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

ÍNDICE DE FIGURAS

página

ÍNDICE DE TABLAS

página

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, envío de la Resolución Universitaria y 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 (O.E.) reconocidas. Entre las actividades destacan: recepción de alumnos nuevos; celebración del día de la carrera; actividades sociales, culturales y deportivas; entre otras [18].

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 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. 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 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. Si esta última es negativa, la devolución del dinero en caso de reembolso o la entrega del dinero por anticipado 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 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 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

Dado que este sistema pretende aliviar todos los inconvenientes *de forma* en que se pueda incurrir tanto al confeccionar la solicitud de fondo por rendir como en su posterior rendición, se estabecen 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, *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 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 cambios de integrantes en las directivas de las O.E. se debe realizar una capacitación para que los encargados puedan realizar las 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" [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 [16], 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 en 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 compresión 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 profundizar lo explicado en la **Sección 1.1** *Contexto del problema*, es que hace mención a los conceptos claves para la comprensión del 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 [18].

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 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. Estas pueden 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 trasferencias de fondos para llevar a cabo actividades por parte de las OE, al momento de que se acepta la solicitud enviada por éstas.

En esta 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.

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 detallada de todos los gastos que incurrieron en una actividad o evento realizado por la OE. En el encabezado del documento se debe indicar lo siguiente: (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, (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 [18].

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 lenguaje 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 ademas 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 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.

- **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 necesarioas 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.

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 al conjunto de medios que se pueden utilizar para llegar a un fin para luego determinar cuál es el más adecuado se le denomina metodología.

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 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 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.

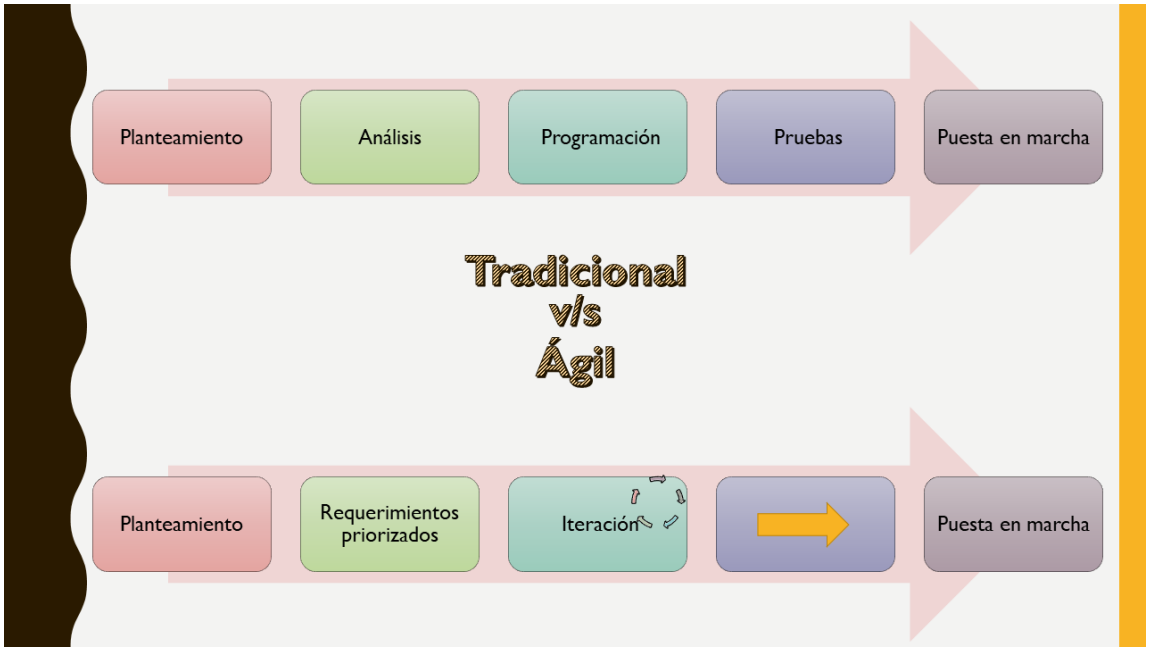

Figura 2.1: Metodología tradicional vs metodología ágil [18].

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 da mayor énfasis en el siguiente ítem de este documento.

Cuadro 2.1: Metodología Tradicional vs Metodología Ágil [18].

	Tradicional	**Ágil**
Enfoque	Predictivo	Adaptativo
Éxito de medición	Conformación de planificar	Valor del negocio
Tamaño del proyecto	Grande	Pequeño
Estilo de gestión	Autocrático	Desentralizada
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 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 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 multidisciplinares 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, pero 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

Scrum[1] 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 [19].

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 seleccionan una lista de historias de usuarios (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.

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 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:

- **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,

[1]Nombre 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.

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 o retrasos:** lista de historias de usuarios que se pueden agregar elementos a los retrasos en cualquier momento. El PO evalúa los retrasos y actualiza las prioridades de esta lista de acuerdo con lo requerido.

  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 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 usuarios son de alto nivel y no pueden tener detalles de implementación o técnicos. Además, están asociados 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 y que 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.

- **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. 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.

2.4.5. 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.

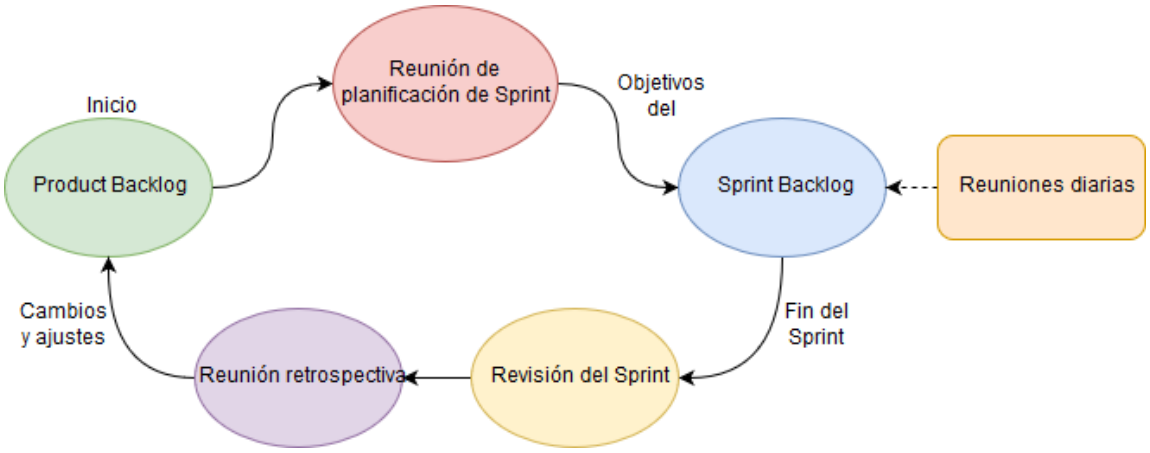

Figura 2.2: Reuniones en el Proceso de Scrum.

- **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; 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 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?

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 contempla muchos sub-ciclos denominados “Entregas” o “Incrementos”, que corresponden a nuevas funcionalidades realizadas y mejoras continuas a lo largo

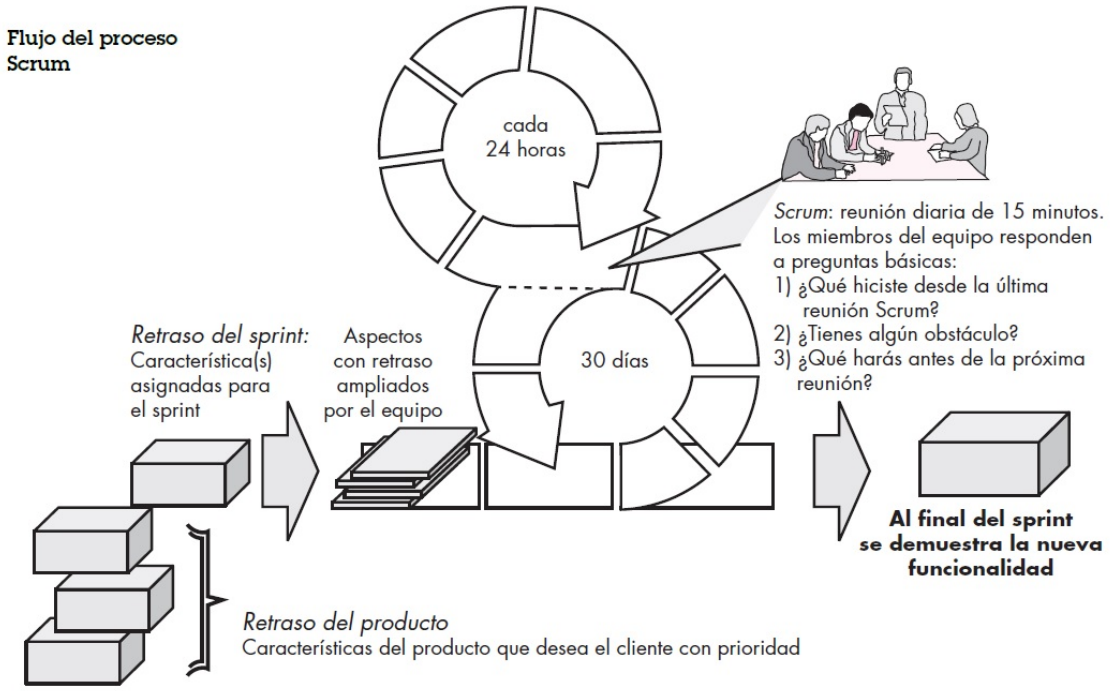

Figura 2.3: Proceso de metodología Scrum.

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 iteración.

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

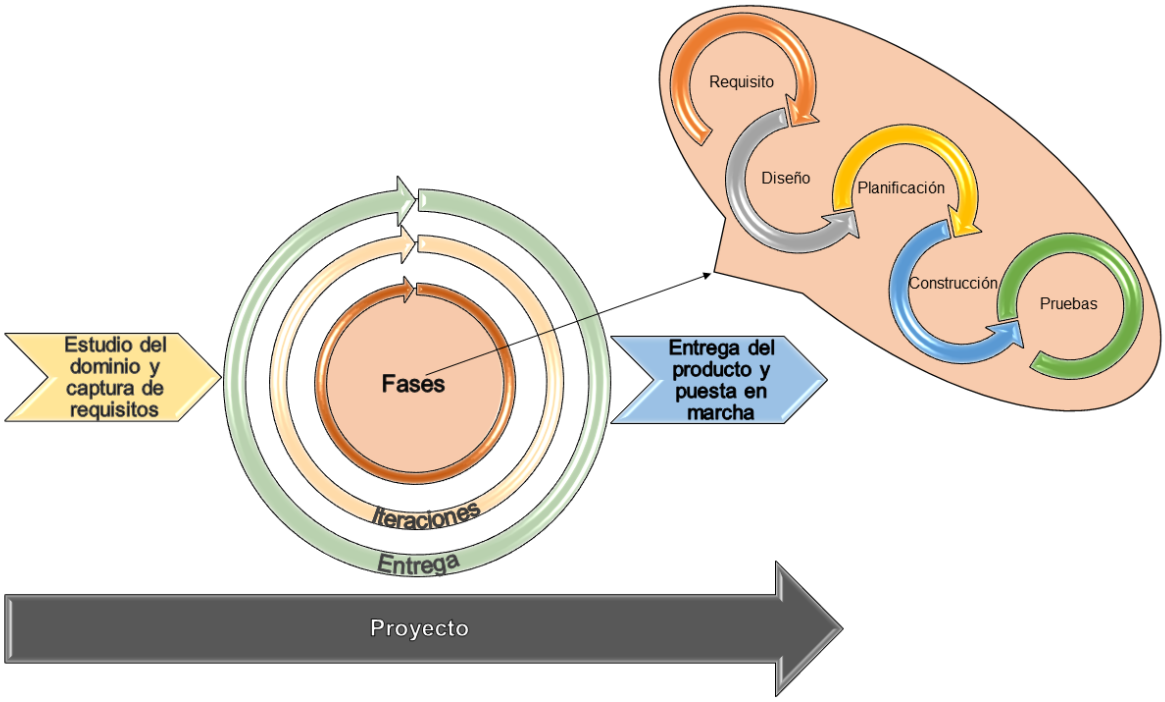

Figura 2.4: Proceso utilizado en el proyecto.

Fase de requisitos: en esta etapa es donde el PO se reune 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 cortas, 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

Para el proyecto, se utiliza como método de evaluación la metodología con experimentos controlados. Esta maneja pruebas que se hacen bajo condiciones totalmente constantes 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], experimentar con la construcción de software permite "Aumentar la compresión 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 tarea que se encuentra en el tablero, aparece como una tarjeta Kanban [10].

[2]El 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 el 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 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 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 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 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 segun 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). Ahora, dado que las OE 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 se conserva. 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 ayuda 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 subdividad 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.

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

Figura 3.2: Proceso de trabajo de un Sprint.

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 sobre 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 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.
- 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 secretarias 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 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 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 discute en detalle cómo se aplica la metodología al proyecto y la planificación de este.

3.2.2.1. El proyecto

Para comenzar el proyecto, lo primero que se realiza es identificar el problema para ver cuál es el enfoque en el que se debe solucionar. Luego se procede a capturar las Historias de Usuarios que representan las características del proyecto. Tras tener una noción general de lo que el proyecto se procede a categorizar las Historias de Usuario creando los sprints. Estos son los que dan vida al proyecto a medida que se van terminando los sprints y cumpliendo con los tiempos planificados y siguiendo las fases que se muestran el en la **Figura 2.4** (en la siguiente sección se habla con más detalle sobre esto).

Para efectos de esta memoria, los sprints están planificado de la siguiente manera:

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.

3.2.2.2. Iteraciones del proyecto

El proyecto comprende de 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 usuarios 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 sprint 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 incurrirán 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**. Específicaente, 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 que son los presidentes y secretarios de finanzas de las OE y las distintas Secretarias 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 realizó fue 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 observó 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 proposito de conocer y conseguir la mayor cantidad de requisitos e información del sistema.

Luego de obtener los requisitos que debe cumplir el sistema se realizó 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 procedió 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 lo fue) 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 tienen planificada realizar en esta, se deben ver en detalle y confirmar con el cliente. También se realizan entrevistas a las Secretarias 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 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 como fue la metodología de desarrollo y como se aplicó 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 realizó.

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 tres copias son distribuidas a la OE interesada, al Gobierno Trasparente y a la Vicerrectoría Estudiantil.

5. 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.

6. 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 y pago.

7. 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.

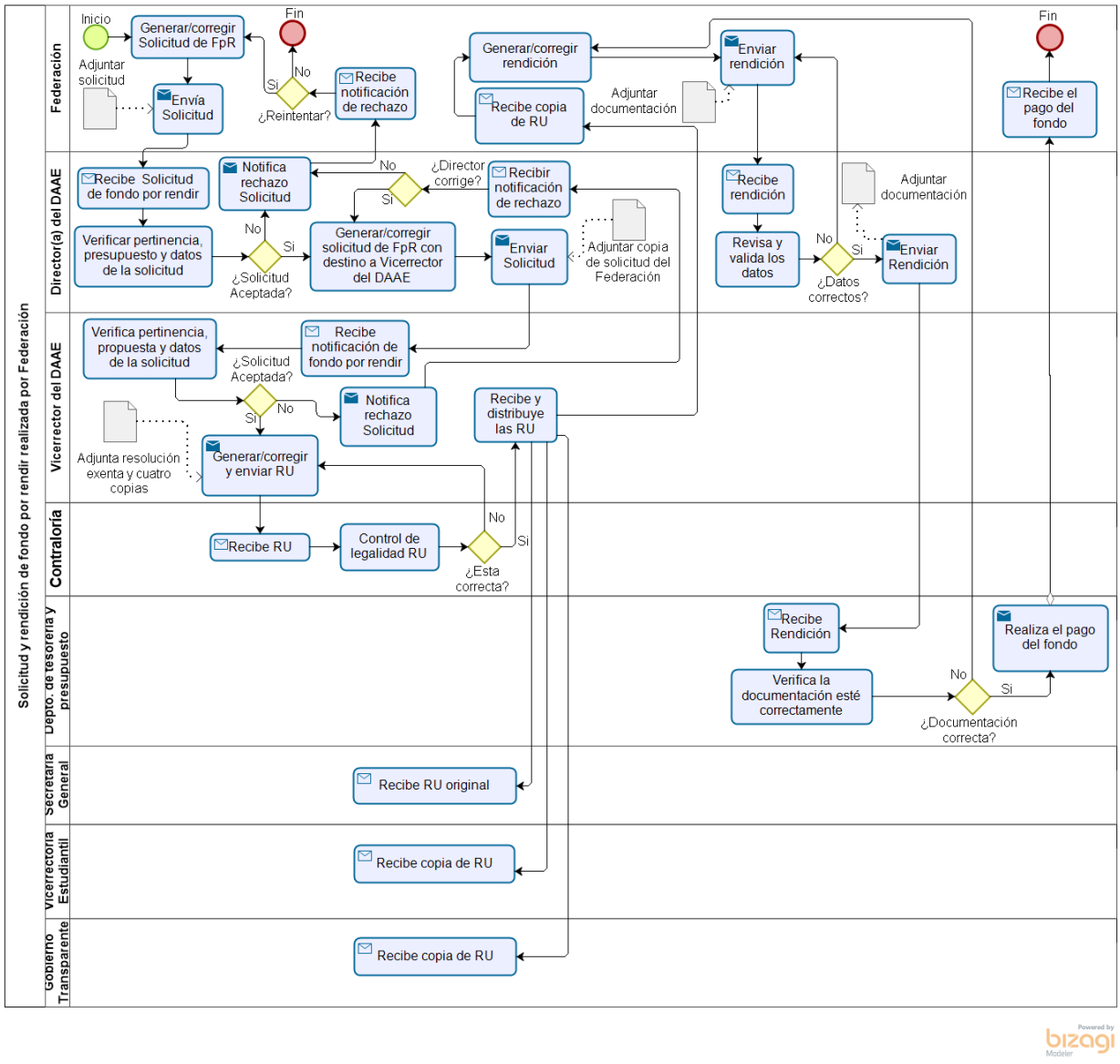

Figura 4.1: Proceso de Fondos por Rendir por parte de Fedeut.

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

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

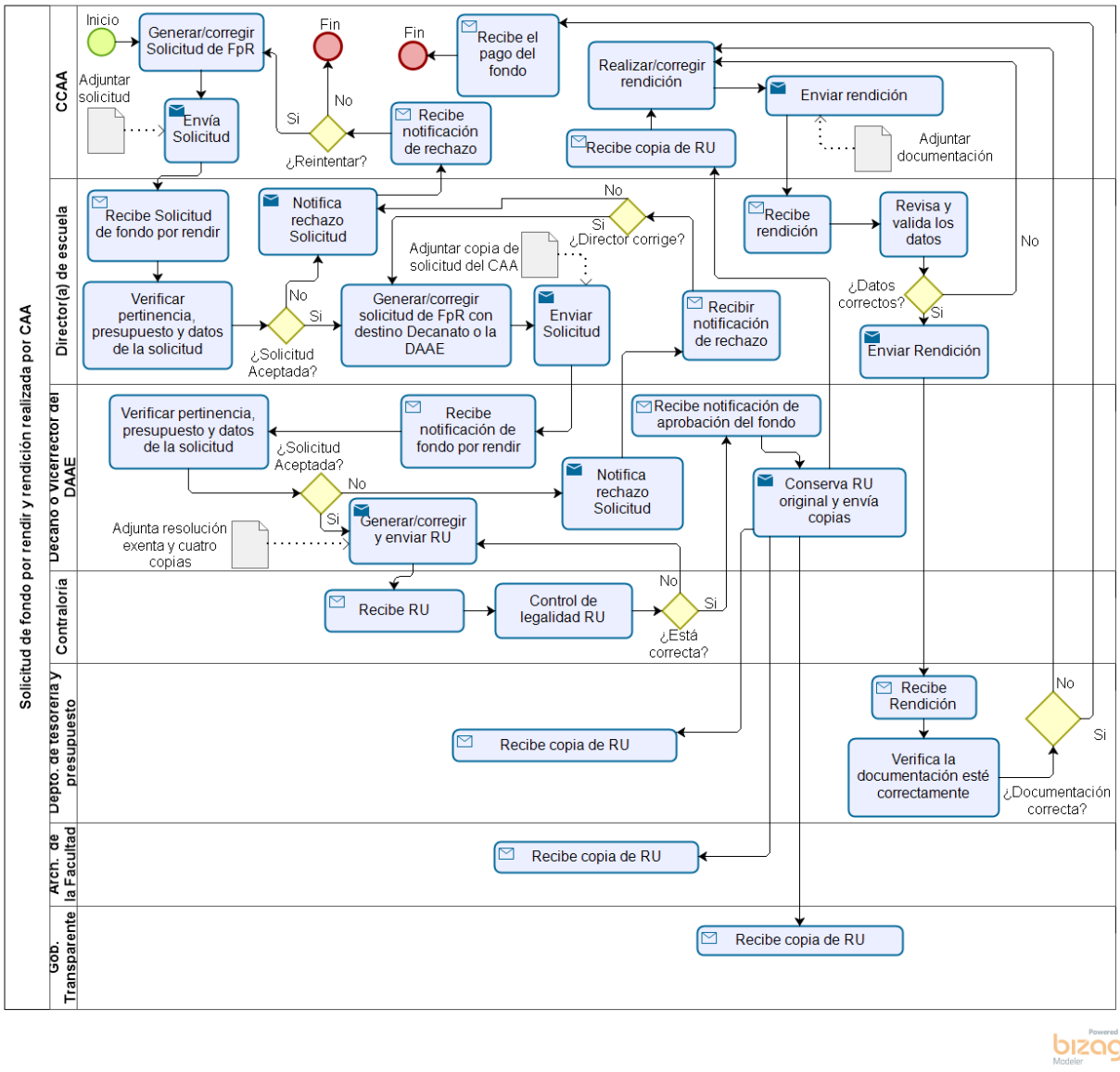

Figura 4.2: Proceso fondos por rendir por parte de un CAA.

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 al Gobierno Trasparente.

5. 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.

6. 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 y pago[1].

7. 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.

[1]Existen dos modalidades de pago en la solicitud de los fondos, pago por anticipado al evento y reembolso de gastos posterior a la rendición. Tradicionalmente, en la Escuela de Ingenería Civil en Computación se utiliza la modalidad de reembolso, que es la que se ilustra en las Figuras 4.1 y 4.2.

- 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**).

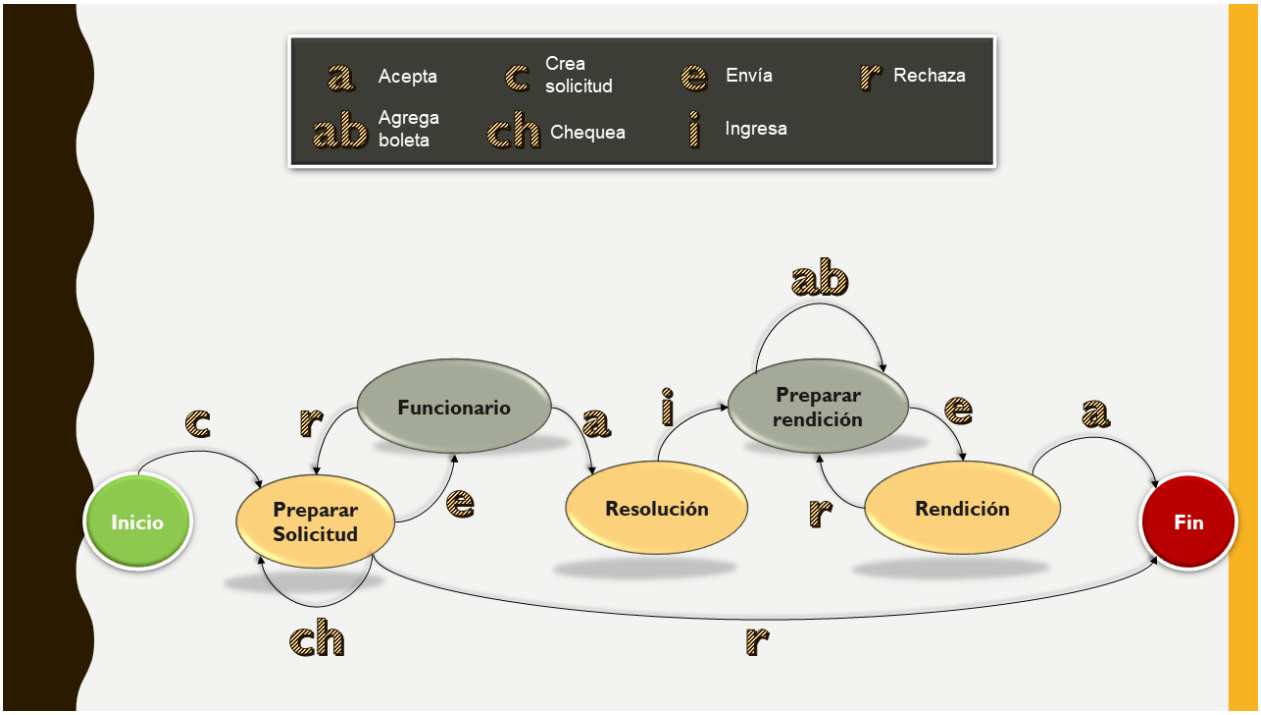

Figura 4.3: Diagrama de estado del proceso que debe simular la aplicación.

El diagrama de estado que se puede observar en la **Figura 4.3** muestra las actividades que sólo se encarga la OE las cuales son la confección de la Solicitud, recibir la Resolución emitida por la Universidad y la posterior Rendición del fondo por rendir. El resto de las actividades que se muestran en las **Figuras 4.1** y **4.2** son responsabilidad de las demás entidades involucradas en el proceso.

A grandes rasgos el sistema debe permitir simular el proceso de una Solicitud de Fondo por Rendir. Este proceso conlleva la confección de una Solicitud e imprimirla en PDF. En caso de que la solicitud fue aceptada, la OE recibe la RU enviada por parte de la Casa de Estudio, la cual debe subir una copia digitalizada de esta al

sistema. Por último, debe confeccionar la Rendición de gastos que incurrieron en el evento e imprimirla en PDF.

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 4.2** y **4.3**. 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 4.2** y **4.3** se redactan s4egun el primero que la menciona.

4.4. Planificación de las iteraciones

Tras realizar el estudio previo de la situación actual de la problematica 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 4.4** muestra cómo se distribuyen las historias de usuarios a través de las diferentes iteraciones.

Cuadro 4.2: Historias de usuario - Secretario de Finanza

Código	Yo, como Secretario de Finanza 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 incurridos de 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 gatos 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 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 este 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 4.3: Historias de usuario - Presidente

Código	Yo, como Presidente quiero...
HU-PDTE-1	tener respaldo de rendición 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 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 como 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 al que se invertirá 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.

Cuadro 4.4: 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 de esta última, las cuales se dieron a conocer en el **Capítulo 4**, se procede a realizar el diseño de la aplicación. Esta busca definir el mapa de navegación, las interfaces de usuarios, la arquitectura, el diagrama de clases y el modelo de datos. 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 mampa 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 busqueda 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**).

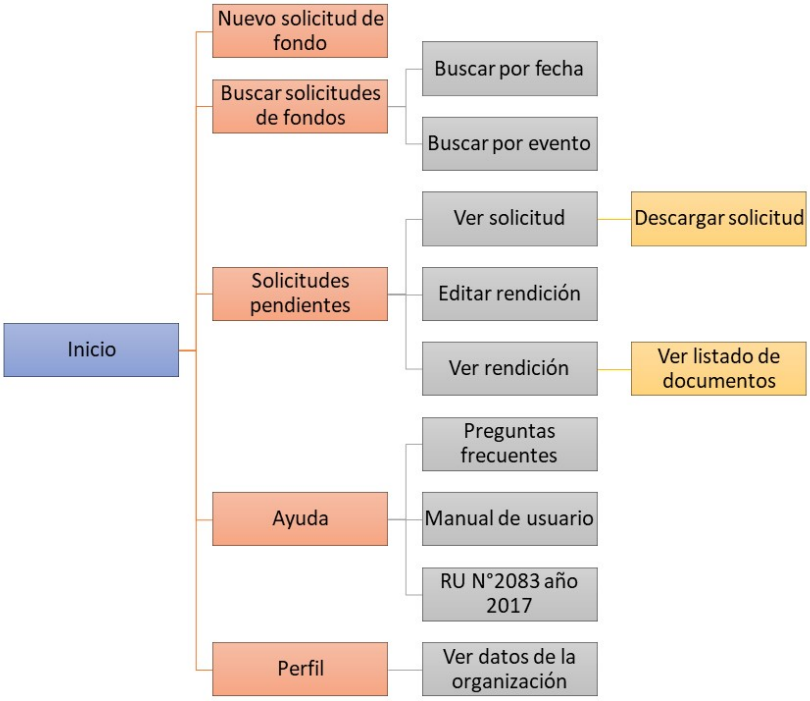

Figura 5.1: Mapa de navegación de la aplicación.

Solicitudes pendientes: vista ayuda al usuario al rápido acceso de 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 ayuda, el botón para cerrar sesión y el botón de perfil de usuario. 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 selecciona el usuario 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 como 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 obserar en la **Figura 5.3**, la arquitectura utilizada para el sistema es de tres capas: cliente, servidor, 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 PC o Laptop. En cuanto al servidor, se puede destacar que el sistema está montado en un servidor diferente al de la base de dato con el fin de que lo datos no vean comprometidos en caso de que vulneraran el servidor web.

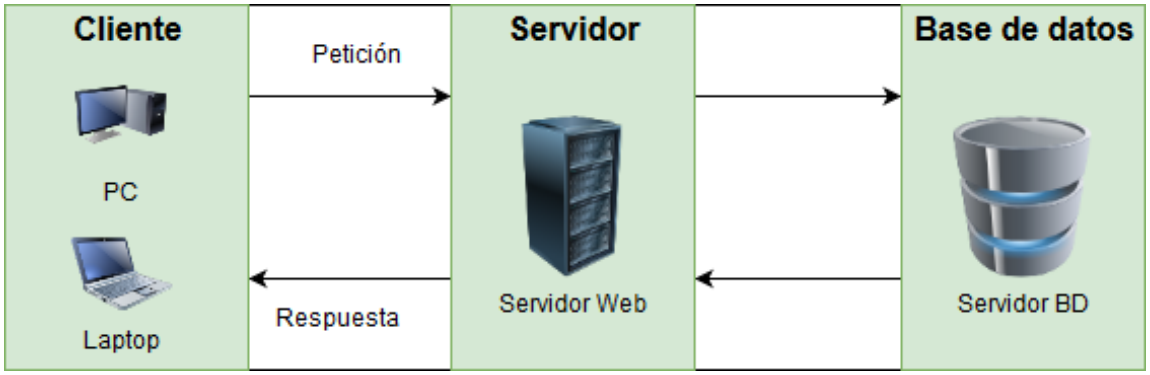

Figura 5.3: Arquitectura física del sistema.

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 la arquitectura lógica la cual para efectos de esta memoria es **Modelo-Vista-Controlado (MVC)**. La elección de esta arquitectura se debe a que separa la interfaz de usuario de la lógica de negocio y el modelo de datos del sistema. Esto se realiza con el fin de evitar acoplamiento y facilitar los futuros mantenimientos dado a 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

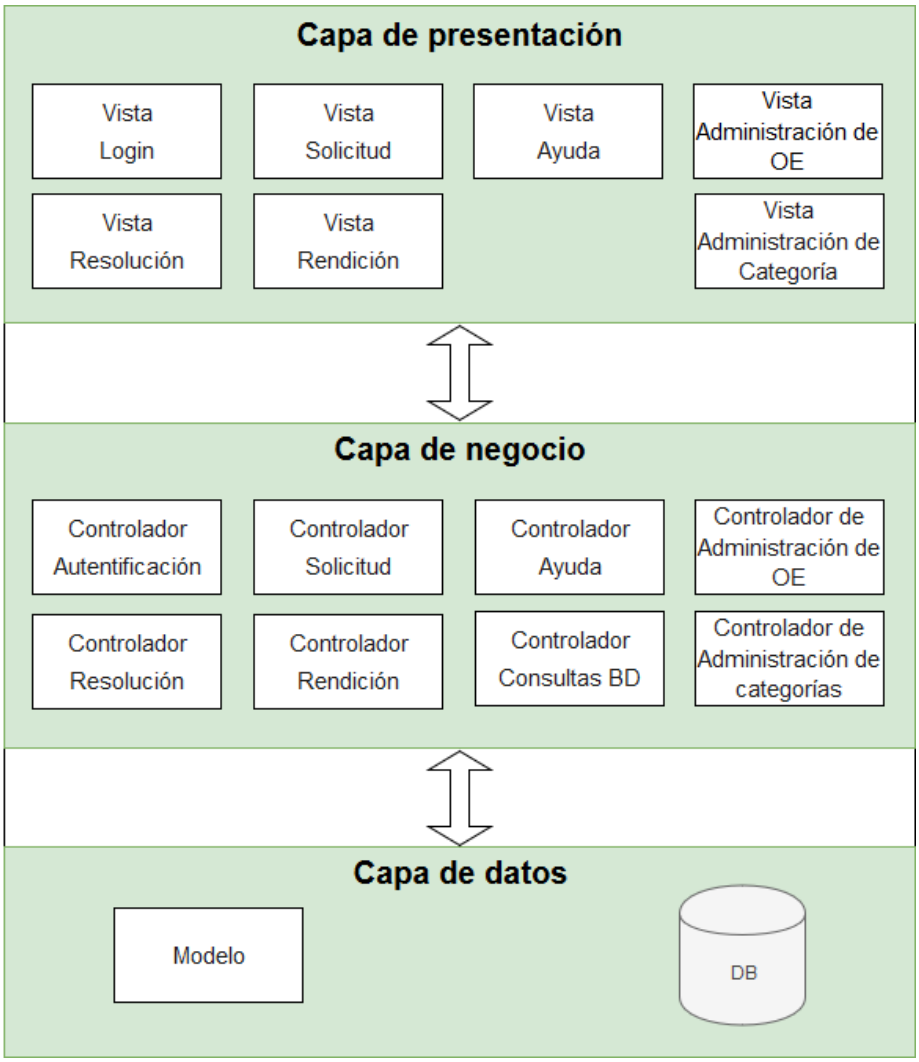

Figura 5.4: Arquitectura lógica del sistema.

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 encuentra en la capa de Modelo y las otras capas no se verán afectadas.

- **Capa de presentación:** Esta capa contempla todas las interfaces de usuarios o vistas que posee el sistema. Cabe destacar que existen dos roles en el sistema, el de usuario comun y la de administrador, es por ello que la distribución de las vistas segun 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 que pueden interactuar ambos usurios 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 Autentificación: encargado de obtener y corroborar los datos de acceso de los distintos usuarios.
  - Controlador Solicitud: encargado de obtener y manipular los datos que confeccionan una solicitud que ingresa la OE interesada para enviarlos al Controlador Consultas BD y asi sean registrados.
  - Controlador Resolución: encargado de obtener y manipular los datos de la Resolución enviada por la Intitución que acredita la aceptación de la Solicitud realizada por la OE.
  - Controlador Rendición: encargado de obtener y manipular todos los datos que conlleva una Rendicion, entre los cuales se considera, 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 reqgistre los datos ingresados.
  - Controlador Ayuda: encargado de obtener la información de manuales, 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, solo obtiene la informacion que se encuentra en el servidor.
  - Controlador de Administracion 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 categorias que serán utilizadas en la Solicitud y en la Rendición para mencionar en que fueron incurridos los gastos.

- Controlador Consultas BD:encargado de obtener los datos enviados desde los controaldores para reenviearlos 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 dato de la aplicación y las conexiones hacia la Base de datos.

5.4. Diagrama de clases

5.5. Modelo de datos

6. Construcción y validación

6.1. Aplicación obtenida

6.2. Pruebas

6.3. Resultados

Glosario

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

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].

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

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].

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].

OE: Organizaciones Estudiantiles. Estas son federación de estudiantes, centros de alumnos o grupos intermedios [15].

CAA: Centro de alumnos. Organización estudiantil que representa a los alumnos de una carrera en particular [17].

Federación de estudiantes: Organización estudiantil que representa a los alumnos de un campus perteneciente a la Universidad [15].

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].

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 [17].

Visar: Dicho de la autoridad competente: Dar validez a un pasaporte u otro documento para determinado uso [11].

Engorroso(sa): adj. Dificultoso, molesto [12].

Burocrático(ca): adj. Perteneciente o relativo a la burocracia [13].

Burocracia: Administración ineficiente a causa del papeleo, la rigidez y las formalidades superfluas [14].

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

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.utalca.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] draw.io. https://www.draw.io/. 15 de mayo de 2019.

[7] Glosario. http://contraloria.utalca.cl/html/documentos.php?cat=29. 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. 25 de Noviembre de 2018.

[9] Instructivo organizaciones estudiantiles utalca. http://vde.utalca.cl/docs/Instructivo_Organizacion 30 de abril de 2019.

[10] kanbanize. https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tablero-kanban/. 15 de mayo de 2019.

[11] Rae. http://dle.rae.es/srv/search?w=visada. 14 de Octubre de 2018.

[12] Rae. http://dle.rae.es/?w=engorroso. 2 de Enero de 2019.

[13] Rae. http://dle.rae.es/?id=6JjuZI8. 2 de Enero de 2019.

[14] Rae. http://dle.rae.es/?id=6Jix9ml. 2 de Enero de 2019.

[15] Vicerrectoria de desarrollo estudiantil. http://vde.utalca.cl/html/organizaciones.html. 9 de Octubre de 2018.

[16] Wish. https://www.wish.com/. 25 de Noviembre de 2018.

[17] Wordreference. https://forum.wordreference.com/threads/resoluci %C3 %B3n-exenta.1164374/?hl=es. 14 de Octubre de 2018.

[18] Universidad de Talca. RU N °2083. pages 1–8, 2017. 9 de Octubre de 2018.

[19] 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...