



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN

**Gestión de fondos para actividades estudiantiles
conducidas por CCAA o la Fedeut Curicó en
acuerdo a la RU N°2083 de 2017**

YORCH SEPÚLVEDA MANRÍQUEZ

Profesor Guía: RODRIGO PAREDES MORALEDA

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Julio, 2019



**UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Gestión de fondos para actividades estudiantiles
conducidas por CCAA o la Fedeut Curicó en
acuerdo a la RU N°2083 de 2017**

YORCH SEPÚLVEDA MANRÍQUEZ

Profesor Guía: RODRIGO PAREDES MORALEDA

Profesor Informante: RENZO ANGLES ROJAS

Profesor Informante: RUTH GARRIDO ORREGO

Memoria para optar al título de
Ingeniero Civil en Computación

El presente documento fue calificado con nota: _____

Curicó – Chile

Julio, 2019

Dedicado a Patricia Manríquez Lisboa y Sergio Sepúlveda Leal, mis padres.

AGRADECIMIENTOS

En primer lugar, quiero agradecer a Rodrigo Paredes por aceptarme como su estudiante memorista, el cual me entrego todo su apoyo y disposición en la culminación de este gran proceso que he realizado en mi vida. En todo momento me oriento, aconsejo y ayudo para que los resultados obtenidos fueran de calidad.

A Ruth Garrido, por aconsejarme y querer la mejor opción para salir adelante en este gran desafío de mi vida. Por su preocupación y apoyo en grandes dificultades que viví dentro de la carrera.

A Marcela Pacheco, por su gran labor en apoyarme con las gestiones académicas que muchas veces son demasiado complicadas. Por esas gratas conversaciones que siempre nos ayudaban salir de todo el estrés y nos alegraba la tarde.

Agradezco a estas dos grandes mujeres que me apoyaron cuando fui dirigente estudiantil, entregando todo su esfuerzo, disposición, pasión y organización para las distintas actividades que realizábamos como Centro de Alumnos fueran las mejores y además disfrutadas por todos los estudiantes de la carrera.

A Cristina Reyes que ha sido un pilar fundamental en todo instante, apoyándome, alentándome a seguir luchando por cumplir mis sueños, sobre todo cuando estaba todo cuesta arriba y desorientado. Gracias por su carisma, su entusiasmo y por los abrazos que me daba cuando más lo necesitaba, compartiendo también mis logros y alegrías. Muchas gracias por todo.

A los profesores José Letelier, Rodolfo Allende, Ruth Garrido, Federico Meza, Luis Silvestre y Yamisleydi Salgueiro por confiar en mis capacidades y darme la oportunidad de cumplir uno de mis grandes logros de ser ayudante y poder entregar mis conocimientos a los alumnos.

A Fernanda Arenas que, si bien al inicio de la carrera nuestro compañerismo no fue del todo bueno, con el tiempo logramos ser los mejores amigos. Logré salir adelante arriesgándolo todo por como lo fue en Física, que lo vivimos al extremo y un sin fin de módulos que tuvimos en conjunto a pesar de las diferentes carreras que estudiamos. De verdad que fueron aventuras extremas que, al recordarlas, son grandes anécdotas por las cuales nos reímos y disfrutamos.

A Daniela Paredes, quien conocí a mitad de carrera y de ahí no nos separamos más. Al igual que con Fernanda, vivimos al extremo y estudiando hasta las últimas

etapas a pesar de todas las trabas que se nos ponían en el camino. También por el gran apoyo que me brindó cuando fuimos dirigentes en el Centro de Alumnos y por lograr nuestra meta que era unir a la carrera.

Gracias al apoyo, compañerismo, amistad, lealtad y trabajo en equipo de estas dos grandes profesionales, he logrado salir adelante de muchas batallas que daba por pedido, ustedes siempre estuvieron ahí cuando más lo necesitaba.

A Patricio Castro que es ha sido un gran amigo y compañero en todo mi proceso universitario y esos largos viajes que realizábamos para llegar a la universidad y también para volver a nuestros respectivos hogares. También por todo el apoyo y animo que me brindaba.

A mi hermano Gregory Sepúlveda por aguantarme cuando estudiaba y soñaba con la materia, mis momentos de estrés pero también por compartir mis logros y alegrías.

Finalmente, a mis Padres, Patricia Manríquez y Sergio Sepúlveda, que han sido un pilar fundamental tanto en mi formación personal como profesional, apoyándome en cada etapa de mi vida, sacándose la mugre para ayudarme a cumplir mis sueños. Ellos han sido mi ejemplo a seguir y creer en que todo es posible con paciencia, constancia, optimismo y esfuerzo.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	IV
Índice de Figuras	VII
Índice de Tablas	IX
Resumen	x
 1. Introducción	 11
1.1. Contexto del proyecto	11
1.2. Definición del problema	14
1.3. Presentación de la solución	14
1.4. Objetivos	15
1.4.1. Objetivo general	15
1.4.2. Objetivos específicos	15
1.5. Alcances	16
1.6. Limitaciones	16
1.7. Proyectos similares	16
1.8. Descripción de contenidos	17
 2. Antecedentes	 19
2.1. Conceptos	19
2.2. Tecnologías utilizadas	21
2.2.1. Frameworks	22
2.2.2. Bibliotecas	22
2.3. Metodología de desarrollo de software	23
2.4. Metodología ágil	25
2.4.1. Scrum	27

2.4.2.	Roles	27
2.4.3.	Aptitudes requeridas	28
2.4.4.	Artefactos	29
2.4.5.	Reuniones	30
2.4.6.	Ciclo de vida	32
2.4.7.	Documentación	35
2.5.	Validación de la aplicación	35
2.6.	Herramientas de planificación	36
3.	Metodología utilizada	37
3.1.	Estudio del entorno	37
3.1.1.	Las personas	37
3.1.2.	La aplicación	38
3.1.3.	Las herramientas	38
3.2.	La metodología	41
3.2.1.	Roles	41
3.2.2.	El proceso	42
3.2.3.	Cómo se trabajó	45
4.	Características del sistema	48
4.1.	Aspectos generales	48
4.2.	Modelos de contexto de solicitudes	49
4.3.	Características generales del software	54
4.4.	Errores frecuentes de la Gestión Manual de Fondos	57
4.5.	Planificación de las iteraciones	59
5.	Diseño de la aplicación	61
5.1.	Mapa de navegación	61
5.2.	Interfaces de usuario	62
5.3.	Arquitectura de la aplicación	63
5.3.1.	Arquitectura física	64
5.3.2.	Arquitectura lógica	64
5.4.	Modelo de datos	67
5.5.	Modelo Relacional	69
5.6.	Algoritmo de solución	70

6. Construcción y validación	74
6.1. Herramientas Utilizadas	74
6.2. Aplicación Obtenida	75
6.3. Pruebas	83
6.4. Resultados	84
7. Conclusiones	86
Glosario	88
Bibliografía	91
Anexos	
A: Pruebas de Caja Negra	94
A.1. Confección de una solicitud	94
A.2. Ingreso de RU	96
A.3. Confección de una rendición	97
B: Encuesta a representantes de las OE	99
B.1. Sección de preguntas generales sobre el sistema	99
B.2. Sección de preguntas sobre las funcionalidades del sistema	102
B.3. Sección de preguntas sobre usabilidad e interfaz de usuario (En escala de 1 (En desacuerdo) a 5 (De acuerdo))	103

ÍNDICE DE FIGURAS

	página
2.1. Metodología tradicional vs metodología ágil [21].	24
2.2. Reuniones en el Proceso de Scrum.	30
2.3. Proceso de metodología Scrum.	33
2.4. Proceso utilizado en el proyecto.	34
3.1. Sprint Backlog.	39
3.2. Proceso de trabajo de un Sprint.	40
4.1. Proceso de Fondos por Rendir por parte de Fedeut.	50
4.2. Proceso fondos por rendir por parte de un CAA.	51
4.3. Diagrama de estado del proceso que debe simular la aplicación.	53
4.4. Proceso fondos con Solicirud de Reembolso por parte de un CAA.	57
4.5. Proceso fondos con Solicitud de Reembolso por parte de Federación.	58
5.1. Mapa de navegación de la aplicación.	62
5.2. Línea de diseño de la aplicación que representa el estándar de los elementos de interfaz.	63
5.3. Arquitectura física del sistema.	64
5.4. Arquitectura lógica del sistema.	65
5.5. Diagrama de clases que detalla el Modelo de Datos.	67
5.6. Modelo Relacional de la Base de Datos del sistema.	69
6.1. Vista de ingreso al sistema.	76
6.2. Vista de recuperar clave.	76
6.3. Vista de búsqueda de solicitudes.	77
6.4. Vista que obtiene los datos principales de una Solicitud.	77
6.5. Vista que obtiene las categorías en que incurren los gastos.	78
6.6. Vista que obtiene los datos de los participantes del evento.	78
6.7. Vista que muestra el resumen de los datos ingresados para la solicitud.	79
6.8. Vista que muestra el resumen de los datos ingresados para la solicitud y obtiene los datos principales de la RU.	79
6.9. Vista que muestra los datos principales de una Rendición.	80

6.10. Vista que muestra el resumen de los documentos declarados por cada participante y los gastos comunes.	81
6.11. Vista que muestra en detalle los documentos registrados que ha presentado un participante cuya actividad ingresada en la solicitud es de tipo grupal.	81
6.12. Vista que obtiene los datos de una boleta o factura.	82
6.13. Tras finalizar la Rendición, se habilitan las opciones de Aceptar o Rechazar, de acuerdo a lo que responda el Depto. de Tesorería y presupuesto.	82
 B.1. Resultado de la pregunta uno.	 99
B.2. Resultado de la pregunta dos.	100
B.3. Resultado de la pregunta tres.	100
B.4. Resultado de la pregunta cuatro.	101
B.5. Resultado de la pregunta cinco.	102
B.6. Resultado de la pregunta seis.	102
B.7. Resultado de la pregunta siete.	103
B.8. Resultado de la pregunta ocho.	103
B.9. Resultado de la pregunta nueve.	104
B.10. Resultado de la pregunta diez.	104

ÍNDICE DE TABLAS

	página
2.1. Metodología Tradicional vs Metodología Ágil [21].	25
4.1. Diferencia en el proceso de Fondos por Rendir entre Fedeut y CCAA.	52
4.2. Diferencia en el proceso de Fondos por Rendir entre Fedeut y CCAA.	54
4.3. Historias de usuario - Secretario de Finanzas	55
4.4. Historias de usuario - Presidente	56
4.5. Errores frecuentes del proceso de Solicitud de Fondos	59
4.6. Planificación de las iteraciones.	60
6.1. Solución a problemas frecuentes	83
A.1. Caja Negra - Registro de datos principales una Solicitud.	94
A.2. Caja Negra - Registro de categorías para una solicitud.	95
A.3. Caja Negra - Registro de participantes para una solicitud.	95
A.4. Caja Negra - Finalizar solicitud.	95
A.5. Caja Negra - Aceptación de solicitud e ingreso de RU.	96
A.6. Caja Negra - Confeccionar una Rendición.	97
A.7. Caja Negra - Registro de un Documento.	98
A.8. Caja Negra - Obtener documento de Rendición.	98
A.9. Caja Negra - Obtener documento de listado de Documentos.	98

RESUMEN

Las Organizaciones Estudiantiles (OEs) de la Universidad de Talca se encargan de gestionar muchas actividades tales como la bienvenida de los alumnos de primer año o el aniversario de carrera, entre otras. Dado lo anterior, la casa de Estudio en su plan estratégico *Visión 2020* destaca el apoyo a estas actividades aportando monetariamente para su gestión. Pero al momento de dar cuentas de los gastos del evento, si es rechazada por Tesorería a la hora de ser revisado, esta no envía una notificación con todos los errores, sino que usualmente notifica sólo el primer error detectado provocando el retraso tanto del reembolso del dinero como de otras actividades planificadas por la OE. Por consecuencia, no se pueden realizar nuevas Solicitudes de Fondos por parte de la OE mientras no culmine la Rendición pendiente.

En este proyecto se desarrolla una plataforma web para dar solución al problema de las rendiciones, utilizando Scrum como metodología de desarrollo de software.

Como primer paso, se realizan encuestas a los clientes con el fin de obtener las características del sistema a desarrollar y el modelo de contexto del mecanismo actual de rendiciones. Luego, estas características son categorizadas de acuerdo con la utilidad que presta (planificación de un *Sprint*). Además, se realiza el diseño de la aplicación la cual comprende la confección del mapa de navegación, el diseño de interfaces, la arquitectura, el modelo relacional del sistema y el algoritmo de solución.

Luego de desarrollar la aplicación se realizan pruebas de caja negra para contrastar que se cumpla con los objetivos, y pruebas de usabilidad que se realizan a los clientes en donde 12 de 16 de ellos expresa que es una herramienta necesaria para asistir en el proceso de Solicitud de Fondo. Por otra parte, 12 de 16 encuestados dicen que la aplicación ayudaría a reducir la causa de rechazo de una rendición.

Para concluir, se obtiene una aplicación que asiste en el proceso de Solicitud y Rendición de un Fondo la cual cumple con los estándares impuestos por la Casa de Estudio. Como consecuencia de lo anterior, permite reducir los errores que podrían rechazar las Rendiciones.

1. Introducción

Las *Organizaciones Estudiantiles(OEs)* de la Universidad de Talca son las encargadas de gestionar muchas actividades para la comunidad estudiantil, tales como la bienvenida de alumnos nuevos, aniversario de carrera, o actividades culturales y deportivas, entre otras. Para ello, dada una actividad, se debe realizar un conjunto de procedimientos tales como la aprobación una de *Solicitud*, envío de su *Resolución Universitaria* y su *Rendiciones de Cuentas*, entre otros. Uno de los procesos más engorrosos es la rendición de cuentas, que es justamente el foco de esta propuesta. En efecto, dado lo engorroso de este proceso, en la práctica se suelen cometer muchos errores pues, por falta de experiencia o por el apuro por cumplir los plazos, es posible incumplir con los requerimientos establecidos por la Universidad a la hora de realizar el detalle y acreditación de los gastos en las actividades realizadas por las OEs. Por lo tanto, en las siguientes páginas se muestra la propuesta de solución para el problema de efectuar correctamente las rendiciones de cuentas, en donde se puede observar los *Conceptos Básicos del Proyecto*, el *Contexto de trabajo* de las OEs, la *Descripción del Problema* y *Trabajos Relacionados*, entre otros.

1.1. Contexto del proyecto

El “Plan estratégico de la Universidad de Talca Visión 2020” destaca el apoyo a todo tipo de actividades emprendidas por las distintas Organizaciones Estudiantiles (OE) reconocidas. Entre las actividades destacan: recepción de alumnos nuevos, celebración del día de la carrera, o actividades sociales, culturales y deportivas; entre otras [19].

Para la realización de estas actividades se debe seguir un procedimiento, el cual

está estipulado en la RU N°2083 promulgada el 12 de diciembre del 2017, la cual dice:

A. Caso Federación y Organizaciones Estudiantiles:

Sólo el Presidente, Tesorero o Secretario de finanzas de la Federación (Fedeut) respectiva o uno de Grupos Intermedios respectivo pueden solicitar por escrito Fondos para realizar actividades. Esta solicitud se envía a quien dirige la Dirección de Apoyo a Actividades Estudiantiles (DAAE), de la Vicerrectoría de Desarrollo Estudiantil (VDE), al menos 20 días antes del inicio de la actividad. Quien dirige la DAAE, una vez que verifica la solicitud enviada por Fedeut o Grupo Intermedio y una vez aprobada, realiza y envía otra solicitud a quien dirige la VDE.

Luego de que quien dirige la VDE verifica y aprueba la solicitud de Fondo, emite una Resolución Exenta en original y cuatro copias las cuales son enviadas a Contraloría de la Universidad de Talca para su respectivo control de legalidad.

Una vez tramitado el acto administrativo, se debe distribuir los documentos anteriormente mencionados de la siguiente forma:

- La Secretaría General archiva la documentación original.
- La primera copia es enviada al Departamento de Tesorería y Presupuesto.
- La segunda para el archivo de la Vicerrectoría Estudiantil.
- La tercera es enviada a la Federación o Grupo Intermedio interesada.
- La cuarta es enviada a la Unidad de Gobierno Transparente.

B. Caso de Centros de Alumnos:

Sólo el Presidente, Tesorero o Secretario de Finanzas del Centro de Alumnos (CAA) respectivo pueden solicitar por escrito Fondos para realizar actividades. Esta solicitud se envía a quien dirige la Dirección de Escuela correspondiente al menos 20 días antes del inicio de la actividad.

El(la) Director(a) de Escuela, una vez que verifica la solicitud enviada por el CAA, eleva la solicitud a quien dirige la Decanatura de la Facultad o a quien dirige la Vicerrectoría de Desarrollo Estudiantil (para el caso de las Escuelas no adscritas a una Facultad).

Luego de que quien dirige la Decanatura de la Facultad o Vicerrectoría de Desarrollo Estudiantil según corresponda, verifique y apruebe la solicitud de Fondo enviada por quien dirige la Dirección de Escuela, emite una Resolución Exenta, en original y cuatro copias, las cuales son enviadas a controlaría universitaria para el respectivo control de legalidad.

Una vez aprobada por Contraloría Interna y totalmente tramitada, es enviada a quien dirige Decanatura o VDE para su distribución de la siguiente forma:

- La documentación original la archiva Decanato o VDE.
- La primera copia es enviada al Departamento de Tesorería y Presupuesto.
- La segunda para el archivo de la Facultad.
- La tercera es enviada al CAA interesado.
- La cuarta es enviada a la Unidad de Gobierno Transparente.

En ambos casos, el Departamento de Presupuesto hace responsable de la actividad presupuestaria correspondiente a la OE (ya sea Fedeut, CAA o Grupo Intermedio) que la solicita.

El Departamento de Tesorería emite el pago a nombre del Presidente, Tesorero o Secretario de Finanzas de la OE que inició el proceso y adjunta la segunda copia de la Resolución al Comprobante de Egreso para la posterior rendición del Fondo.

Al representante de la OE se le asigna las siguientes responsabilidades:

- Supervisar o realizar el cobro del documento de pago.
- Mantener un registro detallado de los gastos establecidos en el presupuesto de la actividad aprobada.
- Ejecutar/Realizar los gastos autorizados para la actividad.
- Rendir el Fondo asignado para la actividad.
- Respalda gastos con la documentación idónea, tal como establece la Resolución Universitaria N°522 de 1992, punto N°12.

1.2. Definición del problema

Tras lo expuesto en la sección anterior, las dificultades surgen en la manera de realizar la rendición, debido a que cuando ésta es creada por el solicitante, no existe ningún método que valide su correcta realización, sino que una vez enviada y revisada por contraloría se obtiene una respuesta. Si esta última es negativa, la entrega del dinero por anticipado o su reembolso posterior, según sea el caso, tarda más de lo planificado.

Por otro lado, el departamento de contraloría se encuentra en Talca y se debe enviar la documentación por valija desde Curicó hasta aquel lugar. Por lo tanto, se pierde tiempo en enviar la documentación y en esperar a que controlaría lo revise. Es más, en la práctica no siempre envían un reporte completo de los errores que tiene la rendición. En efecto, en algunos casos notifican los errores individualmente lo que enlentece mucho el proceso.

Finalmente, cabe destacar que la Federación o CCAA a través de su Presidente o Secretario de Finanzas ¹ pueden realizar una solicitud de fondo, por lo que si se quiere realizar otra actividad y solicitar fondos a la universidad, se debe esperar a que no existan rendiciones pendientes, lo que podría redundar en la no realización de algún evento posterior por meras razones administrativas.

Es por esto que en este trabajo de título se va a abordar el problema, sistematizándolo para aminorar la dificultad del proceso de solicitud de fondos en acuerdo a la RU N°2083 del año 2017. Adicionalmente, también se va abordar la confección de la solicitud de fondo por rendir ya que este proceso está altamente ligado a la rendición del fondo.

1.3. Presentación de la solución

En base al problema descrito anteriormente es que nace este proyecto el cual consiste en implementar un sistema web que permita a las distintas O.E. la confección de solicitudes de fondos por rendir y su posterior rendición, de acuerdo a lo estipulado por la RU N°2083 del año 2017. Cabe señalar que esta RU señala los posibles errores por los cuales una rendición pueda ser rechazada, como por ejemplo, montos superiores a lo solicitado o boletas fuera del plazo en que se realiza la actividad, entre

¹Algunas veces este cargo se llama Tesorero

otros.

Dentro de las ventajas de contar con esta herramienta se tienen las siguientes:

- Los usuarios tienen un respaldo de las solicitudes y resoluciones realizadas.
- Los usuarios tienen un instructivo de cómo realizar una rendición según los estándares que estipula la Universidad.
- El sistema busca el óptimo monto conformado por la suma de boletas ingresadas por el usuario, de tal manera que sea igual o lo más cercana al monto solicitado.
- El sistema permite guardar una rendición incompleta, a la cual se le asigna un estado de pendiente.
- El sistema verifica que la rendición se realice dentro del periodo estipulado por la RU N°2083 del año 2017, la cual dice que son 20 días corridos, contados desde el último acto administrativo de la transferencia de recursos.

1.4. Objetivos

Como este sistema pretende aliviar todos los inconvenientes *de forma* que se puedan cometer tanto al confeccionar la solicitud de fondo por rendir como en su posterior rendición, se establecen los siguientes objetivos:

1.4.1. Objetivo general

- Confeccionar solicitudes de fondos por rendir para actividades estudiantiles y las rendiciones de gastos respectivas, de acuerdo con el marco definido por la RU N°2083 del año 2017.

1.4.2. Objetivos específicos

Para satisfacer el objetivo general, se definen los siguientes objetivos específicos:

- Modelar el proceso de confección de solicitudes de fondos por rendir.
- Modelar el proceso de rendición de gastos.

- Diseñar e implementar una aplicación web que asista al proceso de solicitud de fondos por rendir y su posterior rendición de gastos.

1.5. Alcances

La aplicación determinada, *SimRend*, es un sistema web de fondos por rendir para Federación y CAA de la Facultad de Ingeniería de la Universidad de Talca, en la cual sólo se pueden realizar las siguientes funciones:

- Confeccionar solicitudes de fondos por rendir que detallen en qué se utiliza el dinero, participantes de la actividad, periodo en la que se realiza y su responsable.
- Generar un PDF de solicitud de fondos por rendir.
- Confeccionar la rendición de fondos por rendir que detallen los gastos incurridos de una actividad tras ser aprobada su correspondiente solicitud.
- Gestionar los comprobantes de gastos (boletas o facturas) incurridas en un evento. Esto comprende la validación de fechas, listado de los comprobantes y selección del conjunto de comprobantes que mejor se ajusta al presupuesto expuesto en la solicitud de fondo enviada por la OE a quien dirige la Dirección de Escuela o la DAAE.

1.6. Limitaciones

Dentro de las limitaciones del sistema, se tienen las siguientes:

- El sistema opera bajo la RU N°2083 del año 2017.
- Esta aplicación se excluyen los Grupos Intermedios.

1.7. Proyectos similares

Actualmente existe un formato estándar hecho en Microsoft Excel, el cual se encuentra en la página de Contraloría de la Universidad de Talca [4], pero cada vez que hay cambios de integrantes en las directivas de las OE se debe realizar una

capacitación para que los encargados puedan realizar las rendiciones. Aun así, la capacitación es vaga y las OE deben aprender bajo el proceso.

Una aplicación que realiza procesos similares es el “Sistema de gestión de declaración de gastos en línea (SDGL) del PROGRAMA FONDECYT” [8].

Otras aplicaciones similares en relación con la funcionalidad de “visualizar el estado de una rendición” son aplicaciones de compras online, en donde se pueden visualizar el estado de envío del producto. Algunas de ellas son Wish [17], Correos Chile [5], AliExpress [2] y Amazon [3].

1.8. Descripción de contenidos

El resto de este documento tiene la siguiente organización:

- El capítulo dos hace referencia a los antecedentes necesarios para la comprensión del contexto en que se desarrolla el proyecto.
- En cuanto al capítulo tres, se hace mención de la metodología que se utiliza, además de la forma en que se organizan las iteraciones y el orden en que se desarrollan estas para tener una aproximación del tiempo que conlleva el desarrollo del proyecto.
- En el capítulo cuatro se detalla el modelo de contexto el cual ayuda a la comprensión de la situación actual del sistema y cuál es su enfoque. Tras lo anterior se procede a realizar encuestas para obtener las características del sistema y realizar la planificación general del proyecto.
- El capítulo cinco se enfoca en el diseño de la aplicación. Comienza con el Mapa de Navegación que tiene la aplicación junto con el diseño general de la interfaz. Luego se detalla la arquitectura del sistema, para así diseñar el modelo de datos, modelo relacional de la aplicación y por último el algoritmo de solución para la selección automática de documentación que requiere la rendición.
- El capítulo seis trata sobre la aplicación obtenida tras las iteraciones planificadas en el capítulo cuatro. Además, se menciona las pruebas realizadas y los resultados obtenidos tras el uso de la aplicación por los usuarios.

- Por último, el capítulo siete menciona las conclusiones del proyecto en base al trabajo realizado y los resultados obtenidos. Además, se da a conocer los trabajos a futuros a desarrollar.
- Junto con esto, se incluyen dos anexos que contienen el detalle de las pruebas de las funcionalidades en modalidad de caja negra y los resultados de las pruebas de usabilidad.

2. Antecedentes

Este capítulo hace referencia a los conceptos necesarios para la correcta comprensión del proyecto realizado, tales como el proceso sobre el cual se desenvuelve la aplicación, la metodología de desarrollo de software, en este caso Scrum, y las tecnologías y herramientas a utilizar.

2.1. Conceptos

Para mayor comprensión del *Contexto del problema* esbozado en la **Sección 1.1**, se hace mención a los conceptos claves utilizados en el proceso que rige la aplicación.

Evento Según el “Plan estratégico de la Universidad de Talca Visión 2020”, un evento es una actividad emprendida por las distintas Organizaciones Estudiantiles (OE), tales como recepción de alumnos nuevos, celebración del día de la carrera, actividades culturales y deportivas, y actividades sociales, entre otras [19].

Solicitud Una solicitud es un documento en el cual se pide ayuda económica a quien está a cargo de la Dirección de Escuela (en el caso de CCAA) o a quien dirige la DAAE (en el caso de Federación) para llevar a cabo un evento organizado por la OE, en donde se debe detallar:

- Nombre del evento.
- Fecha de inicio y de término de la actividad.
- Lugar donde se realiza la actividad.

- Monto.
- Categorías en que incurrirán los gastos.
- Datos de los participantes como el nombre y RUT (en caso de que sea una actividad dirigida a un grupo de personas).
- Datos del encargado de la actividad (Presidente o Secretario de Finanzas de la OE).

Resolución Universitaria (RU) Una Resolución puede ser un decreto, una decisión o un fallo que emite una determinada autoridad. Esta puede establecer reglas, voluntades, etc. Si bien, hay una gran variedad de resoluciones, en esta memoria cobra mayor interés las Resoluciones Universitarias en donde se decreta las transferencias de fondos para llevar a cabo actividades por parte de las OE.

En este tipo RU se menciona la suma que se transfiere a la OE, el propósito que se solventará con el dinero, el nombre del evento, la fecha en que ocurre la actividad, el lugar en que se realiza el evento y participantes en específicos en caso de haberlos. Además, se establece a quien serán dirigidos los dineros, en este caso al Presidente o Secretario de Finanzas de la OE.

Cabe notar que cada RU se identifica por su código, correspondiente a la unión del año con su número de documento.

Rendición Una rendición es el documento en el cual se da cuenta detallada de todos los gastos que se realizaron en una actividad o evento realizado por la OE. En el encabezado del documento se debe indicar: (I) número de resolución, (II) unidad (por ejemplo, el nombre de la escuela), (III) nombre Jefe Directo (por ejemplo, Director de Escuela), (IV) responsable del fondo (por ejemplo Presidente o Secretario de Finanzas), (v) RUT del responsable, (VI) total solicitado, (VII) total rendido, (VIII) fono anexo y (IX) email del responsable.

El detalle que debe ir en el documento de rendición para las facturas es: (I) número de documento, (II) fecha, (III) nombre del proveedor, (IV) descripción, (v) monto y (VI) centro de costo.

En cuanto al detalle que debe ir en la sección para las boletas con gastos comunes, se tiene: (I) número de documento, (II) fecha, (III) nombre del proveedor, (IV) descripción y (v) monto.

Mientras que en el detalle que debe ir en la sección para las boletas con gastos individuales se debe indicar: (I) número de documento, (II) fecha, (III) nombre del proveedor, (IV) descripción, (V) monto, (VI) total monto de gastos de todas las Categorías por alumno, (VII) nombre del alumno a reembolsar y (VIII) total a reembolsar por participante.

Además, debe ir un resumen en donde se indique: (I) total de gastos, (II) reintegro por caja y (III) total de rendición.

Otro de los resúmenes que debe existir es el detalle de reembolso por participante que indique: (I) nombre de los participantes, (II) monto total por participante y (III) total de reembolso.

Por último, debe ir la firma del responsable de la rendición y junto con ello debe ir el nombre, el cargo que tiene en la organización y en caso de ser un CAA indicar el nombre de la carrera, si no a la facultad, que pertenece.

Por otra parte, se deben adjuntar los documentos originales que se encuentran en la rendición. En caso de tener gastos superiores a 3 UTM se deben adjuntar tres cotizaciones [19].

2.2. Tecnologías utilizadas

Para resolver el problema mencionado en el Capítulo 1, existen muchas tecnologías dentro de las cuales se encuentra, PHP, Ruby on rails, WordPress y Wix, entre otras. Pero para este proyecto se utiliza ASP.NET Core, cuyo motivo se debe a que es un framework sólido y robusto, tanto para proyectos pequeños, grandes o con gran potencial de crecimiento. Además, está orientado a objetos debido a que C# conforma parte de la tecnología, haciendo que el código sea ordenado y legible. Por otra parte, se puede utilizar para programar aplicaciones móviles, web o incluso para aplicaciones de computadoras de escritorio. Por último, otra de las razones de escoger esta tecnología es porque las empresas del rubro la están solicitando como conocimientos requeridos.

Para entrar un poco más en detalle, se da a conocer pequeñas descripciones de los principales componentes que cuenta esta tecnología, además de las librerías que se utilizan en el proyecto.

- **HTML-Razor:** HTML es un lenguaje de marcado de hipertextos el cual dispone de un conjunto de etiquetas que son insertadas en un documento que

puede mostrar la información a través de un navegador web. Este cuenta además con Razor, que es una sintaxis basada en C# que permite usarse como motor de programación en las vistas HTML.

- **CSS:** Lenguaje de diseño gráfico que describe cómo es la presentación de un documento estructurado escrito en HTML.
- **Javascript:** Lenguaje de programación interpretado, el cual es utilizado en el desarrollo web para implementar páginas web dinámicas.
- **C#:** Lenguaje de programación orientado a objetos el cual fue desarrollado por Microsoft para la plataforma dotNET. Su sintaxis básica proviene de C/C++ y es similar a java, e incluye mejoras cuyo origen son de otros lenguajes.
- **Modelo Vista Controlador (MVC):** Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: Modelo es donde se encuentran los datos que maneja el sistema, la lógica de negocio y los mecanismos de persistencia. La Vista es la información que la aplicación muestra al cliente a través de la interfaz de usuario obtenida de los datos almacenados en el Modelo. Por último, el Controlador se encarga de actuar como puente para el envío de información entre el Modelo y la Vista. Además, se realizan las transformaciones necesarias para adaptar los datos a las necesidades de cada uno.

2.2.1. Frameworks

- **ASP.NET Core:** framework diseñado para la creación de aplicaciones modernas conectadas a Internet, tales como aplicaciones web y APIs Web. Además es multiplataforma y Open Source.

2.2.2. Bibliotecas

DinkToPdf: biblioteca que se utiliza para convertir un documento HTML con CSS en un documento PDF. Esta librería es gratuita y de código abierto. La bi-

bliblioteca necesita como dependencia la biblioteca wkhtmltopdf, que se debe agregar manualmente.

Wkhtmltopdf: es una herramienta de línea de comandos, cuenta con una licencia Open Source (LGPLv3) y su objetivo es convertir páginas HTML o sitios web en archivos PDF.

2.3. Metodología de desarrollo de software

La palabra metodología está compuesta por tres vocablos griegos, los cuales son: metá (“más allá”), odós (“camino”) y logos (“estudio”). Es por ello que del conjunto de medios que se pueden utilizar para conseguir un objetivo, una metodología determina cuales son los más adecuados.

Las metodologías de desarrollo de software ayudan a estructurar, planificar y controlar el proceso de desarrollo de los sistemas de información, con el objetivo de resolver las siguientes preguntas:

- ¿Quién debe hacerlo?
- ¿Qué debe hacer?
- ¿Cuándo debe hacerlo?
- ¿Cómo debe hacerlo?

Este marco de trabajo es un modo sistemático de realizar, gestionar y administrar un proyecto con la posibilidad de llevarlo al éxito. Cada metodología de desarrollo de software tiene ciertas actividades las cuales ayudan a idear, implementar y mantener el producto de software, el cual surge de alguna necesidad hasta cumplir con el objetivo por el cual fue creado. Por lo último mencionado, se propuso la necesidad de metodizar el desarrollo de software, para lo cual se han hecho muchos esfuerzos para mejorarlo con el tiempo. Todo comenzó por los modelos carentes de disciplina y estándares, luego evolucionó a modelos estructurados centrados en el control del proceso (denominados metodologías tradicionales) y por último se dio paso a modelos iterativos y flexibles (metodologías ágiles).

Existe una constante guerra para determinar qué tipo de metodología es mejor que otra, pero surgen preguntas como: ¿existe una mejor que otra? o ¿cuál de ellas llevará al éxito?

Respondiendo a las últimas preguntas mencionadas, no existe una metodología mejor que otra ya que su elección también depende del proyecto en el que se utilicen. En efecto, factores tales como recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. pueden condicionar la elección o aplicación de una metodología específica. Además, no existe metodología que asegure el éxito ya que todas pueden fracasar, lo importante es saber abordar los problemas y decidir qué tipo de metodología ocupar. Cada metodología posee algo bueno y en algunos casos se pueden mezclar, pero depende de las características del proyecto.

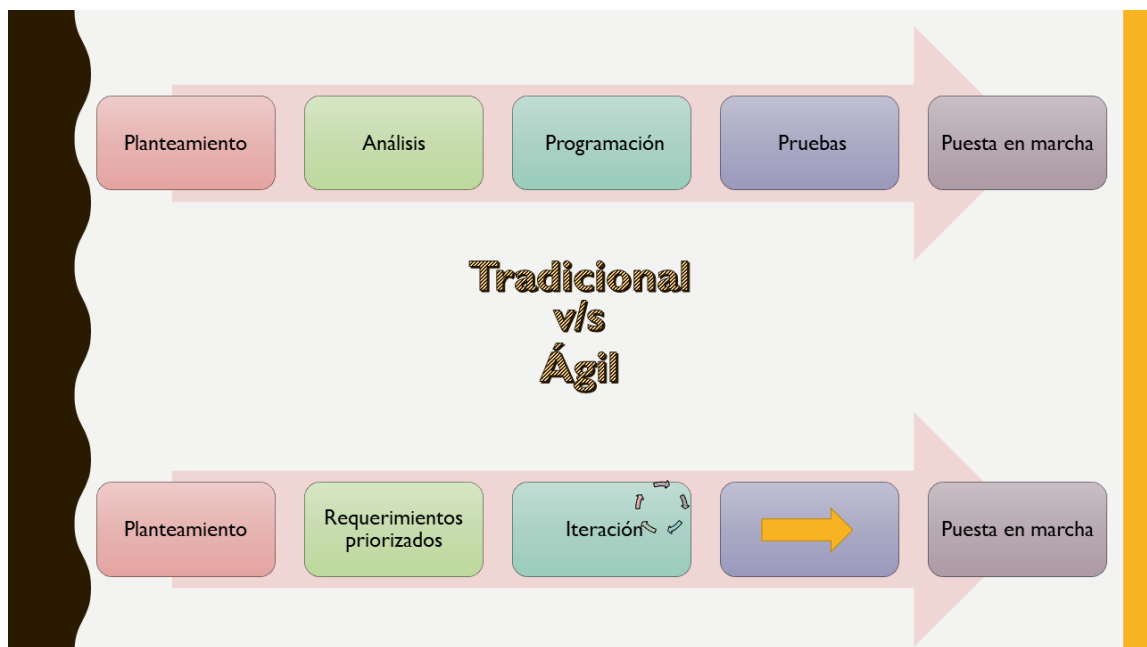


Figura 2.1: Metodología tradicional vs metodología ágil [21].

En cuanto a la elección de la metodología que se utiliza, existen un sinnúmero de factores que diferencian entre una y otra, tal como se puede apreciar en la **Figura 2.1** y en el **Cuadro 2.1**. Es por ello que dada esta información y principalmente por los factores tales como *Cliente*, *Tamaño del equipo*, *Ciclos*, *Perspectiva al cambio* y *Enfoque* es que se optó por elegir una Metodología Ágil de la cual se da mayor detalle en el siguiente ítem de este documento.

Cuadro 2.1: Metodología Tradicional vs Metodología Ágil [21].

	Tradicional	Ágil
Enfoque	Predictivo	Adaptativo
Éxito de medición	Conformación de planificar	Valor del negocio
Tamaño del proyecto	Grande	Pequeño
Estilo de gestión	Autocrático	Descentralizada
Perspectiva para el cambio	Cambio y sostenibilidad	Cambio y adaptabilidad
Cultura	Comandos de control	Liderazgo-Colaboración
Documentación	Pesado	Bajo
Cliente	Interactúa mediante reuniones	Parte del equipo
Énfasis	Orientado a procesos	Orientado a personas
Ciclos	Limitados	Muchos
Planificación por adelantado	Exhaustivo	Mínimo
Retorno de la inversión	Fin del proyecto	A principios del proyecto
Tamaño del equipo	Grandes	Pequeños

2.4. Metodología ágil

Tras las constantes fallas en los proyectos de desarrollo antes del año 2001, es que en este año Kent Beck y otros 16 notables desarrolladores de software, escritores y consultores (grupo conocido como la “Alianza ágil”), firman el “Manifiesto por el desarrollo ágil de software”, el cual define valores y principios que deben estar presente en la metodología ágil, con el fin de mejorar la forma de desarrollar software, cuyos principios son:

- Valorar al individuo y las interacciones del equipo de desarrollo por sobre el proceso y las herramientas.
- Desarrollar software que funciona por sobre conseguir una buena documentación.
- Valorar la colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

Este marco de trabajo aporta gran flexibilidad a los cambios manteniendo las condiciones del proyecto, reduciendo los costes y aumentando la productividad.

Cabe destacar que la metodología ágil se caracteriza por:

- Equipos multidisciplinarios y organizados de acuerdo a las necesidades de cada proyecto.

- La importancia de la comunicación entre los miembros que conforman el proyecto.
- La satisfacción del cliente ante el proyecto a desarrollar a través de la constante comunicación y así lograr evitar fallos y retrasos.
- La gran motivación e implicación del equipo de desarrollo del proyecto. Dado que todos los miembros pueden conocer el estado del proyecto en todo momento, todas las ideas de sus integrantes son de gran importancia.
- La adaptabilidad a que los requisitos cambien incluso en etapas tardías del desarrollo.
- El ahorro de tiempo y de costes dado que se trabaja de una manera más eficaz y sin olvidar el presupuesto acordado junto con el tiempo de entrega definido al comienzo del proyecto.
- Trabajar con mayor velocidad y eficiencia, dado que se realizan entregas parciales y así detectar errores lo antes posible y eliminar características innecesarias del producto.
- Rápido retorno de la inversión debido a las constantes entregas parciales en las cuales se muestra las funcionalidades principales del producto, haciendo que la rentabilidad de la inversión sea de manera más acelerada.

Existen muchas metodologías, cada una con distintas orientaciones acerca de cómo llevar a cabo el proceso de desarrollo de software. Entre ellas se encuentran *Scrum* (marco de trabajo seleccionado para el desarrollo del proyecto, el cual se describe más adelante), XP (eXtreme Programming), Métodos Crystal, FDD (Feature Driven Development), ASD (Adaptive Software Development), RUP (Rational Unified Process), etc.

De todos los métodos ágiles mencionados anteriormente se escoge Scrum. Es por ello que se describe en las siguientes secciones, explicando de que trata, cuáles son los roles que se deben cumplir para que funcione el método, los artefactos que dan vida al proyecto, las reuniones las cuales ayudan a ver la planificación; problemas y avances del producto y, por último, el ciclo de vida que tiene Scrum.

2.4.1. Scrum

Scrum¹ es un método de desarrollo ágil de software creado por Jeff Sutherland y su equipo de trabajo al inicio de la década de los 90 [22].

Su objetivo principal es ser un proceso iterativo e incremental que ayuda a la gestión de proyectos con requisitos cambiantes. Dentro de sus principales cualidades se encuentran las iteraciones y reuniones en el proceso de desarrollo con actividades que ayudan a controlar los cambios y los riesgos del proyecto para elevar la productividad.

Scrum produce resultados en periodos cortos de tiempos (aproximadamente 30 días), delegando al equipo de trabajo la responsabilidad de escoger la mejor forma de trabajar.

La clave del éxito de Scrum son los *sprints*, los cuales pueden durar de dos a cuatro semanas, en donde al iniciar cada uno de estos, se selecciona una lista de historias de usuarios que serán resueltas en dicho sprint⁶ (a esta lista se le conoce como backlog). Por otra parte, dentro del transcurso de cada iteración, diariamente se realizan breves reuniones de aproximadamente quince minutos, las que ayudan a conocer el avance del desarrollo del proceso.

Cabe destacar que las historias de usuarios son pequeñas descripciones de alguna funcionalidad que debe incorporar la aplicación. En la **Sección 2.4.6.2** se da una explicación completa de lo que es una historia de usuario.

2.4.2. Roles

Para llevar una correcta organización en la metodología se debe identificar las actividades que debe realizar cada persona que participa en el proyecto, tanto de forma interna como externa (equipos de trabajo, usuarios e individuos que se relacionan de alguna forma con el proyecto). A este conjunto se le denomina roles.

Una persona puede tener asignado uno o más roles y, a su vez, los roles pueden ser asignados a uno o más individuos. Cabe destacar que las asignaciones se hacen al comienzo del proyecto en base a las características de los sujetos y de los recursos que disponen. Es por ello que a continuación se explican los roles involucrados en el proyecto:

¹Nombre originario de una jugada que tiene lugar durante un partido de rugby en donde se forma un grupo de jugadores alrededor del balón y todos trabajan juntos (a veces con violencia) para moverlo a través del campo.

- **Scrum Master (SM):** es el encargado de que el equipo tenga todas las condiciones necesarias para trabajar sin obstáculos y así cumplir con el sprint. Por otra parte, es quien procura que el proceso siga su curso. Además, supervisa la comunicación dentro del equipo, resuelve desacuerdos dentro del equipo o entre los equipos, y ayuda a asegurar la productividad del equipo. Cabe destacar que el scrum master no necesariamente es el líder para el equipo y es el responsable del “cómo hacer”.
- **Product Owner (PO):** es el encargado de representar a los stakeholders (interesados externos o internos). La tarea más importante del PO es tomar decisiones sobre la realización y asignación de prioridad de las historias de usuarios en la lista de tareas. Además, debe asignar historias de usuarios del proyecto a los integrantes del equipo, ser extrovertido y tener facilidad para expresarse de forma oral y escrita. Por último, para que el PO pueda cumplir con sus funciones debe tener la capacidad de planificación y habilidades de comunicación, además de influir en el equipo con su interés y motivación.
- **Team (equipo):** son los encargados de cumplir con el desarrollo del producto y están reunidos en equipos de cinco a diez personas. Tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos (enfermedad de alguien del equipo, dependencias con otros equipos, presión excesiva del PO, etc.), los cuales deben ser discutidos y realizados al final del sprint. Para ello, debe cumplir con ciertas características tales como la autogestión, la autoorganización y debe ser multifuncional.
- **Stakeholders:** son todas aquellas personas beneficiadas y quienes posibilitan el proyecto. Estos pueden ser usuarios, administradores, técnicos, etc.

2.4.3. Aptitudes requeridas

Para cumplir los roles mencionados en la **Sección 2.4.2** y lograr los objetivos, es necesario tener una buena comunicación con los stakeholders, además de un continuo y rápido aprendizaje (dado a la alta curva de aprendizaje y la poca experiencia). Por

otra parte, es necesario ser autocrítico como también tener la capacidad de recibir críticas constructivas en caso de ser necesario.

2.4.4. Artefactos

- **Backlog:** Lista de historias de usuarios que constituyen al proyecto. El PO evalúa estas historias y le da prioridad de esta lista de acuerdo con lo requerido.

Existen tres tipos de backlogs, el product backlog, el backlog de versión o release backlog y por último el backlog de sprint, los que se muestran con mayor detalle a continuación:

Product backlog: Lista de historias de usuarios o características del proyecto los cuales proporcionan valor comercial para los stakeholders. Estas historias de usuario se caracterizan por ser de alto nivel y no tienen detalles de implementación o técnicos.

Backlog de versión: Consiste en una sub lista de tareas extraídas del backlog del producto, las cuales son priorizadas por el product owner y deben desarrollarse para la siguiente versión del producto. Estas deben ser más detalladas, con mayor precisión en cuanto al tiempo de desarrollo y puede tener observaciones realizadas por los miembros del equipo.

Backlog del sprint: Son tareas que provienen del backlog de versión y que pueden completarse en periodos cortos de tiempo. En cuanto al desarrollo de estas, deben completarse hasta el final del sprint por los equipos que se comprometieron en realizarlas.

Aquellas historias de usuarios que no son completadas en su tiempo planificado, se devuelven al backlog donde son evaluadas nuevamente por el PO.

- **Incremento:** Avance del proyecto que se realiza dentro del sprint en el cual los equipos se comprometieron a desarrollar.
- **Demostración:** Entrega de incremento del software a los stakeholders de tal manera que estos evalúen la funcionalidad implementada. Cabe destacar la posibilidad de que la demostración no contenga toda la funcionalidad

planteada, sino que aquellas funciones susceptibles de entregarse dentro del periodo establecido.

- **Lista de impedimentos:** Son dificultades que impiden la productividad y la calidad del proyecto. El encargado de controlar esta lista, y en lo posible de que este vacía, es el scrum master, por lo cual crea una lista de tareas para tener seguimiento de los impedimentos que requieren resolverse.

2.4.5. Reuniones

En el proceso de Scrum, las reuniones son de suma importancia y en cada paso en el proceso existen tipos de reuniones con sus propias características. Las reuniones principales que destaca Scrum son: *Planificación de sprint*, *Reunión diaria*, *Revisión del sprint* y *Reunión retrospectiva del sprint*. En la **Figura 2.2** se ve el ciclo de Scrum desde el punto de vista de las reuniones.

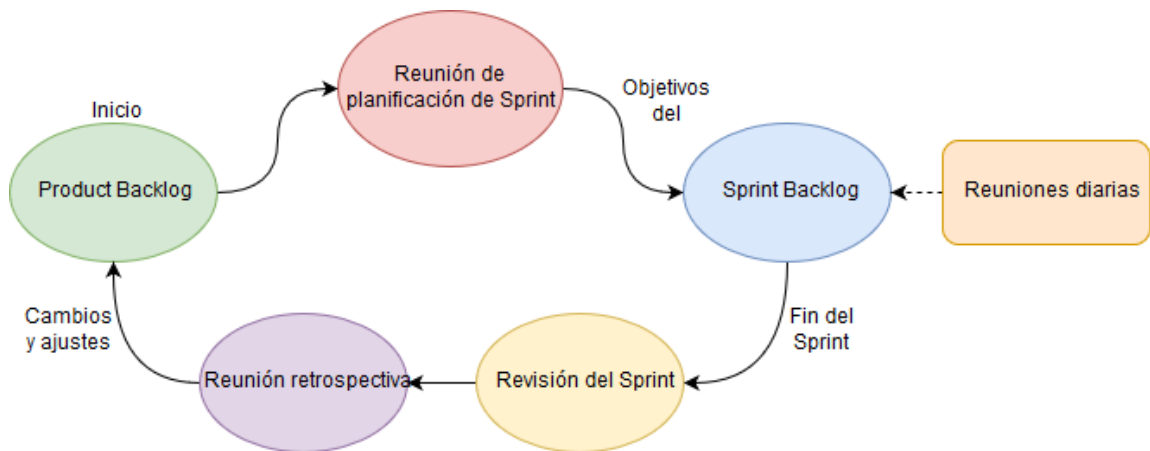


Figura 2.2: Reuniones en el Proceso de Scrum.

- **Planificación del sprint:** Los sprints tienen como objetivo convertir el backlog del sprint en un incremento. Es por ello que antes de comenzar cada uno de ellos, se deben reunir el PO, el scrum master y el equipo para definir en cuál funcionalidad se debe trabajar, asumiendo que los objetivos no van a cambiar durante este sprint. Para ello se debe tener en cuenta la estimación de esfuerzo dedicado a trabajos de soporte o de preparación del ambiente que requiere el proyecto, los requerimientos de recursos de infraestructura o logístico (máquinas, redes, pizarra, etc.), entre otras.

Esta reunión consta de dos partes; en la primera, el equipo junto con el product owner seleccionan la lista de requisitos que se convertirán en el incremento del siguiente sprint según la prioridad hecha por los participantes de la reunión. Mientras que en la segunda parte, se debe planificar el trabajo del sprint, definiendo las tareas que se realizarán en el backlog de este último.

- **Reunión diaria:** Juntas de corto periodo (por lo general quince minutos) las cuales se realizan a diario por el equipo de Scrum, en donde todos los miembros del equipo se deben plantear y responder las siguientes tres preguntas:

- ¿Qué hiciste desde la última reunión?
- ¿Cuáles obstáculos encontraste?
- ¿Qué esperas lograr para la siguiente reunión del equipo?

Durante cada reunión, los integrantes deben reportar sus avances y/o problemas enfrentados durante el trabajo realizado hasta ese momento, por lo que la confianza es fundamental dentro del equipo en esta fase. Cabe señalar que con esta acción de reportar problemas, se espera que los integrantes tengan una reacción de apoyo (referente al problema y con motivo de solucionarlo) y retroalimentación. Esto voto de confianza permite que cada integrante sepa en que está trabajando cada uno.

- **Revisión del sprint:** En esta reunión se inspecciona el trabajo realizado durante el sprint. Para ello, los equipos de trabajo presentan el incremento que se construyó, el cual es supervisado por el PO y los stakeholders. Por otro lado, deben indicar qué resultó bien y mal del incremento, las observaciones que surgieron durante éste y se realiza una demostración del producto que todos pueden ver. Aquí nuevamente se destaca la importancia de la confianza entre el equipo y el SM. Por último, basándose en la inspección se toman decisiones sobre qué realizar en el siguiente sprint, sin descartar la realización de adaptaciones al proyecto.
- **Reunión retrospectiva del sprint:** Proceso cuyo objetivo principal es discutir acerca de los pro y contras del sprint finalizado y determinar si se

requiere modificar algo para una iteración mejor con mayor productividad. Los participantes de esta reunión hacen referencia en el cómo se construyó el incremento anterior y cualquier cosa que afecte al equipo, ya sea procesos, prácticas, comunicación, entorno, artefactos y herramientas. Este tipo de reunión es de gran importancia ya que permite mejorar al equipo de trabajo durante cada etapa del proyecto. Por otra parte, los que pueden participar en esta reunión son el equipo, el PO y el SM, teniendo todo el derecho de opinar abiertamente sobre el trabajo realizado.

2.4.6. Ciclo de vida

Antes de iniciar cada iteración, el equipo de trabajo revisa las tareas pendientes y además selecciona nuevas tareas que formarán parte del nuevo incremento de funcionalidad, el cual se incorpora al software terminada la iteración. Para realizar lo anterior, se debe tener presente la tecnología y los conocimientos que tiene el equipo para seleccionar los requisitos y así decidir en conjunto, cómo implementar la funcionalidad.

En cuanto al desarrollo, se realiza en la fase del sprint, en donde se satisfacen los requisitos definidos en el sprint backlog dependiendo de la funcionalidad que se debe incorporar en el incremento. Para ello, gran parte del tiempo se desarrolla y revisa antes de incorporar el incremento al proyecto, pero a veces sólo se revisa y ajusta actividades anteriormente realizadas ante algún cambio que surge tras las entregas de los incrementos. Posteriormente, pasa por un proceso de revisión para verificar si el incremento cumple con lo requerido. Todo lo mencionado con anterioridad se repiten en iteraciones que duran aproximadamente 30 días a lo largo del proyecto hasta culminar con el producto. Cabe destacar que cuando se comienza un sprint, no se pueden agregar nuevos cambios. Por lo tanto, esto permite trabajar en un ambiente enfocado a corto plazo, pero estable. En resumen, la **Figura 2.3** muestra las fases que se mencionaron de forma general.

2.4.6.1. El proyecto

En ítems anteriores se habla de la metodología que se debe aplicar al proyecto para lograr administrar y organizar los recursos. También se hablan de los roles que tienen los individuos que dan vida al proyecto pero, ¿Qué es el proyecto?

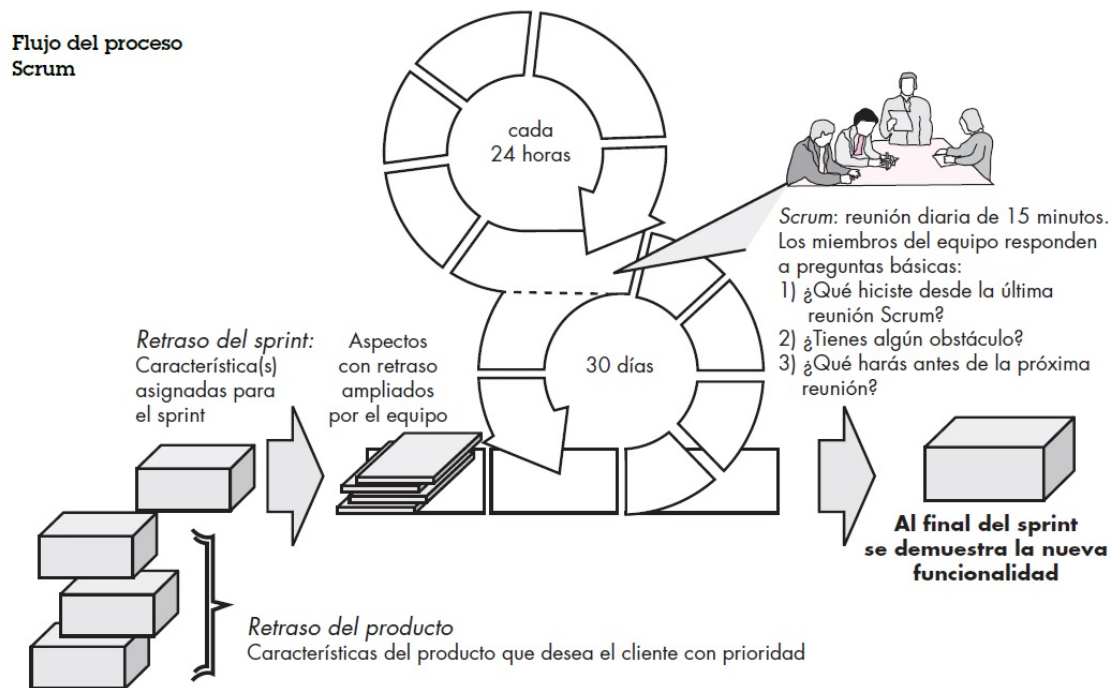


Figura 2.3: Proceso de metodología Scrum.

El proyecto es el ciclo general que ayuda a resolver el problema planteado. Como se aprecia en la **Figura 2.4**, comienza con una reunión de captura de requisitos y finaliza con la entrega del producto junto con la puesta en marcha. Dentro de este ciclo general, se tienen muchos sub-ciclos denominados “Entregas” o “Incrementos”, que corresponden a nuevas funcionalidades realizadas y mejoras continuas a lo largo del proyecto. Para realizar estas entregas, existe otro sub-ciclo que corresponde a las “Iteraciones” o “Sprints” y dentro de esta, están enlazadas un conjunto de fases que se deben llevar a cabo para concluir cada iteración.

2.4.6.2. Las fases

Para cada iteración se cuenta con las siguientes fases: (I) Requisito, (II) Diseño, (III) Planificación, (IV) Construcción y (v) Pruebas. Sin embargo, su presencia y tiempo de dedicación a cada una depende de cada iteración. Por ejemplo, en la primera iteración se incluyen todas las fases debido a que se realiza un prototipo del proyecto. Mientras que existen otras en que solo serán para corregir o modificar alguna funcionalidad. Todo lo anterior depende de las condiciones presentes en la

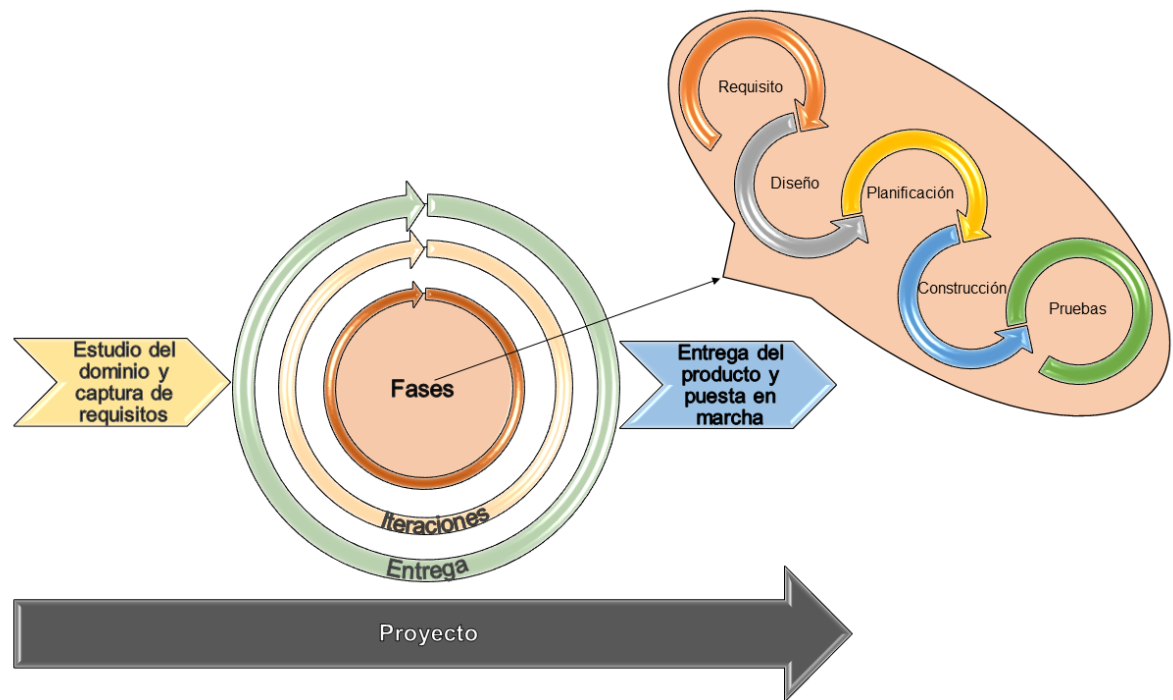


Figura 2.4: Proceso utilizado en el proyecto.

iteración.

A continuación se procede a contextualizar de qué trata cada una de las fases de la iteración:

Fase de requisitos: En esta etapa es donde el PO se reúne con el cliente para obtener información sobre las funcionalidades que tiene el proyecto (etapa comúnmente denominada “Captura de Requisitos”). En la metodología Scrum, para este efecto existe lo que se conoce como *historia de usuario*, en donde, de manera simple y clara se especifica semiformalmente lo que el cliente espera de la aplicación. A mayor abundamiento, las historias de usuario son pequeñas descripciones de los requerimientos de un cliente; y para su redacción se debe considerar la descripción del rol, la funcionalidad y el resultado esperado en una frase corta, cumpliendo con el siguiente formato: **Como <quién> Quiero <qué> Para <objetivo>**. Naturalmente, para una aplicación se pueden definir decenas o miles de historias de usuario, dependiendo de la envergadura del proyecto. Para esta fase se recomienda hacer pautas para mejorar el trabajo con el cliente. Entonces, lo que debe realizar el PO es escribir lo que el cliente quiere como *historias de usuario*.

Fase de diseño: Tras la obtención de requisitos, se procede a realizar el diseño del sistema, el cual incluye la base de datos, la interfaz y la estructura. Cabe destacar que cada diseño debe ser simple de comprender.

Fase de planificación: Luego de tener claro de lo que se quiere al realizar la captura de requisitos y el diseño, se procede a describir las actividades a realizar. Para cada actividad se debe definir su duración, dependencia y la fecha de inicio y de fin. Por último, se debe asignar estas tareas a un responsable de algún equipo de trabajo.

Fase de construcción: Ya asignadas las tareas a los responsables, se procede a codificar e implementar las funcionalidades que logran cumplir con los requisitos planteados en la planificación. La finalidad de esta etapa es desarrollar un incremento como se ha explicado en la **Sección 2.4.4** a través de iteraciones sucesivas.

Fase de pruebas: En esta fase se evalúa la funcionalidad del incremento. Para ello se hacen pruebas de integración y validación junto con el cliente, quien realiza las pruebas de aceptación.

2.4.7. Documentación

Otro punto importante dentro de la metodología utilizada es que se basa en el desarrollo por sobre la documentación. Es por ello que sólo en algunos casos se utiliza esta para tener un seguimiento de los avances, resultado de las pruebas y minutas de reunión. Sin embargo, existe una documentación obligatoria y es el manual de usuario, ya que es considerado como una herramienta de gran provecho para el usuario final.

2.5. Validación de la aplicación

La metodología con experimentos controlados se utiliza como método de evaluación del proyecto. Esta metodología maneja pruebas que se hacen bajo condiciones totalmente constantes aunque en algunos casos puede variar algún factor. Este persigue realizar un muestreo de usuarios conocidos los cuales utilizan el software con pruebas de usabilidad simples.

Por otra parte, los usuarios que realicen las pruebas de usabilidad no necesariamente deben ser los mismos.

Para finalizar, esta metodología de evaluación hace posible la comprensión e identificación de las variables que están siendo utilizadas en la construcción del software. Todo lo anterior se debe ya que, según el artículo *Albert Einstein and Empirical Software Engineering* [1], experimentar con la construcción de software permite “Aumentar la comprensión de lo que hace al software bueno y cómo hacer un software bien”².

2.6. Herramientas de planificación

Si bien existen muchas aplicaciones y herramientas que facilitan la planificación y gestión de proyecto tales como Trello, OpenProj, Open Workbench, GanttProject, etc, se escoge Taiga. Esta aplicación ayuda a gestionar las actividades de los proyectos y para ello se registran junto con sus respectivas historias de usuarios. Luego, cada historia de usuario se agrega a un sprint y se desglosa en pequeñas tareas si así lo requiere, para así ser asignadas a las distintas personas que trabajan en el proyecto.

A cada proyecto se puede asociar personas con algún rol según corresponda (al menos un colaborador). Estas deben tener cuenta de usuario para así ver en detalle el historial de su trabajo, los proyectos en los cuales participa, las tareas y el estado de estas (según en el lugar en que se ubique en el Kanban), entre otras características. Además, se puede priorizar las tareas según el rol que tenga cada persona y así lograr ver cuánta dificultad puede tener. Por otra parte, se puede visualizar la cantidad de colaboradores que tiene el proyecto (si es que los hay).

En cuanto al Kanban, es un tablero utilizado por los equipos de desarrollo del proyecto para mapear y visualizar el flujo del trabajo. Este consta de filas, las cuales representan diferentes tipos de actividades específicas del proyecto denominadas tarjetas kanban; mientras que las columnas ayudan a visualizar el estado en que se encuentra cada actividad. Así mismo, las tareas que se encuentran en el tablero, aparecen como tarjetas Kanban [10].

²El texto original de esta cita es *Gain more understanding of what makes software “good” and how to make software well* [1].

3. Metodología utilizada

Uno de los puntos más críticos al momento de desarrollar un proyecto de software es la elección de la metodología que se utiliza ya que otorga la capacidad de organizar y responder a las necesidades del proyecto. Además de que, una vez elegida la metodología no hay vuelta atrás.

3.1. Estudio del entorno

Antes de comenzar a explicar cómo fue el proceso de la metodología utilizada, se detalla cómo es el ambiente en el cual se realiza el proyecto. Este ambiente contempla el análisis de las personas, las cualidades requeridas de la aplicación y las herramientas utilizadas para el desarrollo del producto.

3.1.1. Las personas

Los roles tales como Scrum Master, Product Owner y Team los cumple una persona, que es el autor de esta memoria. Es por ello que las etapas del proyecto se deben planificar con mucho cuidado para cumplir con el objetivo. Además, existen actividades adicionales al proyecto, por lo cual el tiempo disponible no es óptimo.

Por otra parte, la curva de aprendizaje del autor de esta memoria es alta debido a la falta de conocimiento de la tecnología y la poca experiencia de la metodología de trabajo, pero se cuenta con habilidades tales como el compromiso, la motivación, organización y el autoaprendizaje para sacar el proyecto adelante.

3.1.2. La aplicación

La aplicación solicitada permite asistir al proceso que una OE debe realizar para la solicitud de un fondo por rendir según la RU N°2083 del año 2017 (ver **Sección 1.1**).

Si bien existe documentación de cómo realizar el proceso de fondo por rendir, es muy burocrático seguirlo al pie de la letra por las distintas OE, cometiendo muchos errores (sobre todo las nuevas OE que no tienen experiencia).

Aún cuando el objetivo principal que es cumplir con lo estipulado por la RU anteriormente mencionada se conserva, los requerimientos del sistema que tienen que ver con la facilidad de uso pueden sufrir modificaciones como resultado de las pruebas de usabilidad. Además, es posible que los requisitos cambien a medida que avanza el proyecto. Es por ello por lo que son evaluados y aceptados siempre que su impacto no sea negativo, tanto en el desarrollo como en el tiempo que conlleva para así no atrasar la entrega.

El sistema ayuda a la confección y exportación a un archivo PDF que contiene una solicitud de fondos para realizar una actividad. En caso de ser aprobada esta última, se debe subir al sistema una copia digitalizada de la RU asignada para luego confeccionar la rendición del evento. Tras terminar la elaboración de la rendición, esta es exportada a un archivo PDF junto con un listado de las boletas y/o facturas que deben ser adjuntadas al documento.

En cuanto al listado de documentos, debe estar en el mismo orden en que fueron ingresados al sistema para así ayudar al usuario a chequear que las boletas y/o facturas sigan el orden asignado en el sistema.

3.1.3. Las herramientas

Para el desarrollo de cada Sprint, se utiliza una aplicación gratuita y Open Source llamada Taiga, la cual es una plataforma de gestión de proyectos para diseñadores ágiles.

Para comenzar a utilizar esta plataforma se deben registrar las historias de usuarios con su respectiva prioridad (etapa conocida como Product Backlog). Una vez realizado lo anterior, se proceden a crear los Sprints y por último se añaden las historias de usuarios a cada uno de estos tal como se muestra en la **Figura 3.1** y en caso de ser necesario, una historia del Product Backlog puede ser subdividida para

lograr un mejor enfoque y claridad en el desarrollo.



Figura 3.1: Sprint Backlog.

Dentro de cada Sprint hay un Kanban, el cual ayuda a organizar el desarrollo de cada historia de usuario realizada en el Sprint Backlog, en donde divide el trabajo en pequeñas tarjetas o tickets.

Dado lo anterior es que se divide el proceso de trabajo en cinco columnas como se muestra en la **Figura 3.2**, en donde se encuentra:

- **Nueva:** Se ingresan pequeñas tarjetas o ticket de trabajo que ayudan a realizar el cumplimiento de la historia de usuario.
- **En curso:** Sección en donde se encuentran las tareas que se están realizando.
- **Lista para Testear:** Columna en la que se encuentran las tareas terminadas y en proceso de aceptación.

- **Cerrada:** Área en donde se encuentran las tareas que han sido finalizadas y aceptadas.
- **Necesita Información:** Sección en donde se encuentran las tareas que no han logrado culminarse debido a que falta información para su realización.

SPRINT 2 - RESOLUCIÓN E INICIO DE SESIÓN SIMULADOR DE RENDICIONES PARA CC

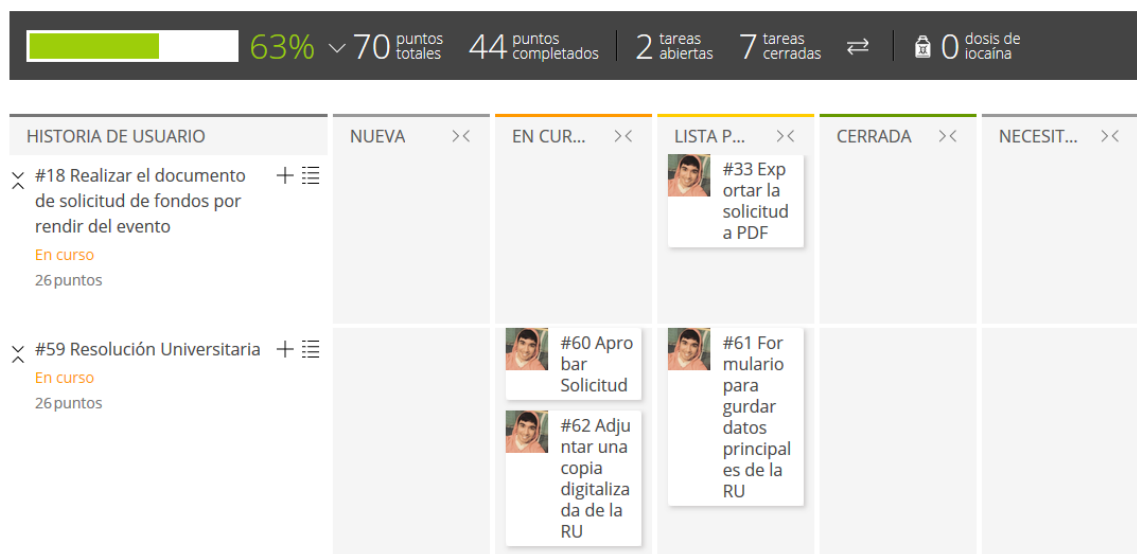


Figura 3.2: Proceso de trabajo de un Sprint.

Para el diseño de base de datos se usa una aplicación llamada draw.io [6] para diseñar el diagrama Entidad/Relación (E/R). Luego de diseñar el diagrama E/R, se procede a crear el modelo relacional y para esto se utiliza StarUML, que permite la creación de estas tablas de forma gráfica y exportar todo esto a código, el cual es importado en MySQL que es donde está alojada la base de datos. StarUML no sólo se utiliza para diseñar la base de datos, sino que también para la creación de la arquitectura lógica y el diagrama de clases.

En cuanto al desarrollo de la aplicación se utiliza el framework ASP.NET Core (ver **Sección 2.2**). Esta tecnología utiliza HTML-Razor para estructurar las vistas que el usuario puede acceder y a su vez, este último utiliza CSS el cual describe cómo es la presentación de la vista. Además, con la ayuda de JavaScript hace que la aplicación sea dinámica. También se utiliza C# debido a que forma parte de la plataforma dotNET. Este ayuda a generar el programa para que opere la plataforma.

Por otra parte, en primera instancia se utiliza el IDE Visual Studio para la creación del proyecto. Pero a lo largo del desarrollo, se cambió de IDE a VSCode, ya que había que utilizar una librería para la creación de archivos PDF que era incompatible con Visual Studio.

Por último, se utiliza Bizzagi Modeler para la creación de diagramas de flujo que permiten comprender el proceso de solicitud de fondos por rendir tanto para Federación como para Centros de Alumnos.

3.2. La metodología

Para llevar a cabo el desarrollo del sistema a implementar con flexibilidad para afrontar cambios en las distintas funcionalidades requeridas en el proyecto, se elige Scrum como metodología de desarrollo de software. Si bien existen muchas metodologías las cuales tienen amplia documentación y han probado ser eficientes para casos en específico, los motivos por los cuales se utiliza la metodología ágil Scrum son:

- Completa participación del cliente y prueba de funcionalidades para obtener retroalimentación.
- Se pueden obtener los cambios y adaptarse al proyecto en cada Sprint.
- La experiencia que el autor ha tenido en el desarrollo de otros proyectos realizados en el proceso de formación académica.
- Como la idea principal es realizar rendiciones y se han aclarado algunos requerimientos, mas no son todos, es que se espera que el proyecto aumente en el proceso de desarrollo. Es por esta incertidumbre que se escoge la metodología Scrum, ya que ayuda al cliente y al equipo de trabajo a definir los requisitos del proyecto de modo incremental.

3.2.1. Roles

Para cumplir con la metodología existen diferentes roles los cuales son los encargados de dar vida al proyecto (ver **Sección 2.4.2**). Pero, para efectos de esta memoria y dada las características del proyecto, es que los roles de Scrum Master, Product Owner y Team los realiza el autor, encargándose de:

- Mantener la comunicación con los stakeholders.
- Definir las historias de usuarios que se deben realizar en cada iteración.
- Implementar funcionalidades que satisfagan las historias de usuarios.

En cuanto a los stakeholders, para este proyecto corresponden a los presidentes y secretarios de finanzas de las OE, además de las secretarías de las distintas Escuelas de la Facultad de Ingeniería de la Universidad de Talca.

3.2.2. El proceso

La gran parte de las metodologías definen sus procesos como una secuencia de pasos. Pero el gran problema es que si se comete un error en alguno de estos, volver atrás para solucionar el problema tiene un costo muy alto. Es por ello que para este proyecto se aplicó un proceso de ciclos en el desarrollo de este y así, en cualquier momento en que surjan problemas se pudiera volver atrás. Además, la metodología es tolerante al cambio siempre y cuando no afecte negativamente al proyecto, es por ello que la presencia de los stakeholders como participantes en el proyecto es de vital importancia. En los siguientes subítems se discute en detalle cómo se aplica la metodología al proyecto y la planificación de este.

3.2.2.1. El proyecto

El nacimiento de este proyecto se inicia cuando el autor de esta memoria se encuentra en calidad de Presidente del CAA de Ingeniería Civil en Computación entre los años 2016 y 2018. Siempre que se realizaba alguna actividad había problemas con las rendiciones. Un problema frecuente era cuadrar la suma de las boletas y/o facturas con el monto solicitado a la Institución para realizar el evento.

Posteriormente, en el momento de la revisión de la rendición por parte del departamento de Tesorería, al primer error que se encontraba se detenía la revisión, se rechazaba la rendición y se notificaba ese error. Posiblemente, la rendición podía tener más errores de alguna boleta fuera de la fecha o de algún gasto incurrido en falta ya que no pertenecía a alguna de las categorías mencionadas en la solicitud, entre otras. Pero debido a que se detiene la revisión al primer error que encuentran, no enviaban toda información, por lo que la OE corregía sólo lo notificado y al momento de realizar la segunda revisión o más, si se encontraban nuevos errores, se

volvía a rechazar. A consecuencia de lo anterior, demoraba la devolución del dinero e impedía que se realizaran nuevas actividades ya que existía una rendición pendiente de aprobación.

Es por ello que el autor lo propuso como tema de *Proyecto de Titulación* y al momento de ser aceptada, comienza con el proceso de formular el proyecto, realizando entrevistas para detectar las historias de usuarios y así tener una planificación general del proyecto, la cual podría modificarse a medida que avanza en este, ya que se podrían requerir nuevas funcionalidades o cambios de prioridades de funcionalidades requeridas con anterioridad. Finalmente, la organización de los sprints del proyecto sigue el proceso que conlleva realizar una Solicitud de Fondo por Rendir por parte de la OE, las cuales se mencionan a continuación:

1. Solicitud.
2. Solicitud y Resolución.
3. Búsqueda.
4. Rendición.
5. Usuarios.
6. Documentación y Panel de ayuda.
7. Detalles.

Por último, tras cumplir con todos los sprints mencionados anteriormente se realiza la entrega del producto y la puesta en marcha para el mes de Octubre del 2019.

3.2.2.2. Iteraciones del proyecto

El proyecto contiene muchas “Entregas”, y cada una de ellas se realiza a través de lo que se denomina “Iteraciones” o “Sprints”, como se menciona en la **Sección 2.4.6.1**. Es por ello que para el desarrollo del sistema se define lo siguiente:

- Cada iteración tiene un tiempo aproximado de **tres** semanas.

- Las iteraciones están compuestas por fases de requisitos, diseño, planificación, construcción y pruebas (ver **Sección 2.4.6.2**).
- Cada iteración tiene un objetivo a cumplir según la funcionalidad del sistema. Para este proyecto se han planificado **siete** sprints.

Para organizar cada sprint que corresponde al desarrollo del sistema, existe una etapa previa la cual busca capturar las historias de usuario que ayudan a tener una visión de lo que es el proyecto, permitiendo realizar el diseño del sistema. Tras lo mencionado anteriormente, se procede al inicio de la etapa de construcción.

Los sprints se organizan de acuerdo con la funcionalidad de la aplicación quedando de la siguiente forma:

1. **Solicitud:** Iteración centrada en la confección e implementación del proceso de Solicitud de Fondos por Rendir. Este entregable debe solicitar al usuario todos los datos que requiere un evento tales como categorías, participantes (si es que es enfocado a personas en específico), los datos principales del evento (nombre, fecha, etc.) y responsable del evento, entre otros.
2. **Solicitud y Resolución:** Iteración centrada en finalizar el proceso de solicitud generando un PDF con la solicitud realizada y el posterior ingreso tanto de los datos principales como de la copia digitalizada de la RU que envía la Universidad a la OE interesada.
3. **Búsqueda:** Iteración encargada de generar el entregable de búsqueda de Solicitudes tanto pendientes como finalizadas a través de fecha de creación de Solicitud o por nombre del evento.
4. **Rendición:** Iteración centrada en la confección de la rendición, la cual cuenta con el ingreso de los documentos (boletas y/o facturas), listado de la documentación según el orden en el que se ingresa para que el usuario las chequee y el documento de la Rendición en PDF.
5. **Usuario:** Iteración en donde se crea al administrador de la aplicación, el cual debe generar a nuevos usuarios y agregar nuevas categorías que se pueden ingresar en la solicitud para detallar en qué se llevaran a efecto los gastos.

6. **Documentación y Panel de ayuda:** Iteración encargada de realizar la ayuda que puede encontrar el usuario en la aplicación, además de la RU por la cual se rige la aplicación.
7. **Detalles:** Iteración encargada de afinar los detalles de la aplicación tales como, visualizar el estado de una solicitud de Fondo por Rendir, tener un respaldo de las rendiciones anteriores realizadas y editar una Rendición dentro de 20 días contados corridos después de haber finalizado el evento.

3.2.2.3. Reuniones

Para cada iteración se deben realizar reuniones, las cuales pueden ser documentadas tal como se menciona en la **Sección 2.4.7**. Específicamente, para efectos de este proyecto de titulación, las documentaciones que se realizan son:

- La captura de nuevos requisitos o modificaciones cuando se está en la reunión con el cliente que son los presidentes y secretarios de finanzas de las OE y las distintas Secretarías de Escuelas de la Facultad de Ingeniería.
- La documentación de algunas reuniones cuando se trabaja con el profesor guía.

Estas reuniones tienen gran importancia para el desarrollo del proyecto. Por ejemplo, las reuniones con el cliente son de vital importancia debido a que se obtienen requisitos o modificaciones de estos últimos y si no son documentados pueden afectar negativamente al proyecto. En cuanto a la reunión con el profesor guía, esta ayuda a ver el progreso del proyecto, tener críticas constructivas y soluciones a problemas que se presentan a lo largo del proyecto.

3.2.3. Cómo se trabajó

Tras detectar el problema, lo primero que se realiza es estudiar cómo realizar el proceso de solicitud de fondo por rendir, el cual se encuentra en la RU N°2083 del año 2017. También se observa cómo las distintas OE realizan este proceso y así detectar las principales falencias que existen. Con todo lo mencionado anteriormente se procede a realizar el diagrama de todo el proceso que conlleva una solicitud de fondo por rendir para las distintas OE a la cual está enfocada la aplicación. Esto con

el propósito de conocer y conseguir la mayor cantidad de requisitos e información del sistema.

Luego de obtener los requisitos que debe cumplir el sistema, se realiza el diseño de la aplicación el cual contempla la línea de navegación, la estructura general, la base de datos y el diseño de interfaz.

Naturalmente, antes de comenzar todo el desarrollo de la aplicación, se procede a realizar una planificación general del proyecto a través de iteraciones. Estas fueron planificadas en secuencia según el proceso que debe tener un fondo por rendir y las funcionalidades que conlleva cada una de estas.

En la primera iteración se deben cumplir todas las fases que tienen las iteraciones según lo estipulado en la **Figura 2.4**, dado a que el proyecto está en su etapa inicial y después, dependiendo de las circunstancias en que se encuentre el proyecto es que se ve si se siguen todas las etapas de las iteraciones.

Si bien existe una planificación inicial, esta se puede modificar (y de hecho así pasó) a medida que avanza el proyecto. Es por ello por lo que, en la etapa de requisitos de las iteraciones, las historias de usuarios que se tienen planificada realizar en esta, se deben ver en detalle y confirmar con el cliente. También se realizan entrevistas a las Secretarías de Escuelas, dado a que asisten a los respectivos Directores de Escuela en la recepción y envío de solicitudes de fondos por rendir junto con sus respectivas rendiciones.

Luego se procede a detallar el diseño según la etapa en que se encuentra enfocada la iteración. En efecto, a partir de la segunda iteración, puede omitirse la fase de diseño cuando no hay cambios en los requisitos. Pero si se efectúan o detectan cambios de requisitos en la iteración anterior, estos deben ser analizados en la fase de requisitos de la iteración actual y luego se debe verificar si afecta o no al diseño para ser implementados.

Tras tener claro lo que se debe realizar, se procede a la etapa de planificación en donde cada historia de usuario se detalla en tareas las cuales darán vida al proyecto. Después, estas tareas pasan a la etapa de construcción en donde comienza a tomar forma el proyecto y tras terminadas todas las funcionalidades planificadas en la iteración, se procede a realizar las pruebas para corroborar que se cumpla con lo estipulado por el cliente y la RU mencionada con anterioridad. Si todo resulta bien, se procede a realizar la entrega. En caso contrario, queda como retraso y se debe terminar en la siguiente iteración junto con las nuevas tareas que se asignan en esta

última.

Dado a que el autor de la aplicación cumple con la mayoría de los roles es que las tareas deben planificarse de tal manera que se deban cumplir en el tiempo de tres semanas por iteración. Además, las preguntas que se deben realizar en las reuniones diarias de Scrum (como se explicó en la **Sección 2.4.5**), en un comienzo se cumplían todos los días en que se trabajaba en el desarrollo de la aplicación. Pero con el avance del proyecto bajan a una vez por semana. Este ajuste se realiza debido a que no se requiere dar información del avance a otras personas debido a que los roles (a excepción del Stakeholder) lo realiza el autor de esta memoria, el cual conoce el avance del proyecto y los problemas que existen.

En cuanto a lo referente con el cliente, se está en comunicación con él al inicio de cada iteración cuando se debe confirmar los requisitos y al final, cuando se realiza la entrega de la iteración para que realicen algunas pruebas controladas. Una vez que se termina la versión beta (versión con funcionalidades principales), se realizan las pruebas con los distintos usuarios.

4. Características del sistema

En capítulos anteriores se ha mencionado de qué trata la aplicación tras plantear el problema. Además, se ha mencionado cómo es la metodología de desarrollo y cómo se aplica al proyecto. Es por ello por lo que en este capítulo corresponde detallar las características que debe tener la aplicación para cumplir con los objetivos que se mencionan en la **Sección 1.4** y así tener mayor dominio de lo que se realiza.

4.1. Aspectos generales

El simulador de rendiciones (SimRend) es un software que aborda los procedimientos llevados a cabo al momento de solicitar un fondo por rendir por las OE con el fin de obtener los medios económicos para realizar algún evento. Este procedimiento se explica en la **Sección 1.1** y dentro de los procesos principales que debe considerar la aplicación son:

- Confeccionar la solicitud de un fondo por rendir.
- Guardar una copia digitalizada de la RU generada tras la aprobación de la solicitud.
- Confeccionar la rendición de fondos solicitados.

Cabe destacar que, sin la aprobación de una solicitud de fondo por rendir no se puede realizar la rendición del fondo solicitado.

4.2. Modelos de contexto de solicitudes

Para concluir los aspectos generales que debe cumplir la aplicación, se realiza un estudio para comprender la situación actual de un proceso de solicitud de fondos por rendir. Si bien existen varias OE, esta memoria se enfoca en dos de ellas, a saber, Federación (de estudiantes del campus Curicó) y CCAA tal como se menciona en la **Sección 1.1**. Al realizar el estudio, se lleva a cabo un diagrama que explica el proceso que conlleva solicitar un fondo por rendir por parte de Federación (ver **Figura 4.1**) y el proceso que deben realizar los CCAA (ver **Figura 4.2**).

El proceso que conlleva la solicitud de un Fondo por Rendir por parte de Federación que representa la **Figura 4.1** es:

1. Elevar una solicitud a la DAAE.
2. Si se acepta la solicitud de Federación por parte la DAAE, este debe elevar una solicitud a quien dirige la Vicerrectoría de la DAAE junto con una copia de la solicitud realizada por Federación.
3. En caso de ser aceptada la solicitud por parte de la Vicerrectoría de la DAAE, esta crea la RU de aprobación de la Solicitud junto con tres copias para luego ser enviada al departamento de Contraloría el cual se encarga de chequear que el documento esté correctamente realizado.
4. En caso de ser aprobado por Contraloría, se envían nuevamente a Vicerrectoría de la DAAE y luego se procede a distribuir los documentos quedando el original en Secretaría General, y las cuatro copias son distribuidas a la OE interesada, al Gobierno Transparente, al Departamento de Tesorería y Presupuesto y a la Vicerrectoría Estudiantil.
5. Tras ser aprobada la solicitud y enviada la RU a los departamentos, Tesorería genera un Cheque al nombre del responsable del evento y lo envía a Federación.
6. Tras recibir la RU por parte de Federación, este se encarga de realizar el evento y después comenzar con el proceso de Rendición de la actividad.
7. Una vez finalizada la Rendición, se envía a quien dirige la DAAE y en caso de ser aprobada se envía la Rendición al departamento de Tesorería y Presupuesto para su posterior revisión.

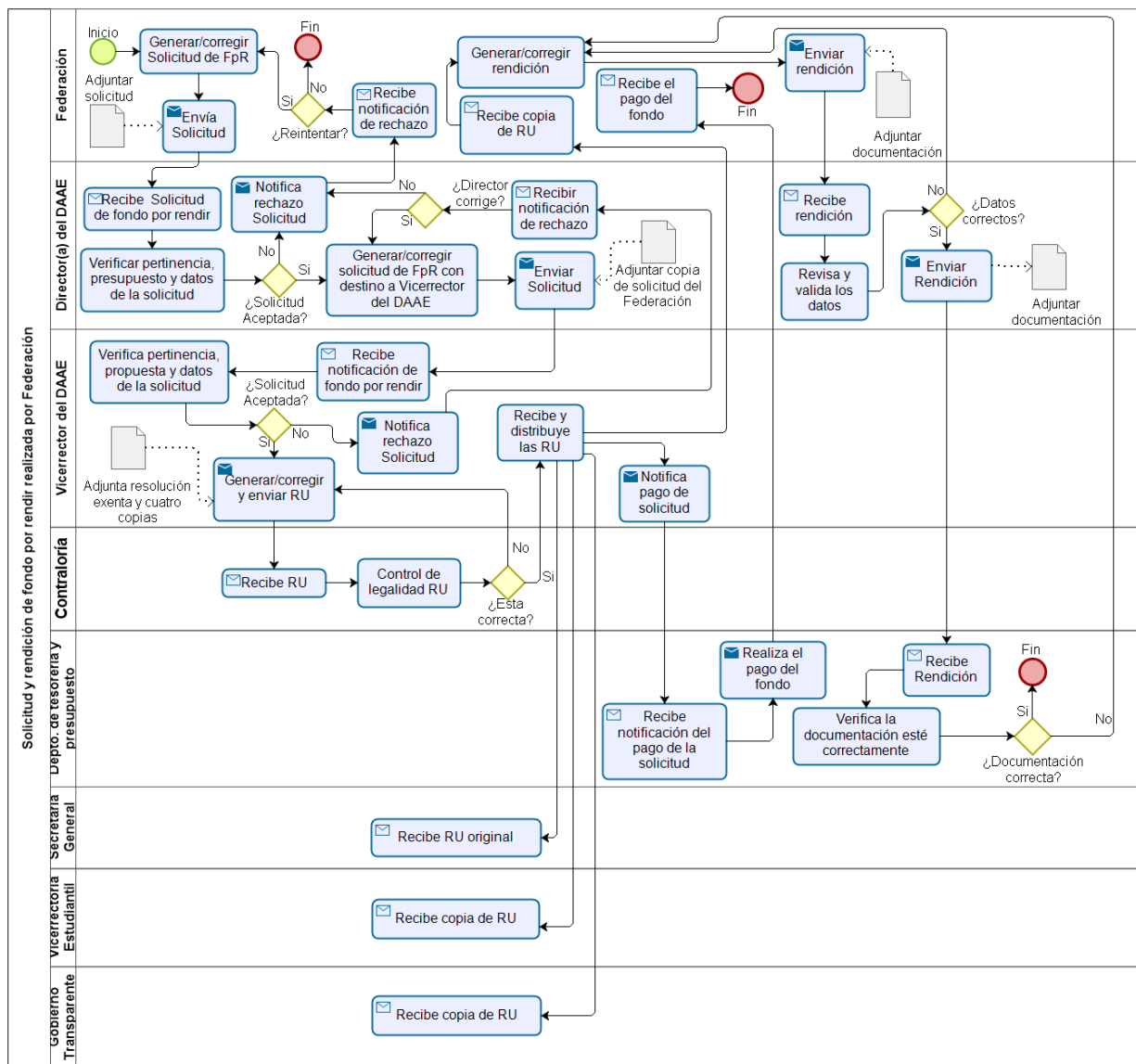


Figura 4.1: Proceso de Fondos por Rendir por parte de Fedeut.

- En caso de ser aprobada, se finaliza el proceso de Solicitud de Fondo por Rendir, en caso contrario, se envía a la OE que realizó la Rendición para ser corregida e iniciar por el proceso de revisión nuevamente hasta cuando sea aprobada.

El proceso que conlleva la solicitud de un Fondo por Rendir por parte de los CCAA que representa la **Figura 4.2** es:

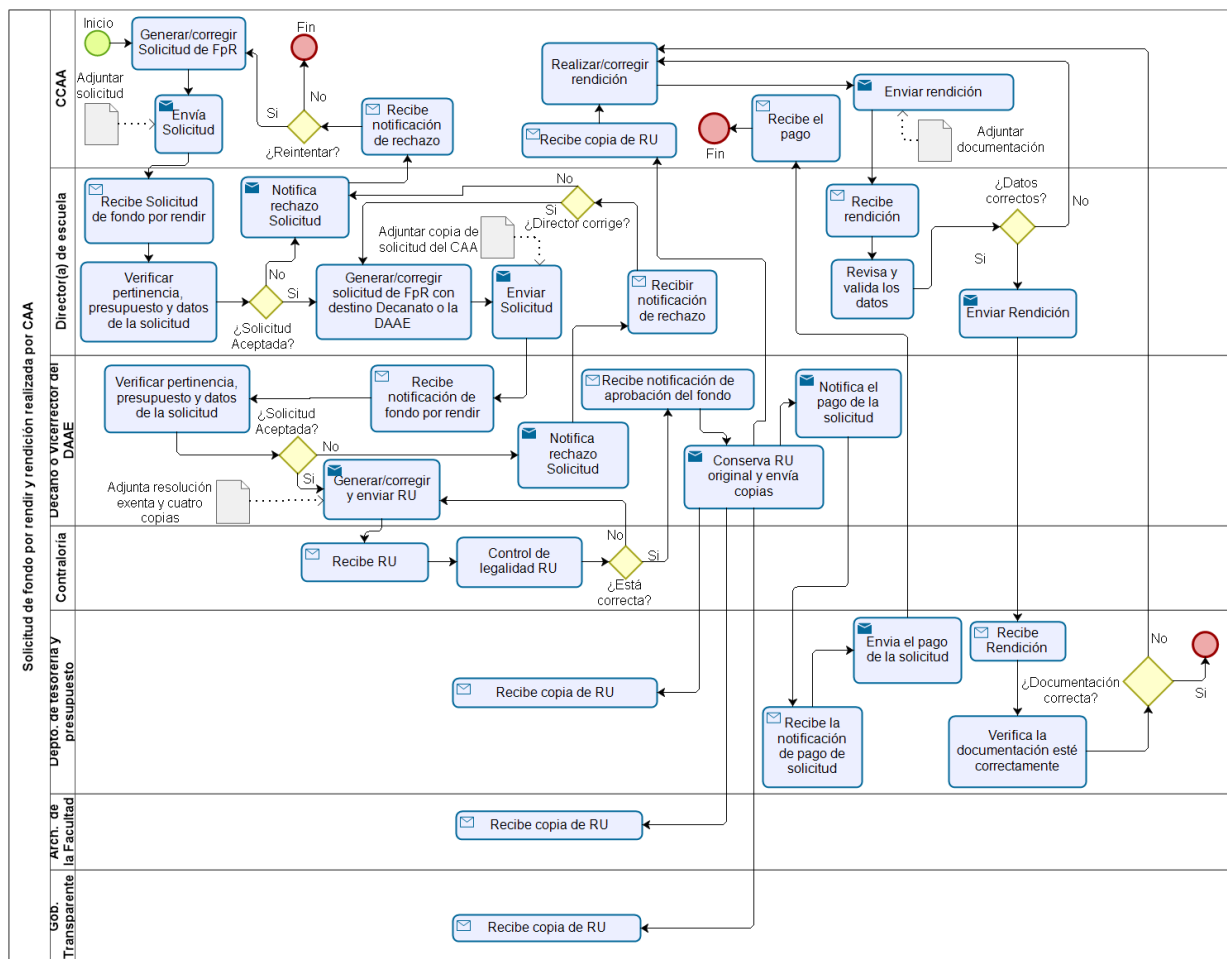


Figura 4.2: Proceso fondos por rendir por parte de un CAA.

1. Elevar una solicitud al Director(a) de Escuela de su respectiva Carrera.
2. Si se acepta la solicitud del CAA por parte de la Dirección de la Escuela, esta debe elevar una solicitud a Decanato junto con una copia de la solicitud realizada por el CAA.
3. En caso de ser aceptada la solicitud por parte de Decanato, este crea la RU de aprobación de la Solicitud junto con cuatro copias para luego ser enviada al departamento de Contraloría, quien se encarga de chequear que el documento esté correctamente realizado.

4. En caso de ser aprobado por Contraloría, se envían nuevamente a Decanato y luego se procede a distribuir los documentos quedando el original en Decanato, y las cuatro copias son enviadas a la OE interesada, al departamento de Tesorería y Presupuesto, al Archivo de la Facultad y Gobierno Transparente.
5. Tras ser aprobada la solicitud y enviada la RU a los departamentos Tesorería genera un Cheque al nombre del responsable del evento y lo envía a la Escuela de la cual proviene la solicitud para ser entregada al respectivo CAA.
6. Tras recibir la RU por parte del CAA, este se encarga de realizar el evento y después comenzar con el proceso de Rendición de la actividad.
7. Una vez finalizada la Rendición, se envía a la Dirección de la Escuela y en caso de ser aprobada se envía la Rendición al departamento de Tesorería y Presupuesto para su posterior revisión.
8. En caso de ser aprobada, se finaliza el proceso de Solicitud de Fondo por Rendir, en caso contrario, se envía a la OE que realizó la Rendición para ser corregida e iniciar por el proceso de revisión nuevamente hasta cuando sea aprobada.

Tal como se aprecia, ambos procesos son similares. En el **Cuadro 4.1** se muestran las diferencias.

Cuadro 4.1: Diferencia en el proceso de Fondos por Rendir entre Fedeut y CCAA.

Fedeut	CCAA
La solicitud de fondos va dirigida a la DAAE.	La solicitud de fondos va dirigida a la Dirección de Escuela.
La solicitud de la DAAE junto con la solicitud de Federación va dirigida al Vicerrectoría de la DAAE.	La solicitud de la Dirección de Escuela junto con la solicitud del CCAA va dirigido a Decanato.
La RU original queda en el archivo de Secretaría General.	La RU original queda en Decanato.

Por su parte, tanto Federación como los CCAA deben cumplir con lo siguiente:

- Ambas OE deben confeccionar una solicitud para inicializar el proceso de fondos por rendir (ver **Sección 2.1**).
- En caso de que la solicitud es aprobada, las OE deben recibir una copia de la RU que aprueba la solicitud.

- Tras obtenida la RU que acredita la aprobación de la solicitud de un fondo por rendir y realizado el evento por el cual se solicita el fondo, se debe confeccionar la rendición de la actividad (ver **Sección 2.1**).

Dado a lo mencionado anteriormente es que se realiza un diagrama de estado, el cual representa el proceso que realiza las OE y por lo tanto el proceso que debe simular la aplicación (ver **Figura 4.3**).

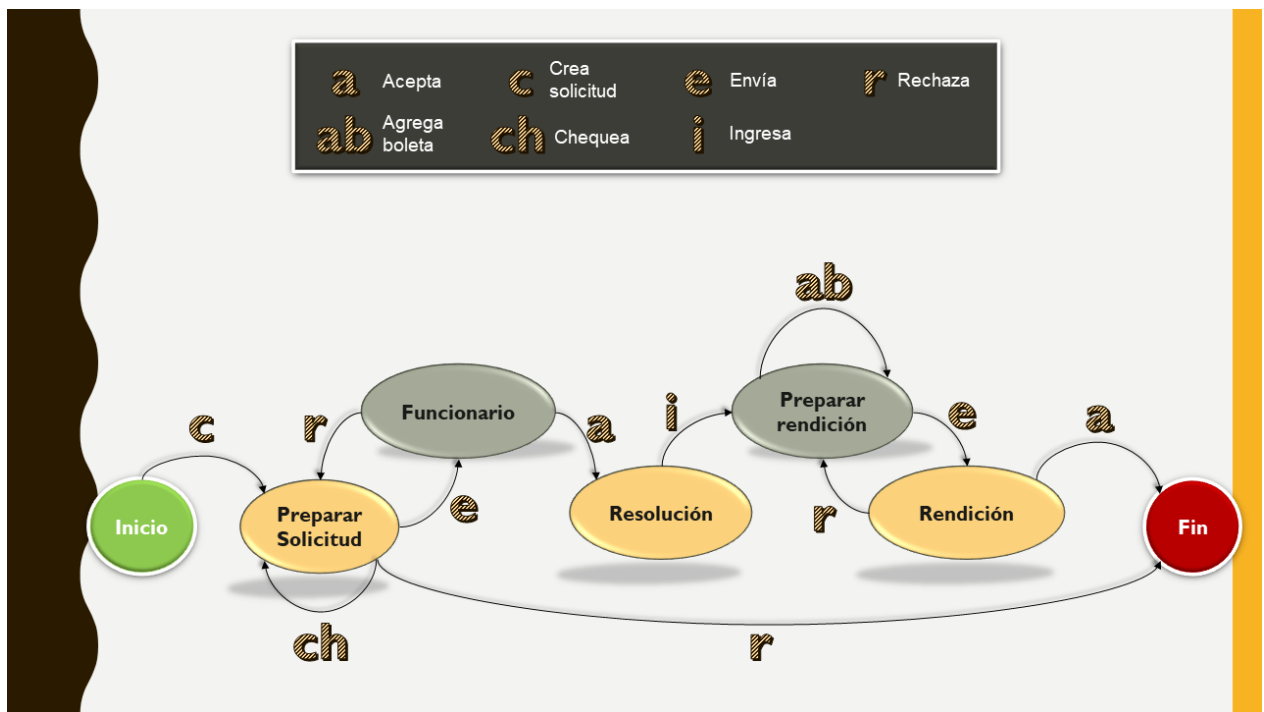


Figura 4.3: Diagrama de estado del proceso que debe simular la aplicación.

El diagrama de estado que se observa en la **Figura 4.3** muestra las actividades que son de responsabilidad exclusiva de la OE, las cuales son la confección de la Solicitud, la recepción de la Resolución emitida por la Universidad y la posterior Rendición del fondo presupestado. El resto de las actividades que se muestran en las **Figuras 4.1** y **4.2** son realizadas por las demás entidades involucradas en el proceso.

A grandes rasgos, el sistema emula el proceso de una Solicitud de Fondo por Rendir desde el punto de vista de la OE. Este proceso conlleva la confección de una Solicitud y su exportación en formato PDF. En caso que la solicitud sea aceptada, la OE recibe la RU respectiva enviada por parte de la Casa de Estudio. Luego, la OE almacena en el sistema una copia digitalizada de la RU respectiva en formato PDF.

Por último, la OE debe confeccionar la Rendición de los gastos efectuados durante el evento y su posterior exportación en formato PDF.

Cabe destacar que no es el único proceso en el cual las OEs realizan declaraciones de gastos. Existe otra forma en la se puede solicitar un Fondo pero con Solicitud de reembolso para las distintas OEs y que se pueden apreciar en las **Figuras 4.4 y 4.5**. Es por ello que en el **Cuadro 4.2** se muestran las diferencias entre un Fondo por Rendir y un Fondo con Solicitud de Reembolso.

Cuadro 4.2: Diferencia en el proceso de Fondos por Rendir entre Fedeut y CCAA.

Fondo por Rendir	Fondo con Solicitud de Reembolso
En caso de ser aprobada la Solicitud, el dinero solicitado llega antes de realizar el evento.	En caso de ser aprobada la Solicitud, la OE debe financiar los gastos al momento de realizar el evento.
Una vez terminada la actividad, se genera el documento en que se declaran los gastos realizado en el evento al cual se le denomina Rendición.	Una vez terminada la actividad, se genera el documento que se declaran los gastos realizados en el evento, pero en este proceso se le denomina Solicitud de Reembolso.
En caso de que en el evento se haya gastado menos de lo presupuestado, se debe devolver el dinero restante a la casa de estudios.	En caso de que en el evento haya gastado menos de lo presupuestado, la Universidad reembolsa el dinero efectivamente gastado.

4.3. Características generales del software

Después conocer los **Aspectos generales** y los **Modelos de contexto de solicitudes**, se procede a presentar los requerimientos del sistema como “Historias de Usuario” las cuales se pueden observar en los **Cuadros 4.3 y 4.4**. Estas fueron obtenidas a través de entrevistas a Presidentes y Secretarios de Finanzas de las respectivas OE, debido a que son los únicos que pueden realizar los procesos de solicitud de fondos por rendir. Además fueron confirmadas por las respectivas Secretarias de Escuelas, ya que ayudan a detectar alguna anomalía en las solicitudes y/o rendiciones antes de ser enviadas a los funcionarios encargados de decidir su aprobación. Cabe destacar que las historias de usuario que se muestra en los **Cuadros 4.3 y 4.4** se redactan según el primer usuario que la menciona, entre el Presidente o el Secretario de Finanzas de las diferentes OE de la Facultad de Ingeniería de la Universidad de Talca.

Cuadro 4.3: Historias de usuario - Secretario de Finanzas

Código	Yo, como Secretario de Finanzas quiero...
HU-SF-1	que sólo se pueda editar una rendición sin llegar a su estado de finalizado dentro del periodo de gracia (20 días contados desde que finaliza el evento), para que no sea rechazada por contraloría por exceso de tiempo.
HU-SF-2	ingresar los gastos realizados en un evento (montos de boletas y/o facturas) para que se calcule el óptimo más cercano al monto solicitado.
HU-SF-3	que para una boleta y/o factura se ingrese la fecha del documento, nombre del proveedor, breve descripción del gasto y monto del gasto para cumplir con lo solicitado con la resolución.
HU-SF-4	ingresar a los participantes de un evento, para detallar los gastos que estos realizaron en el evento (si es el caso).
HU-SF-5	ingresar el rango de tiempo en que ocurre el evento, para no ingresar boletas/facturas que no estén dentro de la fecha.
HU-SF-6	que una vez validada la rendición pueda ser exportada a PDF, para poder ser revisada por el/la Director/a de Escuela o Decano según la OE que corresponda, para poder ser enviada a contraloría.
HU-SF-7	tener un listado de los documentos que debo adjuntar a la documentación requerida en la rendición, en el orden en que se encuentra en esta, para poder chequear que se están adjuntando en el mismo orden.
HU-SF-8	poder editar una rendición en caso de ser rechazada por errores, para poder solucionarlos.
HU-SF-9	que las boletas estén en categorías, para saber en si los gastos son de alojamiento, alimentación, gasto de incorporación e inscripción, etc.
HU-SF-10	guardar los datos de la RU en la cual se aprueba la solicitud de fondos por rendir realizado por la OE, además de guardar una copia digitalizada de esta, para tener un respaldo.

Cuadro 4.4: Historias de usuario - Presidente

Código	Yo, como Presidente quiero...
HU-PDTE-1	tener respaldo de rendiciones anteriores finalizadas, para poder visualizarlas en cualquier instante.
HU-PDTE-2	visualizar y editar rendiciones sin terminar, para ver que cosas faltan y si es posible terminarla.
HU-PDTE-3	visualizar la cantidad de días disponibles para realizar la rendición.
HU-PDTE-4	realizar búsquedas de solicitudes por fecha o por el nombre del evento para poder encontrarlas y verlas en detalle.
HU-PDTE-5	que se valide que las boletas y/o facturas ingresadas estén dentro del periodo del evento, para que no rechacen la rendición.
HU-PDTE-6	adjuntar copias digitales en formato PDF de los documentos solicitados en la rendición (boletas/facturas), para tener un respaldo de lo que se enviará a revisión.
HU-PDTE-7	visualizar el estado de un fondo por rendir, para saber si es factible ejecutar la rendición del fondo.
HU-PDTE-8	visualizar la forma de como realizar una rendición, para saber cómo realizarlas.
HU-PDTE-9	realizar el documento de solicitud de fondos por rendir del evento, para tener un registro de este.
HU-PDTE-10	ingresar en la solicitud el monto del evento, participantes y categorías (alojamiento, alimentación, etc) en el que se utilizará el fondo, encargado del evento y a quien irá dirigida la solicitud, para generar el documento de solicitud de fondos.
HU-PDTE-11	que existan cuentas para federación y centros de alumnos, para poder realizar y visualizar rendiciones.

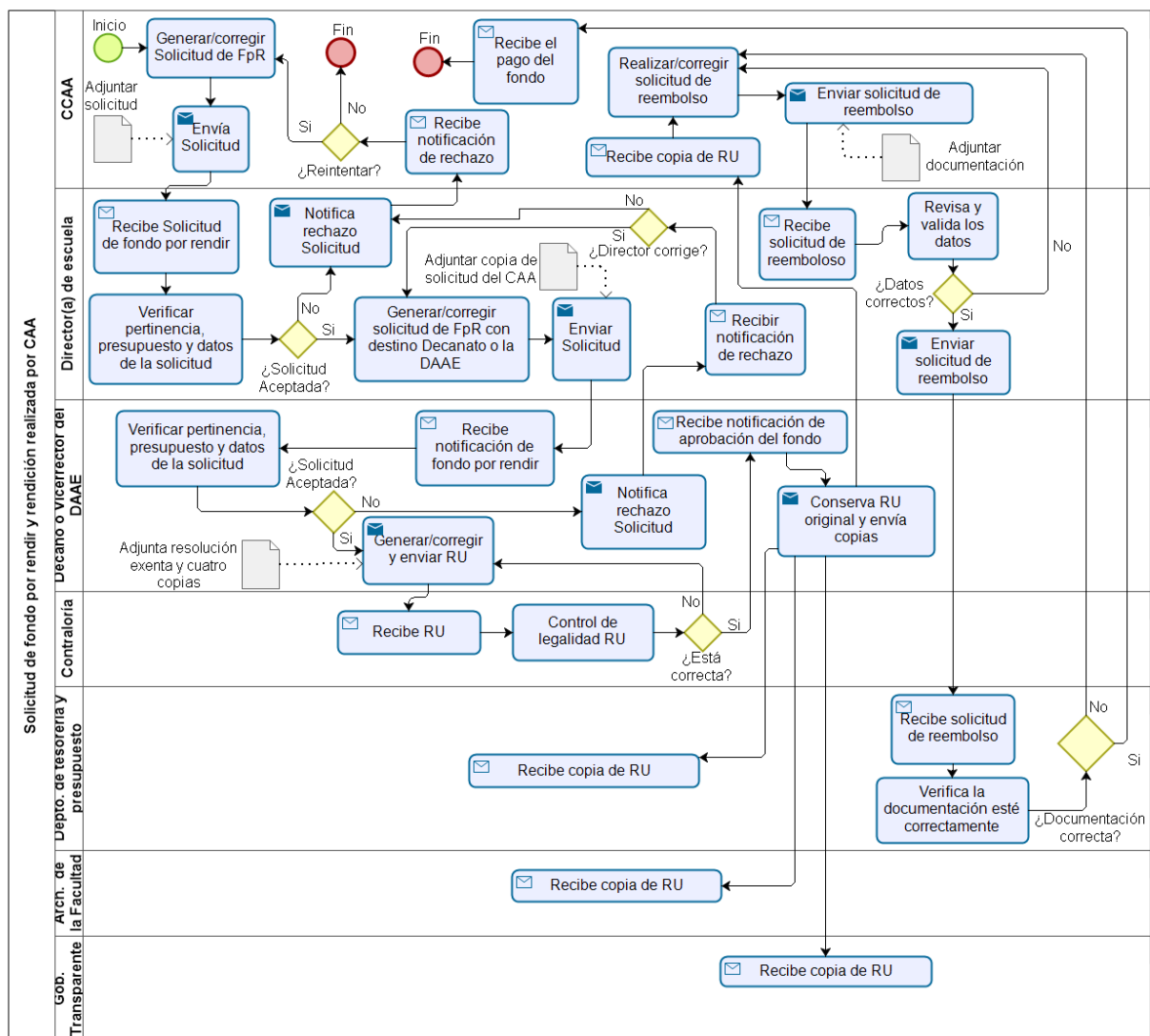


Figura 4.4: Proceso fondos con Solicirud de Reembolso por parte de un CAA.

4.4. Errores frecuentes de la Gestión Manual de Fondos

Luego de conocer el contexto en el que se desenvuelve el problema expuesto en la **Sección 4.2** junto con las historias de usuario expuestas en los **Cuadros 4.3** y **4.4**, se realiza una reflexión acerca de los tipos de errores que existen en el proceso manual de Gestión de Fondos.

A partir de eso, se ha detectado varias fuentes de inconvenientes en el proceso,

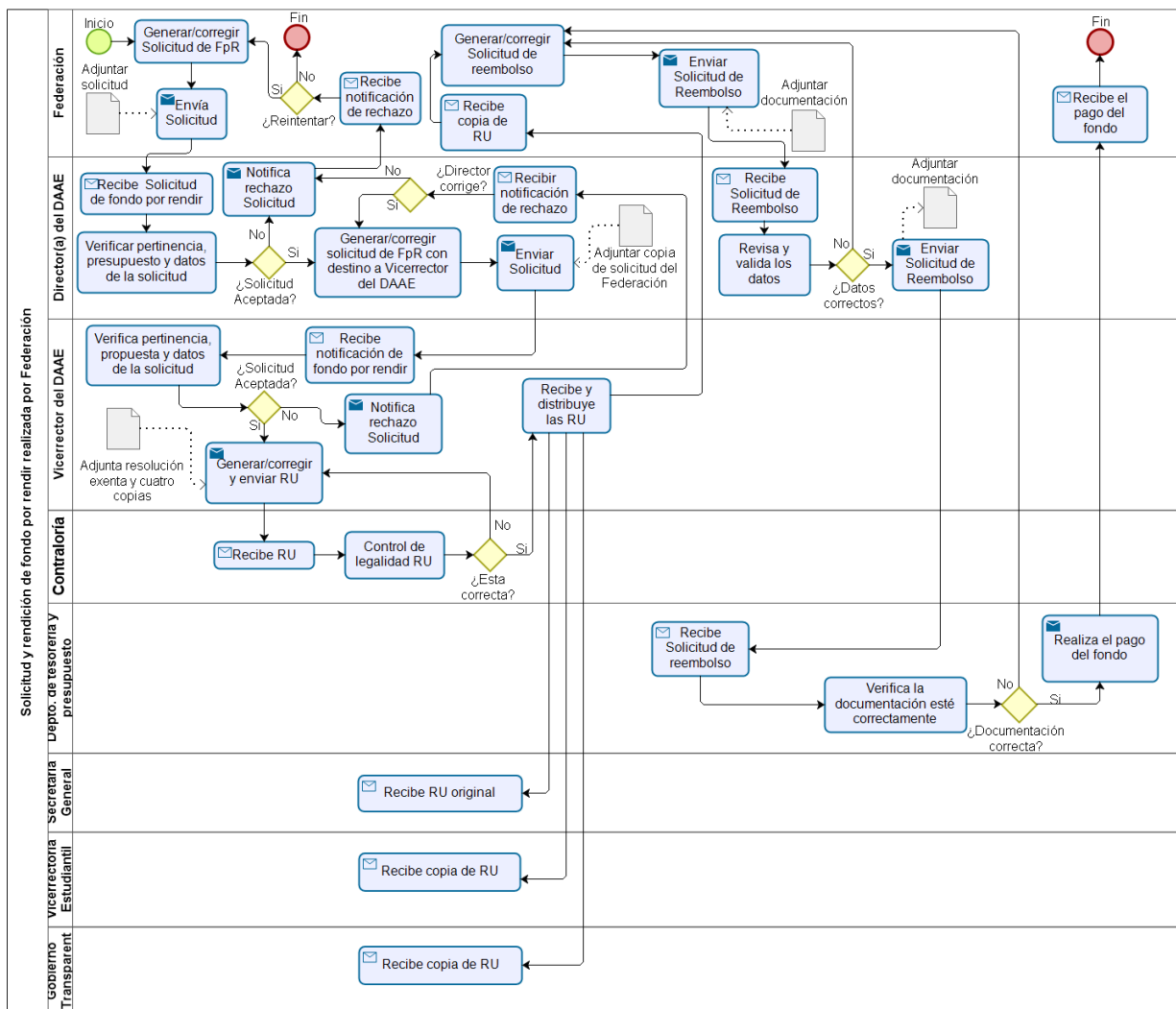


Figura 4.5: Proceso fondos con Solicitud de Reembolso por parte de Federación.

las que se listan a continuación: (I) Rechazo de la declaración de gastos, (II) El monto total de los gastos declarados no es óptimo (en el sentido de que queda dinero disponible que no se utiliza), (III) Asignación incorrecta de gastos, (IV) Declaración de gastos realizada fuera de plazo, (V) Documentación adjuntada sin chequear, (VI) Gastos categorizados incorrectamente, (VII) Falta de respaldo de documentación, (VIII) Formato incorrecto de la declaración de gastos y (IX) Desconocimiento del tipo de declaración de gasto (Solicitud de reembolso/Rendición de gastos).

Por último, el detalle exhaustivo de errores que se considera prevenir con el sistema desarrollado en este trabajo se muestra en el **Cuadro 4.5**, con su respectivo código.

Cuadro 4.5: Errores frecuentes del proceso de Solicitud de Fondos

Código	Problema
PBLM-1	Rendición o Solicitud de Reembolso rechazada por realizarla fuera de plazo de 20 días corridos.
PBLM-2	Suma de boletas y/o facturas no optimizada.
PBLM-3	Asignar un gasto a un participante que no esté en la solicitud.
PBLM-4	Documentación realizada fuera del periodo en que ocurre el evento.
PBLM-5	Falta de chequeo de los documentos que se deben adjuntar en la Rendición/Solicitud de reembolso.
PBLM-6	Rendir gastos fuera de las categorías expuestas en la Solicitud.
PBLM-7	Ausencia de respaldo de la RU y del documento de declaración de gastos de los Fondos Solicitados.
PBLM-8	No respetar el formato de Rendiciones/Solicitud de reembolso.
PBLM-9	No indicar el tipo de declaración de gastos, por ejemplo si es una Solicitud de reembolso o es una Rendición de gastos.

4.5. Planificación de las iteraciones

Tras realizar el estudio previo de la situación actual de la problemática y conocidas las historias de usuario, se procede a implementar la segmentación del proyecto enunciada en la **Sección 3.2.2.2**, para luego planificar el desarrollo de la aplicación. Es por ello que el **Cuadro 4.6** muestra cómo se distribuyen las historias de usuarios a través de las diferentes iteraciones.

Cuadro 4.6: Planificación de las iteraciones.

Iteración	Historia de usuario
1. Solicitud	HU-PDTE-10
	HU-SF-5
	HU-SF-4
2. Solicitud y Resolución	HU-PDTE-9
	HU-SF-10
3. Búsqueda	HU-PDTE-4
4. Rendición	HU-PDTE-3
	HU-PDTE-2
	HU-SF-7
	HU-SF-6
	HU-SF-8
	HU-SF-3
	HU-SF-2
	HU-PDTE-5
	HU-SF-9
5. Usuario	HU-PDTE-11
6. Documentación y Panel de Ayuda	HU-PDTE-6
	HU-PDTE-8
7. Detalles	HU-PDTE-1
	HU-SF-1
	HU-PDTE-7

5. Diseño de la aplicación

Después de realizar el estudio del contexto del problema, conocer los aspectos generales de la aplicación y junto con ello saber las historias de usuarios, las cuales se dieron a conocer en el **Capítulo 4**, se procede a realizar el diseño de la aplicación. Éste busca definir el mapa de navegación, las interfaces de usuario, la arquitectura, el diagrama de clases, el modelo de datos y el algoritmo que se utiliza seleccionar los documentos que se encuentran en la Rendición. Esto se realiza con el fin de buscar una solución simple y rápida que satisfaga las necesidades del usuario de una forma comprensible para el desarrollo de la aplicación.

5.1. Mapa de navegación

Para comenzar con el diseño de la aplicación y tras conocer el dominio de esta, se procede a realizar el mapa de navegación, el cual se puede observar en la **Figura 5.1**. Este mapa consiste en una representación resumida del sitio web que además permite especificar las “pulsaciones” que un usuario debe realizar para que alcance una vista en específico.

A continuación se describen las principales vistas que tiene la aplicación:

Buscar solicitudes de fondos: Vista que ayuda al usuario a realizar la búsqueda de solicitudes realizadas a través del nombre del evento o por fechas.

Nueva solicitud de fondo: Vista que ayuda al usuario a inicializar un proceso de fondo por rendir. En esta fase se realiza la solicitud del evento con toda la información requerida y va dirigida a la Dirección de Escuela o DAAE según sea el caso (ver **Sección 2.1**).

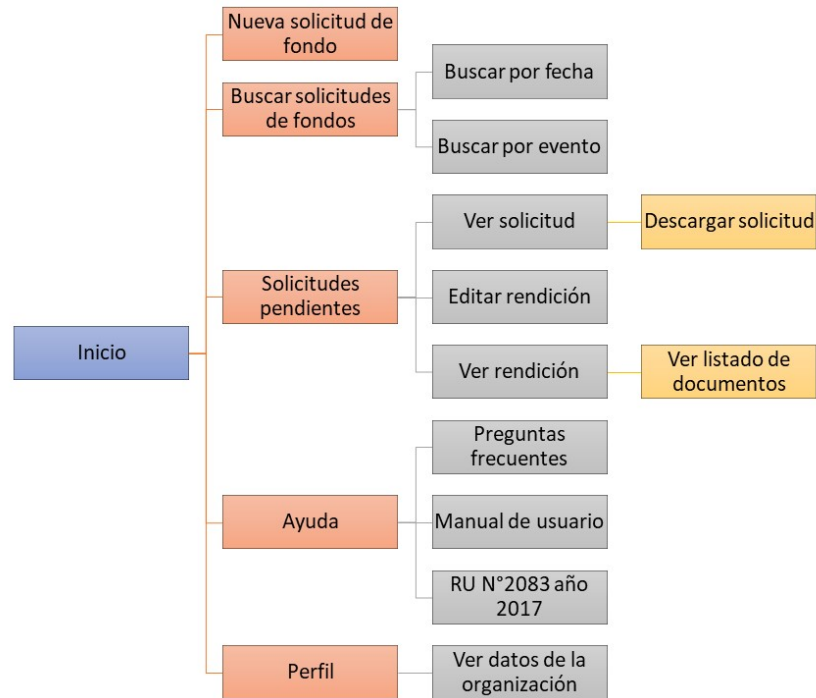


Figura 5.1: Mapa de navegación de la aplicación.

Solicitudes pendientes: Vista que permite al usuario acceder rápidamente a las solicitudes que no están finalizadas. Dependiendo del estado en que se encuentre el fondo por rendir es lo que podrá acceder el usuario. Por ejemplo, si la Solicitud no ha sido aprobada y no se ingresa los datos de la RU que corresponde a la Solicitud, no se puede acceder a realizar la rendición del evento.

Ayuda: Vista en donde se encuentra toda la documentación que requiere el usuario para realizar un proceso de fondos por rendir.

Perfil: Vista que muestra toda la información correspondiente a una OE junto con los datos de la Dirección encargada de la validación de solicitudes y rendiciones de un evento antes de ser enviada a funcionarios superiores.

5.2. Interfaces de usuario

Luego de diseñar el mapa de navegación para conocer el dominio de la aplicación se procede a estandarizar la interfaz de usuario, confeccionando la línea de diseño.

Es por ello que en la **Figura 5.2** se aprecia la posición de los elementos de la vista principal de la aplicación. En la parte superior se encuentra el encabezado y dentro de este último se ubica el logotipo, el botón de perfil de usuario, el botón de ayuda y el botón para cerrar sesión. Debajo del encabezado se encuentra el menú, donde van los íconos para que el usuario tenga el acceso a los distintos módulos de la aplicación. Luego se encuentra el contenido, el cual puede variar según la opción que el usuario seleccione en el menú. Por último, se puede observar el *footer* (pié de página) de la aplicación.

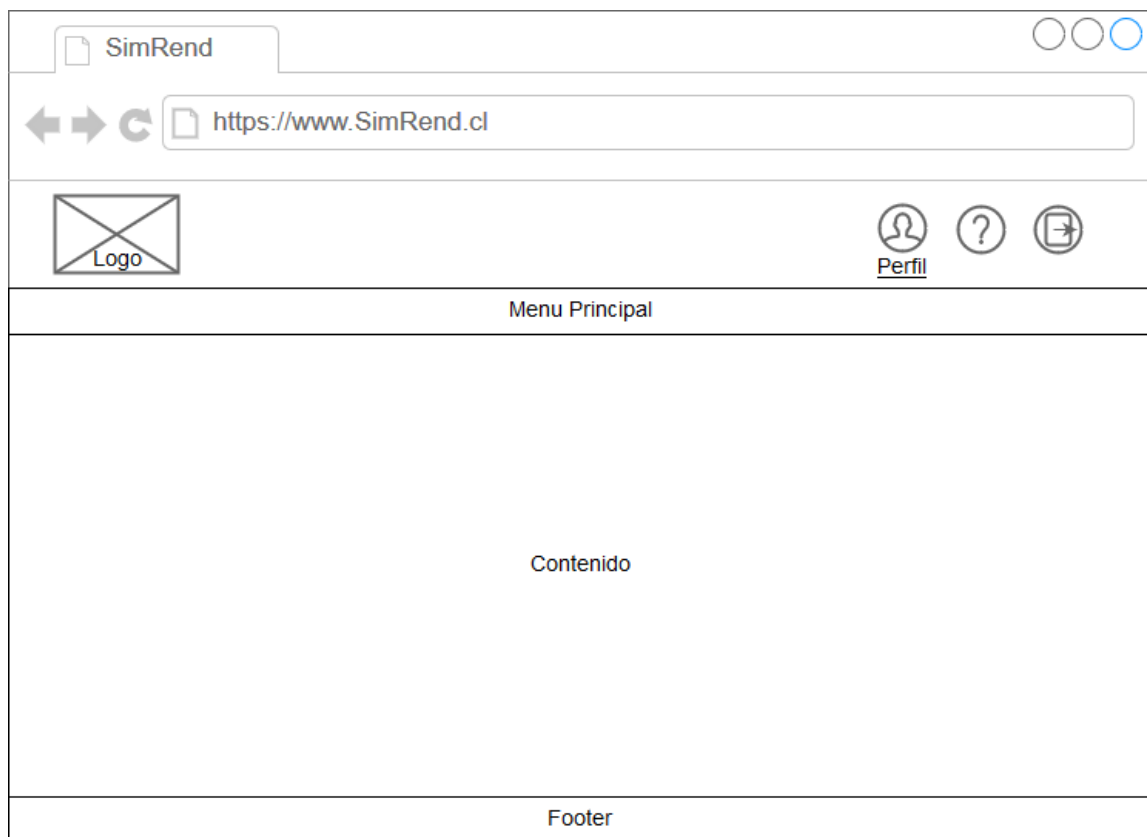


Figura 5.2: Línea de diseño de la aplicación que representa el estándar de los elementos de interfaz.

5.3. Arquitectura de la aplicación

Tras conocer cómo es el funcionamiento del sistema y las distintas vistas que tiene, es que se procede a identificar cómo el sistema interactúa con el cliente tanto

de forma física como lógica. Lo anteriormente mencionado se ve a través del diseño de la Arquitectura del sistema, lo cual se detalla a continuación.

5.3.1. Arquitectura física

Como se puede observar en la **Figura 5.3**, la arquitectura utilizada para el sistema es de tres capas: cliente, servidor y base de datos. Analizando la arquitectura, por el lado de cliente se puede apreciar que sólo se está diseñando para que el sistema sea utilizado a través de un PC o Laptop. En cuanto al servidor, se puede destacar que el sistema está montado en un servidor diferente al de la base de datos con el fin de que los datos no vean comprometidos en caso de que se vulnere el servidor web.

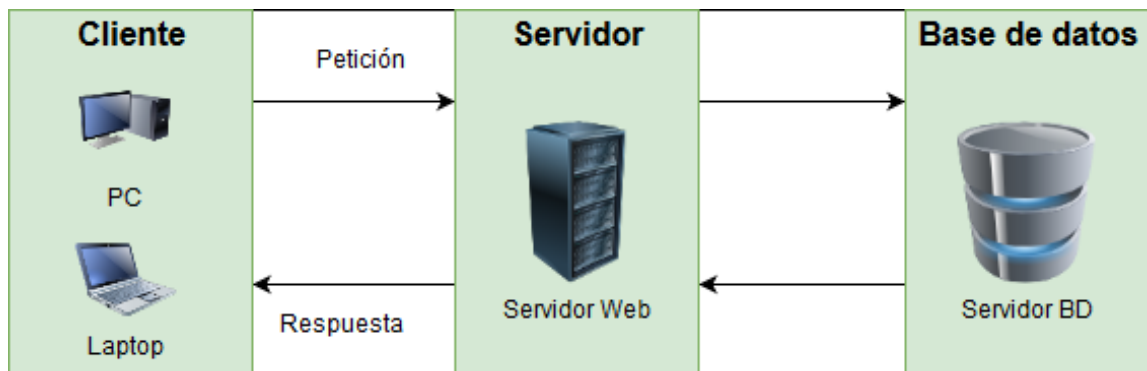


Figura 5.3: Arquitectura física del sistema.

5.3.2. Arquitectura lógica

Si bien se ha mencionado cómo es la arquitectura física que utiliza el sistema, en esta sección se habla acerca de su arquitectura lógica. Para esto se utiliza el **Modelo-Vista-Controlador (MVC)**. Esta arquitectura separa la interfaz de usuario de la lógica de negocio y el modelo de datos del sistema. Esto reduce el acoplamiento y facilita los futuros mantenimientos dado que si se requiere de alguna modificación de una capa, las demás no se verán afectadas o en el caso de que se requiera el reemplazo completo de una capa, sólo se debe revisar las conexiones que tienen las demás capas a la que se está modificando para que funcione correctamente. Por ejemplo, cuando se requiere migrar la Base de datos, sólo se debe reemplazar los datos de conexión que se encuentran en la capa de Modelo y las otras capas no se verán afectadas.

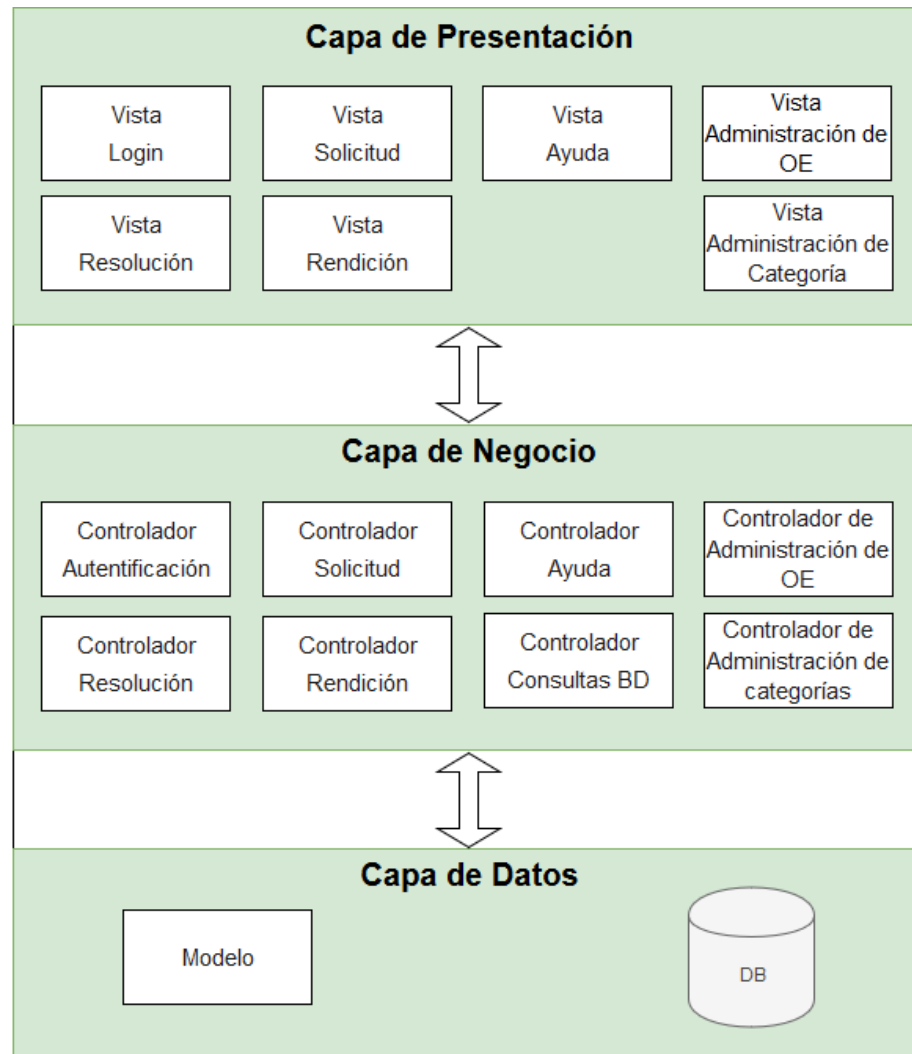


Figura 5.4: Arquitectura lógica del sistema.

La **Figura 5.3.2** muestra la arquitectura lógica de la aplicación. A continuación se explica cada capa.

- Capa de Presentación:** Esta capa contempla todas las interfaces de usuario o vistas que posee el sistema. Cabe destacar que existen dos roles en el sistema, el de usuario común y el de administrador. Es por ello que la distribución de las vistas según el rol queda de la siguiente manera: (I) Vista Solicitud, (II) Vista Resolución y (III) Vista Rendición son exclusivamente del usuario común mientras que (IV) Vista Administración OE y (v) Vista Administración de Categoría son exclusivos del rol del administrador del sistema. Por otra

parte, existen vistas en que pueden interactuar ambos usuarios las cuales son: (vi) Vista Ayuda y (vii) Vista Login.

- **Capa de Negocio:** Esta capa contiene todos los controladores que la aplicación posee y es la capa intermediaria entre la Capa de Presentación y la Capa de Datos. A continuación, se procede a explicar cada uno de los controladores:
 - **Controlador Autenticación:** Encargado de obtener y corroborar los datos de acceso de los distintos usuarios.
 - **Controlador Solicitud:** Encargado de obtener y manipular los datos para confeccionar una solicitud que ingresa la OE interesada para enviarlos al Controlador Consultas BD y así sean registrados.
 - **Controlador Resolución:** Encargado de obtener y manipular los datos de la Resolución enviada por la Institución que acredita la aceptación de la Solicitud realizada por la OE.
 - **Controlador Rendición:** Encargado de obtener y manipular todos los datos que conlleva una Rendición, entre los cuales se considera los datos de las boletas y/o facturas y datos del encargado del evento. Todo esto con el fin de calcular el monto cercano o igual al solicitado por la OE para luego obtener el documento de rendición y a su vez enviar los datos al Controlador Consultas BD para que registre los datos ingresados.
 - **Controlador Ayuda:** Encargado de obtener la información de manuales y preguntas frecuentes del sistema, entre otros, para enviarlos a la vista de ayuda que el usuario solicita. Este controlador no obtiene nada desde la vista de usuario, sólo obtiene la información que se encuentra en el servidor.
 - **Controlador de Administración de OE:** Encargado de crear nuevas cuentas para las nuevas OE y modificar información de las ya existentes.
 - **Controlador de Administración de Categorías:** Encargado de gestionar nuevas categorías que serán utilizadas en la Solicitud y en la Rendición para mencionar en que fueron realizados los gastos.
 - **Controlador Consultas BD:** Encargado de obtener los datos enviados desde los controladores para reenviarlos a la BD. Además, es el encargado

de realizar otras consultas que son solicitadas por los controladores para obtener datos de la BD.

- **Capa de Datos:** Esta capa contempla el Modelo de Datos de la aplicación y las conexiones hacia la Base de datos.

5.4. Modelo de datos

Tras reconocer los diferentes conceptos para la comprensión de la problemática mencionado en la **Sección 2.1** y las relaciones que existen entre ellos, es que se realiza un diagrama de clases (ver **Figura 5.5**) para representar el Modelo de Datos. Es por ello que a continuación se procede a detallar brevemente cada clase que compone el diagrama:

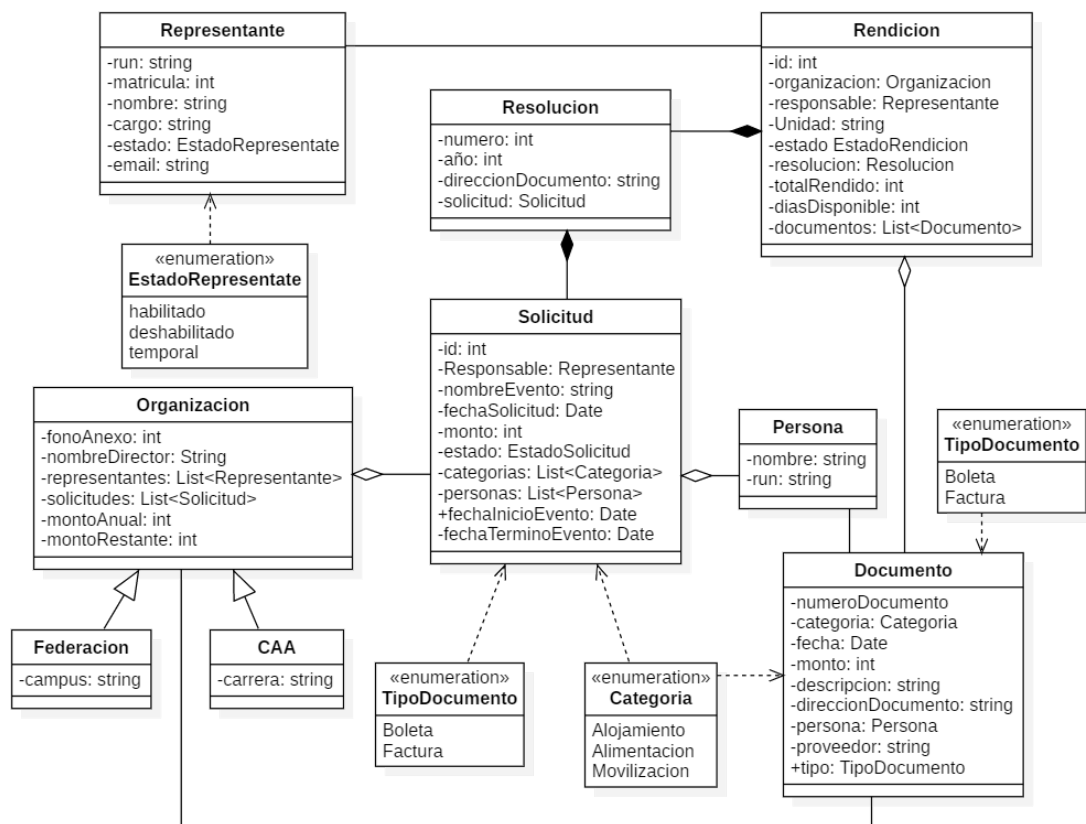


Figura 5.5: Diagrama de clases que detalla el Modelo de Datos.

- **Solicitud:** Representa la solicitud de un Fondo por Rendir por parte de la OE interesada.
- **Persona:** Representa los datos de los participantes de un evento (en caso de que el fondo solicitado sea para una o más personas) y son guardados en un listado en la clase Solicitud.
- **Resolución:** Representa la RU que acepta la solicitud y que es emitida por la Casa de Estudio. Principalmente contiene atributos tales como el número de RU y el año de emisión.
- **Rendición:** Representa la declaración de gastos que se incurrieron en un evento solicitado por la OE. Principalmente contiene los datos de la organización, representante, el número de RU y un listado de documentos que contienen los datos de una boleta y/o factura.
- **Documento:** Representa los datos que contiene una boleta y/o factura, los cuales son guardados en una lista en la clase Rendición.
- **Organización:** Representa los datos que contiene una OE de forma general. Presenta atributos tales como nombre del director a quien va dirigida las Solicitudes que realiza la OE, un listado de los Representantes y un listado de Solicitudes realizadas, entre otros.
- **Federación:** Representa a un usuario de tipo Federación y hereda de la clase **Organización**, por lo que también posee atributos que identifican a una organización. Además, contiene un atributo en específico que es campus al cual representa la Federación.
- **CAA:** Representa a un usuario de tipo CAA y hereda de la clase **Organización**, por lo que también tiene atributos que identifican a una organización. A diferencia de la clase Federación, esta clase contiene el atributo carrera para indicar de dónde proviene este tipo de OE.
- **Representante:** Especifica a uno de los Representantes que pertenece a la organización, el cual puede ser de dos tipos: Presidente(a) o Secretario(a) de Finanzas. Estos se encuentran en una lista que se encuentra en la clase Organización.

5.5. Modelo Relacional

La base de datos que representa la **Figura 5.6** es de tipo relacional, la cual debe contar las tablas y sus respectivas claves primarias, relaciones y restricciones. Este tipo de base de datos puede ser tratada como si estuviera orientada a objetos. Su diseño varía de acuerdo a los cambios que se realizan en el Modelo de Datos.

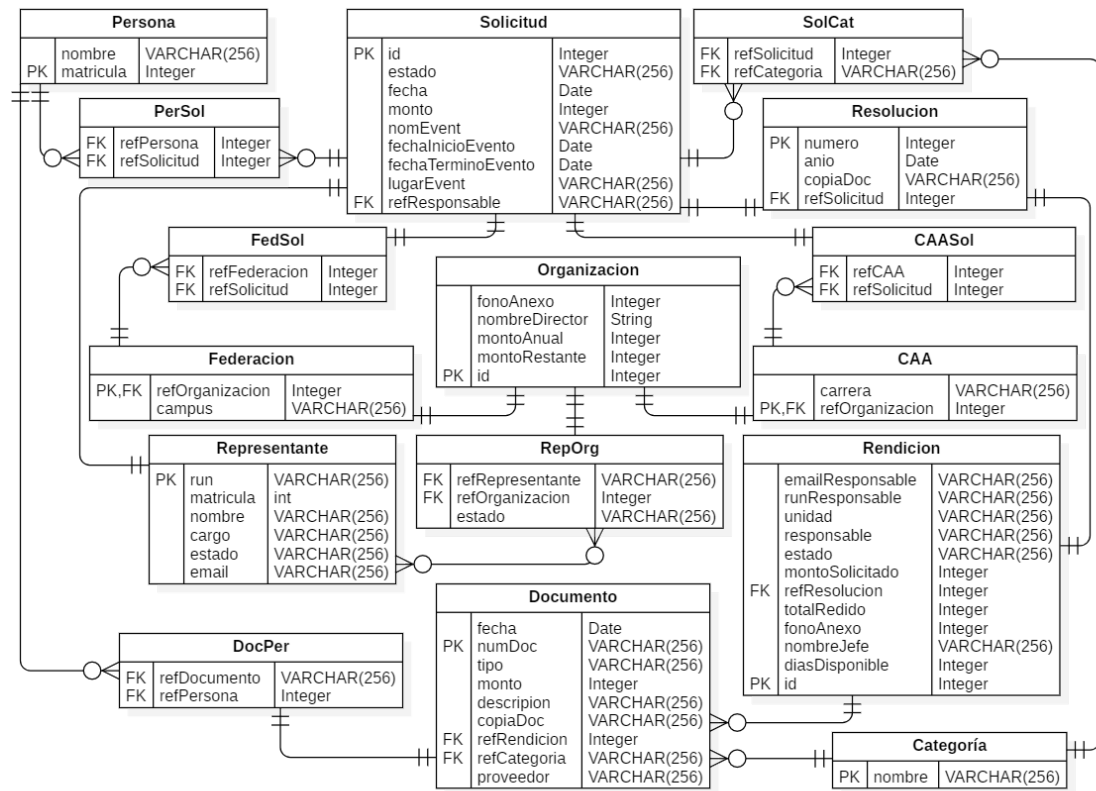


Figura 5.6: Modelo Relacional de la Base de Datos del sistema.

Por otra parte, se guardan cada Categoría que aparecen en el sistema para ser utilizadas en las solicitudes y especificar en que llevaron a cabo los gastos. En cuento a las listas que se realizan en el modelo de datos, se registran en la BD a través de tablas intermedias. Por último, se realizan procedimientos almacenados para el registro de los datos que provienen desde la capa de negocio.

5.6. Algoritmo de solución

Uno de los grandes problemas que se pretende resolver en esta memoria es, disminuir el tiempo en escoger la cantidad de documentación y que se ajuste al monto solicitado.

Inicialmente, se tiende a resolver esta problemática por Fuerza Bruta [20], el cual busca todas las posibles combinaciones y obtiene la mejor opción. Para ello tenemos lo siguiente:

- Dado que hay n elementos, puede haber 2^n posibles combinaciones.
- Luego de tener todas las combinaciones, se deben encontrar todas aquellas que su peso sea menor a la capacidad W de la mochila y que maximice el valor o beneficio.

Sin embargo, el algoritmo que se diseñe para cumplir los puntos mencionados con anterioridad debe tener un tiempo de ejecución de $O(2^n)$ para obtener la solución óptima. Se sabe que, cualquier algoritmo que tenga tiempo mayor al polinomial (como es en este caso que es exponencial) no es posible implementarlo para problemas que tengan grandes valores de n .

Otra solución es resolver el problema con programación dinámica para mejorar su desempeño y dado que, tiene un parecido al problema de la mochila [20] el cual dice, “Dada una serie de elementos (cada uno con un valor p y un peso w), empacar una mochila que soporta un peso máximo W con la finalidad de maximizar el valor de los elementos empacados”. Existen dos versiones del problema:

- **0-1:** Los elementos no se pueden dividir y solo pueden colocar (1) o no colocar (0) dentro de la mochila.
- **Fraccional:** Se pueden tomar valores se pueden dividir para llenar la mochila.

Ahora, dado que los documentos en una Rendición no se pueden dividir, esta memoria se centra en la versión del problema 0-1 y, para definir el problema de una manera mas formal, se puede decir que:

- Se tiene una mochila con capacidad máxima W , y un conjunto S de n elementos.

- Cada elemento i tiene un peso w_i y un valor o beneficio b_i (w_i y W son de tipo entero para acercarlo a la problemática de selección de documentación de la Rendición).
- **Problema:** ¿Qué elementos guardar en la mochila para obtener un máximo valor o beneficio total?

Para ello se tiene que identificar la ecuación recursiva que asocie el problema con los subproblemas (principio de la optimalidad) que es la siguiente:

$$V[i, w] = \begin{cases} V[i - 1, w] & \text{si } w_i > w \\ \max(V[i - 1], V[i - 1, w - w_i] + b_i) & \text{si } w_i \leq w \end{cases} \quad (5.1)$$

Nota: $V[i, w]$ representa el valor máximo de la mochila con capacidad w después de combinar los primeros elementos i

Esto significa que, el mejor subconjunto S_i que tiene un peso total w es:

1. El mejor subconjunto S_{i-1} que tiene un peso total menor o igual a w .
2. El mejor subconjunto S_{i-1} que tiene un peso total menor o igual a $w - w_i$ más el valor o beneficio del elemento i .

Dicho de otra forma, tenemos lo siguiente:

- El mejor subconjunto S_i que tiene peso total menor o igual a w , contiene el elemento i o no.
- Primer caso ($w_i > w$): El elemento i no puede ser parte de la solución, porque si lo fuera, el peso total seria mayor que w , lo cual no es aceptable.
- Segundo caso ($w_i \leq w$): El elemento i puede estar en la solución, y escogemos el caso con el mayor valor.

Dado lo anterior, se implementa el **Algoritmo 1** para resolver el problema con un tiempo de ejecución $O(n * W)$ que podemos compararlo como $O(n^2)$, dado que es la multiplicación de dos variables y es polinomial, cumpliendo así con el principio de la optimalidad.

Tras lo expuesto, se aprecia que este algoritmo se ajusta mucho al problema que plantea resolver esta memoria, en donde la capacidad máxima que tiene la mochila

Algorithm 1 Programación dinamica - *Algoritmo de la mochila 0-1*

Entrada: *documentos*: lista de documentos que serán evaluado, *monto*: cantidad de dinero límite que se puede rendir

Salida: *docsOrdenados*: listado de documentos ordenados (con estado 0 o 1, no seleccionado o seleccionado)

```

1:  $W \leftarrow \text{monto}$ 
2:  $\text{docsOrdenados} \leftarrow \text{sort}(\text{documentos})$ 
3:  $n \leftarrow |\text{docsOrdenados}|$ 
4:  $V \leftarrow \text{new int}[n + 1, W + 1]$ 

5: for  $i \leftarrow 0$ ;  $i \leq W$ ;  $i++$  do
6:    $V[0, i] \leftarrow 0$ 

7: for  $i \leftarrow 0$ ;  $i \leq n$ ;  $i++$  do
8:    $V[i, 0] \leftarrow 0$ 

9: for  $i \leftarrow 1$ ;  $i \leq n$ ;  $i++$  do
10:  for  $w \leftarrow 1$ ;  $w \leq W$ ;  $w++$  do
11:     $w_i \leftarrow \text{docsOrdenados}[i - 1].\text{monto}$ 
12:     $b_i \leftarrow \text{docsOrdenados}[i - 1].\text{monto}$ 
13:    if  $w_i \leq w$  then
14:      if  $(b_i + V[i - 1, w - w_i]) > V[i - 1, w]$  then
15:         $V[i, w] \leftarrow b_i + V[i - 1, w - w_i]$ 
16:      else
17:         $V[i, w] \leftarrow V[i - 1, w]$ 
18:    else
19:       $V[i, w] \leftarrow V[i - 1, w]$ 

20: seleccionarDocumentos( $n, W, V, \text{docsOrdenados}$ );
21: return docsOrdenados

```

(W) , es el monto que se solicita para desarrollar una actividad y, los elementos (i) , son los documentos que se deben evaluar para comprobar si se pueden colocar en la mochila, según su monto $(w_i$ y $b_i)$.

Sin embargo, el algoritmo solo encuentra el beneficio máximo posible que se puede cargar en la mochila (el valor que se encuentra en $V[n, W]$), es por ello que, para conocer los elementos que maximizan el valor, se implementa el **Algoritmo 2**. Dado que, toda la información se encuentra en la matriz (V) y se conoce el máximo beneficio $(V[n, W])$ el algoritmo implementa las siguientes condiciones para la selección de documentos que debe ir en la Rendición:

- Sea $i = n$ y $k = W$
- Mientras i y k sean mayores a cero, se debe realizar lo siguiente:
 - Si $V[i, k]$ es distinto de $V[i - 1, k]$ entonces marca el i -ésimo documento como presente en la Rendición y los se modifican de la siguiente manera:
 $i = i - 1, k = k - w_i$
 - Sino, solo se modifica el valor i de la siguiente forma: $i = i - 1$. Lo anterior asume que, el documento no se debe considerar en la Rendición.

Algorithm 2 Selección de documentos

Entrada: n : es la cantidad de documentos, W : cantidad de dinero límite que se puede rendir, V : es la matriz que representa el valor máximo de la mochila con capacidad w después de combinar los primeros elementos i , $documentos$: lista de documentos ordenados de menor a mayor según su monto

```

1:  $i \leftarrow n$ 
2:  $k \leftarrow W$ 

3: while  $i > 0$  and  $k > 0$  do
4:   if  $V[i, k] \neq V[i - 1, k]$  then
5:      $documentos[i - 1].Estado \leftarrow 1$ 
6:      $k \leftarrow k - documentos[i - 1].Monto$ 
7:    $i \leftarrow i - 1$ 

```

6. Construcción y validación

Tras tener el diseño de la aplicación y la planificación general de proyecto, se da comienzo a la construcción de la aplicación, la cual se detalla en el presente capítulo. Además, se aborda los resultados obtenidos al evaluar el sistema por las distintas OE, los cuales son obtenidos a través de una encuesta.

6.1. Herramientas Utilizadas

Para el desarrollo de la aplicación se debe conocer las herramientas que ayudan a dar vida al proyecto. Es por ello que a continuación detalla el entorno de trabajo en el cual se desenvuelven la construcción y pruebas de la aplicación, los lenguajes de programación y el motor junto con la aplicación para base de datos.

Entorno de trabajo: La estación de trabajo utilizada para la construcción de la aplicación corresponde a un computador que cuenta con un procesador Intel CORE i7 de 2.80GHz, 12Gb de RAM y con conexión a Internet. El sistema operativo es Windows 10 Home. Esta misma estación se utiliza para realizar las pruebas.

En cuanto a los navegadores que se usa para realizar las pruebas se consideran: Mozilla Firefox Quantum 67.0.1 (64-bit) e Internet Explorer. No obstante lo anterior, el navegador recomendado es Mozilla Firefox.

Entorno y lenguajes de programación: Dado a que SimRend es una aplicación Web, se utiliza el framework ASP.NET Core y junto con ello se utilizan los lenguajes C#, JavaScript, HTML y CSS (ver **Sección 2.2**). En primera instancia se utilizó el IDE Visual Studio, pero debido a que hubo un conflicto con el uso de la librería DinkToPDF se migró a VSCode.

La estructura de SimRend está hecha en HTML y el diseño es mejorado utilizando hojas de estilos (CSS), junto con la ayuda de JavaScript, que permite controlar algunos eventos en el sistema. En cuanto a la capa de negocio y el modelo de datos se utiliza C#. Todo lo anterior se une gracias a la arquitectura lógica del proyecto la cual es MVC, que es detallada en la **Sección 5.3.2**.

Motor y aplicación para base de datos: Como motor de base de datos relacional se utiliza MySQL, el cual posee una gran estabilidad y confiabilidad además de fiabilidad e integridad de los datos. Para el diseño del modelo relacional se utiliza StarUML y para la administración de la base de datos se usa Navicat.

6.2. Aplicación Obtenida

A continuación se describe el producto obtenido, siguiendo las diferentes fases de la metodología de trabajo y las herramientas descritas con anterioridad. Además, se indica el código de los inconvenientes que se resuelven expuestos en la **Sección 4.4**.

Autenticación de un usuario en el sistema: Solicitud de acceso, utilizando credenciales de acceso los cuales son ingresados por el usuario. Si las credenciales son válidas, se redirecciona a las interfaces de la OE interesada o a la del Administrador (vea **Figura 6.1**).

Recuperación de acceso a una cuenta de una OE: Solicitud de una nueva clave de acceso en caso de que el usuario la olvide, ingresando el correo su cuenta registrada. Una vez validada esta información se solicita la recuperación de cuenta al servidor, el cual envía un correo con la nueva clave de acceso al usuario (vea **Figura 6.2**).

Búsqueda de solicitudes: En esta etapa de la aplicación se encuentran todas las solicitudes realizadas por la OE, en donde el usuario puede visualizar los resúmenes de la solicitudes y tener la opción de ver en detalle cada una de ellas. Además, se encuentra la opción de realizar búsquedas de solicitudes a través del nombre o la fecha del evento (vea **Figura 6.3**). Esta funcionalidad de la aplicación ayuda a resolver el problema **PBLM-7**.



Figura 6.1: Vista de ingreso al sistema.



Figura 6.2: Vista de recuperar clave.

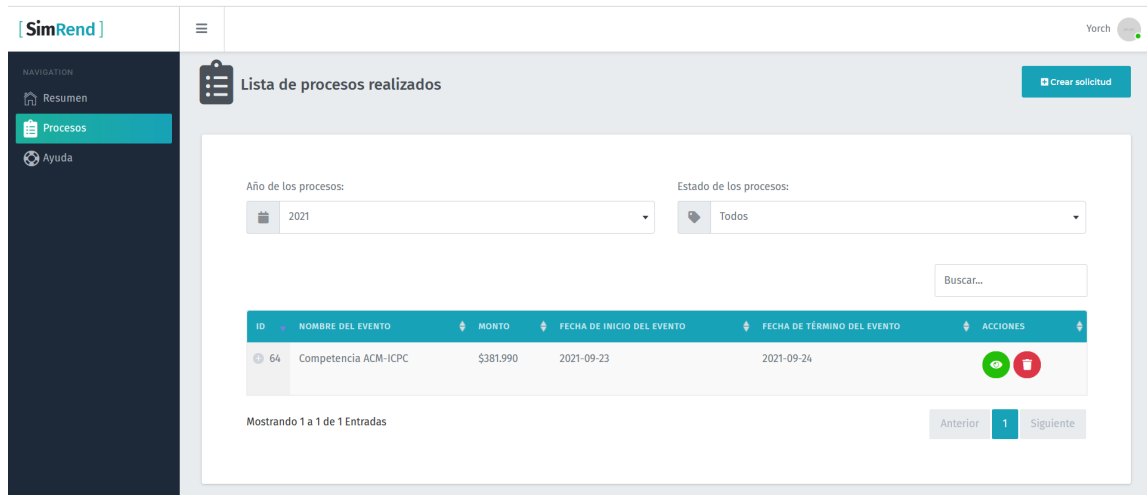


Figura 6.3: Vista de búsqueda de solicitudes.

Solicitud: Esta etapa consta de 4 fases las cuales ayudan a la confección de la solicitud a presentar a la Dirección correspondiente de cada OE. Estas fases se detallan a continuación. Esta funcionalidad de la aplicación ayuda a resolver el problema PBLM-7 y PBLM-9.

Datos principales de la Solicitud: Formulario que solicita el nombre, fecha de inicio, fecha de término, lugar y monto del evento, el responsable a cargo de la solicitud y la cantidad de participantes, la cual puede ser para un grupo de personas específico o masivo (e indeterminado), según sea el caso (vea **Figura 6.4**).

Figura 6.4: Vista que obtiene los datos principales de una Solicitud.

Categoría: En esta etapa se solicitan todas las categoría de los gastos que se realizan dentro del evento para el cual se está realizando la solicitud (vea **Figura 6.5**).

Crear Solicitud

1 Datos principales de la solicitud 2 Gastos del evento 3 Participantes 4 Finalizar

Categoría *

Seleccione una categoría

+ Agregar

CATEGORÍA	ACCIÓN
Alojamiento	

Anterior 1 Siguiente

Anterior Siguiente

Figura 6.5: Vista que obtiene las categorías en que incurren los gastos.

Participante: Formulario que solicita el nombre y RUT de los participantes de la actividad. Esta vista sólo se visualiza si en la etapa de **Datos principales de la solicitud** se indica que es para un grupo de personas específico (vea **Figura 6.6**). Esta funcionalidad de la aplicación ayuda a resolver el problema **PBLM-3**.

Crear Solicitud

1 Datos principales de la solicitud 2 Gastos del evento 3 Participantes 4 Finalizar

RUN del participante:

Nombre participante:

RUN del participante

Nombre del Participante

+ Agregar

NOMBRE	RUN	ACCIÓN
Matias Ignacio Erenchun Marquéz	17.981.314-9	

Anterior 1 Siguiente

Anterior Siguiente

Figura 6.6: Vista que obtiene los datos de los participantes del evento.

Resumen: Etapa que muestra los datos principales de la solicitud y a su vez da la opción de generar el PDF de la solicitud (vea **Figura 6.7**).

Datos Principales	
Nombre del evento	Competencia ACM-ICPC
Lugar del evento	Universidad del Desarrollo, Campus Rector Ernesto Silva Bafallay, San Carlos de Apoquindo, Las Condes, Santiago
Fecha de inicio	2021-09-23
Fecha de término	2021-09-24
Responsable	Yorch Sepúlveda Manríquez
Tipo de actividad	Grupal
Categorías de gastos del evento	
Participantes del evento	
Volver a proceso Volver a procesos Descargar	

Figura 6.7: Vista que muestra el resumen de los datos ingresados para la solicitud.

Resolución: En esta etapa se acepta o rechaza la solicitud. En caso de ser aceptada se procede a ingresar los datos principales de la RU enviada por la Casa de Estudios a la OE interesada, los cuales son el número de la resolución y el año de creación. Además, se solicita que se adjunte una copia digitalizada y en formato PDF de la RU correspondiente a la aceptación de la Solicitud enviada por la OE (vea **Figura 6.8**). Esta funcionalidad de la aplicación ayuda a resolver el problema **PBLM-7**.

Resumen de la solicitud	
Nombre del evento	Competencia ACM-ICPC
Responsable de la solicitud	Yorch Sepúlveda Manríquez
Monto de la solicitud	\$381,990
Fecha de inicio	2021-09-23
Fecha de término	2021-09-24
Lugar del evento	Universidad del Desarrollo, Campus Rector Ernesto Silva Bafallay, San Carlos de Apoquindo, Las Condes, Santiago

Datos de la resolución	
Número de resolución:	<input type="text"/>
Año de la resolución:	<input type="text"/>
Copia del documento:	<input type="text"/>
Volver a proceso Volver a procesos Guardar	

Figura 6.8: Vista que muestra el resumen de los datos ingresados para la solicitud y obtiene los datos principales de la RU.

Rendición: Tras la aceptación de la Solicitud y el ingreso de los datos de la RU que la aprueba, se procede a realizar la Rendición correspondiente al evento. Si bien los datos principales de la Rendición tales como el nombre de la organización, el número de la resolución y el responsable del evento, entre otros, se obtiene automáticamente de los datos registrados en el sistema, el monto total rendido tiene dependencia del registro de documentos (boletas y/o facturas) (vea **Figura 6.9**). Una vez finalizado el proceso de Rendición, está la opción de generar el documento a presentar obedeciendo el formato exigido por la casa de estudio. Esta funcionalidad de la aplicación termina de solucionar los problemas **PBLM-1**, **PBLM-2**, **PBLM-3** **PBLM-4**, **PBLM-7** y **PBLM-8**.

Ver declaración de gastos

Número de resolución:	Fecha límite para declarar gastos:	Días restantes:
264	2021-10-13	20
Monto total solicitado(\$):	Monto total documentos(\$):	Monto total rendido(\$):
381.990	381.750	381.750
Unidad:		
Ingeniería Civil en Computación		
Nombre jefe directo:	Fono anexo:	
Ruth Garrido	0	
Responsable del fondo:	RUN Responsable:	
Yorch Sepúlveda Manríquez	17.824.523-6	
Email:		
yosepulveda7@alumnos.utalca.cl		

Volver a proceso

Documentación

Generar declaración de gastos

Volver a proceso

Cancelar

Finalizar declaración

Figura 6.9: Vista que muestra los datos principales de una Rendición.

Resumen de los gastos realizados por personas: Cuando el tipo de evento ingresado en la Solicitud está destinada a un grupo de personas, se debe declarar los gastos de cada uno de ellos y los gastos que han tenido en común. Es por ello que antes de ingresar la documentación, se muestra un resumen de los registros de documentos (vea **Figura 6.10**). Esta funcionalidad de la aplicación ayuda a resolver el problema **PBLM-3**.

Documentos de una persona: Tras seleccionar a la persona en el *Resumen de los gastos realizados por personas* se procede a mostrar en detalle los documentos que se han registrado de esta y, además se muestran indicadores que ayudan a analizar















+ Documentación			
 TOTAL MONTO DOCUMENTACIÓN \$381.750 <small>88.84% del monto solicitado</small>		 TOTAL MONTO DOCES. RENDIDO \$381.750 <small>88.84% del monto solicitado</small>	
PARTICIPANTES	MONTO DOCUMENTACIÓN(S)	MONTO RENDIDO(S)	ACCIÓN
Carlos Daniel Ríos Moya	8.500	8.500	 
Daniel Eduardo Pérez Bravo	8.950	8.950	 
Documentos en conjunto	328.000	328.000	 
Ignacio Andrés Martínez Hernández	9.000	9.000	 
Matías Ignacio Erenchun Marquáliz	9.500	9.500	 
<div>Anterior 1 2 Siguiente</div>			
 Volver		 Eliminar documentación	

Figura 6.10: Vista que muestra el resumen de los documentos declarados por cada participante y los gastos comunes.

cuanto es la suma total de los documentos registrados para la declaración de gasto, cuanto es la suma total de los documentos seleccionados a rendir, cuanto es la suma de los documentos registrado para declarar del participante y cuanto es la suma de los documentos que se han seleccionado para declarar del participante (vea **Figura 6.11**). Esta funcionalidad de la aplicación ayuda a resolver el problema **PBLM-3**.

Documentación

TOTAL MONTO DOCUMENTACIÓN

\$381.750

88.84% del monto solicitado

TOTAL MONTO GASTOS. RENDIDOS

\$373.250

87.77% del monto solicitado

TOTAL MONTO DOCES. RENDIDOS

\$8.500

2.23% del monto solicitado

TOTAL MONTO GASTOS. RENDIDOS RENDIDOS

\$0

0% del monto solicitado

Responsable de los documentos

Carlos Daniel Ríos Moya

➕ Agregar Documento

ID	NÚMERO DE DOCUMENTO	PRESENCIA	FECHA DEL DOCUMENTO	MONTO(S)	CATEGORÍA DEL GASTO	TIPO DE DOCUMENTO	DOCUMENTADO	ACCIONES
84	70	Metro Santiago	2021-09-23	1.000	Pasajes	Boleta	<div><div></div></div>	<div><div></div><div></div><div></div></div>
83	10280705	Pulmon del Sur	2021-09-24	3.000	Pasajes	Boleta	<div><div></div></div>	<div><div></div><div></div><div></div></div>
82	629953	Andimar	2021-09-23	4.000	Pasajes	Boleta	<div><div></div></div>	<div><div></div><div></div><div></div></div>

Anterior

1

Siguiente

Volver

Eliminar documentación

Figura 6.11: Vista que muestra en detalle los documentos registrados que ha presentado un participante cuya actividad ingresada en la solicitud es de tipo grupal.

Documento: Sección en donde se ingresan los datos de una boleta y/o factura, que corresponden al número del documento, la categoría en que incurrieron los gastos, la fecha de compra de los productos, el monto, una pequeña descripción de los gastos

y el proveedor, entre otros (vea **Figura 6.12**). Esta funcionalidad de la aplicación ayuda a resolver los problema **PBLM-4** y **PBLM-6**.

Agregar documento

Código del documento: Ingrese el código del documento

Fecha del documento: Ingrese la fecha del documento

Descripción del documento: Ingrese la descripción del documento

Tipo de documento: Seleccione el tipo de documento

Proveedor: Ingrese el nombre del proveedor

Monto(\$): Monto del documento

Categoría: Seleccione una categoría

Copia del documento: Seleccione un archivo PDF, JPEG, PNG o JPG

Cancelar Guardar

Figura 6.12: Vista que obtiene los datos de una boleta o factura.

Aceptar o Rechazar una Rendición: Una vez finalizada la Rendición, la aplicación queda a espera de que la Universidad notifique la aceptación o el rechazo de esta. En caso de que se acepte, el proceso finaliza. Mientras que si se rechaza, la aplicación da un plazo de 20 días corridos para modificar la Rendición y su posterior envío al Depto. de Tesorería y Presupuesto (vea **Figura 6.13**).

Ver declaración de gastos

Número de resolución: 254

Fecha límite para declarar gastos: 2021-10-13

Días restante: 15

Monto total solicitado(\$): 381.990

Monto total documentado(\$): 381.750

Monto total rendido(\$): 381.750

Unidad: Ingeniería Civil en Computación

Nombre jefe directo: Ruth Garrido

Fono anexo: 0

Responsable del fondo: Yorch Sepúlveda Matríquez

RUT Responsable: 17.824.523-6

Email: yosepulveda71@alumnos.utaica.cl

Volver a proceso Documentación Declaración rechazada Declaración aceptada

Figura 6.13: Tras finalizar la Rendición, se habilitan las opciones de Aceptar o Rechazar, de acuerdo a lo que responda el Depto. de Tesorería y presupuesto.

Tras lo mencionado con anterioridad se realiza el **Cuadro 6.1**, el cual representa en resumen la funcionalidad de la aplicación obtenida y el/los problema(s) que resuelve mencionados en la **Sección 4.4**.

Cuadro 6.1: Solución a problemas frecuentes

Nombre de la interfaz	Problema que soluciona
Búsqueda de solicitud	PBLM-7
Solicitud	PBLM-7 y PBLM-9
Participante	PBLM-3
Resolución	PBLM-7
Rendición	PBLM-1, PBLM-2, PBLM-3 ,PBLM-4, PBLM-7 y PMBL-8
Resumen de los gastos realizados por personas	PBLM-3
Documentos de una persona	PBLM-3
Documento	PBLM-4 y PBLM-6

6.3. Pruebas

La última fase de las iteraciones es la etapa de prueba, es por ello que para cada una de ellas se realizan Pruebas de Caja Negra. Esto se realiza con la finalidad de corroborar que el sistema cumple con los requerimientos que el usuario solicitó. Principalmente, las pruebas más importantes son las del proceso de solicitud de fondo por rendir, la cual comienza con las fases de confección de la solicitud. Luego se procede a diseñar las pruebas para la aceptación de la solicitud y el ingreso de los datos de la RU enviada por la casa de estudio. Por último, se realizan las pruebas para confección de la rendición correspondiente. Para ello, el sistema debe realizar el proceso de ingreso de los datos para boletas y/o facturas, a las cuales se le denominan Documentos. Además, se debe generar el documento de declaración de gastos del evento. Para más detalles de las Pruebas de Caja Negra vea el **Anexo A**.

Cabe destacar que para los Documentos se valida por un lado que la fecha de emisión del Documento esté comprendida dentro del periodo que comienza con la fecha de emisión de la RU respectiva y termina con la fecha de finalización del evento, y por el otro, que la fecha de declaración de Documento esté dentro del periodo de rendición de gastos (20 días corridos posterior al evento). Adicionalmente, se verifica

que la suma del monto de los Documentos esté dentro del monto solicitado.

En cuanto a las pruebas de usabilidad, se le entregan a las OE una serie de tareas para que interactúen con el sistema. Estas tareas son:

1. Confección de solicitud.
2. Generar PDF de solicitud.
3. Búsqueda de una Solicitud.
4. Ingreso de datos de RU.
5. Confección de Rendición.
6. Generar PDF de Rendición.
7. Generar PDF del listado de documentos.

Para más detalle sobre la pruebas de Caja Negra realizadas, se puede consultar en el **Anexo A**.

6.4. Resultados

La encuesta cuenta con tres secciones y está dirigida a los Presidentes(as) y Secretarios(as) de Finanzas de cada OE (con un total de 16 representantes encuestados). Previo a esta, los representantes tuvieron la oportunidad de probar el sistema. Cabe destacar que estas pruebas están enfocadas en el proceso de una Solicitud de Fondo.

A continuación, se da a conocer el resumen de los resultados de cada sección de la encuesta.

Sección de preguntas generales sobre el sistema 12 de 16 representantes de las OE expresan que es necesaria una herramienta que ayude a asistir en el proceso de realizar una Solicitud de Fondo, mientras que el resto opina lo contrario. Por otra parte, 9 de 16 de los encuestados indican que el sistema ayudaría a reducir el tiempo que demanda realizar una declaración de gasto, mientras que 7 de 16 confían en el sistema entre un 60 % a un 80 %.

7 de 16 encuestados indican que la mayoría de las personas pueden aprender a utilizar el sistema rápidamente, mientras que el resto demuestra ciertas dudas, pero aun así confía en el sistema.

Por último, 12 de 16 encuestados indican que la aplicación ayudaría a reducir la causa de rechazo de una declaración de gasto, mientras que el resto reporta que no está completamente en acuerdo con esta afirmación, pero confían en el sistema entre un 60 % a un 80 %.

Sección de preguntas sobre las funcionalidades del sistema 12 de 16 representantes de las OE indican que las funcionalidades del sistema son suficientes para que la aplicación sea útil para ellos. Por otra parte, al momento de realizar las pruebas existieron fallas al realizar una declaración de gastos, las cuales fueron resueltas con posterioridad. Sin embargo, el 11 de 16 representantes indica que la aplicación logra realizar una declaración de gastos fiable, mientras que el resto indica que solo a veces.

Sección de preguntas sobre usabilidad e interfaz de usuario 13 de 16 representantes indican que la aplicación es fácil de usar y que no requiere de instrucciones previas, mientras que el resto indican un pequeño desacuerdo. Por otra parte, 14 de 16 usuarios indican que la distribución de los elementos en la pantalla es adecuada, mientras que el resto está de acuerdo entre un 60 % y 80 %.

Por último, 12 de 16 representantes indican que el proceso completo de realizar una Solicitud de Fondo es claro, mientras que el resto indica que la claridad de este proceso es entre un 60 % y 80 %.

Para más detalle sobre la encuesta realizada, se puede consultar en el **Anexo B**.

7. Conclusiones

A través de este documento se dan a conocer el proceso que conlleva el desarrollo de este *Proyecto de Titulación*. En un comienzo se detalla el proceso que se debe realizar en una Solicitud de Fondo por Rendir junto con los conceptos principales que ayudan a la comprensión para el desarrollo de esta memoria. Luego se procede a describir la problemática que busca resolver este proyecto: dado lo engorroso que es la confección y la aceptación de una Rendición, ya que los representantes de las OE no siempre siguen el proceso estipulado para la aceptación y su posterior devolución de dinero que cubren los gastos incurridos en un evento (ya sea por inexperiencia o falta de claridad en los detalles del proceso), es que se desarrolla un sistema que **Confecciona solicitudes de fondos por rendir para actividades estudiantiles y las rendiciones de gastos respectivas, de acuerdo con el marco definido por la RU N°2083 del año 2017.**

Esta memoria detalla todo el proceso que conlleva el desarrollo de la aplicación. Esto comienza con la captura de requisitos y la planificación general del proyecto, la cual se modificó dado que surgía problemáticas y/o nuevos requisitos. Luego, tras la planificación se comienza con el diseño de la aplicación y su posterior desarrollo que se realizaban a través de incrementos. Por último, se realizaban las pruebas de los incrementos para verificar si cumplía con el objetivo de cada iteración.

Por último, se realizan pruebas de Caja Negra para medir los objetivos planteados en este documento. Además, se realizaron pruebas de Usabilidad que debió ser contestado por los representantes de las distintas OE.

Tras realizar las diferentes pruebas a los usuarios de la aplicación, estos indican que la herramienta logra asistir en el proceso de una Solicitud de Fondo por Rendir, reduciendo los posibles errores tales como el ingreso de boletas y/o facturas fuera

del periodo del evento, gastos incurridos sin categorización y monto excesivo en comparación al solicitado, entre otros. Esto no implica que el usuario pueda cometer otros errores que produzcan el rechazo de la Rendición, ya que el factor humano no puede ser controlado por la aplicación.

Para finalizar, este proyecto entrega un sistema funcional el cual puede ser utilizado por los distintos CCAA y la Federación de Estudiantes de la Universidad de Talca, Campus Los Niches, Curicó. Pero además existe trabajo futuro que es posible realizar con este Software los cuales son:

- Implementar el sistema de modo tal que pueda ser utilizado por los diferentes navegadores. Por ejemplo, Safari y Opera, entre otros.
- Implementación de funcionalidades para que Grupos Intermedios puedan realizar Solicitudes de Fondos por Rendir.
- Lanzar el sistema como una herramienta oficial para las OE de los demás Campus que pertenecen a la Casa de Estudios.
- Realizar pruebas de seguridad.
- Listado para chequeo de los documentos que se deben adjuntar en la Rendición/Solicitud de reembolso.
- Habilitar o deshabilitar Instituciones.
- Habilitar o deshabilitar Campus.
- Habilitar o deshabilitar Tipos de OE.
- Habilitar o deshabilitar OE.
- Habilitar o deshabilitar Categorías.
- Realizar vista de resumen en donde se muestran indicadores tales como el número de solicitudes realizadas, procesos sin terminar, monto total utilizado del presupuesto anual, gráficos de gastos realizados por mes, entre otros.
- Realizar vista de ayuda en donde se encuentre la RU de cómo realizar el proceso de solicitud de fondos, manual de usuario de la aplicación, preguntas frecuentes y contacto para informar cualquier problema que pueda tener la aplicación.

Glosario

Burocracia: Administración ineficiente a causa del papeleo, la rigidez y las formalidades superfluas [11].

Burocrático(ca): adj. Perteneciente o relativo a la burocracia [12].

CAA: Centro de alumnos. Organización estudiantil que representa a los alumnos de una carrera en particular [18].

DAAE: Dirección de Apoyo a Actividades Estudiantiles. Unidad perteneciente a la Vicerrectoría de Desarrollo Estudiantil (VDE) que promueve el desarrollo integral de los estudiantes, mediante la entrega de herramientas complementarias a su formación académica, que les permitan adquirir las competencias necesarias para lograr un proyecto de vida personal y profesional exitoso, haciendo de la etapa universitaria una experiencia más enriquecedora [7].

Decano: Autoridad superior de la Facultad y dirige todos los asuntos académicos, administrativos y financieros de ella [7].

Director de Escuela: Académico que con dedicación preferente, está encargado de la gestión del plan de estudios de la(s) Carrera(s) a su cargo. Es designado por el Rector a proposición del Decano y depende jerárquicamente de éste ultimo [7].

Engorroso(sa): adj. Dificultoso, molesto [13].

Escuela: Unidad básica de administración de uno o más programas docentes afines de pregrado. Esta a cargo de un(a) Director(a) de Escuela, quien depende jerárquicamente de un(a) Decano(a), o del Vicerrector(a) de Pregrado cuando se trate de carreras no adscritas a una Facultad. Excepcionalmente una Escuela puede depender del Rector(a) [7].

Facultad: Unidad académica que, en conformidad con el Estatuto y las Ordenanzas de la Universidad, agrupa a un cuerpo de personas asociadas con el propósito de enseñar e investigar en áreas afines del conocimiento superior. Una Facultad está dirigida por un Decano [7].

Federación de Estudiantes: Organización estudiantil que representa a los alumnos de un campus perteneciente a la Universidad [16].

Fondos: Recursos económicos que la Universidad de Talca dispone para la realización de diversas iniciativas y/o actividades, entre ellas, las estudiantiles [7].

Fondo por Rendir: Sumas de dinero solicitadas y puestas a disposición de los jefes de unidades y proyectos con presupuesto asignado y disponible, que necesiten atender gastos de carácter especial, imprevistos o urgentes, sujetos todos ellos a rendición posterior [7].

Grupos Intermedios: Grupos de estudiantes que se reúnen con un objetivo y/o interés en común, los cuales son reconocidos por la Universidad de Talca [9].

OE: Organizaciones Estudiantiles. Estas son federación de estudiantes, centros de alumnos o grupos intermedios [16].

Resolución: Puede ser un decreto, una decisión o un fallo que emite una determinada autoridad. Estas pueden establecer reglas, voluntades, etc.

Resolución Exenta: Comprenden aquellas Resoluciones en que no es necesario que sean visadas por la Contraloría General de la República [18].

RU: Resolución Universitaria.

Visar: Dicho de la autoridad competente: Dar validez a un pasaporte u otro documento para determinado uso [14].

Principio de optimalidad: aplicado en programación dinámica, consiste en que una secuencia óptima de decisiones que resuelve un problema, debe cumplir la propiedad de que cualquier subsecuencia de decisiones, que tenga el mismo estado final, debe ser también óptima respecto al subproblema correspondiente. [15].

Bibliografía

- [1] Albert Einstein and empirical software engineering. <https://pdfs.semanticscholar.org/589a/8858795db3919bb1ca05ab44640ce038b76e.pdf>. Última visita: 14 de Octubre de 2018.
- [2] Aliexpress. <https://cl.aliexpress.com/>. Última visita: 25 de Noviembre de 2018.
- [3] Amazon. <https://www.amazon.es/>. Última visita: 25 de Noviembre de 2018.
- [4] Contraloría universidad de talca. <http://contraloria.ugal.cl/html/documentos.php?cat=26>. Última visita: 9 de Octubre de 2018.
- [5] Correos chile. <https://www.correos.cl/SitePages/home.aspx>. Última visita: 25 de Noviembre de 2018.
- [6] draw.io. <https://www.draw.io/>. Última visita: 15 de mayo de 2019.
- [7] Glosario. <http://contraloria.ugal.cl/html/documentos.php?cat=29>. Última visita: 9 de Octubre de 2018.
- [8] Instrucciones generales para la rendición de cuentas de los fondos asignados a través del beneficio complementario de beca doctorado nacional y para extranjeros sin residencia definitiva en chile: Asignación anual para gastos operacionales del proyecto de tesis doctoral etapas 2017. http://www.conicyt.cl/becasconicyt/files/2017/02/manual_rendicionGOP-2017.pdf. Última visita: 25 de Noviembre de 2018.
- [9] Instructivo organizaciones estudiantiles utalca. <http://vde.ugal.cl/docs/Instructivo-Organizaciones.Estudiantiles-Utalca.pdf>. Última visita: 30 de abril de 2019.

- [10] kanbanize. <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tablero-kanban/>. Última visita: 15 de mayo de 2019.
- [11] Rae. <http://dle.rae.es/?id=6Jix9ml>. Última visita: 2 de Enero de 2019.
- [12] Rae. <http://dle.rae.es/?id=6JjuZI8>. Última visita: 2 de Enero de 2019.
- [13] Rae. <http://dle.rae.es/?w=engorroso>. Última visita: 2 de Enero de 2019.
- [14] Rae. <http://dle.rae.es/srv/search?w=visada>. Última visita: 14 de Octubre de 2018.
- [15] Real academia de ingeniería a inicio. <http://diccionario.raing.es/es/lema/principio-de-optimalidad-de-bellman>. Última visita: 15 de mayo de 2019.
- [16] Vicerrectoria de desarrollo estudiantil. <http://vde.utalca.cl/html/organizaciones.html>. Última visita: 9 de Octubre de 2018.
- [17] Wish. <https://www.wish.com/>. Última visita: 25 de Noviembre de 2018.
- [18] Wordreference. <https://forum.wordreference.com/threads/resoluci%C3%B3n-exenta.1164374/?hl=es>. Última visita: 14 de Octubre de 2018.
- [19] Universidad de Talca. RU N °2083. pages 1–8, 2017. Última visita: 9 de Octubre de 2018.
- [20] Fernan. 6 Dynamic Programming. pages 1–31, 2011. Última visita: 15 de mayo de 2019.
- [21] Alba López Gil. Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software. page 95, 2018. Última visita: 14 de Abril de 2019.
- [22] Roger S. Pressman. *Ingeniería del software. Un enfoque practico*. Mc Graw Hill, 2010. Última visita: 11 de Octubre de 2018.

ANEXOS

A. Pruebas de Caja Negra

A.1. Confección de una solicitud

Cuadro A.1: Caja Negra - Registro de datos principales una Solicitud.

Código	Prueba 1	
Precondiciones	- - -	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none">■ Ingresar datos principales de un evento.■ Seleccionar responsable.■ Ingresar la cantidad de participantes.	<ul style="list-style-type: none">■ Datos principales registrados en la BD.	<ul style="list-style-type: none">■ Datos registrados exitosamente.
Observaciones	Para la cantidad de participante existen 2 opciones: para un grupo de personas en donde se debe ingresar el número que participan en la actividad, mientras que existe la posibilidad de que sea una actividad masiva.	

Cuadro A.2: Caja Negra - Registro de categorías para una solicitud.

Código	Prueba 2	
Precondiciones	Prueba 1	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Agregar una o más categorías que indican en que se realizan los gastos del evento. 	<ul style="list-style-type: none"> ■ Categorías registradas en la BD. 	<ul style="list-style-type: none"> ■ Datos registrados exitosamente.
Observaciones	- - -	

Cuadro A.3: Caja Negra - Registro de participantes para una solicitud.

Código	Prueba 3	
Precondiciones	Prueba 1, Prueba 2	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Agregar a uno o más participantes del evento. 	<ul style="list-style-type: none"> ■ Participantes registrados en la BD. 	<ul style="list-style-type: none"> ■ Datos registrados exitosamente.
Observaciones	La cantidad de participantes a agregar, depende del número de participantes ingresado en la Prueba 1 .	

Cuadro A.4: Caja Negra - Finalizar solicitud.

Código	Prueba 4	
Precondiciones	Prueba 1, Prueba 2, Prueba 3	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Finalizar solicitud y generar PDF. 	<ul style="list-style-type: none"> ■ Mensaje de la operación. ■ Obtener Documento de solicitud en formato PDF 	<ul style="list-style-type: none"> ■ Mensaje de solicitud finalizada. ■ Documento de Solicitud en formato PDF
Observaciones	- - -	

A.2. Ingreso de RU

Cuadro A.5: Caja Negra - Aceptación de solicitud e ingreso de RU.

Código	Prueba 5	
Precondiciones	Prueba 4	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Aceptar solicitud. ■ Ingresar datos de RU. ■ Adjuntar copia digitalizada de RU en formato PDF. 	<ul style="list-style-type: none"> ■ Datos registrados en la BD. ■ Mensaje de la operación. 	<ul style="list-style-type: none"> ■ Mensaje de solicitud finalizada.
Observaciones	- - -	

A.3. Confección de una rendición

Cuadro A.6: Caja Negra - Confeccionar una Rendición.

Código	Prueba 6	
Precondiciones	Prueba 5	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Registrar Documentos. ■ Finalizar Rendición. 	<ul style="list-style-type: none"> ■ Documentos registrados en la BD. ■ Mensaje de la operación. ■ Selección de Documentos para realizar la declaración de gastos. ■ Cálculo de monto cercano o igual al solicitado. ■ Datos registrados en la BD. ■ Mensaje de la operación. 	<ul style="list-style-type: none"> ■ Documentos registrados exitosamente. ■ Mensaje de éxito. ■ Listado de Documentos seleccionados para realizar la declaración de gastos. ■ Monto cercano o igual al solicitado. ■ Datos registrados en la BD exitosamente.
Observaciones	- - -	

Cuadro A.7: Caja Negra - Registro de un Documento.

Código	Prueba 7	
Precondiciones	Prueba 6	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Datos del documento. 	<ul style="list-style-type: none"> ■ Datos registrados en la BD. ■ Mensaje de la operación. 	<ul style="list-style-type: none"> ■ Datos registrados exitosamente. ■ Mensaje de éxito.
Observaciones	---	

Cuadro A.8: Caja Negra - Obtener documento de Rendición.

Código	Prueba 8	
Precondiciones	Prueba 6, Prueba 7	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Generar Rendición. 	<ul style="list-style-type: none"> ■ Mensaje de la operación. 	<ul style="list-style-type: none"> ■ Mensaje de éxito. ■ Rendición en formato PDF
Observaciones	---	

Cuadro A.9: Caja Negra - Obtener documento de listado de Documentos.

Código	Prueba 9	
Precondiciones	Prueba 6	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ■ Generar listado de Documentos. 	<ul style="list-style-type: none"> ■ Mensaje de la operación. ■ Listado de Documentos en formato PDF. 	<ul style="list-style-type: none"> ■ Mensaje de éxito.
Observaciones	---	

B. Encuesta a representantes de las OE

B.1. Sección de preguntas generales sobre el sistema

1. ¿Cree usted que es necesaria una herramienta que ayude a las OE a realizar el proceso de Solicitud de Fondo?

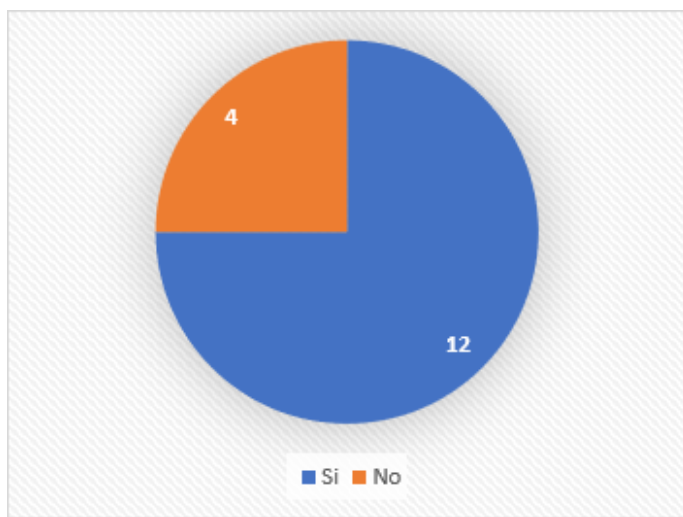


Figura B.1: Resultado de la pregunta uno.

2. ¿Cree usted que esta herramienta puede ayudar a reducir el tiempo que demanda realizar una Rendición? (En escala de 1 (No es de ayuda) a 5 (Es de mucha ayuda))

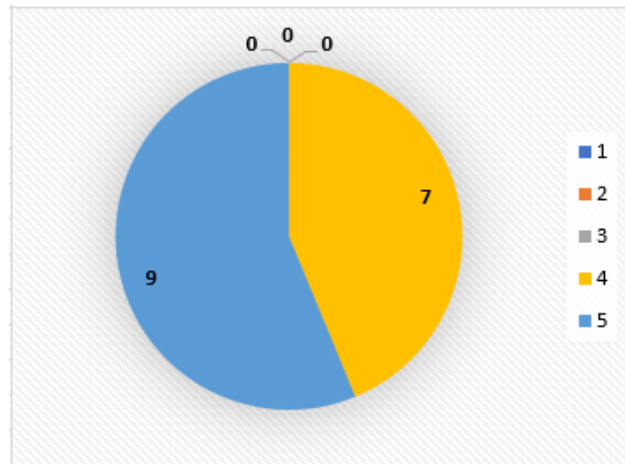


Figura B.2: Resultado de la pregunta dos.

3. ¿Cree usted que la mayoría de las personas pueden aprender a usar el sistema rápidamente? (En escala de 1 (En desacuerdo) a 5 (De acuerdo))

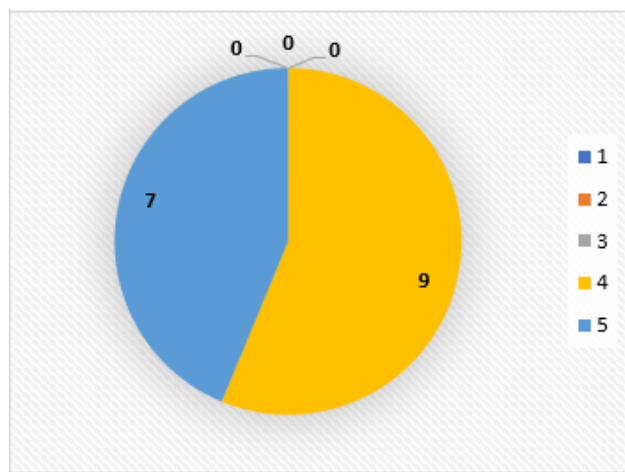


Figura B.3: Resultado de la pregunta tres.

4. ¿Cree usted que gracias a la aplicación se puede reducir la causa de rechazo de una Rendición? (En escala de 1 (En desacuerdo) a 5 (De acuerdo))

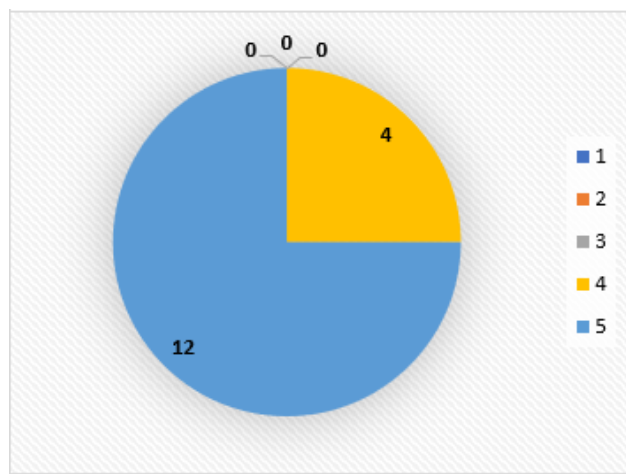


Figura B.4: Resultado de la pregunta cuatro.

B.2. Sección de preguntas sobre las funcionalidades del sistema

5. ¿Cree usted que las funcionalidades son suficientes para que la aplicación sea útil?

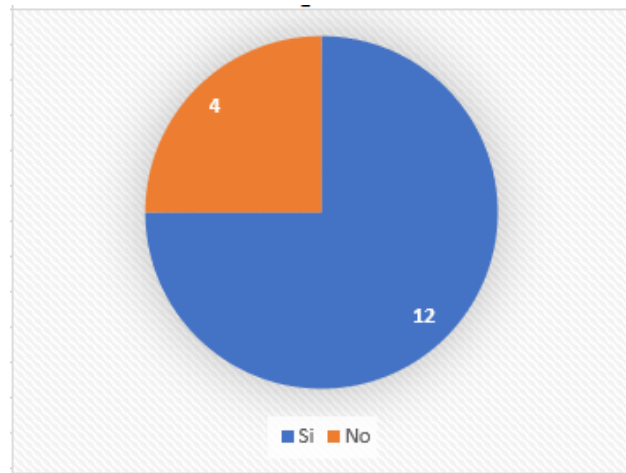


Figura B.5: Resultado de la pregunta cinco.

6. ¿El sistema logra realizar una Rendición fiable? (Siempre, A veces, nunca)

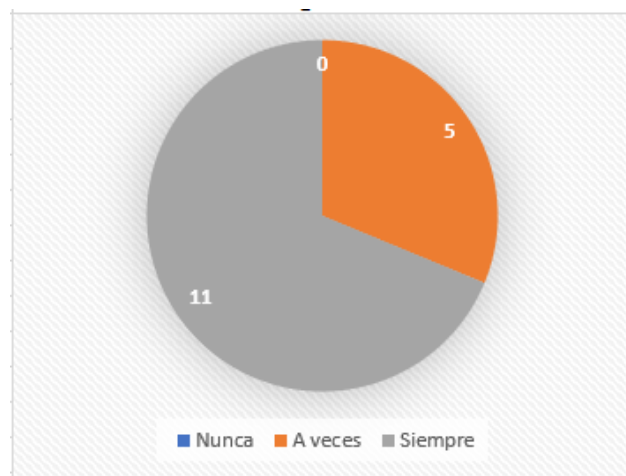


Figura B.6: Resultado de la pregunta seis.

B.3. Sección de preguntas sobre usabilidad e interfaz de usuario (En escala de 1 (En desacuerdo) a 5 (De acuerdo))

7. ¿La aplicación es fácil de usar?

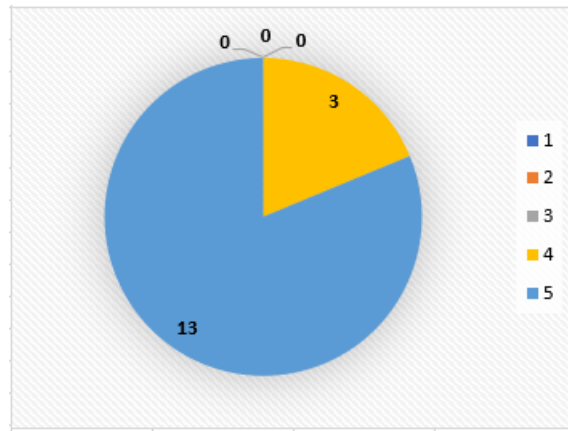


Figura B.7: Resultado de la pregunta siete.

8. ¿Se entiende cómo utilizar la aplicación sin necesidad de instrucciones?

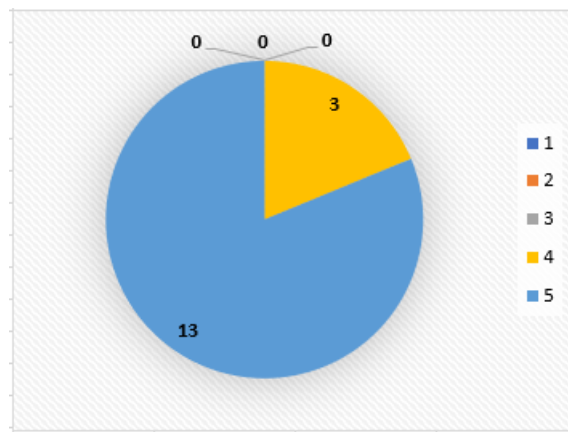


Figura B.8: Resultado de la pregunta ocho.

9. ¿La distribución de los elementos en la pantalla es adecuada?

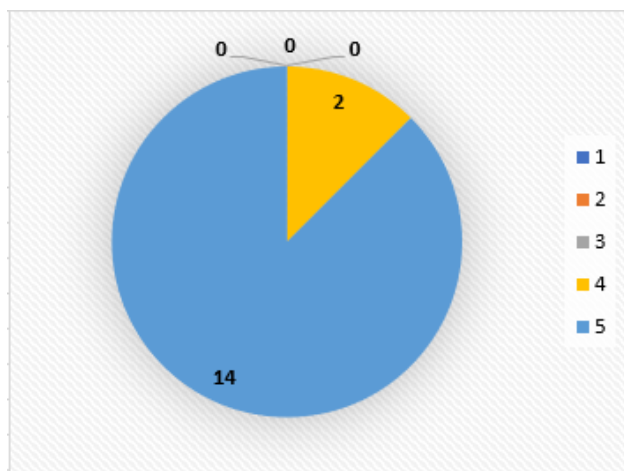


Figura B.9: Resultado de la pregunta nueve.

10. ¿El proceso completo de realizar una Solicitud de Fondo es claro?

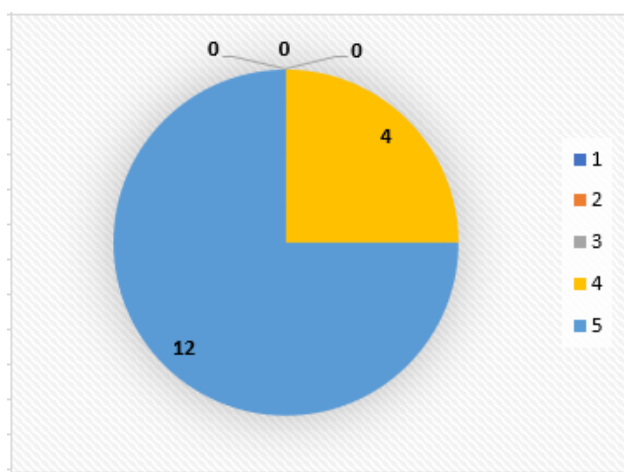


Figura B.10: Resultado de la pregunta diez.