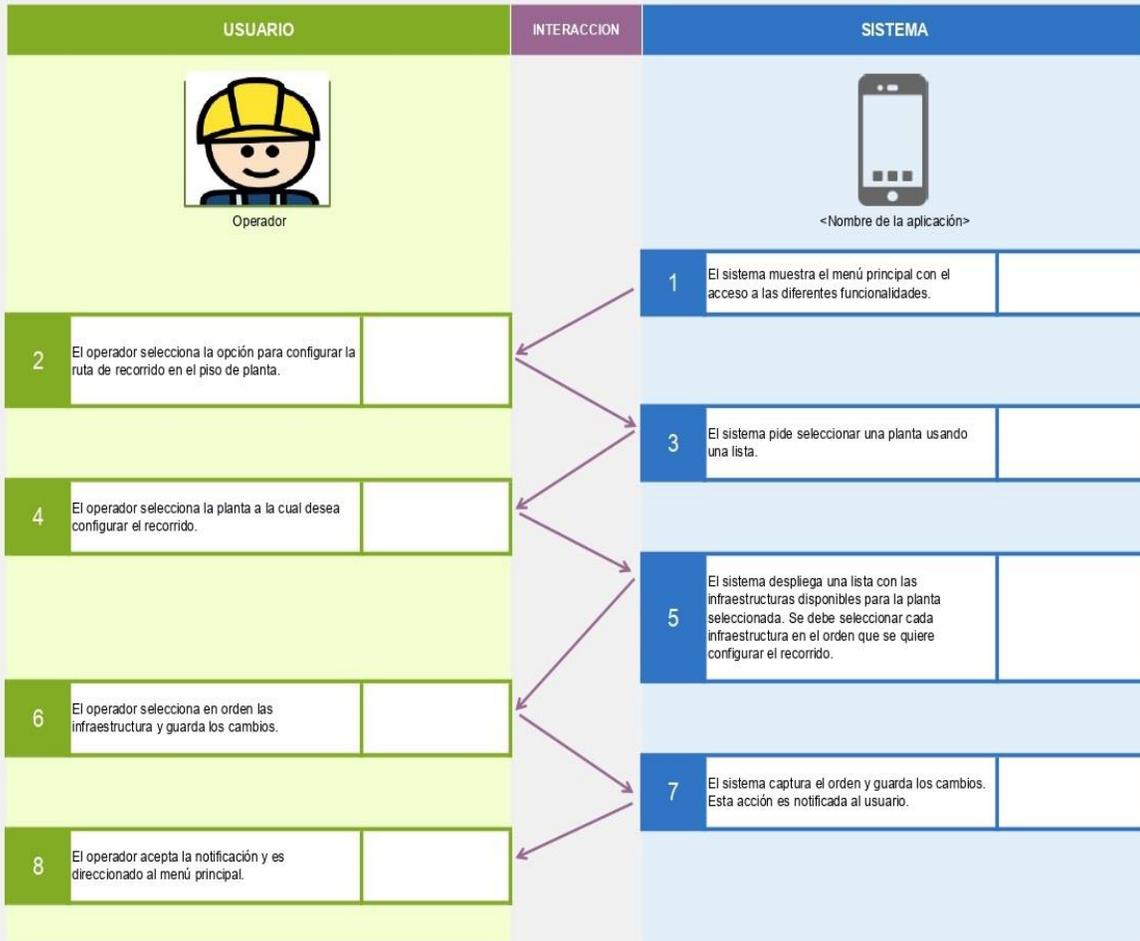


Diagrama de Casos de Uso

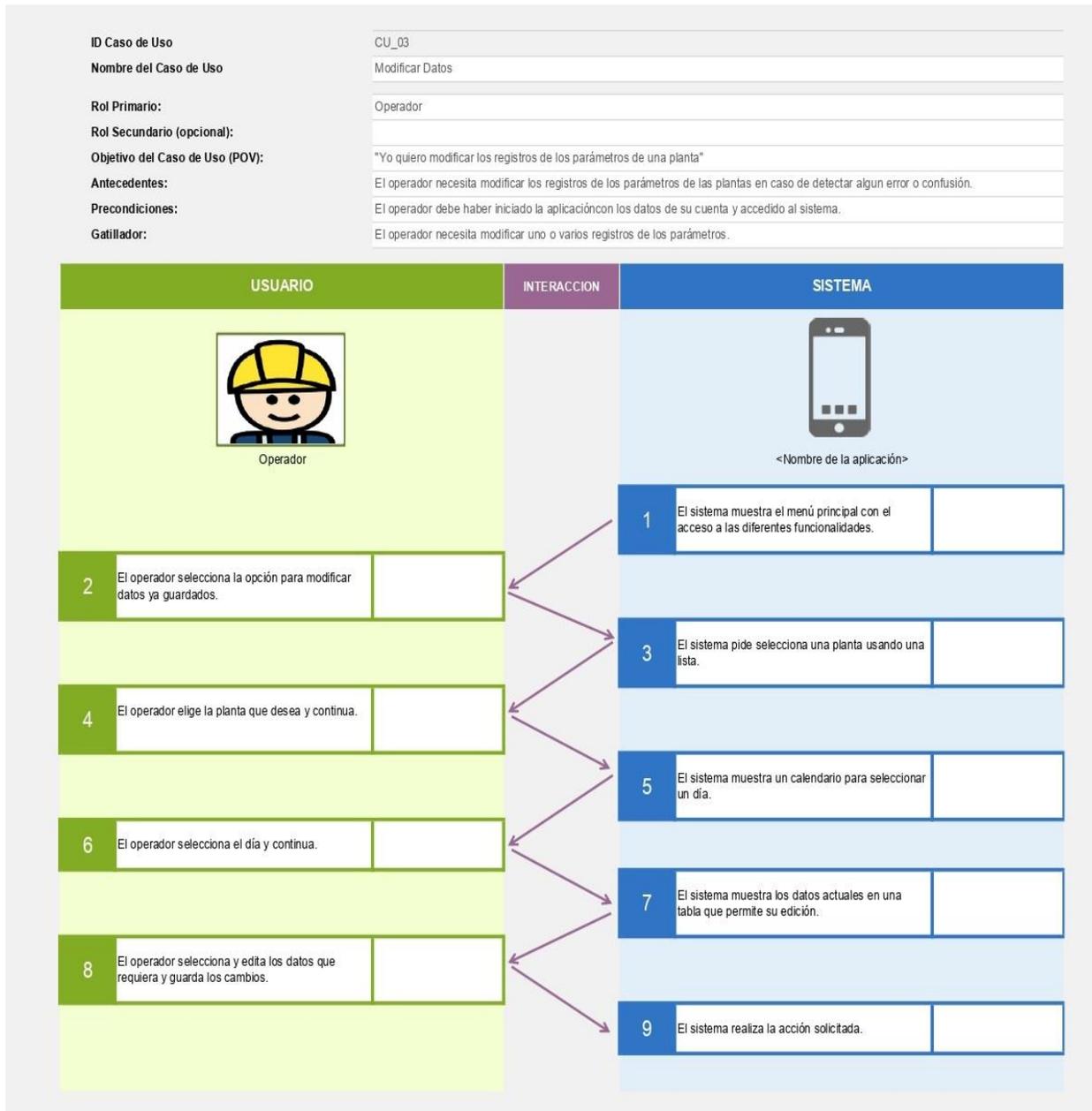


Diagrama de Casos de Uso

ID Caso de Uso	CU_04
Nombre del Caso de Uso	Generar Reportes
Rol Primario:	Operador
Rol Secundario (opcional):	
Objetivo del Caso de Uso (POV):	"Yo quiero obtener reportes con información útil para realizar mi trabajo"
Antecedentes:	El operador necesita reportes de parámetros.
Precondiciones:	El operador debe haber iniciado la aplicación con los datos de su cuenta y accedido al sistema.
Gatillador:	El operador necesita saber algun dato en general de los parametros que debe medir. Por ejemplo: frecuencia, obligatoriedad, justificabilidad, etc.

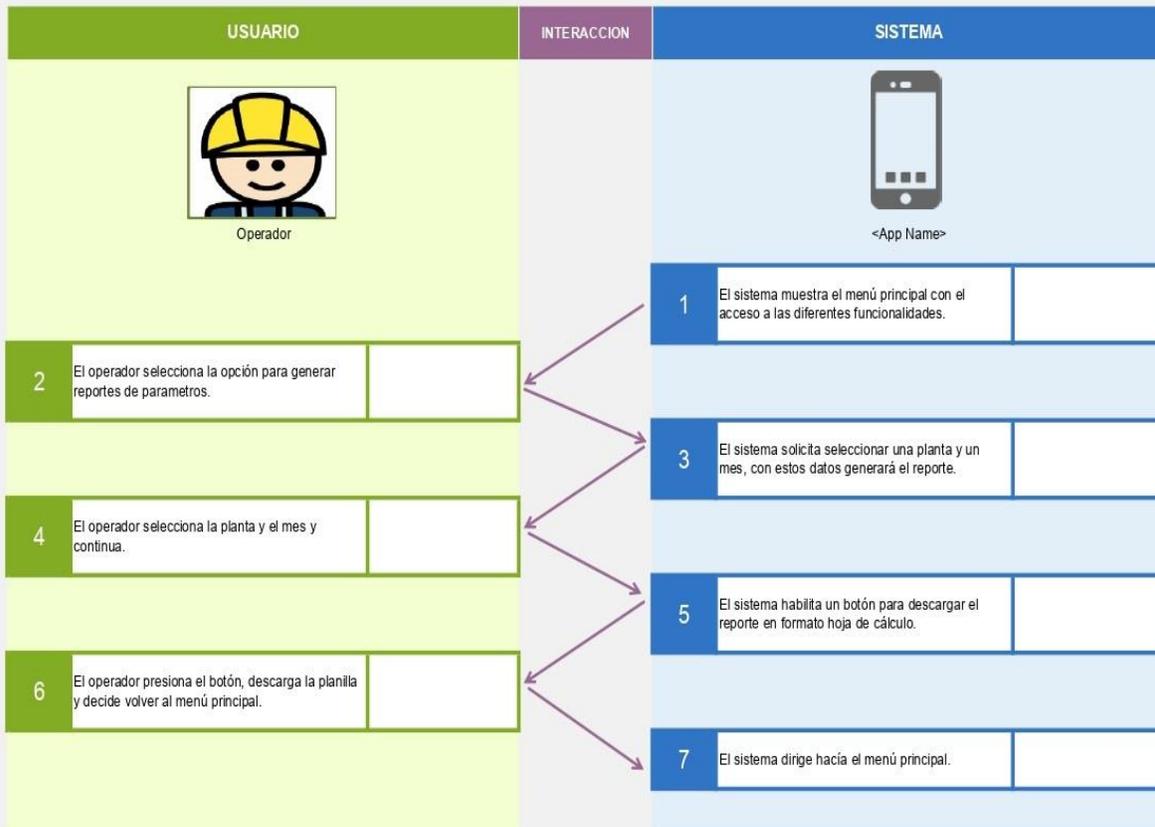


Diagrama de Casos de Uso

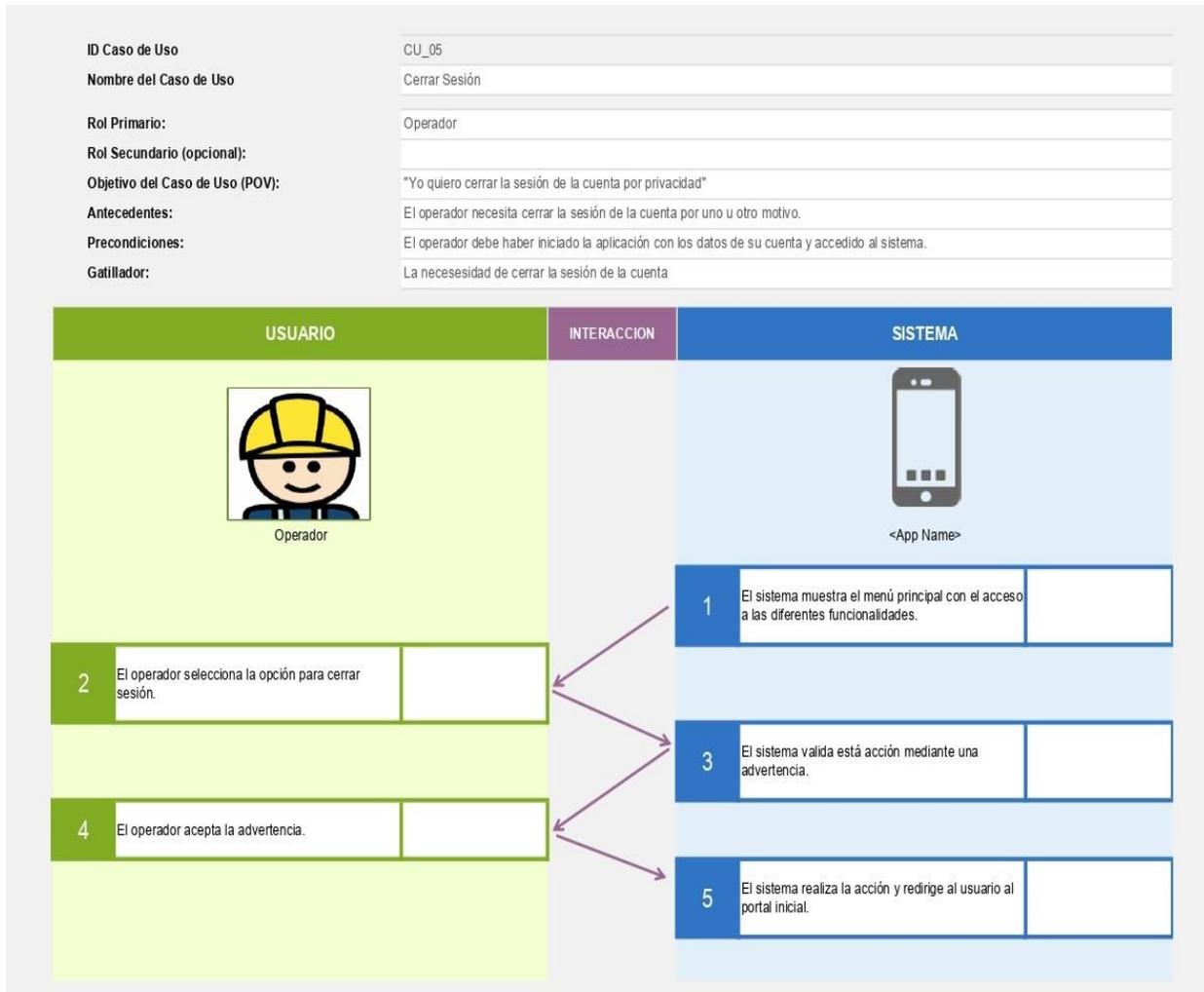
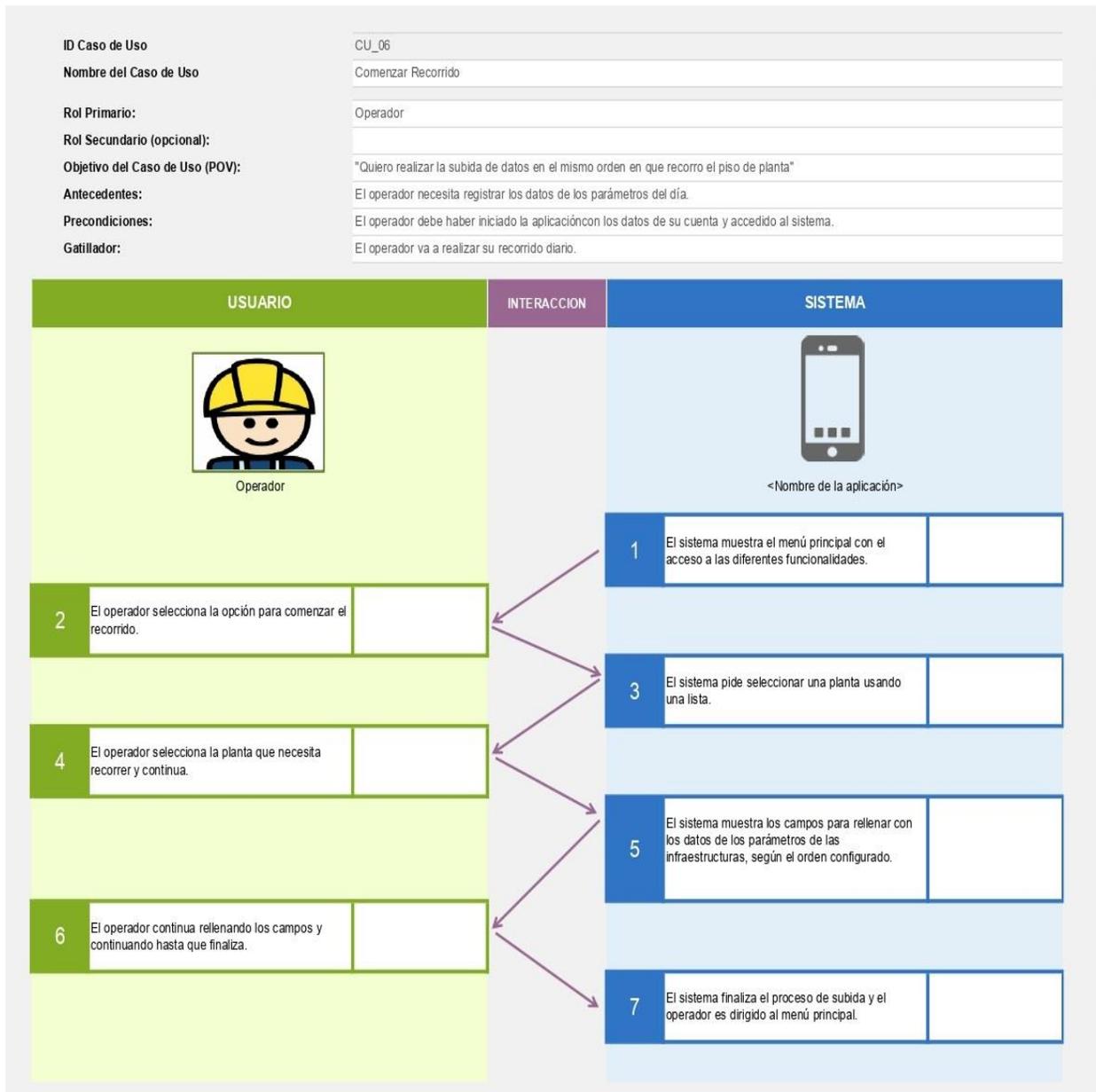
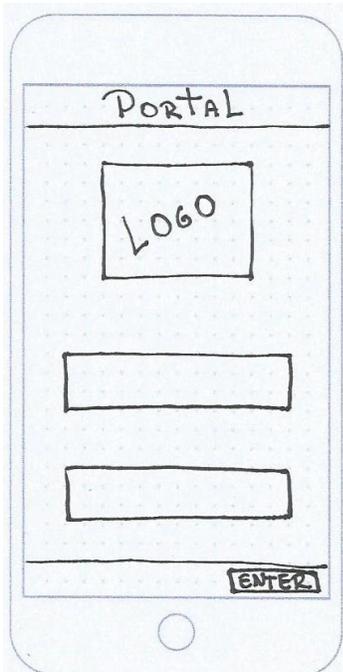


Diagrama de Casos de Uso



7.8 Anexo 8 – Bocetos

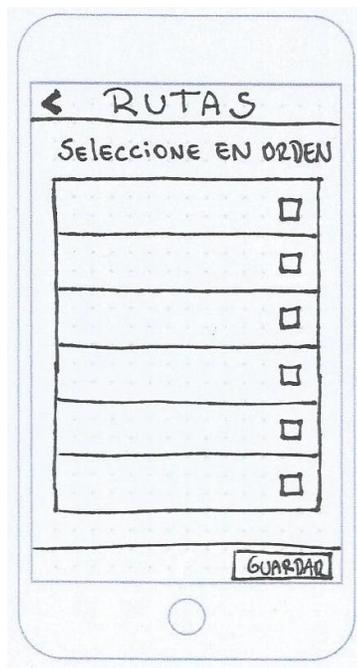
Portal



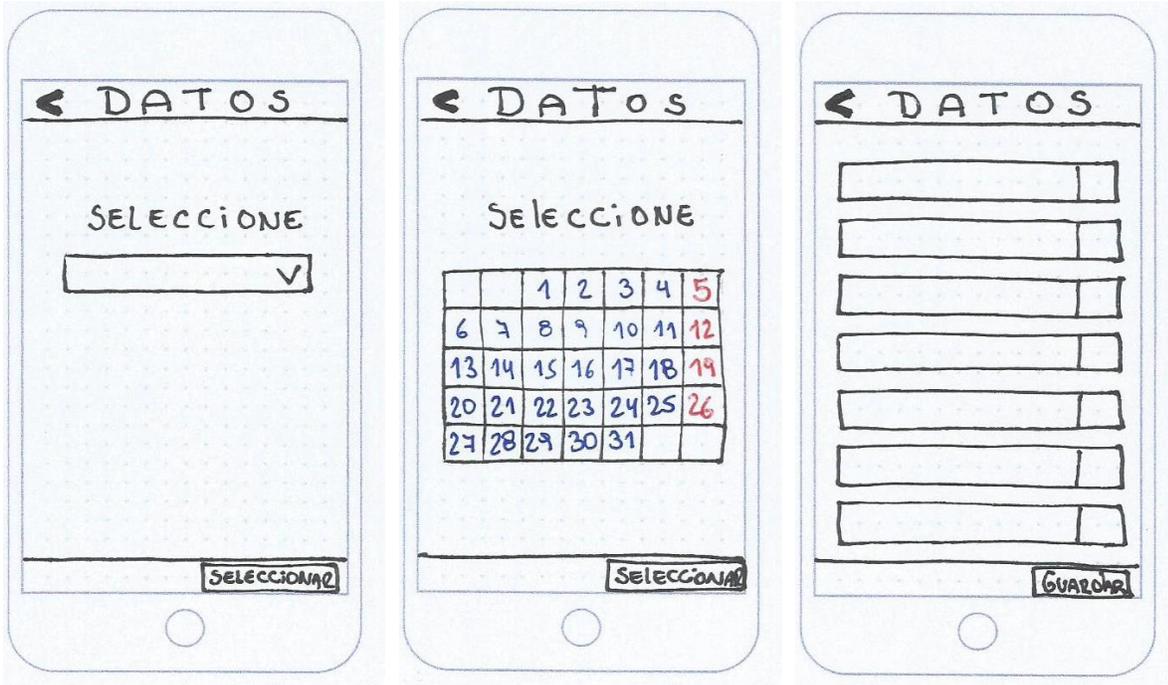
Menú Principal



Rutas



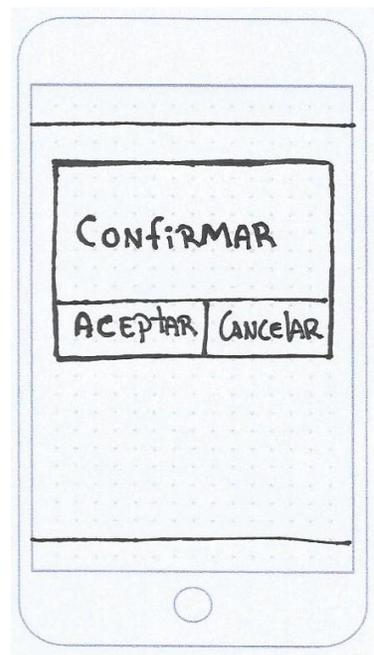
Datos



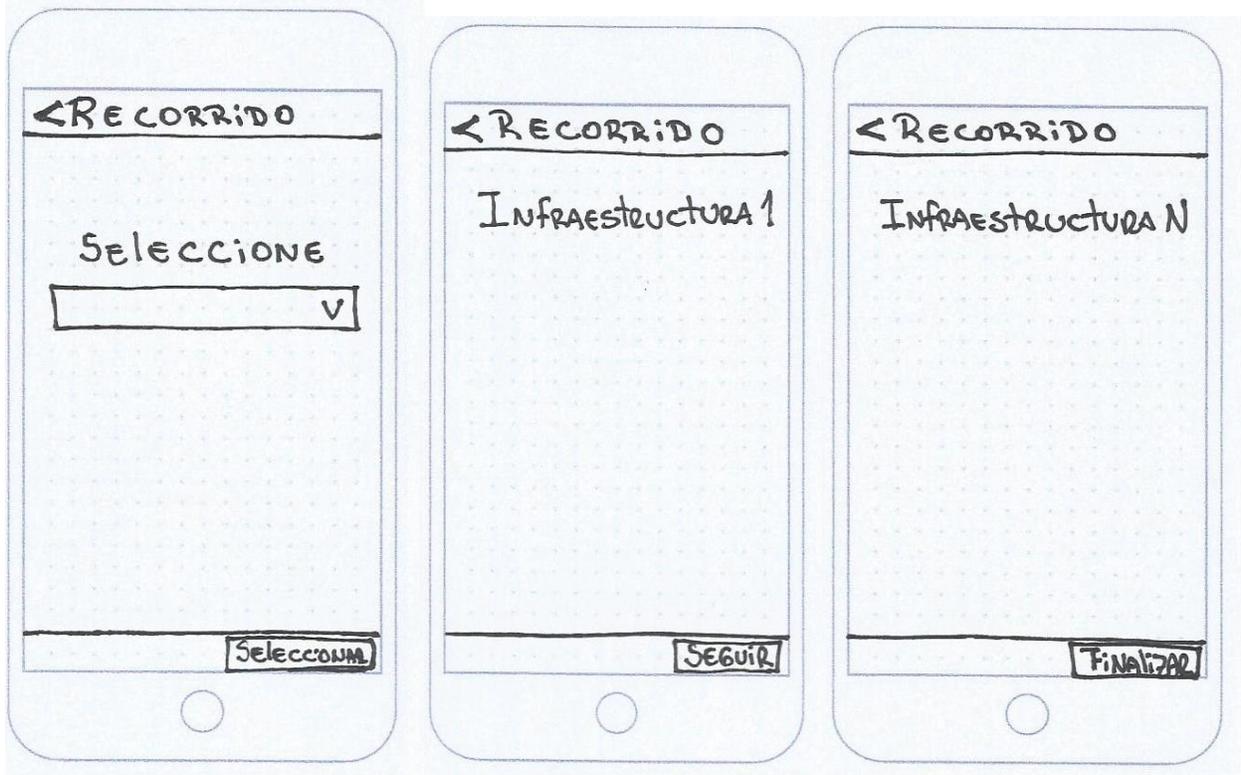
Reportes



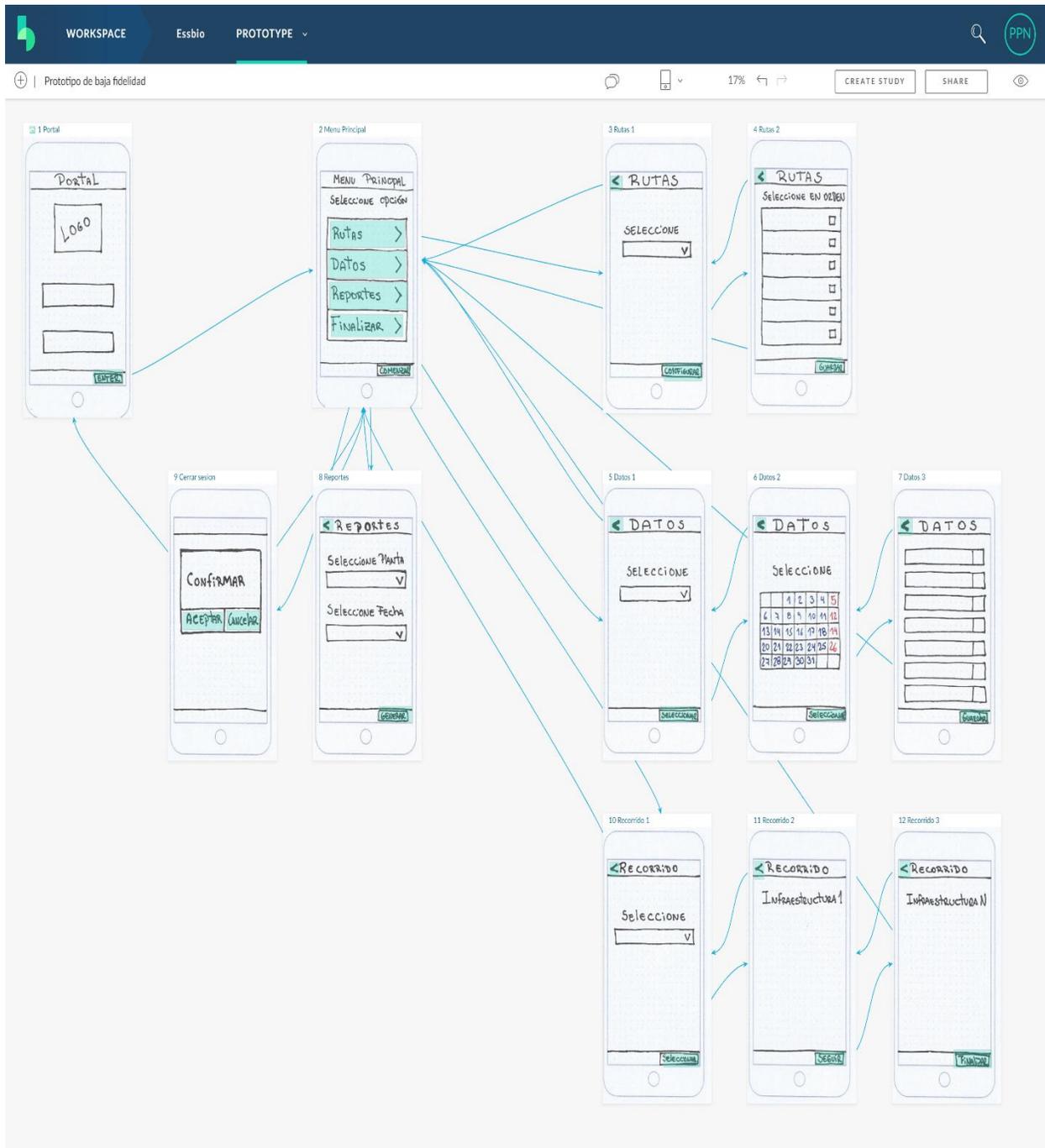
Cerrar Sesión



Recorrido



7.9 Anexo 9 – Mapa de navegación del prototipo de baja fidelidad



7.10 Anexo 10 – Capturas

Portal

Portal

Nombre de Usuario

Contraseña

Entrar

The screenshot shows a mobile application interface for a portal. At the top, the title 'Portal' is centered. Below the title is a large, stylized blue starburst logo. Underneath the logo are two input fields: the first is labeled 'Nombre de Usuario' and the second is labeled 'Contraseña'. At the bottom right corner, there is a blue button with the text 'Entrar'.

Menú-Principal

Menú Principal

Seleccione opción

Rutas

Datos

Reportes

Finalizar

Comenzar

The screenshot shows a mobile application interface for a main menu. At the top, the title 'Menú Principal' is centered. Below the title is a grey bar with the text 'Seleccione opción'. Underneath this bar are four menu items: 'Rutas', 'Datos', 'Reportes', and 'Finalizar'. At the bottom right corner, there is a blue button with the text 'Comenzar'.

Rutas

Rutas

Seleccionar Planta:

Configurar

The screenshot shows a mobile application interface for the 'Rutas' screen. At the top, the title 'Rutas' is centered with a back arrow on the left. Below the title is a grey bar with the text 'Seleccionar Planta:'. Underneath this bar is a dropdown menu. At the bottom right corner, there is a blue button with the text 'Configurar'.

Rutas

Seleccione en Orden

Infraestructura 1

Infraestructura 2

Infraestructura 3

Infraestructura N

Guardar

The screenshot shows a mobile application interface for the 'Rutas' screen. At the top, the title 'Rutas' is centered with a back arrow on the left. Below the title is a grey bar with the text 'Seleccione en Orden'. Underneath this bar are four menu items: 'Infraestructura 1', 'Infraestructura 2', 'Infraestructura 3', and 'Infraestructura N'. At the bottom right corner, there is a blue button with the text 'Guardar'.

Datos

< Datos

Seleccionar Planta:

[Dropdown menu]

Seleccionar

< Datos

Seleccionar Fecha:

< agosto 2021 >

	dom.	lun.	mar.	mié.	jue.	vie.	sáb.
30	1	2	3	4	5	6	7
31	8	9	10	11	12	13	14
32	15	16	17	18	19	20	21
33	22	23	24	25	26	27	28
34	29	30	31	1	2	3	4

Seleccionar

< Datos

Infraestructura 1

Parámetro 1:
[Dato Registrado]

Parámetro 2:
[Dato Registrado]

Parámetro 3:
[Dato Registrado]

Observacion:
[Dropdown menu]

Seguir

< Datos

Infraestructura N

Parámetro 1:
[Dato Registrado]

Parámetro 2:
[Dato Registrado]

Parámetro 3:
[Dato Registrado]

Observacion:
[Dropdown menu]

Guardar

Reportes

Reportes

Seleccionar Planta:

Seleccionar Mes:

Generar

Cerrar-Sesión

Menú Principal

Seleccione opción

Rutas

Finalizar

Comenzar

Cerrar Sesión

¿Está seguro que desea confirmar esta acción?

Salir Cancelar

Recorrido

Recorrido

Seleccionar Planta:

Seleccionar

Recorrido

Infraestructura 1

Parámetro 1:

Parámetro 2:

Parámetro 3:

Observacion:

Seguir

Recorrido

Infraestructura N

Parámetro 1:

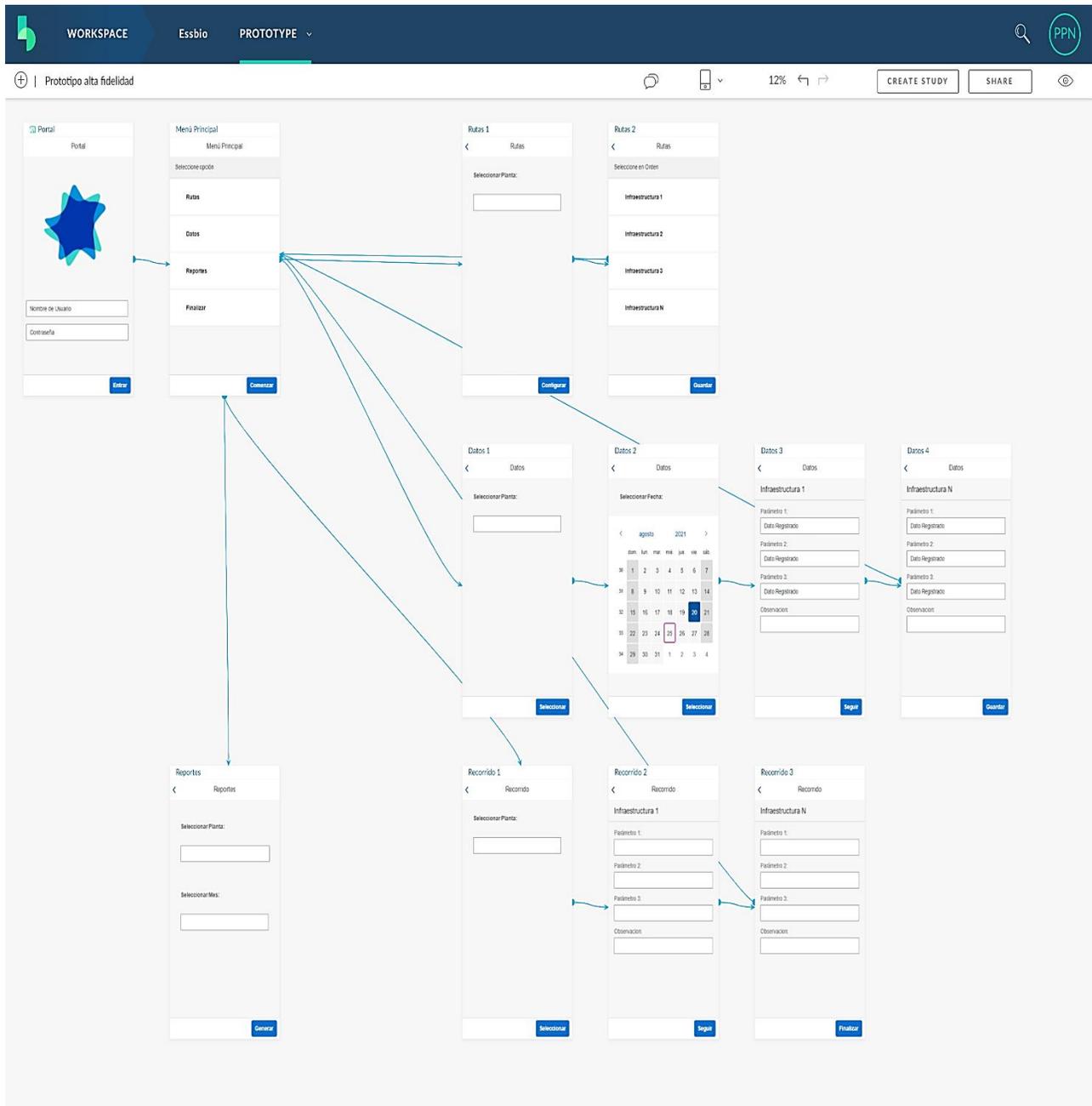
Parámetro 2:

Parámetro 3:

Observacion:

Finalizar

7.11 Anexo 11 – Mapa de navegación del prototipo de alta fidelidad



7.12 Anexo 12 – Capturas prototipo de alta fidelidad en vivo

Portal

Menú Principal

Seleccione opción

Rutas

Datos

Reportes

Finalizar

Nombre de Usuario

Contraseña

Entrar

Comenzar

< Rutas

Seleccionar Planta:

CABRERO AS

CABRERO AP

Configurar

< Rutas

Seleccione en Orden

Infraestructura 1

Infraestructura 2

Infraestructura 3

Infraestructura N

Guardar

< Datos

Seleccionar Planta:

CABRERO AS

CABRERO AP

Seleccionar

< Datos

Seleccionar Fecha:

< agosto 2021 >

	dom.	lun.	mar.	mié.	jue.	vie.	sáb.
30	1	2	3	4	5	6	7
31	8	9	10	11	12	13	14
32	15	16	17	18	19	20	21
33	22	23	24	25	26	27	28
34	29	30	31	1	2	3	4

Seleccionar

< Datos

Infraestructura 1

Parámetro 1:

Parámetro 2:

Parámetro 3:

Observacion:

Seguir

< Datos

Infraestructura N

Parámetro 1:

Parámetro 2:

Parámetro 3:

Observacion:

Guardar

< Reportes

Seleccionar Planta:

CABRERO AP

Seleccionar Mes:

MAYO

Generar

< Reportes

Seleccionar Planta:

CABRERO AP

Reportes

¿Esta seguro que desea descargar este reporte?

Descargar Cancelar

Generar

Menú Principal

Seleccione opción

Rutas

Datos

⊞ Cerar Sesión

¿Está seguro que desea confirmar esta acción?

Salir Cancelar

Finalizar

Comenzar

7.13 Anexo 13 – Confirmación de Participación en el curso en OpenSap



Confirmation of Participation



Patricio Pardo Novoa

Date of birth: April 4, 1993

E-mail: pardonovoapatricio@gmail.com

has participated in the following openSAP course:

Design Your First App with Build

Instructors: Ben Ceaser, Jaideep Godara and Gaby Lui

This four-week online course was held from October 4 through November 9, 2017. It comprised 2-4 hours of learning effort per week.

The course covered the following topics:

- Design Thinking: Starting with Empathy
- Point of View and Idea Generation
- Build: Prototype and Feedback
- Taking Your Vision into the World

Walldorf, Germany, November 2017

openSAP
open.sap.com

Fecha de descarga: 25-08-2021.

Este curso fue realizado entre los meses de marzo y abril de 2021.