UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

INTEGRACIÓN DE HERRAMIENTAS DE APOYO AL DESARROLLO DE SOFTWARE EN CURSOS DE PREGRADO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

SEBASTIÁN ANDRÉS DONOSO DÍAZ

PROFESOR GUÍA:
SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:
BENJAMIN BUSTOS CÁRDENAS
JOSÉ BENGURIA DONOSO

SANTIAGO DE CHILE
2022

Resumen

Durante más de 10 años, en el curso CC5401 (Ingeniería de Software II) los alumnos han utilizado herramientas de software ad hoc para llevar adelante los proyectos que realizan durante un semestre. Particularmente, utilizan una herramienta de conformación de equipos de trabajo, una para coevaluar la labor de sus compañeros de equipo, y otra para gestionar los requisitos de los proyectos que llevan adelante. Estos sistemas requieren manejar de forma integral y coordinada a los usuarios (alumnos, profesores y auxiliares) y equipo de trabajo, en pos de mantener la coherencia entre ellos.

Estos sistemas fueron desarrollados en distintos momentos, por distintas personas, sin conexión entre ellos. Por lo tanto, para utilizarlos en conjunto, el profesor del curso y/o los auxiliares deben cargar o traspasar la información de usuarios y equipos de forma manual, lo cual es una labor tediosa, que insume bastante tiempo y eventualmente genera errores. Esta labor manual requiere también expertise por parte de quienes la ejecutan. Estas restricciones limitan su uso en otros cursos, por parte de otros profesores, o en otras instituciones educativas.

Para ayudar a paliar estas limitaciones, en este trabajo de memoria se desarrolló un sistema integrador de usuarios y equipos de trabajo, el cual se acopló a través de una API a los sistemas antes mencionados, permitiendo la evolución autónoma de las partes, al mismo tiempo que se mantiene la capacidad de operación conjunta de ellos.

El sistema integrador consume información automáticamente del data lake de docencia, y alimenta a los otros sistemas, ya sea con información primaria, como con información procesada (por ejemplo, equipos de trabajo o roles de los miembros de dichos equipos). En ese sentido, el nuevo sistema representa la única fuente de verdad para los sistemas legados antes mencionados.

El sistema fue evaluado en dos aspectos importantes: 1) el correcto traspaso de información de los cursos y grupos formados entre los sistemas, y 2) que los sistemas legados sigan manteniendo su funcionalidad actual. Para lo primero fue necesario hacer un seguimiento paso a paso de la consistencia de los datos en cada traspaso, así como verificar que los datos quedarán bien almacenados y listos para ser utilizados en cada sistema. Para lo segundo, se hicieron diversas pruebas y se comparó el comportamiento del sistema modificado y el que estaba en producción. Como resultado, se encontró que el comportamiento de ambos sistemas era equivalente, y no se identificaron problemas causados por las modificaciones realizadas.

Dado que el sistema integrador automatiza gran parte de las labores manuales, el costo y la cantidad de errores asociados a la labor de traspaso de usuario y equipos entre sistemas es prácticamente cero. Por lo tanto, podemos decir que los objetivos inicialmente planteados en la memoria fueron alcanzados. Esto abre diversas oportunidades para que dichas herramientas puedan ser utilizadas en otros cursos, por otros equipos de cátedra.

El sistema implementado es una solución *ad hoc* a la problemática actual, y por lo tanto puede ser mejorado, o incluso integrado a los sistemas legado en un futuro.

Agradecimientos

A mi familia, que siempre me dieron apoyo y fuerzas para seguir adelante incluso cuando todo se veía feo, ayudando aunque no supieran como a este testarudo hijo y hermano con el que tenían que convivir.

A Sergio, mi profesor guía que siempre me traía de vuelta a la tierra cuando mi cabeza se ponía a volar o se angustiaba por tonteras durante estos meses de trabajo.

A mis amigos de sección, las tardes de risas, papitas y estudios fueron una buena motivación para seguir a pesar de lo duro de los ramos.

A la gente maravillosa del DCC que he conocido en estos años, las tardes de Smash en la salita, los juegos de rol, los paseos y karaokes hicieron que el infierno del plan común valiera la pena soportarlo.

A mis queridos Feeders, a los que el destino nos unió por coincidencia el 2013 por un juego y aun no nos separan, que nos acompañamos en los momentos buenos y malos, en las aventuras locas y en las fomes y en mis juegos de mesa eternos que salían mal.

A la Fabi y a la Ale, por ser pilares importantes en mi vida, patitos debuggers, sicologas, fuentes de memes y risas en estos años turbulentos que nos han tocado vivir.

A la Javi, mi compañera en esos primeros años tan difíciles que me tocaron vivir cuando entre a la universidad. Y aunque la vida y las decisiones de cada uno nos terminaron separando, sin su compañía y la falta de esta, no hubiera podido juntar las fuerzas para avanzar y seguir creciendo como persona.

A mi abuela Nona, que siempre orgullosa de su nieto no pudo alcanzar a verme titulado. Te amo abuela.

Y a todos los demás que no alcanzo a nombrar en una sola hoja de agradecimientos, ¡Al fin me titularé! :D

Tabla de Contenido

Capítulo 1

Introducción

En el curso CC5401 (Ingeniería de Software II), al igual que en otros cursos del Departamento de Ciencias de la Computación de la Universidad de Chile, los alumnos trabajan en equipo realizando un proyecto durante un semestre; para ello, se apoyan en diversas herramientas de software. En el caso particular del curso antes mencionado, los alumnos deben utilizar tres herramientas en distintas etapas del proyecto:

1. *Conformación de Equipos (https://equipos.dcc.uchile.cl/login/)*. A través de esta herramienta los alumnos realizan un test psico-social. Los resultados del test permiten conformar equipos de trabajo entre personas psico-socialmente compatibles, y además balancear los equipos en términos de los atributos de sus miembros. Esta herramienta fue desarrollada en la tesis de Magíster en Ciencias de la Computación del Sr. Luis Silvestre [4].
2. *MainReq (https://mainreq.dcc.uchile.cl/)*. Esta aplicación permite gestionar los requisitos, el diseño, los casos de prueba, y la relación entre ellos. Por lo tanto, dicha aplicación se usa para documentar y hacerle seguimiento a un proyecto. Esta herramienta fue desarrollada como parte de la memoria de ingeniería civil en computación del Sr. Francisco Hafón [2].
3. *Coevaluaciones (http://coevaluaciones.dcc.uchile.cl/).*Esta aplicación permite que los miembros de un equipo puedan evaluar de forma anónima el desempeño de sus pares utilizando una rúbrica predefinida. Se realizan tres coevaluaciones durante la realización del proyecto. Los resultados de éstas son informados a los miembros que corresponda (destinatarios de la evaluación), a modo de retroalimentación. Dicho *feedback* es entregado de forma automática por el sistema. Esta herramienta fue desarrollada como parte de la memoria de ingeniería civil en computación del Sr. Roberto Riquelme [3].

1.1. Problema abordado

Estos sistemas fueron desarrollados en distintos momentos, y diseñados para operar como islas (sin comunicación entre ellos). Debido a eso, para poder utilizar estas herramientas, cada

semestre el profesor del curso (y/o sus ayudantes) deben ingresar a mano los integrantes de los cursos y agruparlos según los equipos a los que fueron asignados, lo cual representa un esfuerzo de configuración cada vez más importante considerando el creciente número de estudiantes del curso. Por otra parte, la información de entrada requerida por cada herramienta (es decir, la lista de alumnos del curso) es diferente entre ellas, lo cual también dificulta el proceso de alimentación.

Además de lo antes mencionado, este esfuerzo de configuración para permitir que los equipos utilicen dichas herramientas, limita su uso en otros cursos, e inclusive, en otras universidades. Particularmente, la herramienta de conformación de equipos y la de coevaluaciones podrían ser utilizadas en cualquier curso donde los alumnos trabajen en equipos ejecutando un proyecto. En semestres pasados ha habido interés de profesores del DCC, así como de otros Departamentos (inclusive de otras universidades) por utilizar algunas de estas herramientas. Sin embargo, el esfuerzo y el conocimiento requerido para realizar la configuración inicial, ha impedido llegar a utilizarla en la práctica, excepto en el curso CC5401 donde se tiene el *know-how* respecto a cómo hacer esta tarea.

Este trabajo de título busca crear una solución que integre el flujo de datos que comparten los sistemas mencionados anteriormente, incorporando además la base de datos de docencia como un cuarto actor en este proceso y congregar todas las interacciones entre los distintos sistemas en una nueva plataforma. La idea de esta plataforma es simplificar el acceso a usuarios no expertos a cada uno de los sistemas mencionados y ampliar su uso más allá del curso CC5401, además de permitir la fácil integración de sistemas y herramientas adicionales en el futuro en caso de necesitar.

Estos sistemas forman parte de un mismo flujo de datos compartido, que se lleva a cabo en forma manual (Figura 1). Particularmente, la información que es generada en el sistema de creación de equipos, se utiliza luego para alimentar los sistemas de coevaluaciones y de seguimientos del proyecto (MainReq). La alimentación manual y las diferencias de formatos entre los inputs requeridos por cada sistema, acarrean problemas para usuarios no expertos en el sistema; esto es, profesores y/o auxiliares de los cursos que desean usarlas.

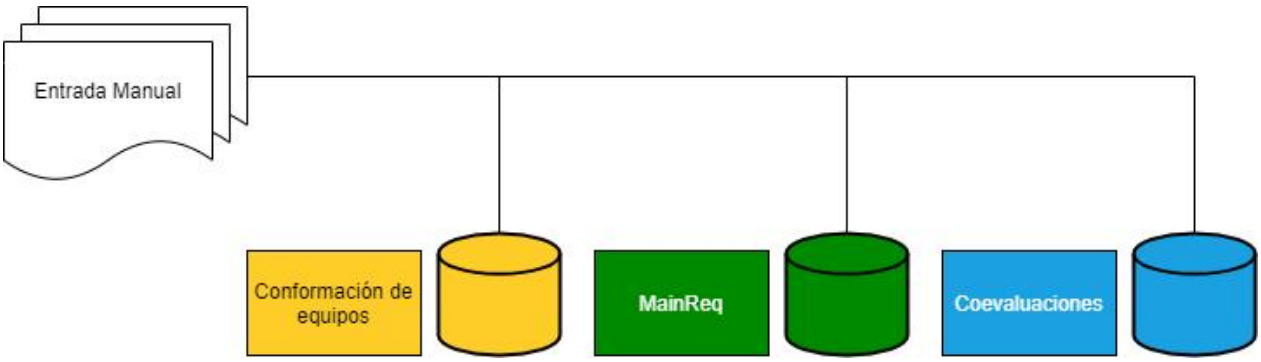

Figura 1.1: Escenario de trabajo (situación actual)

Por otra parte, existe ya disponible la primera versión de una plataforma de gestión de la docencia del DCC. Esta plataforma obtiene la información de cursos y estudiantes del Departamento directamente desde la plataforma U-Campus, y los almacena en una base de datos local. Esa iniciativa abre una oportunidad para resolver muchos problemas, y particu-

larmente el declarado en esta memoria, pues la lista de estudiantes de un curso es consumida directamente desde dicha base de datos.

Dado que las herramientas que son parte de la integración fueron desarrolladas de manera independiente entre ellas y hace varios años, se consideran actualmente como sistemas legados, ya que se encuentran en funcionamiento actualmente sin grandes cambios y solo se realiza mantenimiento de éstas cuando ocurre algún error mayor.

Respecto a la disponibilidad de los sistemas, estos se encuentran alojados en los servidores del DCC. Por temas operativos, y también para optimizar el uso de los servidores, frecuentemente los administradores de sistemas del Departamento bajan algunos de estos servidores (los sacan de línea) para hacerles actualizaciones o mantención al software; esto también se hace cuando se requiere levantar sistemas o productos de software particulares, a solicitud de distintos miembros de la unidad. En caso de que los sistemas estén abajo al momento de necesitarlos, el profesor del curso (o quien lo quiera usar) debe mandarle un mensaje al personal del área de sistemas del DCC, para que ellos lo vuelvan a levantar.

Respecto al uso de estos sistemas, el sistema de conformación de equipos se usa una única vez (al inicio del semestre) durante el curso CC5401. Esto se hace con el fin de realizar la conformación de los equipos de trabajo semestrales. Por su parte, el sistema de coevaluaciones se utiliza después de cada iteración de desarrollo del proyecto semestral, que en general son tres por semestre. Por último, el sistema MainReq se usa de manera recurrente durante todo el semestre, debido a que es la principal herramienta de trabajo de los alumnos durante el semestre, por lo que debe estar siempre disponible. Sólo uno de estos sistemas utiliza el sistema U-pasaporte para validar la identidad de los usuarios, el resto usa soluciones *ad hoc* para autenticar usuarios.

1.2. Objetivos de la memoria

El objetivo general de este trabajo de memoria es realizar una integración del flujo de datos que utilizan los distintos sistemas que se usan en el ramo CC5401: Ingeniería de Software II. Se busca esto, con el fin de simplificar el acceso a las herramientas y abrirlas a otros usuarios (profesores y auxiliares de otros cursos del DCC) que requieran utilizarlas. Para llevar a cabo la integración planteada, se propuso una nueva plataforma integradora de los sistemas.

Para alcanzar el objetivo general, se han definido los siguientes objetivos específicos.

1. Identificar las necesidades de datos de usuarios de cada uno de los sistemas, para llevar a cabo la integración entre ellas.
2. Crear endpoints en cada sistema para facilitar el intercambio de información entre ellos y la plataforma integradora.
3. Crear un servicio en la plataforma integradora, desde donde manejar la información suministrada desde y hacia los distintos sistemas.
4. Validar funcionalidad y utilidad de la plataforma integradora.

1.3. Estructura del documento

En el próximo capítulo se presenta un análisis de los sistemas legados, así como una descripción del *Data Lake* de Docencia del DCC. En el capítulo 3 se presenta la concepción de la solución que incluye la definición de los perfiles de usuario, los principales requisitos del sistema, su arquitectura y el modelo de datos del mismo. En el capítulo 4 se describe la implementación del sistema, incluyendo las principales interfaces del mismo, y su funcionamiento. En el capítulo 5 se explica la evaluación realizada al sistema desarrollado, y los resultados obtenidos. Por último, el capítulo 6 presenta las conclusiones y el trabajo a futuro.

Capítulo 2

Marco de Trabajo

A continuación se describen los tres sistemas que fueron integrados en el marco de esta memoria: *Conformación de Equipos, MainReq y Coevaluaciones*. También se explica brevemente el *Data Lake* desde donde se va a obtener la información de los estudiantes que participan en el curso, y los *end-points* asociados a dicho *Data Lake*.

2.1. Sistema de Conformación de Equipos

Como se mencionó anteriormente, el sistema corresponde a una aplicación Web[1] que apoya los procesos de diagnóstico de desarrolladores (en este caso, alumnos del curso CC5401), y el armado de equipos de trabajo cohesivos en base al diagnóstico anterior [4]. Para ello, dicha aplicación implementa un módulo de diagnóstico que permite, en base a una encuesta, identificar el perfil socio-técnico de los desarrolladores. Además, este sistema cuenta con una heurística que permite armar equipos de desarrollo de software en base al perfil socio-técnico de los potenciales miembros.

Dado que el sistema fue desarrollado en el año 2012, los usuarios de la plataforma se manejaban de forma aislada en el sistema. En el año 2017 el sistema realiza una actualización, la cual permite integrar el sistema con la plataforma U-Pasaporte para la validación de los usuarios y obtener de forma automática datos del usuario. Sin embargo, esa versión mantiene el manejo local de la información de cada uno de los usuarios de la plataforma.Este sistema está desarrollado en el *framework web* Django en la versión 1.11.29.

En la Figura 2.1 se observa el flujo de trabajo esperado para la creación y utilización de las funciones de la plataforma. La figura también nos muestra dos perfiles de usuario: el *administrador* y el *usuario normal*.

El *administrador* corresponde a un miembro del equipo de cátedra. Éste es el encargado de la creación de los usuarios utilizando el rut de ellos, y usa un pequeño *script* para cargar

[1] `https://equipos.dcc.uchile.cl/`

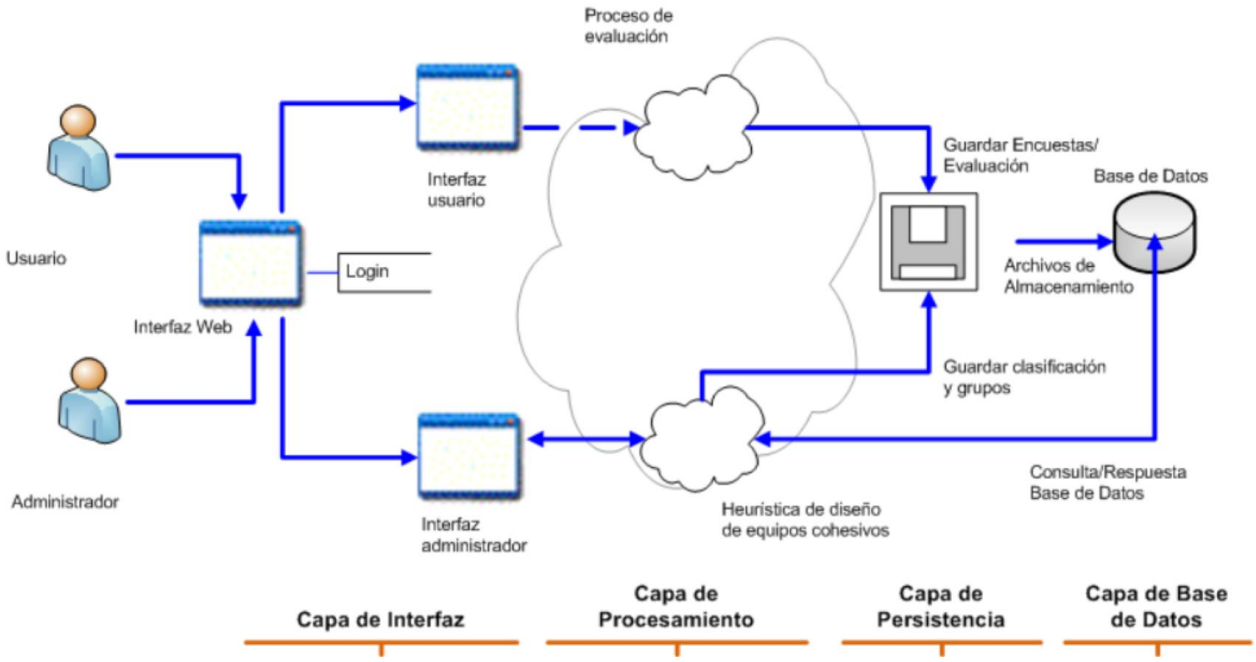

Figura 2.1: Diseño del sistema de conformación de equipos (obtenido de [4])

dichos usuarios directamente a la base de datos. El administrador es también el encargado de confirmar o reorganizar los equipos que la aplicación genera, y exportar los datos en una planilla de cálculo o equivalente.

El *usuario normal* corresponde a un estudiante del curso. Éste ingresa a la plataforma utilizando las credenciales de U-Pasaporte, y responde el cuestionario que esté disponible en la plataforma; es decir la encuesta socio-técnica antes mencionada.

Como se muestra en los casos de uso de la Figura 2.2, las respuestas de los usuarios es uno de los insumos que usa el sistema para la formación de los equipos. Este proceso se ve afectado también por la cantidad de alumnos y equipos que se requieran formar en el curso para el semestre en cuestión.

La salida que entrega el sistema toma en cuenta los puntos anteriores, y propone un ordenamiento en base a ellos. El administrador puede organizar de forma manual los equipos en caso de que lo estime conveniente, y luego exporta la información a una planilla Excel para ser distribuida a los estudiantes.

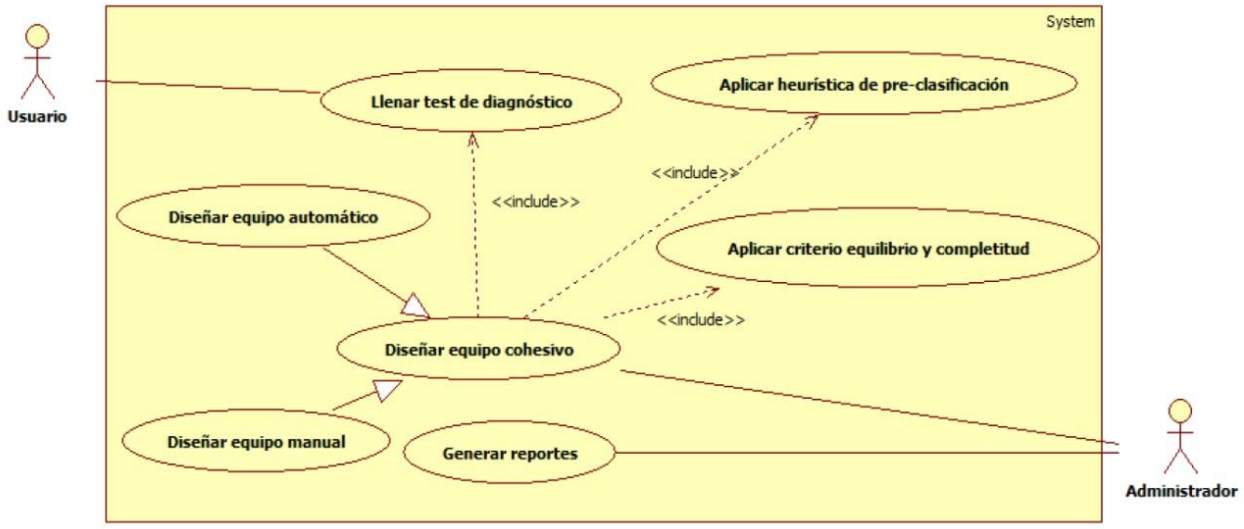

Figura 2.2: Modelo casos de uso del sistema de conformación de equipo (obtenido de [4])

2.2. Sistema Mainreq

MainReq[2] es una herramienta inicialmente concebida para la administración de requisitos. Ésta fue desarrollada específicamente para dar visibilidad y mantener la trazabilidad de los requisitos de usuario y de software en los proyectos ejecutados por los estudiantes del curso CC5401.El sistema está desarrollado en el *framework web* Django en la versión 1.5.2.

En la Figura 2.3 se indica el flujo de trabajo esperado para la formalización de los requisitos en un proyecto de software. La figura muestra tres tipos de usuarios: *administrador*, *miembro del equipo de desarrollo*, y el *jefe de proyecto*. El primero corresponde al equipo de cátedra, y tiene acceso a toda la funcionalidad del sistema, incluyendo crear y eliminar proyectos.

Los *miembros del equipo* de desarrollo pueden crear/modificar/eliminar/consultar requisitos, e información generada por el sistema a partir de estos; por ejemplo, las matrices de trazado. Estos usuarios corresponden a alumnos del curso CC5401, los cuales son agrupados en equipo y asignados a un proyecto.

El *jefe de proyecto* es un miembro del equipo de desarrollo, que tiene capacidades adicionales; por ejemplo, puede asignar tareas a los miembros del equipo y monitorear su cumplimiento. El flujo de trabajo que llevan a cabo estos usuarios y los servicios disponibles dentro de la plataforma, para la gestión de los requisitos, se muestra en la Figura 2.4.

Para ingresar a un nuevo usuario a la plataforma, se solicitan los correos de los miembros del curso, para luego enviarles un correo con el nombre de usuario y una contraseña por defecto. Luego, estos usuarios tienen la posibilidad de cambiar su contraseña, una vez que acceden a la plataforma MainReq. La información de los usuarios de MainReq se maneja de forma local (en la base de datos de la plataforma), y no es compartida con las otras

[2] `https://mainreq.dcc.uchile.cl/`

herramientas antes mencionadas.

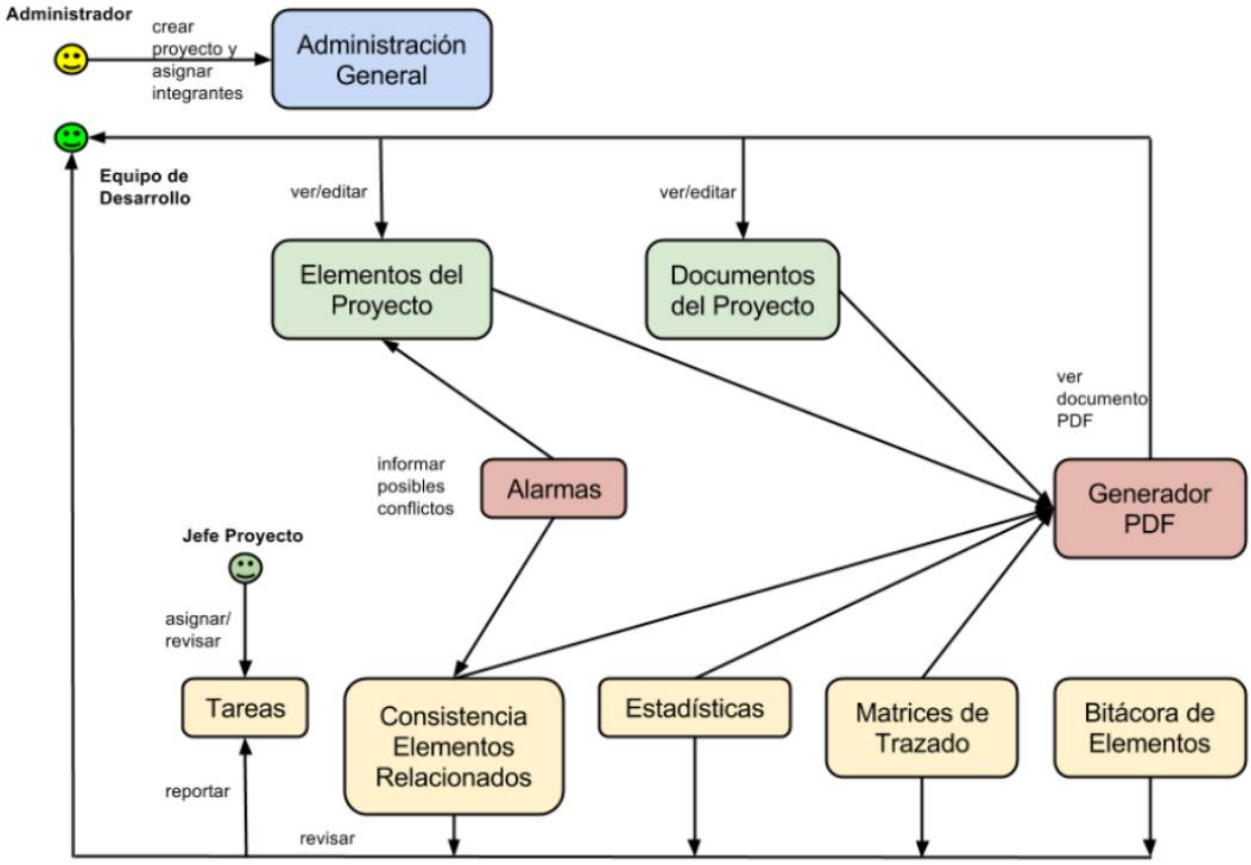

Figura 2.3: Componentes e interacción entre componentes en el Sistema MainReq (obtenido de [2])

2.3. Sistema Coevaluaciones

Como se mencionó anteriormente, el sistema de coevaluaciones[3] es una aplicación Web utilizada para realizar una evaluación entre pares. A través de una encuesta, cada miembro de un equipo de trabajo califica, de forma anónima, el desempeño de sus compañeros en términos de los siguientes ítems: compromiso con el proyecto, cumplimiento de tareas, iniciativa para lograr el éxito del proyecto, comunicación con el resto del equipo, coordinación entre sus tareas y las de sus pares, calidad del trabajo realizado, apoyo a sus compañeros, y capacidad de admitir equivocaciones y recibir críticas. Además, el sistema solicita indicar fortalezas, debilidades y sugerencias para cada uno de los pares. Esta información es agrupada, anonimizada y entregada de forma individual (como retroalimentación) a cada miembro del equipo.

En la Figura 2.5 se indica el flujo de trabajo esperado para el proceso de una coevaluación. La figura también muestra dos tipos de usuarios: el *profesor* y el *alumno*. El *profesor* corresponde a un miembro del equipo de cátedra y tiene acceso a todas las funcionalidades especificadas en la figura. Éstas se centran principalmente en la creación y configuración de los cuestionarios que los alumnos responderán en su coevaluación.

Por otra parte, el *alumno* corresponde a un estudiante del curso en donde se está utilizando la plataforma. Este tipo de usuario puede responder y revisar las coevaluaciones que ha hecho, y también los comentarios anonimizados que fueron dados por sus compañeros de equipo.

Para la administración de los usuarios, la plataforma maneja de forma local la información en una base de datos propia, sin relación alguna con los usuarios almacenados en UCampus o alguna otra plataforma. Todos los usuarios deben ser agregados de forma manual por parte de un miembro del cuerpo docente.

2.4. Data Lake de Docencia

El *Data Lake* de docencia[4] es una plataforma desarrollada como parte de la memoria de ingeniería civil en computación del Sr. Nicolás Caracci [1]. La Figura 2.6 ilustra a grandes rasgos la estructura del sistema.

Los módulos ETL (por sus siglas en inglés *Extract, Transform and Load*) son los encargados de obtener la data directamente desde la plataforma UCampus (u otras fuentes de información), realizar el pre-procesamiento y filtrado de la información antes de cargarla al *Data Lake*. Desde aquí se generan una serie de indicadores y métricas de interés que sirven de insumos para la gestión de la docencia del departamento, por ejemplo, la demanda de cada uno de los cursos impartidos en pregrado, y el departamento de origen de los alumnos, entre otros. Más detalles sobre estos indicadores está disponible en [5].

[3] `http://coevaluaciones.dcc.uchile.cl/`

[4] `https://w3.dcc.uchile.cl/warehouse/`

Es importante destacar que la información contenida en el *Data Lake* es un fiel reflejo de la información almacenada en la plataforma UCampus. El objetivo central de la plataforma es poder manejar de forma local esta información, permitiendo la obtención de métricas e indicadores que no están disponibles en la plataforma UCampus. Es por esto que la estructura interna del *Data Lake* es muy similar a la presente en UCampus. La Figura 2.7 muestra el modelo de dominio del *Data Lake*, en lo que respecta al ámbito de docencia.

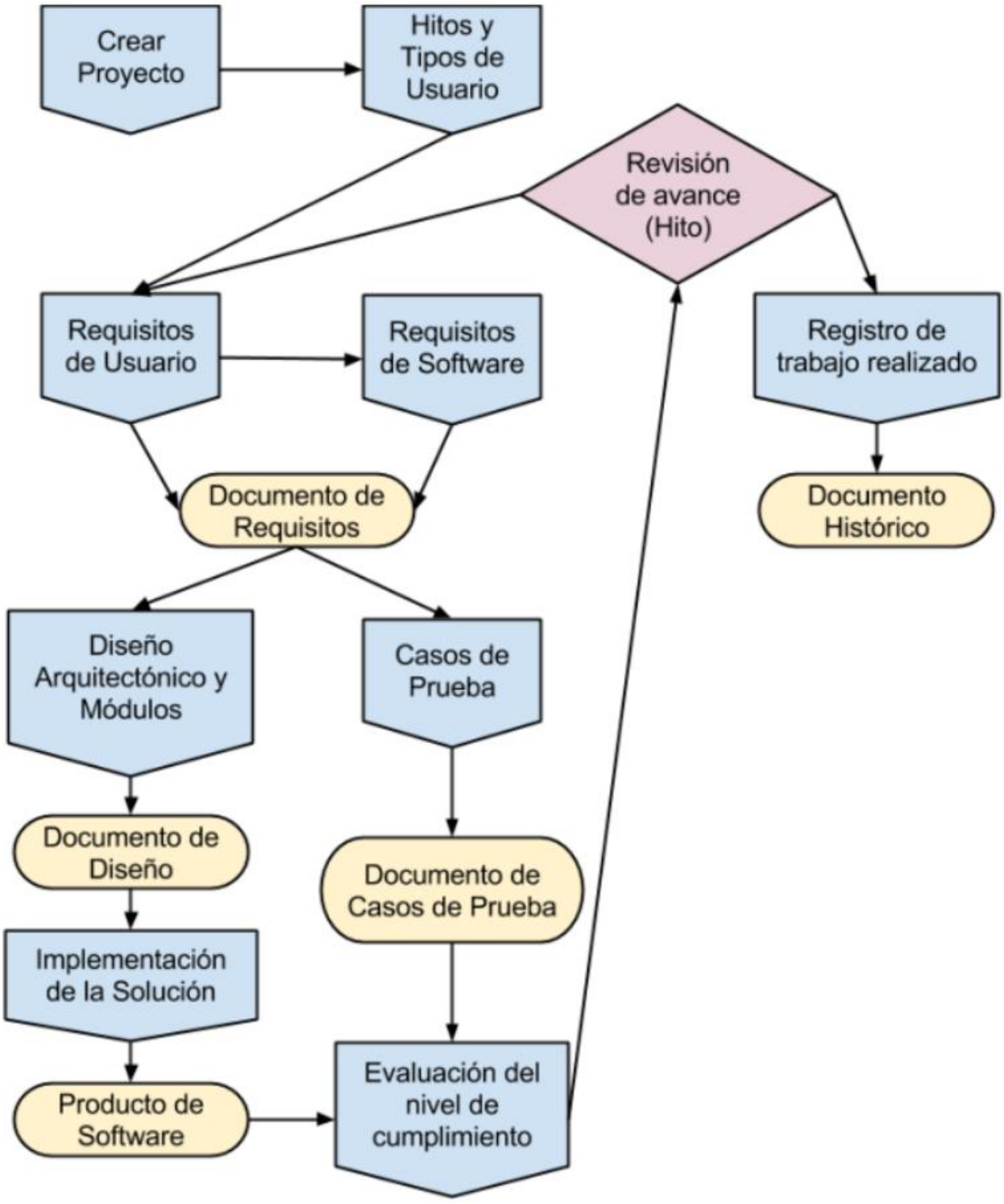

Figura 2.4: Proceso de gestión de requisitos (obtenido de [2])

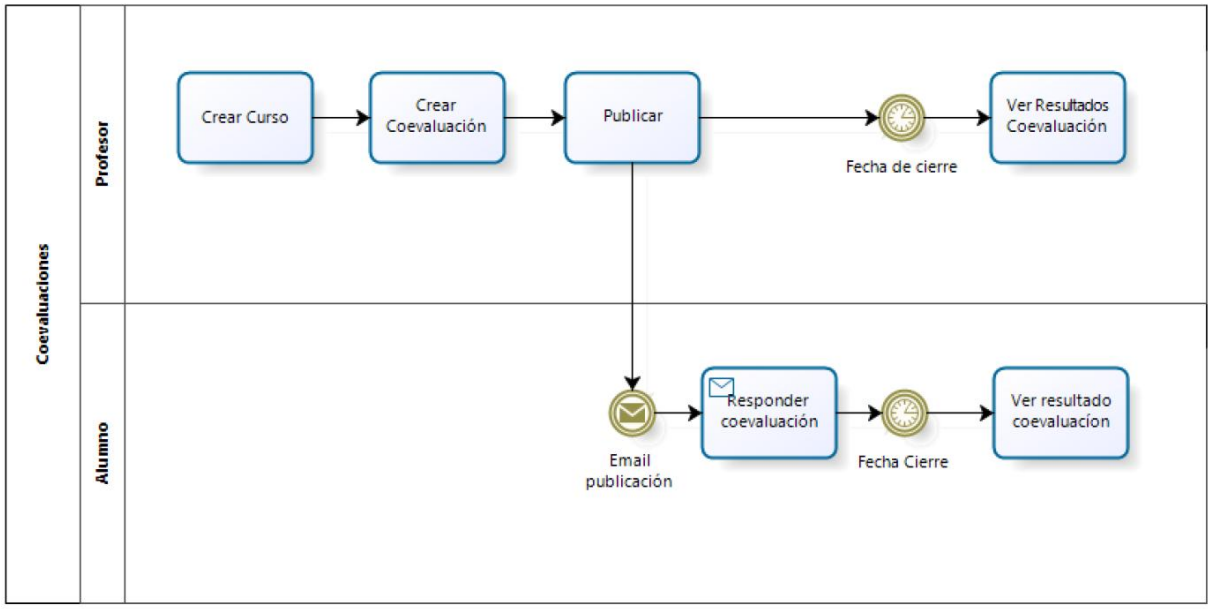

Figura 2.5: Proceso de una coevaluación (obtenido de [3])

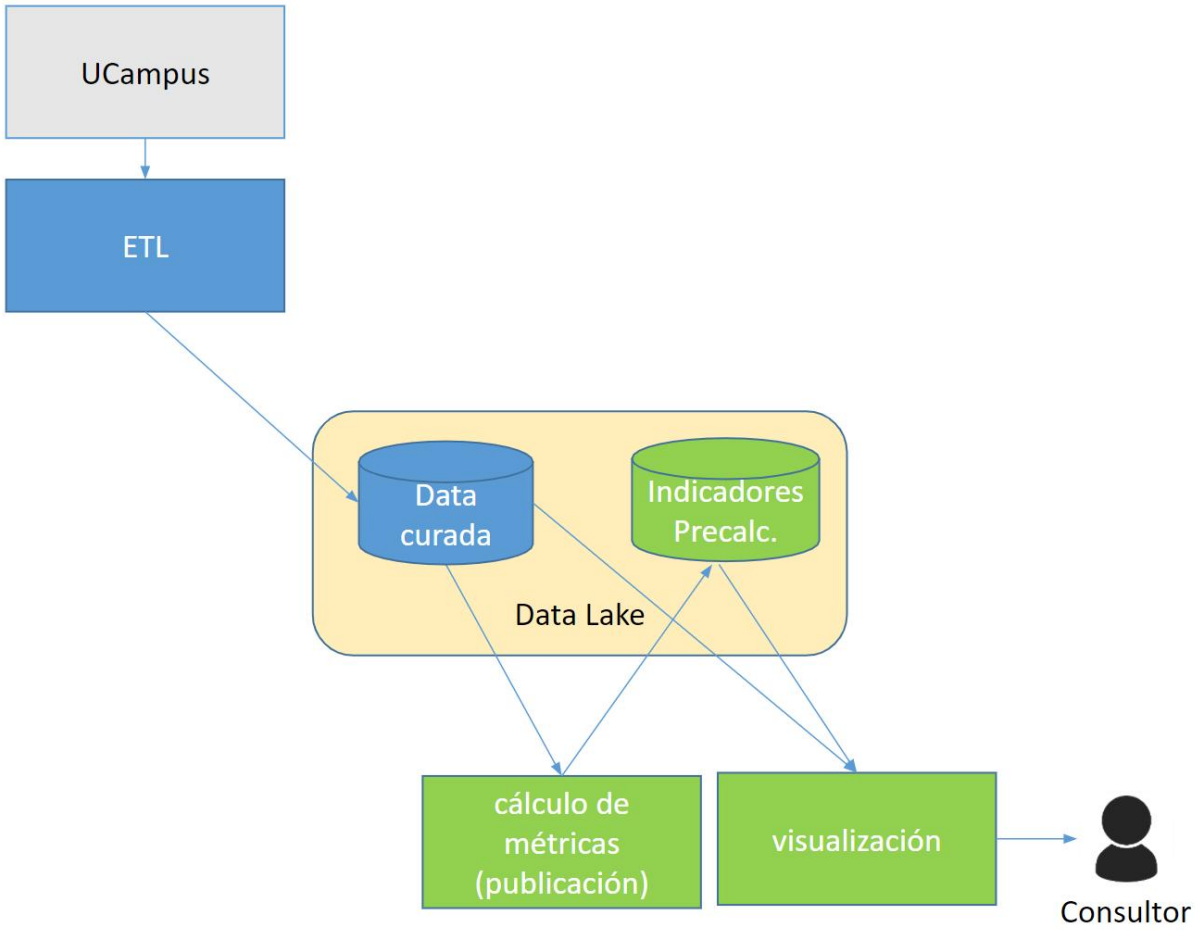

Figura 2.6: Estructura de alto nivel de la estructura del *Data Lake* de docencia

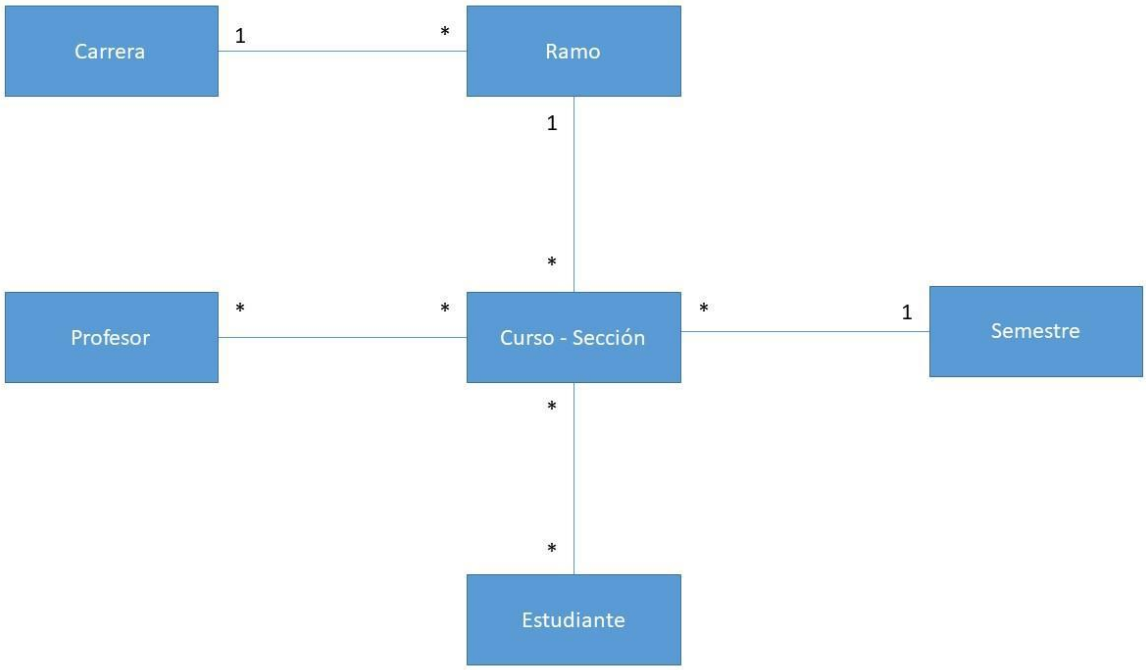

Figura 2.7: Modelo de dominio de docencia representado en el *Data Lake* del DCC (obtenida de [5]).

Capítulo 3

Concepción de la Solución

Dado que cada uno de los sistemas es una isla, la primera labor realizada consistió en una exploración exhaustiva de los mismos y de su documentación. Se detectaron varios cambios que se requerían realizar sobre la lógica desarrollada originalmente por sus creadores. El único sistema que se mantiene cercano a su concepción, es el sistema de gestión de requisitos (MainReq). Tanto el sistema de coevaluaciones, como el de conformación de equipos, han sufrido cambios desde su concepción, fundamentalmente para actualizar a las tecnologías utilizadas a versiones más recientes, pero manteniendo el *core* de cada uno.

Debido a que estos sistemas están en producción en el DCC, se mantuvieron reuniones con el encargado del área de desarrollo de sistemas del DCC, para que la integración de usuarios planteada en la memoria permitiera mantener a estos sistemas en producción. Esta persona solicitó mantener el código fuente actualizado, en un repositorio interno para los proyectos del departamento. Además, propuso la idea de actualizar en parte las tecnologías de los sistemas a intervenir, debido a que durante el proceso de exploración se descubrieron algunas librerías sin soporte, y con vulnerabilidades conocidas en dichos sistemas. Finalmente se descartó gran parte de esta actualización, debido a que el alcance del trabajo de título iba a crecer demasiado.

A continuación se presentan los perfiles de usuario soportados, los principales requisitos de la solución, y la estrategia de integración adoptada para hacerle frente a las necesidades antes descritas.

3.1. Perfiles de Usuarios Soportados

El sistema soporta 2 tipos de usuarios en estos momentos: un usuario *administrador* y un usuario *docente*. El usuario administrador tiene acceso total a las funcionalidades presentes en la plataforma mencionadas anteriormente, así como también agregar al sistemas nuevos ramos que deseen ser utilizados en la plataforma. Por otro lado, también es el encargado de la creación de tokens para poder acceder a la información disponible a través de la API de la plataforma.

El usuario docente representa a un profesor de cátedra o profesor auxiliar de un curso que esté en la plataforma. Estos pueden revisar el listado de alumnos de los cursos disponibles en la plataforma, así como también los equipos en caso que estén formados. Finalmente, puede iniciar el proceso de formación de equipos para un curso determinado que no tenga un proceso ya iniciado en el sistema de conformación de equipos directamente desde la plataforma integradora.

3.2. Principales Requisitos de la Solución

A continuación se indican los requisitos generales para la plataforma integradora, y también aquellos para los sistemas legados que están involucrados en este trabajo:

- *Automatizar el flujo de los 3 sistemas*: Como se menciona en el objetivo general de la memoria, el requisito principal es automatizar la entrada y salida de la información de usuarios de cada uno de los sistemas legado, utilizando como principal fuente de datos el *data lake* de docencia del DCC.
- *Utilizar UCampus para autentificación*: Dado que la plataforma UCampus es común para todos los miembros del departamento, se utilizará dicha plataforma para realizar la autenticación de usuarios.
- *Facilitar el uso y acceso para usuarios no técnicos*: Los sistemas actualmente requieren conocimiento técnico para poder configurarlos, e incluso acceso directo a las máquinas donde estos están corriendo. Con la integración de usuarios aquí planteada se busca evitar esto, y simplificar la utilización de dichos sistemas.
- *Mantener las funcionalidades originales de los sistemas*: Dado que estos son sistemas legados, la intervención de ellos es poco conveniente y se intentará evitar la modificación de la funcionalidad principal de los mismos, tanto como se pueda. Esto significa que solo se modificarán los sistemas para poder agregarles la carga automática de los datos desde la plataforma a desarrollar, más no se quitará ninguna de las funcionalidades actuales disponibles en dichos sistemas.

3.3. Estrategia de Integración de los Sistemas Legados

La solución implementada involucra una adecuación de los sistemas actuales, para que estos puedan acoplarse al flujo de trabajo que se propone implementar. Este flujo es manejado por un nuevo sistema de software (integrador), que coordina la interacción entre las tres plataformas, y el flujo de información entre ellas (Figura 3.1).

El nuevo sistema usa el *data lake* de docencia del DCC como fuente de datos primaria, simplificando la interacción de los usuarios con cada uno de los sistemas legados. De esa manera, estos últimos pueden ser utilizados por usuarios no expertos, y así se abre la posibilidad de usar estos sistemas en otros cursos.

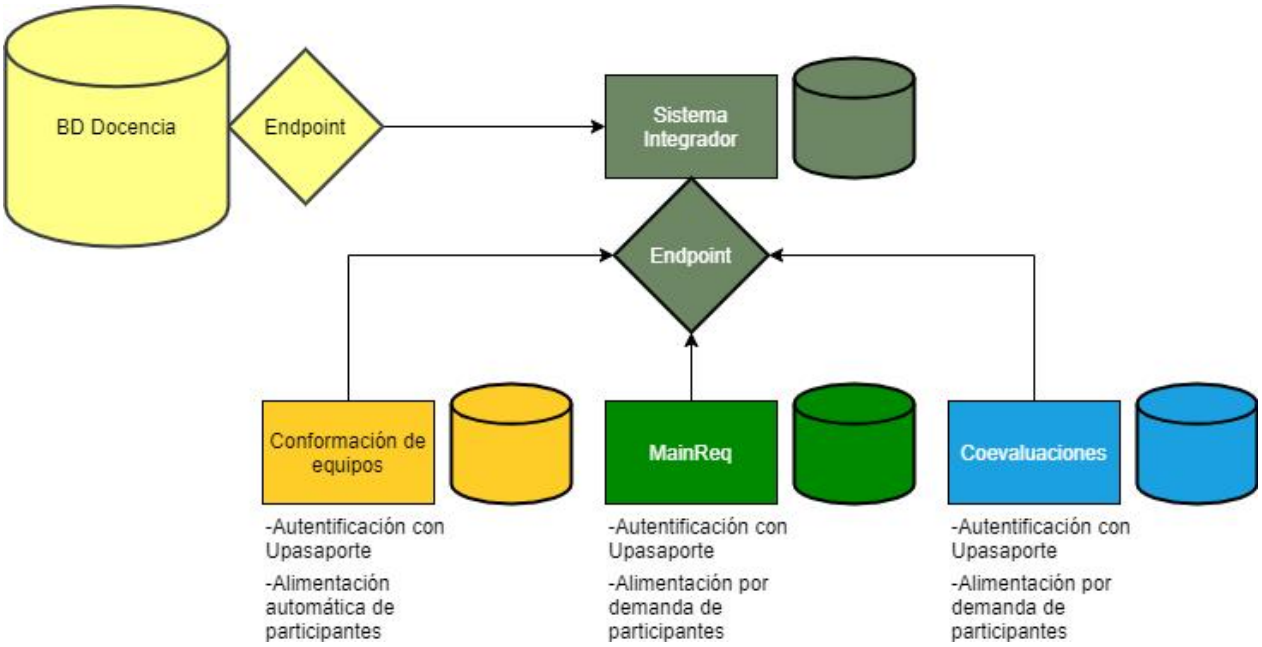

Figura 3.1: Arquitectura de sistema integrador propuesto

Para llevar a cabo la integración planteada, se analizaron los sistemas legado y se identificaron los puntos de alimentación de datos para cada uno de ellos. En el caso del sistema de conformación de equipos, además se requirió identificar la salida que debe ser entregada para que los otros sistemas la consuman. Además de lo antes mencionado, se determinó la posición de estos sistemas en el flujo de datos, y la información que se necesita en cada sistema para realizar la integración de usuarios.

Para realizar la interacción en sí, se consideró inicialmente crear *endpoints* en cada una de las plataformas. Estos *endpoints* son capaces de recibir y disponibilizar la información presente en cada uno de sus sistemas, y son para mantener sincronizados los cambios que pudieran ocurrir en cada una de las plataformas. Esta idea finalmente fue desechada para los sistemas Mainreq y Coevaluaciones, debido a que la herramienta con la que se iba a desarrollar estos *endpoints* no posee soporte para versiones tan antiguas de Django. Además, se requería, la creación a mano de dichos *endpoints*, debido a la falta de *expertise* del memorista y a la escasa documentación existente para las versiones de cada uno de los sistemas. Esto generaría la posibilidad de vulnerabilidades innecesarias en sistemas que, de por sí ya son vulnerables, optando entonces por una aproximación más pasiva, que recupere los datos desde el sistema integrador.

Dicha aproximación se realizó de la siguiente manera para ambos sistemas. Entre todas las vistas que tiene la plataforma, se identificó el mejor lugar para agregar un botón para una nueva vista desde donde manejar el proceso de integración. Estas vistas se modelaron siguiendo el patrón diseño de interfaces del sistema, para que los nuevos componentes no se vieran fuera de lugar con respecto al resto.

El sistema de conformación de equipos entonces fue la excepción, ya que entre las dependencias que utiliza la plataforma se encontraba dicha herramienta, y por lo tanto, no

fue necesario intervenir las interfaces. A continuación se explica brevemente cada uno de los macro-componentes de la solución.

3.4. Arquitectura del Sistema Integrador

La Figura 3.2 muestra la arquitectura de la aplicación, la cual es bastante simple. El sistema integrador obtiene de forma automática, o a petición de un usuario, información del *Data Lake* de docencia. Esto se hace por medio de la conexión del sistema a la API dispuesta por el *Data Lake*. Luego de procesar la información y guardarla en la base de datos, ésta se disponibiliza a través de una API hecha en Django Rest Framework. La información se entrega en formato JSON, listo para ser utilizado. El sistema también envía esta información de vuelta al *Frontend* para mostrarla a los usuarios.

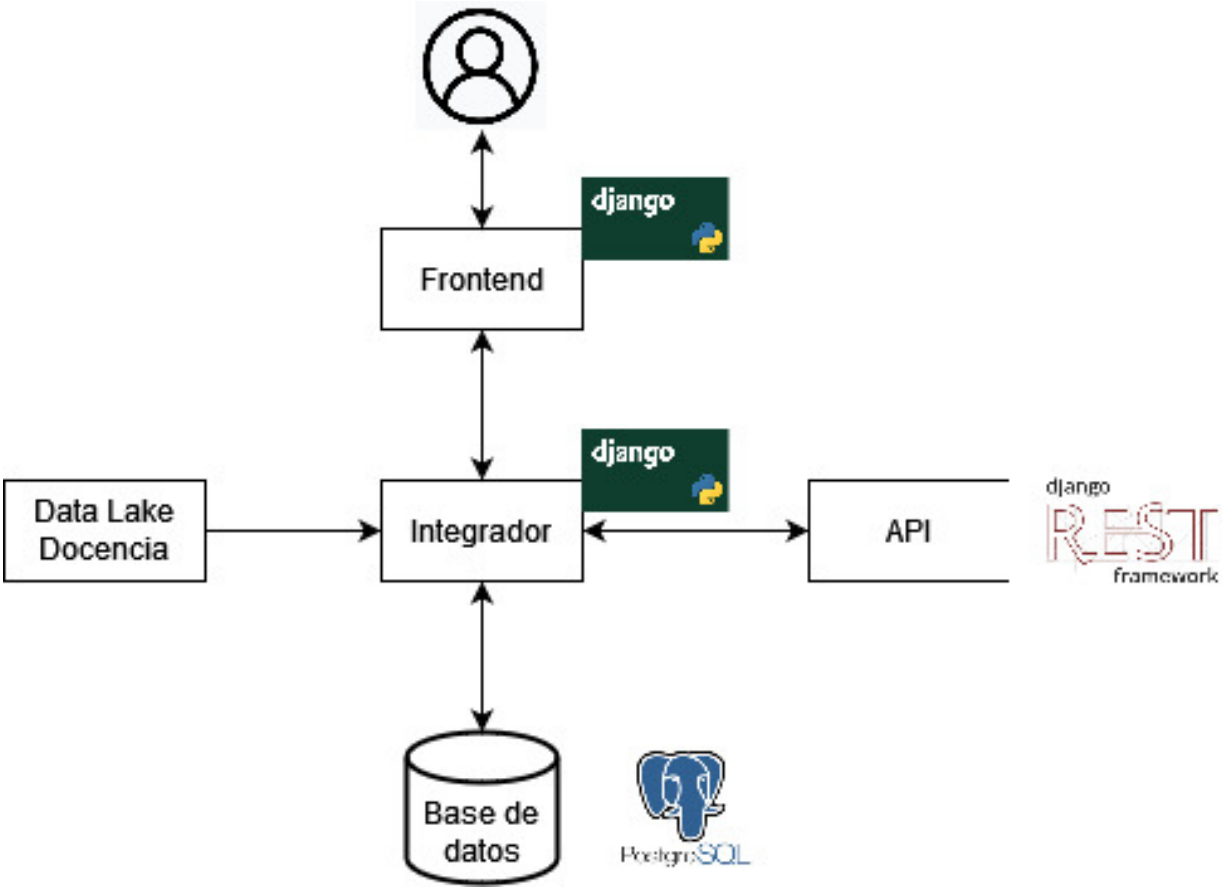

Figura 3.2: Arquitectura del escenario de trabajo del sistema integrador

En este desarrollo se utilizó Docker[1] como contenedor base para la plataforma. La aplicación web fue realizada usando el *framework* web Django, con Postgresql como base de datos. Para la creación y el manejo de *endpoints* se uso el *framework* Django-Rest[2] . Para el *frontend*

[1] `www.docker.com`
[2] `www.django-rest-framework.org`

se usó Bootstrap 4[3] y los templates de Django.

Los sistemas legados usan el *framework* web Django, cada uno está en una versión diferente 1.x.x de dicho *framework* así como Python 2.7.

En el caso del sistema de Conformación de Equipos, se utilizó Django Rest Framework en la versión que ya estaba instalada. Este *framework* se usó para la creación y manejo de la API, que es la que permite disponibilizar la información de la base de datos.

Los sistemas Mainreq y Coevaluaciones utilizaron la librería Requests[4] , fundamentalmente para implementar la comunicación con el sistema integrador. También se usó esta librería para realizar una adaptación de las funciones existentes, las cuales manejaban la carga manual de los datos en las respectivas plataformas; ahora dicha funcionalidad corresponde a carga automática de la información.

[3] `https://getbootstrap.com/`

[4] `https://docs.python-requests.org/en/latest/`

Capítulo 4

Implementación del Sistema Integrador

En este capítulo se presentan las capas que componen el sistema integrador, así como las modificaciones que se le hicieron a los otros sistemas intervenidos.

4.1. Capa de Datos

La Figura 4.1 corresponde al modelo de datos del sistema integrador. Dada la decisión de intervenir lo menos posible los otros sistemas, no se realizó ningún cambio en los modelos de datos de los mismos. Las tablas Ramo, Curso y Periodo corresponden a copias locales de información disponible en el *Data Lake* de docencia. A continuación una descripción de cada una de las tablas presentes en la figura.

Las tablas *Django User Model* y *Django Group Model* corresponden a las tablas que mantiene Django (por defecto), para el manejo de usuarios y equipos dentro de su plataforma. Se prefirió utilizar esta opción con la intención de simplificar el manejo de permisos a los distintos usuarios de la plataforma. Otra razón para usarla, es debido a la integración que tiene el sistema con el Portal DCC, el cual crea un usuario de Django por cada usuario que ingresa a la plataforma.

La tabla *Test* corresponde a un respaldo en el sistema de los tipos de cuestionarios existentes en el sistema de conformación de equipos. Estos se usan en el proceso de creación de un nuevo proceso de conformación de equipos desde la plataforma integradora.

La tabla *Confequi* contiene la información correspondiente a un proceso de conformación de equipos, empezado desde el sistema integrador. Ésta guarda además información para poder identificar y extraer los equipos formados, luego de terminado el proceso de conformación de equipos.

La tabla *GrupoPersistente* guarda una referencia al nombre del equipo original por defecto, proveniente desde el sistema de conformación de equipos, y el nombre actual del equipo. El nombre actual puede ser modificado dentro del sistema integrador.

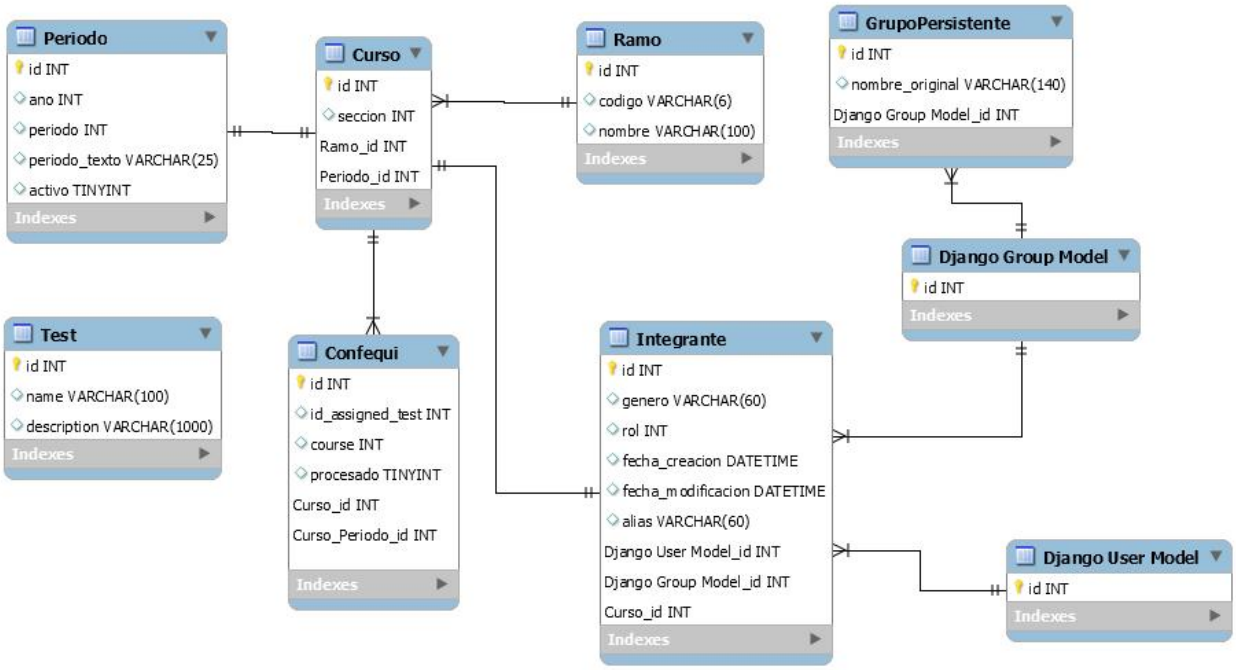

Figura 4.1: Modelo de datos del sistema integrador.

La tabla *Ramo* guarda la información de todos los ramos que hayan sido agregados al sistema. La tabla *Curso* corresponde a la instancia semestral de un ramo.

La tabla *Integrante* es la representación particular de un usuario de Django de la plataforma asociado a un curso en particular. Éste puede ser miembro del cuerpo docente de dicho curso o un estudiante.

Finalmente, la tabla *Período* almacena información sobre un semestre e identifica si es el semestre actual o no.

4.2. Capa de Lógica de Negocio

La Figura 4.2 muestra, a grandes rasgos, la interacción que hay entre los sistemas. El primer paso consiste en hacer Login a través del Portal DCC, para así permitirle a los usuarios autorizados acceder al sistema. Si el usuario es autorizado para acceder, entonces se le muestra la vista *Home* del sistema integrador.

Al agregar un nuevo ramo al sistema integrador, éste generará una solicitud para traer todos los cursos (con sus respectivos integrantes) según el periodo (semestre) activo en la plataforma. Luego esta información se procesa en el sistema integrador, y se deja disponible para utilizarse por alguno de los sistemas legados. Este proceso se repetirá automáticamente todos los semestres, por cada ramo existente en la plataforma.

En caso de querer iniciar un nuevo proceso de conformación de equipos, se debe seleccionar

un curso de los disponibles en el sistema, y en ese caso, se envía la información necesaria para comenzar el proceso en el sistema correspondiente. Luego se guarda, en el integrador, un identificador generado en la plataforma de conformación de equipos.

Una vez completado el proceso de conformación de equipos, se genera una solicitud desde el integrador para recuperar los equipos formados para los cursos presentes en la plataforma. Una vez recuperada la información de los equipos, se actualiza dicha información en la plataforma y se deja disponible a través de la API, para que ésta pueda ser consumida por los sistemas Mainreq y Coevaluaciones.

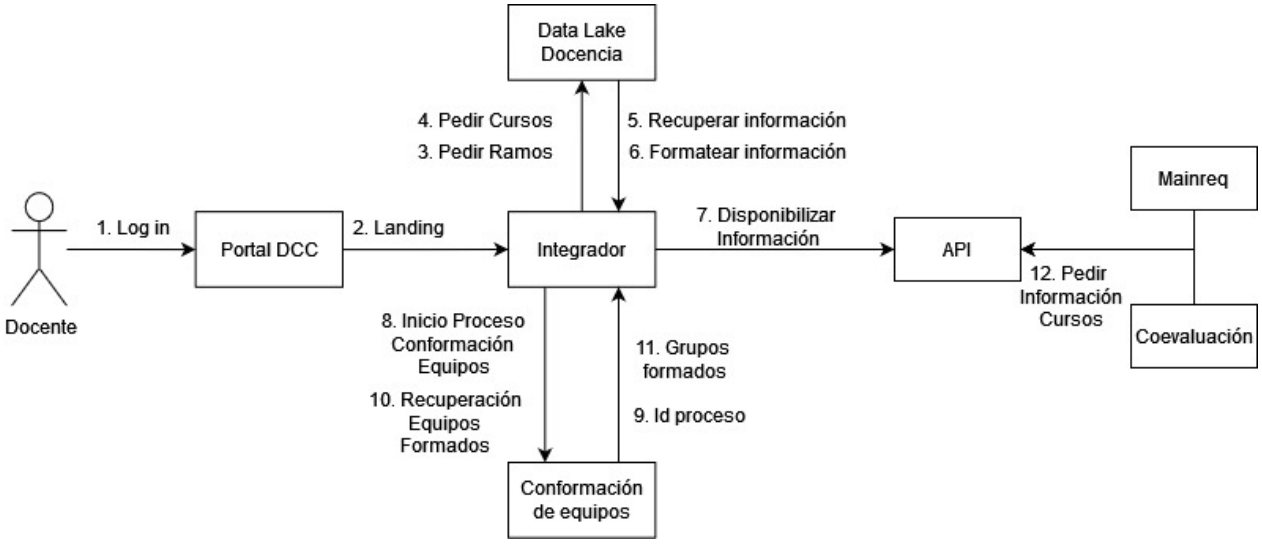

Figura 4.2: Dinámica de la interacción entre los sistemas legados y el integrador.

4.3. API del Sistema

Para la comunicación entre los sistemas, se crearon dos APIs, una para el sistema integrador y la otra para el sistema de conformación de equipos. Ambas APIs operan bajo un esquema similar al presentado en la Figura 4.3. A continuación se describen los *endpoints* de cada una de estas APIs.

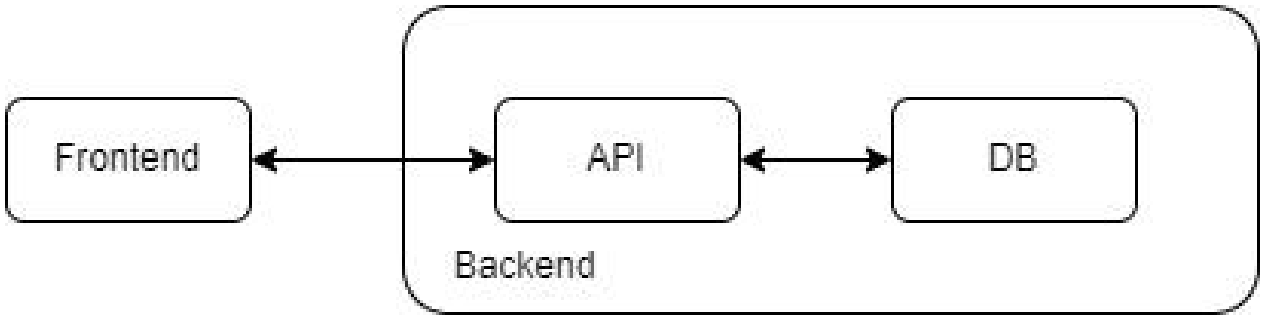

Figura 4.3: Interacción entre la API y los otros macrocomponentes del sistema (obtenida de [5]).

4.3.1. API del Sistema Integrador

A continuación se describen brevemente los *endpoint*s de la API implementada como parte del sistema integrador.

Lista de Ramos (método GET)

Este *endpoint* sirve para obtener todos los ramos existentes dentro del sistema. La URL correspondiente a este *endpoint* es *"/api/ramo"*, y permite filtrar la información basado en el siguiente atributo:

- *codigo_ramo*: código del ramo que se está buscando (por ejemplo, CC5401).

El *endpoint* retorna una lista con los ramos siguiendo la siguiente estructura:

- *Id*: El id del ramo en el *Data Lake* de docencia.
- *Codigo*: El código del ramo.
- *Nombre*: El nombre del ramo.

Ejemplo:

```
{
      "id": 5174,
      "codigo": "CC5401",
      "nombre": "Ingenier a de Software II"
 }
```

Lista de integrantes (método GET)

Este *endpoint* sirve para obtener todos los integrantes del curso, correspondiente al período actual. La URL correspondiente a este *endpoint* es la siguiente: *"/api/integrante/"*; ésta permite filtrar la información basado en el siguiente atributo:

- *curso_id*: Id del curso cuya información de los integrantes se desea recuperar.

El *endpoint* retorna una lista con los integrantes, siguiendo la siguiente estructura:

- *Rut*: Rut del integrante.
- *Nombres*: Primer y segundo nombre del integrante.
- *Apellidos*: Primer y segundo apellido del integrante.
- *Email*: Email de la persona.
- *Rol*: Rol desempeñado por la persona en el curso (profesor, auxiliar, ayudante, estudiante).
- *Grupo*: Equipo al que pertenece la persona.

- *Curso*: Id del curso al que pertenece.
- *Alias*: Primer nombre y primer apellido de la persona.

Ejemplo:


```
{
        "rut": 700001588,
        "nombres": "Alejandra Josefina",
        "apellidos": "Antilef Buschmann",
        "email": "aantbus@fake.dcc.uchile.cl",
        "rol": 0,
        "grupo": "Sin Equipo",
        "curso": 1001079,
        "alias": "Alejandra Antilef"
}
```


Lista de Cursos (método GET)

Este *endpoint* sirve para obtener todos los ramos existentes dentro del sistema, que correspondan al período actual. La URL correspondiente a este *endpoint* es *"/api/curso/"* y permite filtrar la información basado en el siguiente atributo:

- *codigo_ramo*: Este es el código del ramo que se está buscando (por ejemplo, CC5401).

El *endpoint* retorna una lista con los cursos siguiendo, la siguiente estructura:

- *Id*: El id del curso en el *Data Lake* de docencia.
- *Ano*: El año en que fue dictado el curso.
- *Semestre*: El semestre en que fue dictado el curso. Éste puede ser otoño, primavera o verano.
- *Seccion*: La sección del curso de la cual se quiere recuperar la información.
- *Codigo_ramo*: El código del ramo.

Ejemplo:


```
 {
        "id": 1001762,
        "ano": "2016",
        "semestre": 2,
        "seccion": 1,
        "codigo_ramo": "CC5402"
  }
```

4.3.2. API de Conformación de Equipos

A continuación se describen brevemente los *endpoint*s de la API implementada como parte del sistema de Conformación de Equipos.

Lista de Tests (método GET)

Este *endpoint* sirve para obtener todos los tipos de cuestionarios (tests) que existen actualmente en el sistema. La URL correspondiente a este *endpoint* es *"/api/test/"*. El *endpoint* retorna una lista con los cuestionarios con la siguiente estructura:

- *Id*: Id del tipo de cuestionario.
- *Name*: Nombre del cuestionario.
- *Description*: Descripción del cuestionario.

Ejemplo:


```
{
        "id": 1,
        "name": "Cuestionario basado en Habilidades T cnicas y Liderazgo",
        "description": "Balancea los conocimientos de los alumnos en distin
}
```


Lista de Equipos (método GET)

Este *endpoint* sirve para obtener todos los equipos existentes dentro del sistema. La URL correspondiente a este *endpoint* es *"/api/team/"* y permite filtrar la información basado en el siguiente atributo:

- *assigned_test*: Id del cuestionario en donde están almacenadas las respuestas de los integrantes del curso.

El *endpoint* retorna una lista con los equipos con la siguiente estructura:

- *Id*: id del equipos formado.
- *Members*: diccionario con el nombre, rut y rol de los miembros del equipo.
- *Data*: Corresponde al nombre del equipo asignado por el sistema.
- *Assigned_test*: Id del cuestionario en donde están almacenadas las respuestas de los integrantes del curso.
- *User*: id del usuario que armó el equipo en el sistema.

Ejemplo:


```
{
        "id": 75
```

```
        "members": [
            "700001724, Valentina Jabre  : Alumno",
            "700001723, Alexis Contreras  : Alumno",
            "700001201, Barbara Salazar  : Alumno",
            "700001745, Cristi n Romero  : Alumno",
            "700001682, Christian Rojas  : Alumno",
            "700001554, Catalina Tapia  : Alumno",
            "700001578, Angel Aburto  : Alumno"
        ],
        "data": "Equipo 1",
        "assigned_test": 51,
        "user": 328
}
```


Lista de Cuestionarios Asignados (método GET)

Este *endpoint* sirve para obtener todos los procesos de conformación de equipos que han sido iniciados en el sistema, y el estado en que se encuentra el proceso. La URL correspondiente a este *endpoint* es *"/api/assigned_test"*.

El *endpoint* devuelve una lista con todos los cuestionarios en el sistema con la siguiente estructura:

- *Id*: Id del cuestionario.
- *Start Dat*e: Fecha de inicio del cuestionario.
- *End Date*: Fecha de cierre del cuestionario.
- *Active*: Muestra si el cuestionario está activo.
- *Processing*: Muestra si el cuestionario se está procesando.
- *Published*: Muestra si el cuestionario fue publicado.
- *Test*: Id del cuestionario elegido.
- *Coure*: Id de curso asociado al cuestionario.

Ejemplo:


```
 {
        "id": 54,
        "start_date": "2022-02-07T00:00:00-03:00",
        "end_date": "2022-02-08T23:59:59-03:00",
        "active": true,
        "processing": false,
        "published": false,
        "test": 1,
        "course": 50
  }
```

Configuración de equipos (método POST)

Este *endpoint* sirve para recibir la información proveniente del sistema integrador; particularmente, a través de éste se reciben los datos para iniciar un nuevo proceso de conformación de equipos. La URL correspondiente a este *endpoint* es */api/configurar_confequi*. Se deben entregar los siguientes campos como parámetros:

- *Fecha_inicio*: Fecha de inicio del proceso de conformación de equipo.
- *Fecha_termino*: Fecha de término del proceso de conformación de equipo.
- *Cuestionario*: Id del cuestionario elegido para ser utilizado.
- *Codigo_curso*: Id del curso en el sistema integrador que está iniciando el proceso.
- *Ramo*: Diccionario con información detallada del curso; se requiere indicar el semestre, año, sección, nombre y código del ramo.
- *Integrantes*: Lista con la información de los integrantes del curso que iniciará el proceso. Cada integrante debe venir con el nombre completo, rut, alias (primer nombre, primer apellido) y el rol que desempeña en el curso.

Ejemplo:


```
{
    "fecha_inicio": "2021-10-13",
    "fecha_termino": "2021-10-14",
    "cuestionario": "3",
    "codigo_curso": "1002755",
    "ramo": {
        "semestre": 2,
        "ano": "2016",
        "codigo_ramo": "CC5614",
        "nombre_ramo": "Herramientas para el Trabajo en Equipo",
        "seccion": 2
    },
    "integrantes": [
        {
            "rut": 700001375,
            "rol": 0,
            "nombre": "Palma Chia Carolina Fadia",
            "alias": "Carolina Palma"
        },
        {
            "rut": 700001201,
            "rol": 0,
            "nombre": "Salazar Fuentealba Barbara Giovanna",
            "alias": "Barbara Salazar"
        }
    ]
}
```

Configuración de equipos (método PUT)

Este *endpoint* sirve para recibir la información proveniente del sistema integrador; particularmente recibe los datos para actualizar los integrantes de un curso, y usualmente se utiliza al iniciar o durante el proceso de conformación de equipos. La URL correspondiente a este *endpoint* es */api/configurar_confequi"*, y se le deben entregar los siguientes campos como parámetros:

- *Course*: Id del curso que se va a actualizar.
- *Integrantes*: Lista con los datos de los integrantes a agregar. Ésta contiene los datos descritos
- anteriormente.

El *endpoint* retorna un código 200 en caso de procesar sin problemas la actualización de los integrantes.

Ejemplo:

```
{
    "course": 46,
    "integrantes": [
        {
            "rut": 700001375,
            "rol": 0,
            "nombre": "Palma Chia Carolina Fadia",
            "alias": "Carolina Palma"
        },
        {
            "rut": 700001201,
            "rol": 0,
            "nombre": "Salazar Fuentealba Barbara Giovanna",
            "alias": "Barbara Salazar"
        }
    ]
}
```

4.4. Capa de Presentación

El diagrama presentado en la Figura 4.4 muestra el esquema de navegación de la plataforma integradora.

En azul están las vistas que sólo pueden ser accedidas por el usuario administrador. El resto de las vistas pueden ser accedidas, tanto por el usuario administrador, como por el usuario docente. A continuación se presenta una descripción más detallada de las vistas más importantes en la aplicación.

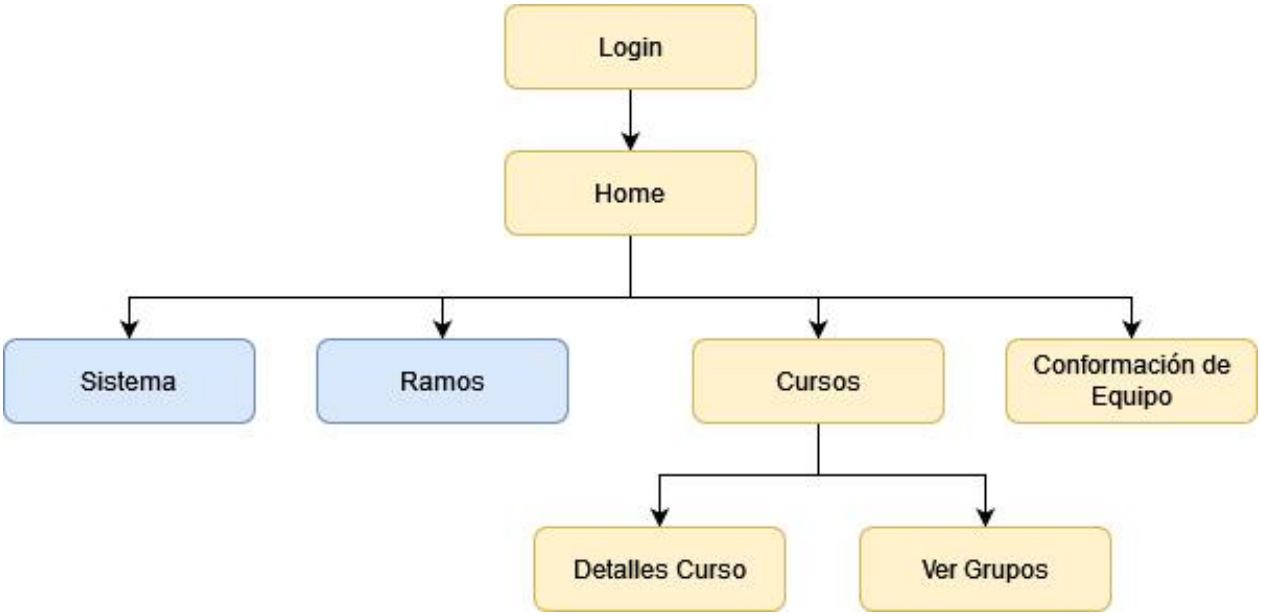

Figura 4.4: Diagrama de navegación del sistema integrador.

4.4.1. Vista: Home

La Figura 4.5 muestra la landing page a la que el usuario llega al momento de ingresar a la plataforma, desde la vista de un administrador. En la parte superior a la izquierda tenemos una barra de navegación, la cual cuenta con botones para desplazarnos entre las vistas disponibles. Arriba a la derecha se encuentra el nombre del usuario, junto con los botones para cerrar sesión; esto permite acceder al administrador de Django, y una vista para ver configuraciones del sistema.

Figura 4.5: Vista Home (Administrador)

En la Figura 4.6 tenemos la vista Home desde la vista de un Docente. Los cambios más notables en comparación a la vista del administrador es la falta de los botones de Admin y Sistema arriba a la derecha, y el botón de Ramos en la barra de navegación a la izquierda.

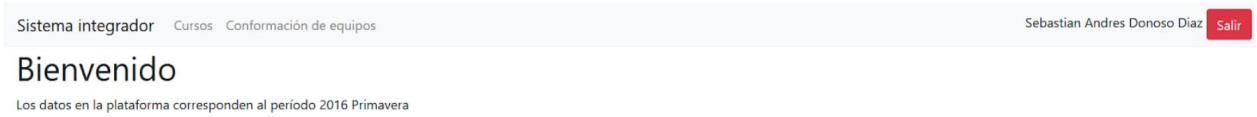

Figura 4.6: Vista home (Docente)

4.4.2. Vista: Ramos

En la Figura 4.7 observamos la lista de ramos ingresados en la plataforma, los cuales están identificados en la tabla central con el código y el nombre. Arriba de la tabla se encuentra un botón de *Agregar Ramo*, que despliega un modal para agregar un nuevo ramo al sistema, tal como se ejemplifica en la Figura 4.8.

Figura 4.7: Vista de Ramos.

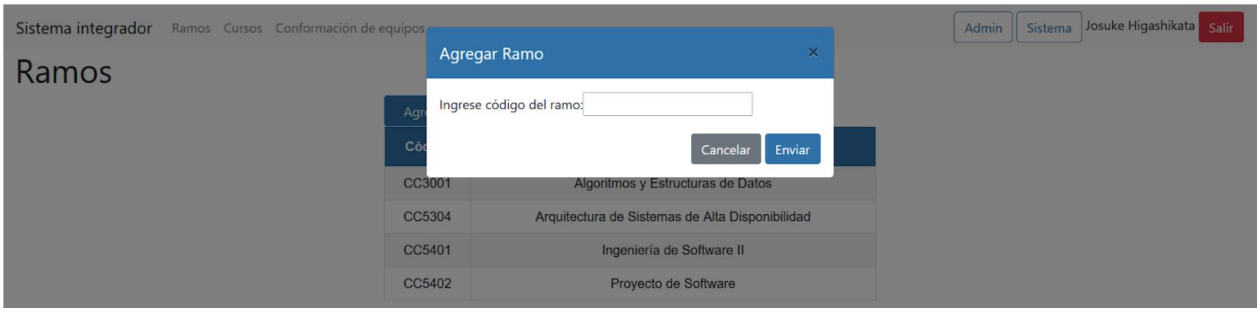

Figura 4.8: Modal para agregar un ramo.

4.4.3. Vista: Cursos

En la Figura 4.9 vemos una tabla con todos los cursos del semestre actual, asociados a algún ramo dentro de la plataforma. Estos se identifican por su nombre, el código del ramo asociado y la sección respectiva. La vista también muestra si el curso ya tiene equipos formados, y da la opción de ver una lista con los integrantes, tanto del cuerpo docente como de los estudiantes.

Figura 4.9: Vista Cursos.

4.4.4. Conformación de Equipos

El proceso de conformación de equipos utilizando esta vista involucra dos pasos. El primero implica elegir el curso, e iniciar el proceso de conformación de equipos, tal como se muestra en la Figura 4.10. En el segundo paso se completa la sección del curso que queremos utilizar, el cuestionario y las fechas de inicio y término del periodo en que queremos tener abierto el cuestionario que responderán los alumnos, tal como se muestra en la Figura 4.11.

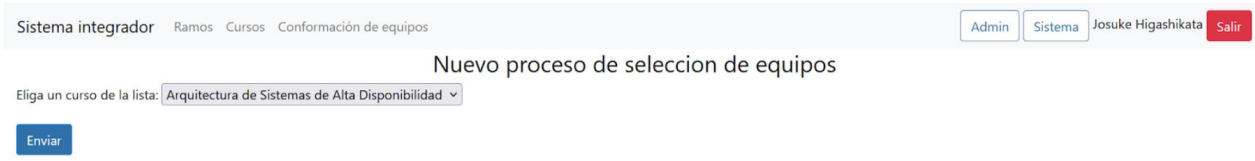

Figura 4.10: Vista conformación de equipos.

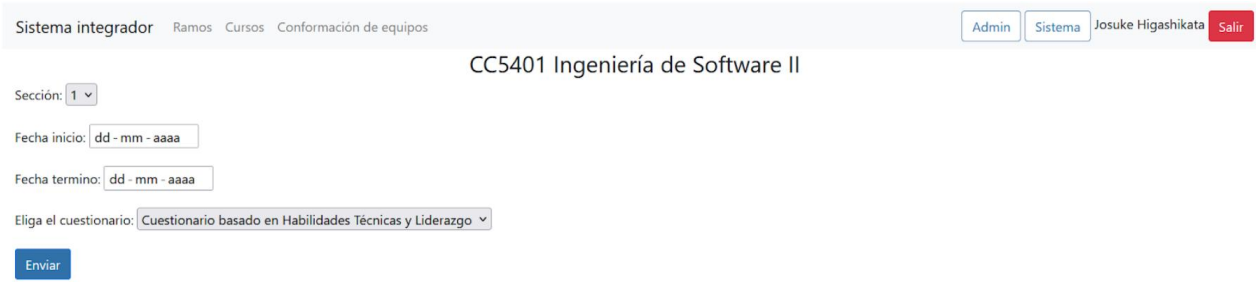

Figura 4.11: Vista de formulario para crear un nuevo proceso de selección de equipos.

4.4.5. Vista: Integrantes Curso

En la Figura 4.12, la vista es una continuación de la vista de curso, en donde se muestran (en una tabla) los alumnos y miembros del cuerpo docente con nombre y cargo respectivo.

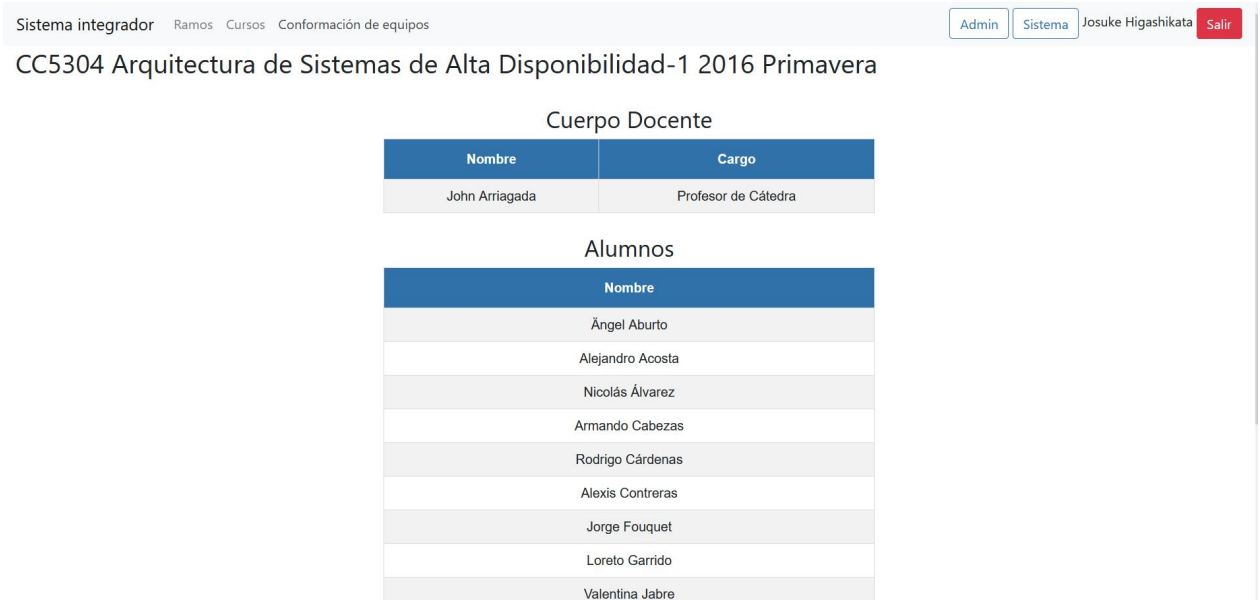

Figura 4.12: Vista de integrantes de un curso.

4.4.6. Vista: Equipos

La Figura 4.13 nos muestra varias tablas contiguas que representan a un equipo, y los integrantes respectivos de cada uno. En la versión actual de la plataforma no hay forma de cambiar la distribución de los alumnos en los equipos, por lo que se necesita reorganizarlos en el sistema de conformación de equipos. Lo que sí se puede realizar, tal como se indica en

Figura 4.13: Vista de los equipos formados para un curso.

la Figura 4.14, es cambiar el nombre asociado a cada equipo dentro de la plataforma. Esto debido a que cuando se crean los equipos, se les asigna un nombre genérico por defecto.

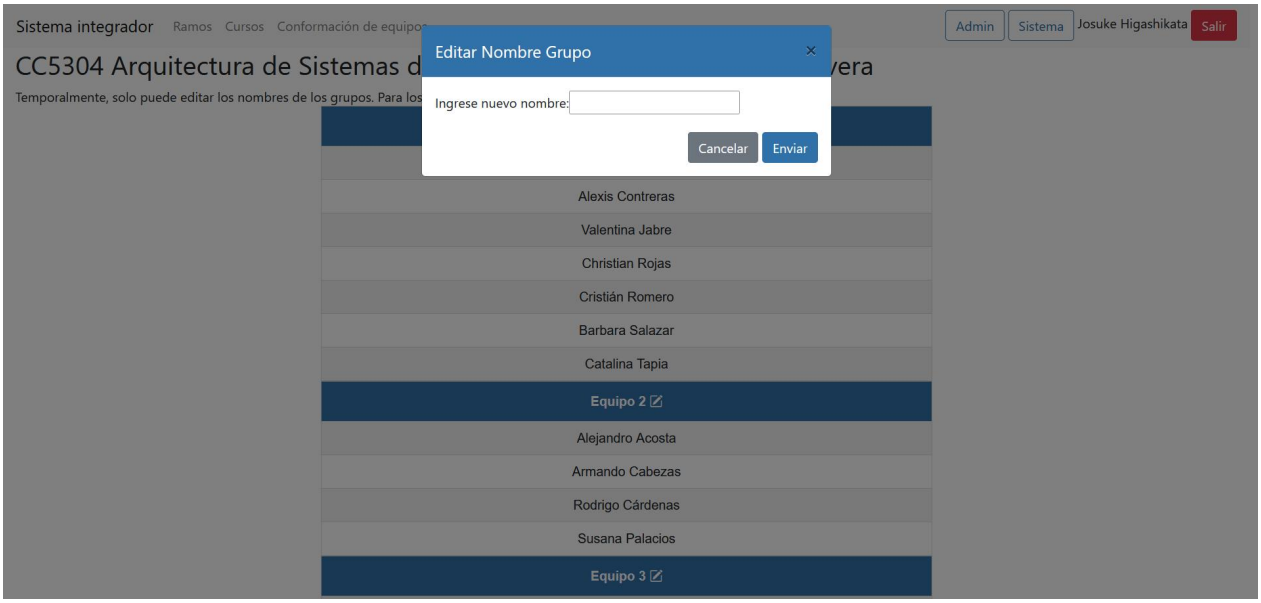

Figura 4.14: Modal para cambiar el nombre de un equipo.

4.5. Cambios en los Sistemas Legados

A continuación se indican los principales cambios realizados a los sistemas legados, para que puedan operar con el nuevo sistema integrador.

4.5.1. Sistema de Conformación de Equipos

Como se mencionó anteriormente, no fue necesario intervenir las interfaces de este sistema ya que tenía las librerías necesarias para la creación de la API. Esto implicó que no se agregaron vistas nuevas al mismo, y los cambios fueron únicamente a nivel de *backend*.

En este caso la API permite recibir la información de los cursos y sus respectivos integrantes, en donde se utiliza la lógica de los *scripts* de carga existentes, para así cargar correctamente los datos a la plataforma. Esta API también deja disponible datos necesarios para comunicarle al sistema integrador, los grupos que fueron formados dentro de la plataforma.

4.5.2. Sistema Mainreq

En el sistema Mainreq, se agregó el botón *"Cargar Alumnos"* en la vista a la que llega el usuario de tipo administrador después de ingresar con sus credenciales, como se muestra en la Figura 4.15. Este botón nos lleva a la vista mostrada en la Figura 4.16. Ésta permite agregar cualquier curso que se encuentre en la plataforma integradora, usando como identificador el código del curso. Al hacer click ese botón se cargan automáticamente los alumnos

almacenados en la base de datos de la plataforma integradora.

Además, se deja en claro las instrucciones de cómo operar la plataforma en esta vista, dado que no existía un estándar único para cargar los datos a la plataforma antes del desarrollo de este trabajo de título.

Figura 4.15: Landing page modificado (vista de administrador).

Figura 4.16: Vista para cargar datos de cursos y alumnos.

4.5.3. Sistema Coevaluaciones

En el sistema de Coevaluaciones, se agregó el botón *"+Cargar Curso"*, particularmente en la vista de Cursos del usuario Profesor, tal como se muestra en la Figura 4.17. Este botón nos lleva a la vista presentada en la Figura 4.18. Ésta solo tiene un *dropdown* con los ramos que se encuentran permitidos (cargados) en la plataforma.

Al apretar este botón, el sistema trae la información de todos las secciones del curso en cuestión del semestre en curso. La vista también muestra un pequeño *banner* indicando si se logró cargar el curso o no.

Figura 4.17: Vista de cursos para un perfil de usuario profesor.

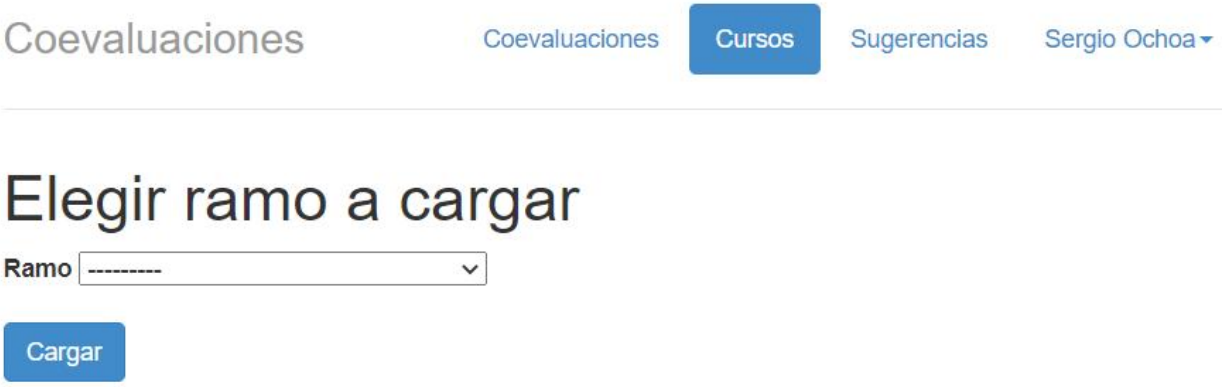

Figura 4.18: Vista para cargar cursos al sistema.

Capítulo 5

Evaluación del Sistema

En esta sección se describen las evaluaciones hechas para revisar la consistencia de los datos transferidos y almacenados en los distintos sistemas, y el correcto funcionamiento de los sistemas legados y del integrador.

5.1. Consistencia de los datos a través de los sistemas

Para comprobar que los datos se cargan correctamente a lo largo de cada una de las etapas del flujo, se realizaron una serie de pruebas en cada uno de los intercambios de información, así como también se revisó la consistencia de dicha información. A continuación se describen brevemente las pruebas realizadas en los distintos intercambios de información, y los resultados obtenidos.

5.1.1. Carga datos desde *Data Lake*

En la primera parte de esta prueba se tomó un ramo con código DCC, y se consideraron dos semestres de dictado para éste. Se elige uno de los semestres como actual (el semestre activo), y luego se solicitan los datos correspondientes al ramo seleccionado (curso, sección). Luego se cambia el semestre activo al siguiente, y se vuelve a solicitar los mismos datos asociados al ramo. Dado que los semestres son diferentes, se espera que el listado de integrantes entre ambas versiones del ramo sean diferentes, y correctos respecto a la información que se maneja en la plataforma UCampus.

La segunda parte de la prueba corresponde a tomar un ramo con código DCC, pero que es una instancia de un curso creado en el sistema. Luego se procede a eliminar la totalidad o una parte de la información del curso. Se registran los datos eliminados, y luego se vuelve a solicitar la información al *Data Lake*, para ver si el servicio complementa (repone) la información eliminada en forma manual.

Los resultados para ambas partes de la prueba fueron positivos, y se cumplieron las expectativas que se tenían. Los datos fueron restaurados correctamente para la segunda parte de la prueba, sin generar duplicidad de información en la base de datos. De igual forma, los resultados en la primera prueba muestran que cursos de distintos semestres tienen distinta composición de integrantes, y las listas son correctas respecto a lo que informa UCampus.

5.1.2. Carga y actualización automática desde *Data Lake*

Las siguientes pruebas comprobaron la capacidad del sistema de manejar un cambio de semestre, y la consecuente actualización de los datos en el sistema. También se realizaron cambios a la lista de integrantes de un curso en particular, ya sea agregando o quitando miembros a dicha lista.

Para que la prueba se considere exitosa, se necesita la actualización del semestre activo en que se encuentra la plataforma, así como también la obtención de los cursos nuevos asociados a los ramos en la plataforma. En caso que se modifique la lista de integrantes, ya sea agregando o eliminando miembros de los cursos, estos deben verse reflejados en el listado de integrantes del curso.

En ambos, los scripts de carga automática programados funcionaron perfectamente, y cargaron la información tal como se esperaba.

5.1.3. Sistema de Conformación de Equipos

La primera parte de la evaluación busca cargar los datos de un curso, cualquiera que esté presente en el sistema integrador, al sistema de conformación de equipos. Con esta información, luego se ejecuta el proceso de conformación de equipos. Esto último implica la creación de los integrantes en el sistema, la asignación correcta de roles de cada uno de ellos ,y la creación del proceso en sí asociado al curso recién cargado.

Luego de terminado el proceso de conformación de equipos, la segunda parte consiste en hacer que el sistema integrador recupere los grupos formados en la plataforma, y los deje disponibles a través la API para ser consultados.

Para la primera parte, se verifica con la plataforma integradora que los integrantes estén en el sistema, además de que estos deben tener el rol correspondiente (jefe de proyecto, desarrollador o ingeniero de calidad). También se verifica que se creó un nuevo proceso de conformación de equipos con los datos provenientes de la plataforma integradora.

El resultado exitoso de esta operación genera un id que contiene los datos del proceso, identificando las fechas, el tipo de cuestionario elegido para caracterizar a los participantes, y el curso involucrado. Todo esto es almacenado por la plataforma integradora, para luego poder recuperar por demanda la información que se requiera. Al consultar manualmente la API del sistema, se comprueba que el *id* corresponde a los datos del proceso ingresados anteriormente.

Para la segunda parte, una vez terminado el proceso y formados los equipos, se pide la información de estos usando el id anteriormente guardado. Se comprueba que los grupos formados en el sistema son los mismos que los que se disponibiliza en la API.

5.1.4. Intercambio de información hacia Mainreq y Coevaluaciones

Las pruebas realizadas para ambos sistemas fueron simples. Estas consistieron en cargar los datos desde el sistema integrador, asegurarse que los datos cargados en la base de datos eran consistentes con los datos recibidos desde el sistema integrador, y finalmente, probar que los usuarios creados eran funcionales en los login respectivos.

Para el sistema de Mainreq se observa una carga correcta de los datos. En caso de existir choques de nombre de usuario y no poder ser manejados por el sistema de forma automática, estos se notifican al profesor o administrador que está cargando la información para cargarlo manualmente.

Por otra parte, al momento de crear los usuarios, se manda un correo con las credenciales, que funcionan sin problemas para ingresar a la plataforma con uno de estos nuevos usuarios. Los usuarios son asignados a los equipos según estaban distribuidos en la plataforma integradora.

Para el sistema de Coevaluaciones se observa que los cursos son agregados al sistema de manera automática. En este sistema es posible editar los grupos, tanto su nombre como los miembros pertenecientes a cada uno de ellos. También es posible crear un proceso de coevaluación con el curso recién agregado. Los usuarios cargados en el sistema pueden solicitar un cambio de contraseña, y también ingresar a la plataforma para realizar una coevaluación.

5.2. Funcionamiento de los sistemas legados

Para comprobar que los sistemas legados funcionan correctamente, se crearon cuentas en cada uno de los sistemas, representando a los distintos tipos de usuarios. Además, se probaron varias acciones frecuentemente realizadas por cada uno de ellos. Luego se comprobaron los resultados, y se revisó que las interacciones fueran consistentes con lo que estaba en producción en ese momento.

Un detalle importante es que todos los cambios y pruebas fueron realizadas en un ambiente de desarrollo, debido a problemas técnicos y de permisos para poder subir los cambios a producción antes de terminado el plazo de entrega de este trabajo de título. A continuación se detallan las pruebas realizadas para cada sistema y los resultados obtenidos.

5.2.1. Pruebas al sistema de conformación de equipos

Para comprobar la estabilidad del sistema, se enviaron credenciales del tipo alumno y administrador al cuerpo docente del curso CC5401 (Ingeniería de Software II), quienes se ofrecieron para responder la encuesta que se utiliza en el curso. Para el caso del administrador del sistema, se revisaron las funcionalidades de creación, modificación y eliminación de procesos de conformación de equipos, así como también la organización de los alumnos dentro de dichos equipos.

Los miembros del cuerpo docente que probaron el sistema no encontraron ningún problema en el proceso de respuesta al cuestionario. Tampoco se encontraron problemas al guardar las respuestas y formar los equipos de forma automática, en base a las respuestas de los participantes. Usando las credenciales de administrador tampoco se encontraron diferencias respecto al funcionamiento tradicional de la plataforma.

5.2.2. Pruebas al sistema de Coevaluaciones

Para comprobar la estabilidad del sistema, se le solicitó a un grupo reducido de personas, (estudiantes del DCC habían utilizado el sistema coevaluaciones), responder un proceso de coevaluación ficticio. Éste fue creado usando los datos obtenidos desde el sistema integrador, por lo que también era necesario evaluar el comportamiento del sistema con los datos cargados.

Los usuarios respondieron la coevaluación como estudiantes, y no encontraron errores en el proceso. Un usuario profesor encontró un error para editar los grupos de cada curso, ya que la plataforma daba un error 500. Se comprobó que en producción con las mismas condiciones se generaba el mismo error, por lo que es un error del sistema y no de los cambios implementados. El resto del sistema funcionaba sin problemas.

5.2.3. Pruebas al sistema Mainreq

Para comprobar el nivel de estabilidad del sistema, se creó un proyecto con unos pocos integrantes en distintos roles. Además, se probaron las funcionalidades más importantes del sistema, como por ejemplo, la creación de hitos, requisitos y la generación de documentos. No se observaron errores durante el flujo normal de trabajo en la plataforma por parte de los usuarios que probaron el sistema.

Capítulo 6

Conclusiones y Trabajo Futuro

Este trabajo de memoria involucró la integración de usuarios de tres sistemas legados: *conformación de equipos, coevaluaciones y MainReq* (administrador de requisitos de proyectos de software). La integración se realizó a nivel de datos y servicios, por lo que se creó un sistema integrador, pero además se modificaron los sistemas legados para que pudieran interactuar con la nueva aplicación.

El objetivo de esta integración fue facilitar el uso conjunto de las herramientas antes indicadas, evitándole a los académicos de un curso, tener que realizar labores manuales asociadas al manejo de usuarios y equipos de trabajo. Estas labores manuales no sólo eran tediosas y generaban errores, sino que su ejecución requería cierto conocimiento por parte de quienes las realizaban.

El sistema implementado desacopló el manejo de usuarios y equipos de trabajo, y lo centralizó en el "sistema integrador", que ahora representa la fuente de verdad para los otros tres sistemas. Este desacoplamiento involucró la definición de APIs, que permiten a los sistemas legados evolucionar de manera autónoma, y mantener los mecanismos de interacción con las otras aplicaciones colaboradoras.

El sistema integrador se evaluó en términos de las funcionalidades implementadas necesarias para cumplir los objetivos propuestos de la memoria. Estas funcionalidades fueron encontradas adecuadas para resolver el problema inicial, aunque se dejó de lado la evaluación de la usabilidad del sistema, debido a que la mayor parte del trabajo corresponde al desarrollo del *back-end* y por un retraso importante en la migración de los cambios desde el ambiente de desarrollo al de producción.

El plan original consideraba realizar pruebas con los miembros del cuerpo docente en donde completaran tareas para probar las funcionalidades del sistema utilizando la técnica de *thinking aloud* para poder recolectar *feedback* y ayudar a encontrar cualquier error adicional no detectado tanto en los sistemas intervenidos como en el desarrollado.

Las pruebas se realizarían en el ambiente en producción de los sistemas en dos fases. La primera contemplaba correr con datos de prueba las interacciones entre los sistemas. La

segunda contemplaba utilizar datos reales cuando fueran a utilizar los sistemas en el curso CC5401.

Durante el proceso de migración de los cambios realizados en los sistemas, uno de los planes del área de desarrollo del DCC consistía en montar los sistemas legado en contenedores de Docker para poder manejarlos con mayor facilidad, plan que fallo debido a la aparición de numerosos *bugs* en la transición. Esto retraso la migración de los cambios necesarios para que los sistemas pudieran comunicarse entre si y dado que los sistemas ya estaban siendo utilizados, se decidió posponer las pruebas hasta terminado el semestre de Otoño 2022, en vista que los sistemas son utilizados por otros ramos del DCC y realizar cambios en dichos sistemas podría generar inestabilidades indeseadas.

En el desarrollo de la memoria y la intervención en los sistemas legados, se encontraron los siguientes *bugs* y particularidades:

- **Integrador:** Se puede crear mas de un proceso de conformación de equipos para cada curso. Esto genera problemas al momento de sincronizar los grupos. Es mejor resolver esta situación en el sistema de conformación de equipos eliminando los procesos adicionales.
- **Coevaluaciones:** Se encontró que agregar el miembros en el rol de ayudantes a un curso genera problemas de duplicación de dicho curso. Este problema genera problemas en el código en donde espera 1 entidad y esta recibiendo 2, botando el sistema. Esto se evidencia en *dropdowns* al momento de iniciar una nueva coevaluación, en donde aparece duplicado el curso en cuestión y también al momento de intentar editar un grupo de alumnos de dicho curso. En ambos casos el sistema se cae.
- **Mainreq:** No se puede eliminar ningún dato de la plataforma debido a restricciones impuestas por el desarrollador original.

  Existen ocasiones en donde el sistema al intentar generar un pdf para algún documento, este falla y retorna un error. Esto es debido a un *timeout* en alguno de los proceso, pero no pudo determinarse el origen.

Dentro de las consideraciones para el trabajo futuro asociado a este trabajo de título, se recomienda realizar una actualización de tecnologías a los sistemas que fueron intervenidos, ya sea haciendo el esfuerzo de actualizar los ya existentes o crearlos desde cero, pero tomando siempre como guía a los existentes.

Parte de las problemáticas que se encontraron en este trabajo de título se debieron a la falta de compatibilidad de los sistemas para ser corridos en el computador local del memorista, y complicaciones varias que surgieron cuando se realizó la copia de las máquinas en la que corrían los sistemas en los servidores del DCC, ya que se debieron llevar a cabo una serie de cambios manuales que no eran del todo claros a primera vista. En ese sentido, cuando se vaya a trabajar con sistemas legados que son muy diferentes entre ellos, se recomienda hacer una investigación profunda sobre el funcionamiento y las tecnologías usadas en cada uno de los sistemas, buscar la documentación de éstas y del sistema (en caso de existir). Finalmente, se deben comprobar vulnerabilidades descubiertas y resueltas en versiones posteriores, ya que ésta fue una de las limitantes importantes al elegir el tipo de aproximación a los sistemas.

Es importante destacar que el sistema desarrollado en esta memoria corresponde a una solución *ad hoc* a la problemática presentada en el contexto de trabajar con sistemas legados. Una actualización de los sistemas legados podría significar dejar obsoleto el sistema desarrollado, así como también generar nuevas oportunidades para agregar otros sistemas y funcionalidades a éste. Por lo tanto, la estabilidad del sistema integrador dependerá de los planes que tenga el Departamento respecto a la evolución de estos sistemas legados.

Bibliografía

[1] Nicolás Caracci. Sistemas de alimentación de información docente para el dcc. Memoria de ingeniería civil en computación, DCC, FCFM, Universidad de Chile., 2020.

[2] Francisco Hafón. Reingeniería de una herramienta de administración de requisitos: Mainreq. Memoria de ingeniería civil en computación, DCC, FCFM, Universidad de Chile., 2014.

[3] Roberto Riquelme. Sistema de evaluación del desempeño de los miembros de un equipo de desarrollo de software. Memoria de ingeniería civil en computación, DCC, FCFM, Universidad de Chile., 2014.

[4] Luis Silvestre. Diseño de equipos de desarrollo de software en escenarios universitarios. Tesis de magíster en ciencias mención computación, DCC, FCFM, Universidad de Chile., 2012.

[5] Ignacio Vallejos. Plataforma de monitoreo de la docencia de pregrado del dcc. Memoria de ingeniería civil en computación, DCC, FCFM, Universidad de Chile., 2021.