



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

Profesor Guía: RODRIGO PAREDES MORALEDA

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Enero, 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

Profesor Guía: RODRIGO PAREDES MORALEDA

Profesor Informante: PROFESOR INFORMANTE 1

Profesor Informante: PROFESOR INFORMANTE 2

Memoria para optar al título de
Ingeniero Civil en Computación

El presente documento fue calificado con nota: _____

Curicó – Chile

Enero, 2019

Dedicado a ...

AGRADECIMIENTOS

Agradecimientos a ...

TABLA DE CONTENIDOS

| | página |
|--|-------------|
| Dedicatoria | I |
| Agradecimientos | II |
| Tabla de Contenidos | III |
| Índice de Figuras | VI |
| Índice de Tablas | VII |
| Resumen | VIII |
| 1. Introducción | 9 |
| 1.1. Contexto del proyecto | 9 |
| 1.2. Definición del problema | 12 |
| 1.3. Presentación de la solución | 13 |
| 1.4. Objetivos | 13 |
| 1.4.1. Objetivo general | 13 |
| 1.4.2. Objetivos específicos | 14 |
| 1.5. Alcances | 14 |
| 1.6. Limitaciones | 14 |
| 1.7. Proyectos similares | 15 |
| 1.8. Descripción de contenidos | 15 |
| 2. Antecedentes | 16 |
| 2.1. Conceptos | 16 |
| 2.2. Tecnologías utilizadas | 20 |
| 2.2.1. Frameworks | 20 |
| 2.2.2. Bibliotecas | 21 |
| 2.3. Metodología de desarrollo de software | 21 |
| 2.4. Metodología ágil | 22 |
| 2.4.1. Scrum | 24 |

| | |
|---|-----------|
| 2.4.2. Roles | 25 |
| 2.4.3. Artefactos | 26 |
| 2.4.4. Reuniones | 27 |
| 2.4.5. Ciclo de vida | 29 |
| 2.5. Validación de la aplicación | 30 |
| 2.6. Herramientas de planificación | 31 |
| 3. Metodología utilizada | 32 |
| 3.1. Estudio del entorno | 32 |
| 3.1.1. Las personas | 32 |
| 3.1.2. La aplicación | 33 |
| 3.1.3. Las herramientas | 33 |
| 3.2. La metodología | 35 |
| 3.2.1. Roles | 36 |
| 3.2.2. Aptitudes requeridas | 37 |
| 3.2.3. El proceso | 37 |
| 3.2.4. Documentación | 40 |
| 3.2.5. Cómo se trabajó | 41 |
| 4. Características del sistema | 42 |
| 4.1. Requisitos del sistema | 42 |
| 5. Diseño de la aplicación | 43 |
| 5.1. Diseño | 43 |
| Glosario | 46 |
| Bibliografía | 48 |
| Anexos | |
| A: El Primer Anexo | 51 |
| A.1. La primera sección del primer anexo | 51 |
| A.2. La segunda sección del primer anexo | 51 |
| A.2.1. La primera subsección de la segunda sección del primer anexo | 51 |

| | |
|---|-----------|
| B: El segundo Anexo | 52 |
| B.1. La primera sección del segundo anexo | 52 |

ÍNDICE DE FIGURAS

| | página |
|--|--------|
| 2.1. Metodología tradicional vs metodología ágil [15]. | 22 |
| 2.2. Reuniones en el Proceso de Scrum. | 27 |
| 2.3. Proceso de metodología Scrum. | 30 |
| 3.1. Sprint Backlog. | 34 |
| 3.2. Proceso de trabajo de un Sprint. | 35 |
| 3.3. Proceso utilizado en el proyecto | 38 |
| 5.1. Proceso de Fondos por Rendir por parte de Fedeut. | 44 |
| 5.2. Proceso fondos por rendir por parte de CAA. | 45 |

ÍNDICE DE TABLAS

| | página |
|--|--------|
| 2.1. Metodología Tradicional vs Metodología Ágil [15]. | 23 |

RESUMEN

Aquí va el resumen (en Castellano)...

1. Introducción

Las Organizaciones Estudiantiles 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, actividades culturales y deportivas, entre otras. Para ello se debe realizar un conjunto de procedimientos tales como la aprobación de Solicitud y envío de la Resolución Universitaria, 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 incurridos en las actividades realizadas por las Organizaciones Estudiantiles. 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 Organizaciones Estudiantiles, 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 reconocidas. Entre las actividades destacan: recepción de alumnos nuevos; celebración del día de la carrera; actividades sociales, culturales y deportivas; entre otras [15].

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) o uno de Grupos Intermedios 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 documentación original la archiva Secretaría General.
- 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.

Este proceso se puede apreciar con mayor detalle en la **Figura 5.1**.

B. Caso de Centros de Alumnos

Sólo el Presidente, Tesorero o Secretario de finanzas del Centro de Alumnos (CAA) 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.

Quien dirige la Dirección 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.

Se puede apreciar con mayor detalle en la **Figura 5.2**.

En ambos casos, el Departamento de Presupuesto hace responsable de la actividad presupuestaria correspondiente a la O.E. (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 O.E. y adjunta la segunda copia de la Resolución al Comprobante de Egreso para la posterior rendición del Fondo.

Al representante de la O.E. 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.
- Respaldar 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, y si esta última es negativa, la devolución del dinero 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 sólo el Presidente, Tesorero o Secretario de Finanzas de Federación o CAA 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 de fondo.

1.3. Presentación de la solución

En base al problema descrito anteriormente es que nace este proyecto el cual trata de un sistema web que permita a las distintas O.E. la confección de solicitud 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, boletas fuera del plazo en que se realiza la actividad, entre otros.

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

- Los usuarios tienen un respaldo de las 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 contado desde el último acto administrativo de la transferencia de recursos.

1.4. Objetivos

Dado que este sistema pretende aliviar todos los inconvenientes de forma en que se puedan incurrir 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

- Modelar el proceso de confección de solicitudes de fondos por rendir.
- Modelar el proceso de rendición de gastos.
- Desarrollar una aplicación web que asista al proceso de solicitud de fondos por rendir y su posterior rendición de gastos.

1.5. Alcances

La aplicación, *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 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 las boletas y selección del conjunto de boletas que mejor se ajusta al presupuesto expuesto en la solicitud de fondo enviada por la O.E. 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 cambio de directivas de las O.E. se debe realizar una capacitación para que los encargados puedan realizar rendiciones. Aun así, la capacitación es vaga y las O.E. 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” [7].

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 [13], 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, los requisitos del sistema, además de la forma que se priorizan estos últimos para así saber una aproximación del tiempo que conlleva el desarrollo del proyecto.

2. Antecedentes

Este capítulo hace referencia a los conceptos necesarios para la correcta comprensión del proyecto realizado, tales como el proceso el cual se desenvuelve la aplicación, la metodología, tecnologías y herramientas a utilizar.

2.1. Conceptos

Para profundizar lo explicado en el contexto del problema que se encuentra en la Sección 1.1 del Capítulo 1, es que hace mención a los conceptos claves para la comprensión del proceso el cual se rige la aplicación.

Evento Un evento según el “Plan estratégico de la Universidad de Talca Visión 2020” es una actividad emprendida por las distintas Organizaciones Estudiantiles reconocidas tales como recepción de alumnos nuevos, celebración del día de la carrera, actividades culturales y deportivas, actividades sociales, entre otras [15].

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

- Nombre del evento.
- Fecha de inicio y de término de la actividad.
- Lugar a realizar la actividad.

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

Resolución Universitaria(RU) Una Resolución puede ser un decreto, una decisión o un fallo que emite una determinada autoridad. Estas pueden establecer reglas, voluntades, etc. pero para efectos de la universidad, se habla de Resolución Universitaria en donde se decreta las trasferencias de fondos para llevar a cabo actividades a las cuales se realizarán por la Organización Estudiantil al momento de que se acepta la solicitud enviada por este.

En este documento se menciona la suma que se transfiere a la Organización Estudiantil, el propósito que se solventara 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 haberlo.

Además, se establece a quien serán dirigido los dineros, en este caso al Presidente o Secretario de Finanzas de la Organización estudiantil.

Por otra parte, el código de una RU es la unión del año con su número de documento.

Rendición Una rendición es el documento en el cual se da cuenta de todos los gastos que incurrieron en una actividad o evento realizado por la organización universitaria detalladamente.

En el encabezado del documento se debe detallar lo siguiente:

- Número de Resolución
- Unidad (por ejemplo, el nombre de la escuela)
- Nombre Jefe Directo (por ejemplo, Director de escuela)
- Responsable del Fondo

- Rut del responsable
- Total solicitado
- Total rendido
- Fono anexo
- email del responsable

En el detalle que debe ir en el documento de rendición para las facturas son:

- Número de documento
- Fecha
- Nombre del proveedor
- Descripción
- Monto
- C. Costo

En cuanto al detalle que debe ir en la sección para las boletas con gastos comunes son:

- Número de documento
- Fecha
- Nombre del proveedor
- Descripción
- Monto

Mientras que en el detalle que debe ir en la sección para las boletas con gastos individuales son:

- Número de documento

- Fecha
- Nombre del proveedor
- Descripción
- Monto
- Total monto de gastos de todas las Categorías por alumno
- Nombre del alumno a reembolsar
- Total reembolso por participante

Además, debe ir un resumen en donde se indique:

- Total de gastos
- Más Reintegro por caja
- Total de rendición

Otro de los resúmenes que debe existir es el detalle de reembolso por participante que indique:

- Nombre de los participantes
- Monto total por participante
- Total de reembolso

Y por último debe ir la firma del responsable de la rendición y junto con ello debe ir el nombre, el cago que tiene en la organización y en caso de ser una C.A.A. indicar el nombre de la carrera, sino a la facultad que pertenece.

Por otra parte se deben adjuntar los documentos originales se encuentran en la rendición y en caso de tener gastos superiores a 3 UTM se deben adjuntar tres cotizaciones [15].

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, Wix, entre otras. Pero para este proyecto se utilizó ASP.NET Core cuyo motivo se debe a que es un lenguaje sólido y robusto, tanto para proyectos pequeños, grandes o con gran potencial de crecimiento. Además, está orientado a objeto 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 dará a conocer pequeñas descripciones de los principales componentes que cuenta esta tecnología, además de las librerías que se utilizó en el proyecto.

- **HTML-Razor:** HTML es un lenguaje de marcado de hipertextos el cual dispone de un conjunto de etiquetas las cuales son insertadas en un documento que puede mostrar la información a través de un navegador web. Este cuenta además con Razor, el cual 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 será 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.

2.2.1. Frameworks

- **ASP.NET Core:** es la nueva versión modular del framework .NET, la cual es multiplataforma y Open Source diseñado para la creación de aplicaciones

modernas conectadas a Internet, como aplicaciones web y APIs Web.

2.2.2. Bibliotecas

2.3. Metodología de desarrollo de software

La palabra metodología es compuesta por tres vocablos griegos, las cuales son: metá (“más allá”), odós (“camino”) y logos (“estudio”); es por ello por lo que dado lo anterior, metodología son los medios que se pueden utilizar para llegar a un fin para luego determinar cuál es el más adecuado.

Las metodologías de desarrollo de software ayudan a estructurar, planificar y controlar el proceso de desarrollo en 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 llevar al éxito un proyecto. Cada metodología de desarrollo de software tiene ciertas actividades las cuales ayudan a idear, implementar y mantener el producto de software el cual surge desde alguna necesidad hasta cumplir con el objetivo por el cual fue creado. Es por esto por lo que, tras lo último mencionado, se propuso la necesidad de metodizar el desarrollo de software, a 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 la otra, pero surgen preguntas como: ¿existe una mejor que la otra? o ¿Cuál de estas llevará al éxito?

Respondiendo a las últimas preguntas mencionadas, no existe una metodología mejor que otra ya que depende del en el que se utilicen como por ejemplo recursos

técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. 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.

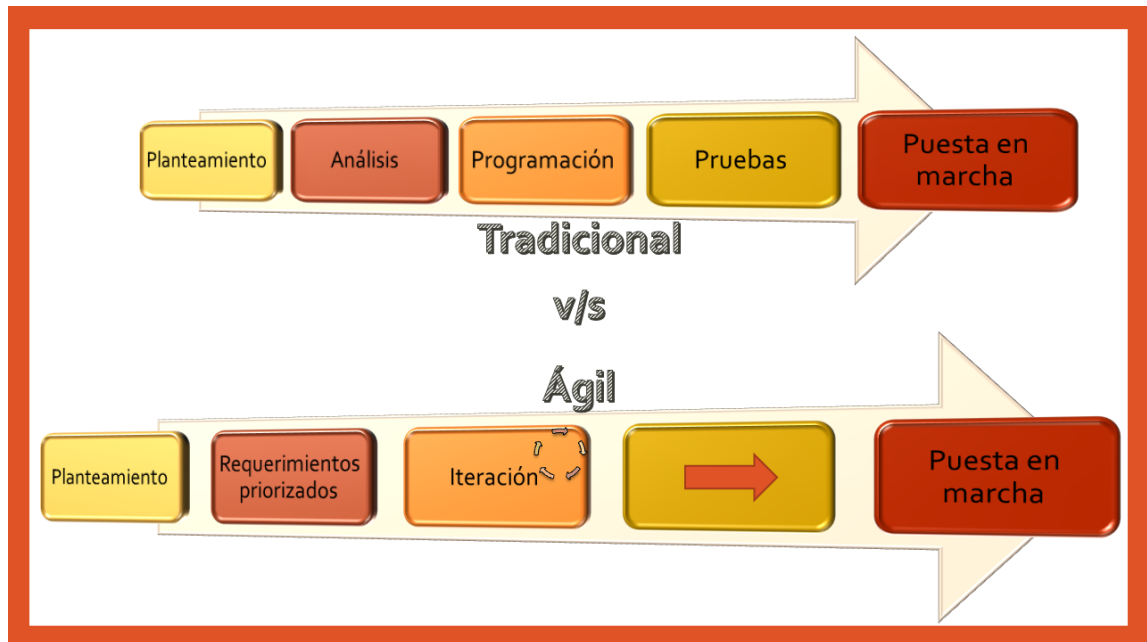


Figura 2.1: Metodología tradicional vs metodología ágil [15].

En cuanto a la elección de la metodología que se utiliza, existen un sin fin de factores que diferencia 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 la Metodología Ágil de la cual se dará mayor énfasis en el siguiente ítem de este documento.

2.4. Metodología ágil

Tras las constantes fallas en los proyectos de desarrollo antes del año 2001, es que en esta fecha 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

Cuadro 2.1: Metodología Tradicional vs Metodología Ágil [15].

| | 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 |

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 más que 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 con 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, de modo que 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 a que se trabaja de una manera más eficaz y sin olvidar el presupuesto acordados junto con el tiempo de entrega definido al comienzo del proyecto.
- Trabajar con mayor velocidad y eficiencia, dado a 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.

Existe muchas metodologías, pero con distintas orientaciones acerca de cómo llevar a cabo el proceso de desarrollo de software, entre las cuales se encuentran Scrum (Marco de trabajo seleccionado para el desarrollo del proyecto el cual se dará énfasis 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 por lo que se dará énfasis en los siguientes ítems 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 [16].

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.

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.

En resumen, la clave del éxito de Scrum son los sprint, los cuales pueden durar de dos a cuatro semanas, en donde al iniciar cada uno de estos, se seleccionan una lista de historias de usuarios (backlog). Por otra parte, dentro del transcurso de cada iteración, diariamente se realizan breves reuniones de aproximadamente 15 minutos, las que ayudan a conocer el avance del desarrollo del proceso.

2.4.2. Roles

Para llevar a cabo el desarrollo del proyecto existen roles fundamentales dentro de la metodología Scrum, los cuales son:

- **Scrum Master (SM):** 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, 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 los requisitos en la lista de tareas. Además, debe asignar requisitos 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):** 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, 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. Artefactos

- **Backlog o retrasos:** Existen tres tipos, los cuales son el product backlog (retraso del producto), el backlog de versión o release backlog y por último el backlog de sprint (Retraso de sprint), los que se verán con mayor detalle a continuación:

Product backlog: lista de requisitos o características del proyecto los cuales proporcionan valor comercial para los stakeholders. Estos requisitos son de alto nivel y no pueden tener detalles de implementación o técnicos. Además, están asociado a una estimación de tiempo y son especificados según los criterios de la organización.

Backlog de versión: consiste en una sub lista de tareas extraídas del backlog del producto, las cuales son priorizadas por el product owner en donde deben desarrollarse para la siguiente versión del producto. Estas deben ser más detalladas, con mayor precisión en cuanto al tiempo en que demorará en desarrollarse 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.

En general, es una lista de requisitos o características del proyecto ordenadas por prioridad, los cuales proporcionan valor comercial para los stakeholders. Además, en cualquier momento se pueden agregar elemen-

tos a los retrasos. Por otra parte, el product owner evalúa los retrasos y actualiza las prioridades de esta lista de acuerdo con lo requerido.

- **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 éste evalúe 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. Para ello, el encargado de esto no suceda es el scrum master, por lo cual crea una lista de tareas para tener seguimiento de los impedimentos que requieren resolverse.

2.4.4. 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.

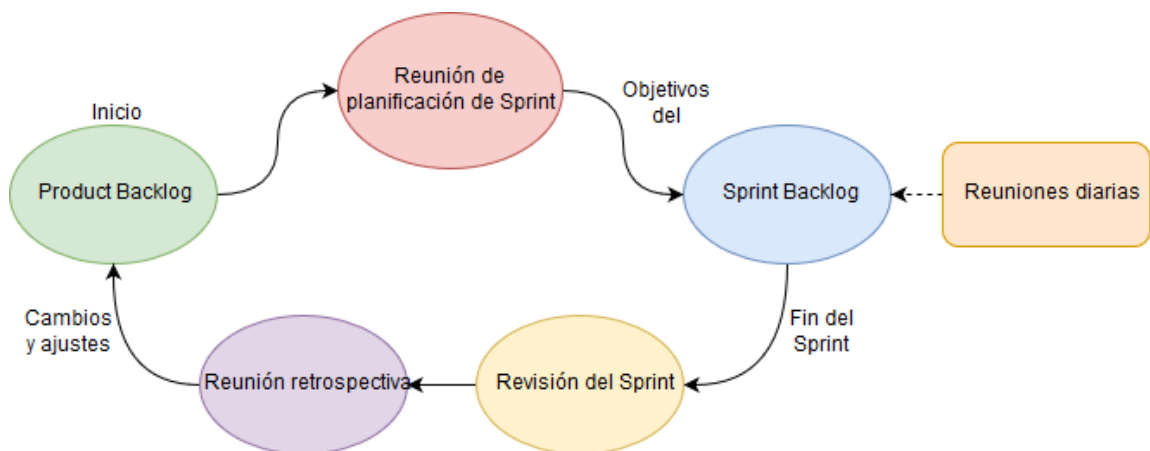


Figura 2.2: Reuniones en el Proceso de Scrum.

- **Planificación del sprint:** Los sprint 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 product owner, el scrum master y el equipo para definir en cual funcionalidad se debe trabajar, asumiendo que los objetivos no van a cambiar durante este. 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 (maquinas, redes, pizarra, etc.), entre otras.

Esta reunión consta de dos partes; la primera, el equipo junto con el product owner seleccionan la lista de requisitos que se convertirá 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 15 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 trabajos presentan el incremento que construyó, el cual es supervisado por el product owner y los stakeholders. Por otro lado, deben indicar que resultó bien y mal del incremento, las observaciones que surgieron durante este 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 Scrum Master. 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 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, se deben discutir. 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 product owner y el scrum master, teniendo todo el derecho de opinar abiertamente sobre el trabajo realizado.

2.4.5. Ciclo de vida

Antes de iniciar cada iteración, el equipo de trabajo revisa las tareas pendientes y además selecciona nuevas tareas que formará 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.

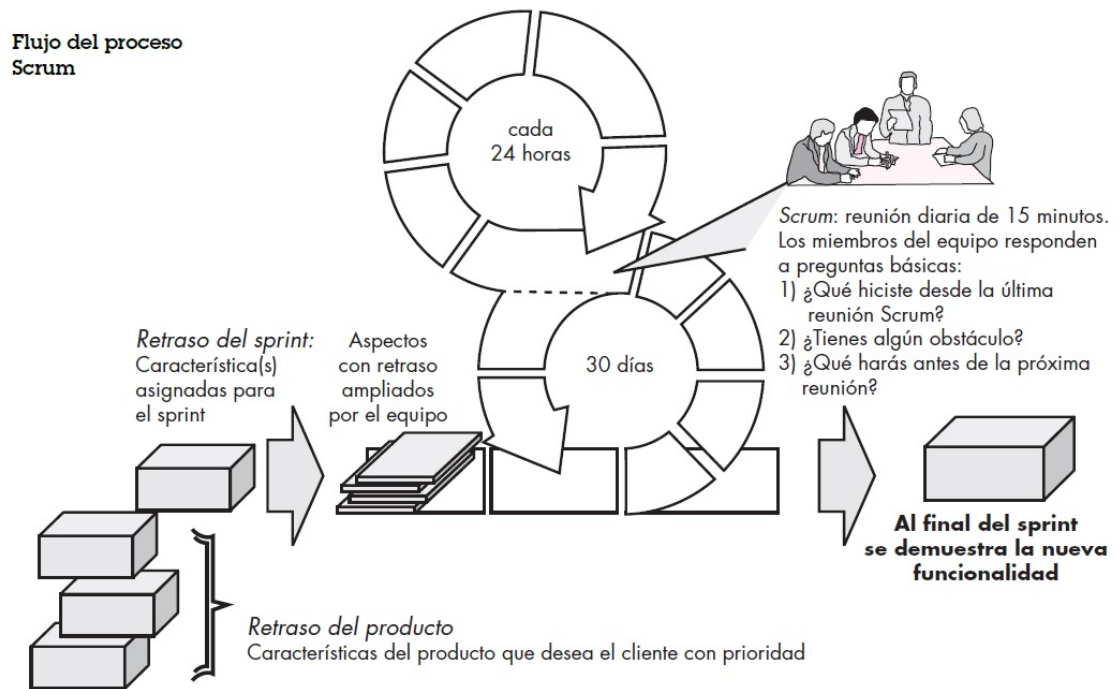


Figura 2.3: Proceso de metodología Scrum.

2.5. Validación de la aplicación

Para el proyecto, se utiliza como método de evaluación la metodología con experimentos controlados. Esta maneja pruebas que se hacen bajo condiciones totalmente constante y en algunos casos, puede variar algún factor. El objetivo principal es realizar un muestreo de usuarios conocidos los cuales utilizan el software con pruebas de usabilidad simples, además del cliente.

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

Para finalizar, el motivo por el cual se escoge esta metodología de evaluación se debe a que 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], experimen-

tar 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 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 cuanta dificultad puede tener. Por otra parte, se puede visualizar la cantidad de colaboradores que tener el proyecto (si es que los hay).

²*Gain more understanding of what makes software “good” and how to make software well*

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 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 como es el ambiente en el cual se realizó el proyecto. Este contempla el análisis de las personas, las cualidades 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 roles los cumple una persona 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 lo óptimo.

Por otra parte, la curva de aprendizaje 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 auto-aprendizaje para sacar el proyecto adelante.

3.1.2. La aplicación

La aplicación solicitada permite realizar el proceso que una O.E. 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 una documentación de cómo realizar el proceso de fondo por rendir, es muy burocrático seguirlo al pie de la letra por las distintas O.E., cometiendo muchos errores (sobre todo las nuevas O.E. que no tienen experiencias). Ahora, dado que las O.E. son los principales usuarios de la aplicación, es que los requerimientos del sistema son cambiantes en cuanto a la facilidad de uso, pero el objetivo principal que es cumplir con lo estipulado por la RU anteriormente mencionada. Además, es posible que los requisitos cambiaran a medida que avanza el proyecto. Es por ello por lo que eran 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 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 el mismo orden que se ingresó. Este listado ayuda al usuario a chequear que las boletas sigan el orden en que se asignó 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.

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



Figura 3.1: Sprint Backlog.

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 estan 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 finalizada y aceptadas.
- **Necesita Información:** sección en donde se encuentran las tareas que no ha logrado culminarse debido a que falta información para su realización.

Para el diseño de base de datos se usa una aplicación llamada draw.io para diseñar el diagrama Entidad/Relación y luego de diseñar el diagrama E/R se procede crear el modelo relacional, es por que aparece StarUML que permite la creacion de estas tabas de forma gráfica y exportar todo esto a código, el cual es importado en MySQL que es donde está alojada la base de datos.

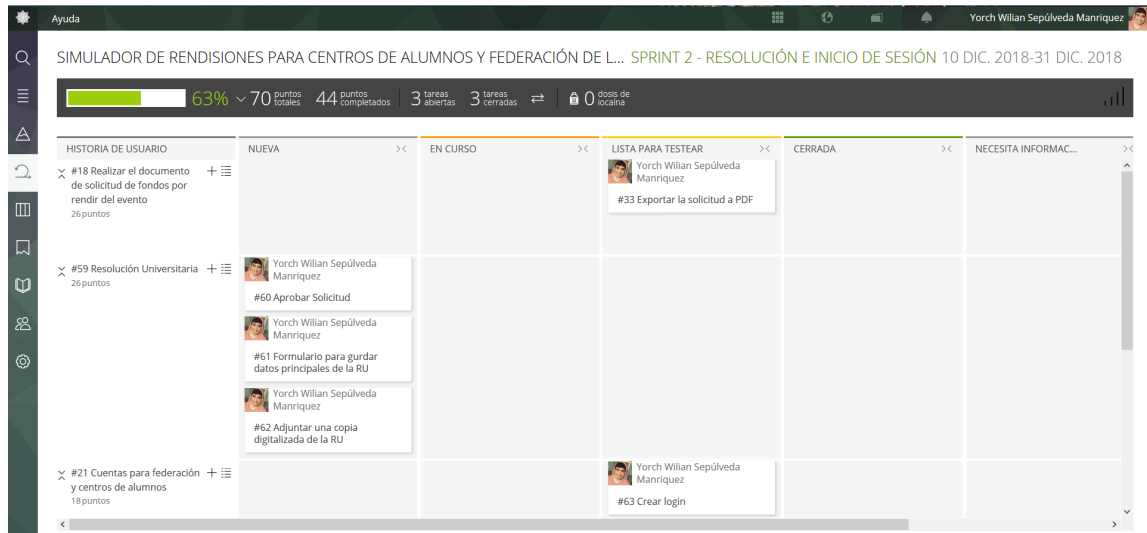


Figura 3.2: Proceso de trabajo de un Sprint.

StarUML no solo se utilizó para diseñar la base de datos, sino que también para la creación de la arquitectura lógica y el diagrama de clases. Este diseño se utilizó en un framework (ver Sección 2.2) el cual ayudó a que se desarrollará la aplicación. Por otra parte, se utiliza primeramente el IDE Visual Studio para la creación del proyecto, pero a lo largo del desarrollo de este último se cambió de IDE a VSCode ya que había que utilizar una librería para la creación de archivos PDF y que Visual Studio no podía utilizar debido a que entraba en conflicto.

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

3.2. La metodología

Existen muchas metodologías las cuales tienen amplia documentación y han probado ser eficientes para casos en específicos. 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.

- 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.

Cabe destacar que a pesar de que Scrum está diseñado para trabajar en equipos, para efectos de esta memoria, se tuvo que ajustar cada rol a sólo un individuo, excepto el de los stakeholders los cuales ayudan a asegurar que el software desarrollado cumpla con las expectativas.

En cuanto a la duración de cada Sprint, se determina en tres semanas, tiempo relativamente razonable para realizar iteraciones de productos utilizables.

3.2.1. Roles

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

Una persona puede tener asignado uno o más roles y a su vez estos últimos pueden ser asignados a una 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 por lo que a continuación se dará mayor énfasis en explicar los roles involucrados en el proyecto:

- **Product Owner:** Es quien administra el proyecto y dirige al equipo de trabajo, controlando que todo marche bien. Además, debe establecer estrategias para mitigar los riesgos que se presentan. Es por ello que debe tener cualidades de democrático, participativo y colaborador.
- **Scrum Master:** encargado de trabajar junto al team ayudándoles en la toma de decisiones, escrituras y optimización del código, haciendo que el proceso siga su curso. Por otra parte, resuelve conflictos que existen en el equipo es por ello por lo que debe supervisar la comunicación entre los integrantes.

- **Team:** encargado de la codificación de los componentes a desarrollar en la iteración. Debe tener conocimientos de los aspectos técnicos involucrados, ser colaborativo, comunicativo autocrítico y capaz de aprender rápidamente.
- **Stakeholders:** son los individuos que interactúan con la aplicación que se realiza, ya que de ellos nacen los requerimientos del sistema a parte de la RU en la cual está enfocado este proyecto. Es por ello que su participación es de gran importancia para mantener un desarrollo rápido y ágil.

3.2.2. Aptitudes requeridas

Para cumplir los roles mencionados en la Sección 3.2.1 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.

3.2.3. El proceso

La gran parte de las metodologías realizan sus procesos como 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 por lo 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 por lo que la presencia de los stakeholders como participante en el proyecto es de vital importancia. Es por ello que en los siguientes subítems se hablará en detalle de cómo se aplica la metodología al proyecto y la planificación de este.

3.2.3.1. 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. Y por último, se hace mención del conflicto que se tiene en caso de que se deba volver atrás por algún error cometido a lo largo del proyecto pero, ¿Qué es el proyecto?

El proyecto es el ciclo general que ayuda a resolver el problema planteado, el cual como se logra apreciar en la **Figura 3.3.** 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 contempla muchos sub-ciclos denominados “Entrega”, que corresponde a nuevas funcionalidades realizadas y mejoras continuas a lo largo del proyecto. Pero para realizar estas entregas, existe otro sub-ciclo que corresponde a las “Iteraciones” y dentro de esta, están enlazadas un conjunto de fases que se deben llevar a cabo para concluir cada iteración.

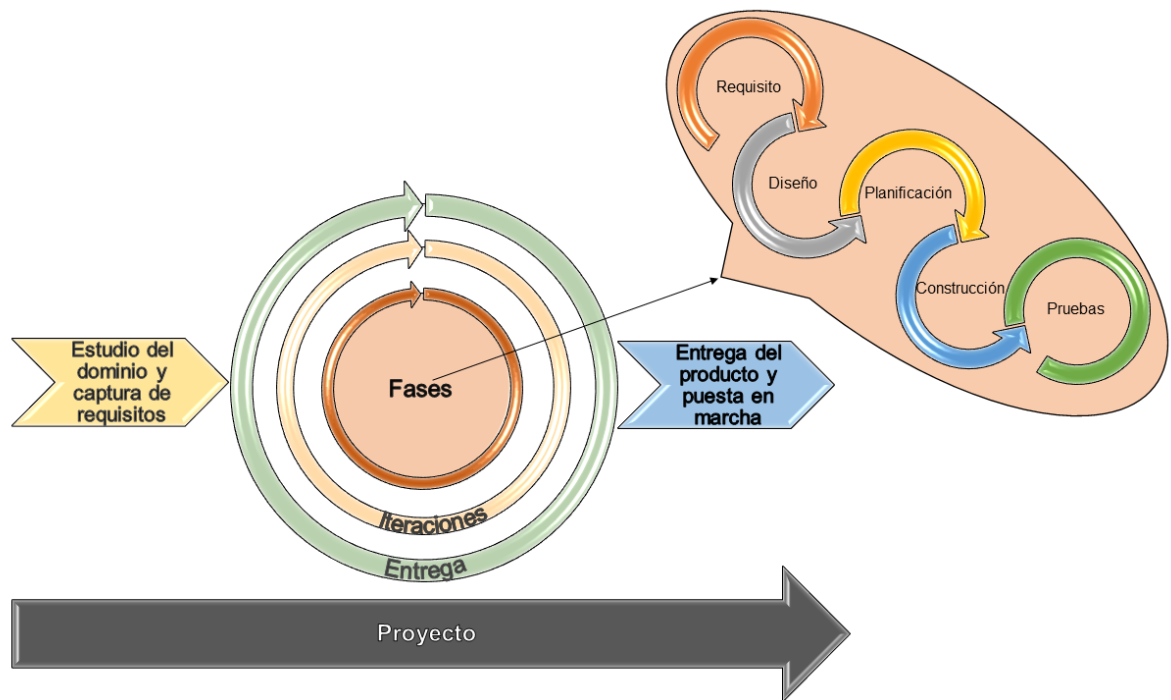


Figura 3.3: Proceso utilizado en el proyecto

3.2.3.2. La iteración

Como se explicó en el punto anterior, cada proyecto comprende de muchas “Entregas”, y cada una de ellas se realiza a través de lo que se denomina “Iteraciones” como se logra apreciar en la **Figura 3.3.** Este ciclo ayuda a tener control sobre el proceso de desarrollo y a disminuir los riesgos. Por otra parte, permite adaptar los cambios de requisitos en caso de haberlo.

En cuanto a la duración y estructura de cada iteración depende de las características del proyecto. Para efecto de esta memoria, cada iteración tiene un tiempo aproximado de tres semanas y está compuesta por fases (Requisitos, Diseño, Planificación, Construcción y Pruebas) las cuales se hablará en más detalle en el siguiente ítem.

3.2.3.3. Las fases

Como se mencionó en el ítem anterior, cada iteración cuenta con las siguientes fases:

- Requisito
- Diseño
- Planificación
- Construcción
- 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 iteración.

Se ha hablado de que existen fases, pero no se sabe que se realiza en cada una de ellas, es por ello por lo que a continuación se procede a contextualizar de que trata cada una:

Fase de requisitos: en esta etapa es donde el Product Owner se junta con el cliente para obtener información sobre las funcionalidades que tiene el proyecto (etapa comúnmente denominada “Captura de Requisitos”). Para esta fase se recomienda hacer pautas para mejorar el trabajo con el cliente.

Fase de diseño: tras la obtención de requisitos, se procede a realizar el diseño del sistema, la 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, la fecha de inicio y de fin. Y por último asignar estas tareas a un responsable de algún equipo de trabajo. Pero para efectos de esta memoria y como se ha mencionado con anterioridad, todas estas tareas recaen solo una persona que es el autor del proyecto.

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 2.4.3 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.

3.2.4. Documentación

Otro punto importante dentro de la metodología utilizada es que se basa en el desarrollo por sobre la documentación, es por ello que solo 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.

3.2.4.1. Reuniones

Tal como se dijo en el punto anterior, es que a veces se debe documentar las reuniones. Para efectos de esta memoria, las documentaciones que se realizan son:

- La captura de nuevos requisitos o modificaciones cuando se está en la reunión con el cliente.
- 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 las de vital importancia debido a que se cómo

se expuso anteriormente, 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.

3.2.5. Cómo se trabajó

4. Características del sistema

4.1. Requisitos del sistema

En esta sección se presentan los requerimientos del sistema realizados en Historias de Usuarios.

5. Diseño de la aplicación

5.1. Diseño

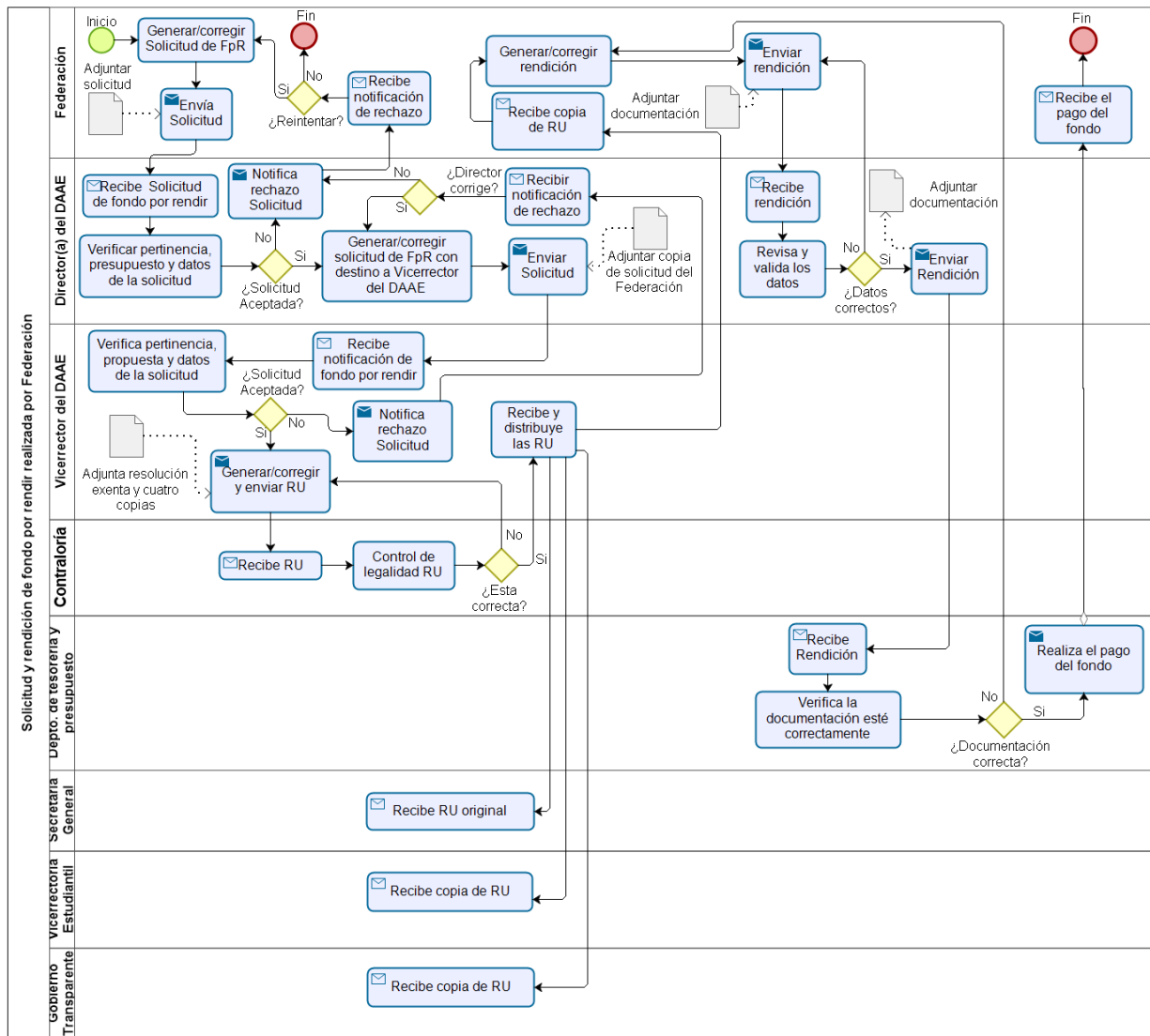


Figura 5.1: Proceso de Fondos por Rendir por parte de Fedeut.

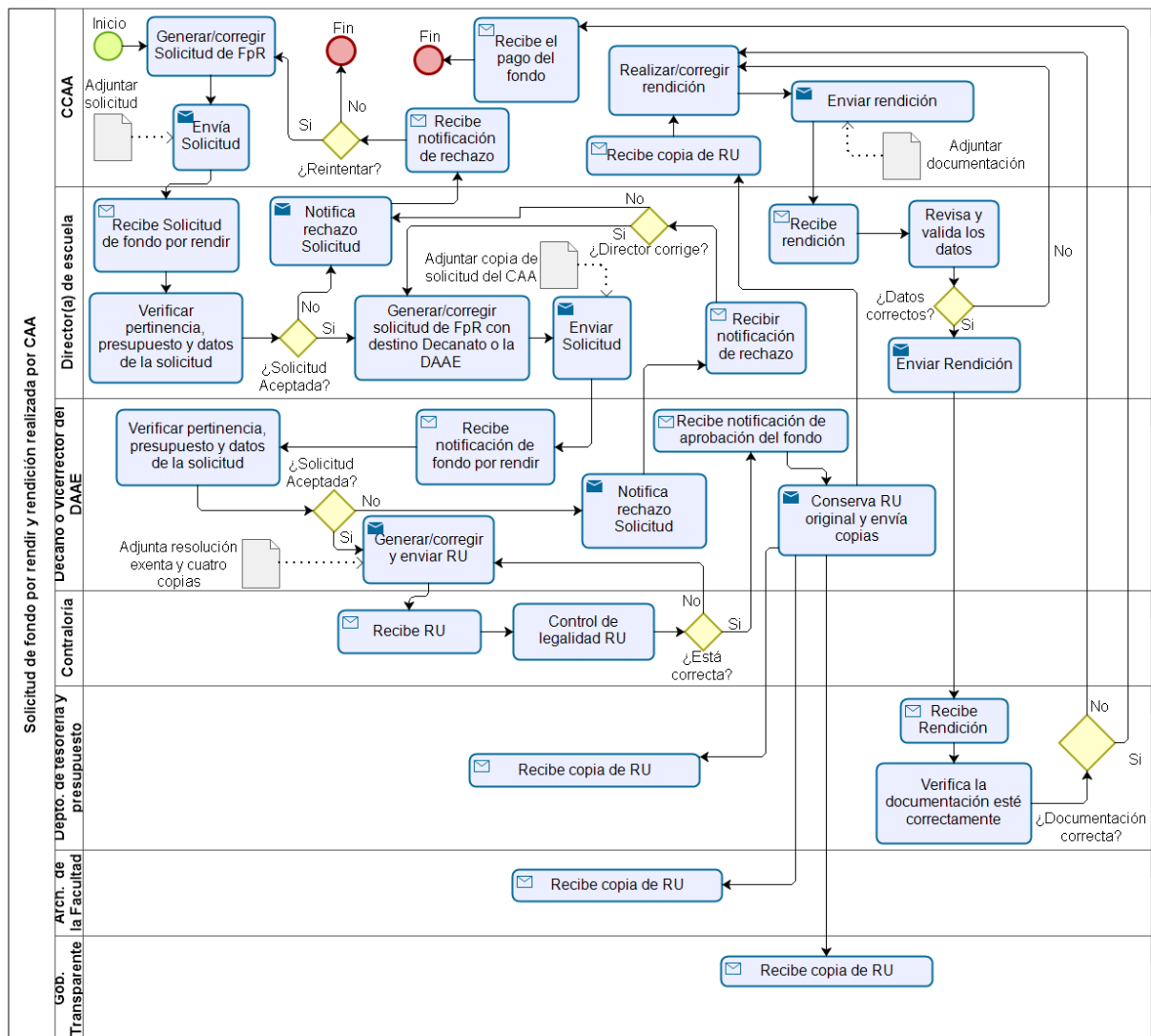


Figura 5.2: Proceso fondos por rendir por parte de CAA.

Glosario

Fondos: recursos económicos que la Universidad de Talca dispone para la realización de diversas iniciativas y/o actividades estudiantiles [6].

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 [6].

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 [6].

Decano: autoridad superior de la Facultad y dirige todos los asuntos académicos, administrativos y financieros de ella [6].

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) [6].

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 [6].

O.E.: Organizaciones Estudiantiles. Estas son federación de estudiantes, centros de alumnos o grupos intermedios [12].

CAA: Centro de alumnos. Organización estudiantil que representa a los alumnos de una carrera en particular [14].

Federación de estudiantes: Organización estudiantil que representa a los alumnos de un campus perteneciente a la Universidad [12].

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 [6].

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

RU: Resolución Universitaria.

Resolución Exenta: comprenden aquellas en que no es necesario que sean visadas por la Contraloría General de la República [14].

Visar: Dicho de la autoridad competente: Dar validez a un pasaporte u otro documento para determinado uso [8].

Engorroso(sa): adj. Dificultoso, molesto [9].

Burocrático(ca): adj. Perteneciente o relativo a la burocracia [10].

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

Bibliografía

- [1] Albert einstein and empirical software engineering.
<https://pdfs.semanticscholar.org/589a/8858795db3919bb1ca05ab44640ce038b76e.pdf>.
14 de Octubre de 2018.
- [2] Aliexpress. <https://cl.aliexpress.com/>. 25 de Noviembre de 2018.
- [3] Amazon. <https://www.amazon.es/>. 25 de Noviembre de 2018.
- [4] Controlaría universidad de talca. <http://contraloria.utralca.cl/html/documentos.php?cat=26>.
9 de Octubre de 2018.
- [5] Correos chile. <https://www.correos.cl/SitePages/home.aspx>. 25 de Noviembre de 2018.
- [6] Glosario. <http://contraloria.utralca.cl/html/documentos.php?cat=29>. 9 de Octubre de 2018.
- [7] 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.
25 de Noviembre de 2018.
- [8] Rae. <http://dle.rae.es/srv/search?w=visada>. 14 de Octubre de 2018.
- [9] Rae. <http://dle.rae.es/?w=engorroso>. 2 de Enero de 2019.
- [10] Rae. <http://dle.rae.es/?id=6JjuZI8>. 2 de Enero de 2019.

- [11] Rae. <http://dle.rae.es/?id=6Jix9ml>. 2 de Enero de 2019.
- [12] Vicerrectoria de desarrollo estudiantil. <http://vde.utalca.cl/html/organizaciones.html>. 9 de Octubre de 2018.
- [13] Wish. <https://www.wish.com/>. 25 de Noviembre de 2018.
- [14] Wordreference. <https://forum.wordreference.com/threads/resoluci%C3%B3n-exenta.1164374/?hl=es>. 14 de Octubre de 2018.
- [15] Universidad de Talca. RU N °2083. pages 1–8, 2017. 9 de Octubre de 2018.
- [16] Ph.D. Roger S. Pressman. *Ingenieria del software. Un enfoque practico*. Mc Graw Hill, 2010. 11 de Octubre de 2018.

ANEXOS

A. El Primer Anexo

Aquí va el texto del primer anexo...

A.1. La primera sección del primer anexo

Aquí va el texto de la primera sección del primer anexo...

A.2. La segunda sección del primer anexo

Aquí va el texto de la segunda sección del primer anexo...

A.2.1. La primera subsección de la segunda sección del primer anexo

B. El segundo Anexo

Aquí va el texto del segundo anexo...

B.1. La primera sección del segundo anexo

Aquí va el texto de la primera sección del segundo anexo...