



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

MEJORA DE LA GOBERNABILIDAD Y LA CAPACIDAD DE INTEGRACIÓN DEL SISTEMA
U-PROYECTOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

JOAQUÍN IGNACIO PÉREZ ARAYA

PROFESOR GUÍA:
SERGIO OCHOA DELORENZI

PROFESOR CO-GUÍA:
JUAN ARRIAGADA CANCINO

MIEMBROS COMISIÓN:
JOSÉ BENGURIA DONOSO
MARÍA CECILIA BASTARRICA PIÑEYRO

SANTIAGO DE CHILE
2023

Resumen

El proceso de gestión y administración de los proyectos del Departamento de Ciencias de la Computación es una parte vital del funcionamiento del mismo, para facilitar dicho proceso, en 2014 se implementó el sistema U-Proyectos, que permite gestionar los proyectos de los distintos académicos del departamento. Éste además permite registrar las visitas de académicos externos y los depósitos y/o transferencias de dinero asociados a proyectos de investigación y vínculo con el medio.

La última actualización de este sistema fue realizada en 2017, por lo que el sistema ha ido quedando obsoleto en términos tecnológicos, y limitado en términos de gobernabilidad y capacidad de integración con otros sistemas del DCC. Tampoco existe documentación extensiva que permita conocer la arquitectura interna o su despliegue en los ambientes de desarrollo y en el productivo.

El objetivo de esta memoria es restaurar este sistema para que sea fácilmente extensible, siguiendo los lineamientos definidos por el área de desarrollo del departamento, ayudando a mejorar su gobernabilidad y capacidad de integración con otros sistemas del DCC.

Por lo tanto, en esta memoria se documentó de forma extensiva el sistema, se modificó el código fuente para que fuese armonioso levantar un ambiente de desarrollo, se actualizaron las versiones de las tecnologías usadas en la implementación, se incorporó un mecanismo de autenticación basado en U-Pasaporte, y se añadió una API básica de fácil extensión para permitir el acceso de consulta desde otros sistemas.

Para evaluar esta memoria, y así la mejora de la gobernabilidad y capacidad de integración del sistema, se realizó una evaluación comparativa de ambas versiones del sistema, considerando a los niveles de cumplimiento de los requisitos de gobernabilidad e integración definidos por el DCC.

Los resultados obtenidos muestran una clara mejora de la gobernabilidad del sistema. Dos de los puntos importantes para lograr esto, son la extensiva documentación creada, y la implementación de Docker para facilitar enormemente la puesta en producción de cualquier tipo de ambiente de la plataforma.

A futuro se esperan realizar cambios de las librerías PHP utilizadas, dado que algunas están deprecadas. Además, se buscará extender la API para soportar un mayor número de casos de uso. Finalmente, se restaurará la funcionalidad de envíos automáticos de correos.

Tabla de Contenidos

Resumen	i
1. Introducción	1
1.1. Situación actual	2
1.2. Objetivos de la memoria	3
1.3. Estructura del documento	4
2. Análisis del escenario a Intervenir	5
2.1. Identificación del problema y contexto abordado en U-Proyectos	5
2.2. Análisis del sistema U-Proyectos	7
2.3 Arquitectura del sistema	15
2.4. Nivel de cumplimiento de los lineamientos de desarrollo del DCC	20
2.5. Desafíos abordados para extender el U-Proyectos	21
2.6. Extensiones al sistema U-Proyectos	21
3. Ajustes al sistema U-Proyectos	23
3.1. Ajustes al modelo de datos	23
3.2. Ajustes al software	24
4. Extensión al sistema U-Proyectos	28
4.1 Actualización de PHP	28
4.2 Tablero del producto	29
4.3 Inclusión del mecanismo de autenticación de usuarios con U-Pasaporte	30
4.4 Descripción de la API de U-Proyectos	31
5. Evaluación del trabajo realizado	32
5.1 Proceso de evaluación seguido	33
5.2 Resultados de la evaluación	33
5.3 Nivel de cumplimiento de los lineamientos de gobernabilidad e integración con otros sistemas	38
6. Conclusiones y trabajo a futuro	42
7. Bibliografía	43
Anexos	44
Anexo A: Requisitos de gobernabilidad e integración de sistemas del DCC	44
Anexo B: Readme del código	47
Anexo C: Documentación legado	49

1. Introducción

El Departamento de Ciencias de la Computación (DCC) ha desarrollado múltiples sistemas de información, los cuales se han ido evolucionado a través de múltiples memorias de título, desarrollos del curso de Ingeniería de Software II (CC5401), y también por medio de iniciativas personales de académicos del Departamento. Uno de estos sistemas es el de Gestión de Proyectos del DCC (conocido como U-Proyectos), creado bajo el contexto del curso de CC5401 en 2014. Este sistema fue extendido en al menos una oportunidad en el mismo curso, y luego extendido nuevamente en 2015 por un memorista del departamento.

U-Proyectos es una herramienta web que permite gestionar, analizar y documentar la información, tanto estadística como financiera de proyectos de los académicos del departamento. Además, permite gestionar los investigadores que participan o han participado en dichos proyectos, independiente si éstos forman parte del DCC o no; incluso, existe una funcionalidad dedicada al registro de investigadores temporales del DCC (académicos visitantes y post-doctorandos).

A grandes rasgos, la plataforma puede realizar las siguientes tareas con los proyectos: 1) ingresar proyectos, los cuales tienen asociado un título, identificador, estado, fechas de inicio y término, investigadores e información financiera, 2) editar y eliminar proyectos ya ingresados, y 3) revisar proyectos, imprimirlos y generar estadísticas.

Asimismo, el sistema permite llevar un registro manual de las transferencias de dinero realizadas por distintas organizaciones hacia el DCC, permitiendo realizar búsquedas, exportar la información financiera como plantillas de cálculo. Estos registros son únicamente para fines de reportería financiera interna, la plataforma no mueve el dinero entre centros de costo y/o cuentas bancarias.

Adicionalmente, la aplicación dispone de la funcionalidad de hacer seguimiento de los dineros que deberían llegar al DCC en fechas determinadas. Llevando cuenta de los depósitos pendientes, depósitos por confirmar y los informes de depósitos.

Los roles principales de la plataforma son los siguientes: la Secretaría de Investigación, el Área de Administración y la Dirección del DCC. La Secretaría de Investigación hace uso de la aplicación para registrar las transferencias de dinero por las organizaciones financiadoras hacia el DCC (ANID, CORFO, etc.); estas transferencias se realizan indirectamente a través de la Casa Central de la Universidad. El área de Administración y el Director del DCC consultan los informes financieros de los proyectos, y sacan estadísticas de estos para apoyar su proceso de toma de decisiones.

Los proyectos de investigación son parte fundamental del funcionamiento del departamento, por lo que este sistema de información es una herramienta vital para el control financiero de dichos proyectos. Sin este sistema, la gestión de estos proyectos sería realizada por medio de planilla de cálculo.

1.1. Situación actual

Esta plataforma es uno de múltiples sistemas de información creados por iniciativas de académicos, que al ser desarrollados por múltiples estudiantes del departamento en varias iteraciones, terminan siendo productos confiables y robustos, pero no disponen de información rigurosa que dé cuenta de los usuarios, los procesos, la arquitectura, y contenidos actuales de cada sistema.

Con respecto al sistema U-Proyectos¹, su única documentación formal es la memoria de Carlos Dettoni [1]. En consecuencia, para mejorar la gobernabilidad del sistema, se hace necesario revisar el código fuente de la aplicación, para identificar su arquitectura y modelo de datos actual, los servicios que ésta provee, y el nivel de validez de las indicaciones para poner la aplicación en producción. Esta situación de incerteza crea una barrera de entrada considerable para realizar mejoras, incorporar nuevas funcionalidades al sistema y/o dar soporte a los usuarios del sistema.

La información general respecto al diseño se puede extraer del documento de memoria [1], sin realizar un análisis exhaustivo al código fuente que está en producción. El stack tecnológico utilizado consiste en una base de datos MySQL, un frontend implementado con HTML5, CSS3 y JavaScript (que utiliza las librerías de jQuery y jQueryUI), y un backend escrito en PHP 5.5 montado en un servidor Apache2. El modelo de datos es relacional, y debe ser revisado en forma detallada. La arquitectura básica del sistema utiliza el patrón MVC (Modelo-Vista-Controlador) [4].

Como se mencionó antes, esta herramienta fue extendida por última vez a mediados del 2017, durante el trabajo de reingeniería realizado por un memorista. En consecuencia, el software lleva aproximadamente 7 años sin mantenimiento correctivo y/o evolutivo. Por otra parte, la versión de PHP que se está utilizando dejó de tener soporte en 2016 [3]. Debido a que este es un sistema importante para el departamento, y en vista de lo expuesto anteriormente, la plataforma necesita una actualización profunda si se pretende mantenerla en producción por algunos años más.

Los tres factores mencionados anteriormente son un riesgo para esta herramienta, puesto que, si no se resuelven oportunamente, estos harán que se requiera re-implementar completamente del sistema pues no habrá mucho que rescatar, excepto las ideas y algún diseño del sistema legado. Si la herramienta no está disponible o es

¹ Su URL actual en un ambiente de pruebas es: uproyectos-test.dcc.uchile.cl

vulnerada por algún ataque externo, afectará al proceso de monitoreo, y eventualmente la toma de decisiones del DCC en esta materia.

Para prevenir lo anterior, a principios del año 2021 se creó el Área de Desarrollo del Software del DCC. Dicha área definió lineamientos que permitirán un aumento significativo de la gobernabilidad e integración de los sistemas del Departamento (véase el Anexo I). Bajo este contexto se inserta este trabajo de memoria.

1.2. Objetivos de la memoria

El objetivo principal de esta memoria es mejorar la gobernabilidad y capacidad de integración del sistema U-Proyectos, de manera que éste se integre con el resto de las aplicaciones en el ecosistema de servicios del DCC y facilitar su mantención y extensión. El logro de este objetivo mejorará y facilitará los desarrollos futuros que se realicen sobre la aplicación, extendiendo así la vida útil de este software.

Para cumplir con el objetivo general, se definieron los siguientes objetivos específicos:

- Llevar el sistema a un nivel de gobernabilidad que permita desarrollos futuros armoniosos, haciendo que éste cumpla con los Requisitos de Gobernabilidad e Integración de Sistemas del DCC (véase Anexo I).
- Extender el sistema con la finalidad de dejar en evidencia la mejora de las capacidades de gobernabilidad e integración de la nueva versión de U-Proyectos.

Para medir el grado de éxito del trabajo de esta memoria se evaluaron los siguientes aspectos del trabajo realizado:

- El aumento de gobernabilidad e integralidad del proyecto, considerando los puntos enumerados en el documento de Requisitos de Gobernabilidad e Integración (Anexo I).
- La mejora del impacto del sistema, a partir de la extensión realizada. Para ello se realizará una evaluación conjunta con el área de sistemas y el área de desarrollo del DCC para determinar la mejora, considerando también el aumento de las necesidades de integración de U-Proyectos, con otras aplicaciones y servicios del ecosistema del DCC.

1.3. Estructura del documento

La estructura de la presente memoria es la que se indica a continuación. En el capítulo 2 se muestra un análisis y documentación de la aplicación legada según los requisitos definidos por el DCC para mantener un buen nivel de Gobernabilidad e Integración de sus Sistemas. En el capítulo 3 se describen los cambios y ajustes realizados a la plataforma para mantener y restaurar la operatividad de ésta. En el capítulo 4 se detallan las extensiones realizadas al sistema para demostrar el aumento de gobernabilidad e integración de éste. En el capítulo 5 se realiza una evaluación extensa respecto al impacto causado por los cambios y/o extensiones de la aplicación, en el ámbito de la gobernabilidad e integración de la plataforma. Finalmente, en el capítulo 6 se presentan las conclusiones y se indica el posible trabajo a futuro a abordar para mejorar la aplicación.

2. Análisis del escenario a Intervenir

En esta sección se presenta el análisis previo a la intervención al sistema, listando y documentando todas las principales partes presentes en éste, y generando documentación nueva en el caso de no existir anteriormente. El detalle de la funcionalidad de la aplicación legada está reportada en [1].

2.1. Identificación del problema y contexto abordado en U-Proyectos

A continuación se presenta el tablero de exploración del problema y contexto abordado, considerando la versión legada de la aplicación U-Proyectos (Fig. 1). Luego se explican los distintos componentes (secciones) del tablero.

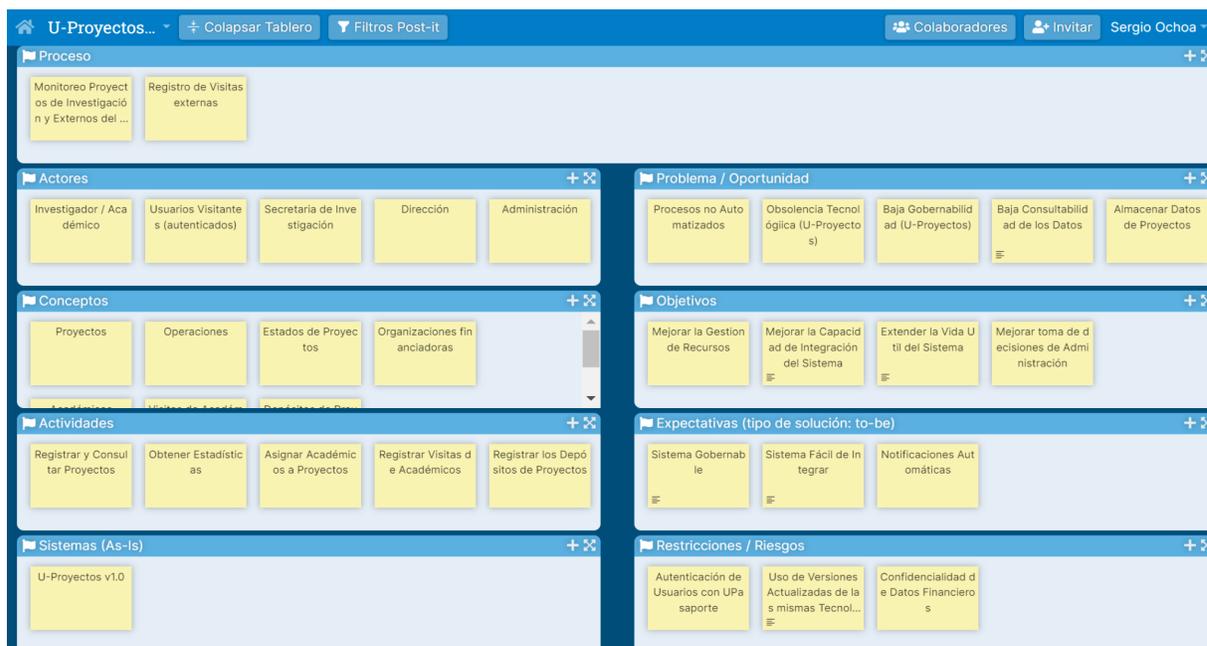


Figura 1. Tablero de exploración del problema y contexto para U-Proyectos

2.1.1 Procesos apoyados

Cada elemento de esta sección del tablero corresponde a un proceso o macro-actividad que apoya el sistema. En este caso, los principales usuarios son la secretaria de investigación y el personal del área de administración del DCC, que registran y consultan la información acerca de: 1) proyectos realizados por académicos del DCC, 2) visitas de investigadores externos (visitantes, post-doctorandos y alumnos de postgrado de intercambio).

A pesar de contar con estos dos macro-servicios, el grueso de la actividad del sistema tiene que ver con la gestión de proyectos; por ejemplo, el Análisis Financiero, el Monitoreo de Transferencias y Depósitos, el Registro de Visitas Externas, y la Documentación asociada a estos.

2.1.2 Actores

Cada elemento de esta sección corresponde a un perfil o tipo de usuario distinto, que participa directa o indirectamente en el proceso. A continuación se describen algunos de estos (ver detalles en [1]):

- *Investigadores/Académicos*. Estos son los responsables de los proyectos o colaboradores en estos.
- *Secretaría de Investigación*. Ésta es quien debe registrar/actualizar los datos de los proyectos, sus actividades, y obtener reportes del sistema.
- *Administración*. Estos usuarios registran y monitorean los fondos (la información financiera en general) de los proyectos administrados por el Departamento, y también por aquellos que manejan fondos de manera externa, pero que contribuyen financieramente al DCC. Por ejemplo, los proyectos Fondecyt o Milenios.
- *Visitantes*. Son usualmente investigadores de otras universidades que vienen al DCC a realizar estadías de trabajo.

2.1.3 Problema / Oportunidad

Las oportunidades que se han identificado en este proceso son los siguientes: 1) la existencia de procesos manuales (pero automatizables) que apoyan el manejo de los proyectos, 2) la falta de visualizaciones de datos de proyectos, y 3) la oportunidad de almacenamiento de más datos asociados a los proyectos.

Por otra parte, los problemas asociados al software de soporte a este proceso (es decir, al sistema legado) son: 4) la poca gobernabilidad, 5) la obsolescencia tecnológica del mismo y 6) su baja capacidad de integración con otros sistemas servicios del DCC.

2.1.4 Conceptos

Dentro del proceso se manejan los siguientes conceptos (o entidades de datos): proyecto, académico, operaciones financieras, fondos, depósitos, organizaciones financiadoras, y visitas de académicos (más detalles en [1]).

2.1.5 Objetivos

Los objetivos principales de este proceso son Mejorar la Gestión de Recursos del Departamento y Mejorar la toma de decisiones de la Administración del Departamento, a partir de estas metas, nace y se desarrolla U-Proyectos.

Y particularmente, al intervenir el sistema de U-Proyectos, se tienen estos objetivos focalizados en la plataforma en sí: Mejorar la Capacidad de Integración del Sistema y Extender la vida útil del Sistema.

2.1.6 Actividades

Las actividades asociadas a este proceso son: Registrar y Consultar Proyectos, Registrar Visitas de Académicos Externos, Obtener Estadísticas, Asignar Académicos a Proyectos y Registrar los Depósitos de Proyectos. Las vistas de la aplicación donde se realizan estas actividades se mencionan en el Análisis del Sistema en la siguiente sección de este capítulo.

2.1.7 Expectativas (tipo de solución)

A nivel de aplicación, la expectativa de la solución es crear notificaciones automáticas para alertar los depósitos y/o transferencias pendientes. A nivel de proyecto (esta memoria) es tener un sistema gobernable y fácil de integrar.

2.1.8 Soluciones

La solución existente de apoyo a este proceso es únicamente el sistema U-Proyectos.

2.1.9 Restricciones / Riesgos

El principal riesgo de este proceso es que la plataforma contiene datos sensibles del departamento, como por ejemplo los datos financieros de los proyectos. La autenticación de usuarios utilizando U-Pasaporte y el uso de versiones actualizadas de las mismas tecnologías, son restricciones que la plataforma debe abordar.

2.2. Análisis del sistema U-Proyectos

Tal como se mencionó anteriormente, U-Proyectos es una plataforma web que tiene como principal objetivo llevar un registro de los diversos proyectos de investigación del Departamento. A continuación se describen brevemente las vistas de los distintos tipos de usuario soportados por el sistema.

Este análisis se obtiene a partir del documento del memorista Carlos Dettoni [1], y de una revisión de las características de la plataforma. En esta sección se mencionan los

roles existentes en la aplicación, y posteriormente, una descripción resumida de cada una de las vistas que tiene la plataforma.

La aplicación contiene permisos para los distintos tipos de usuarios, quienes tienen acceso a diferentes vistas según su rol, en la Tabla 1 se muestran los roles existentes y su nivel de acceso a la plataforma.

Tabla 1. Roles soportados por la plataforma, y sus vistas permitidas

Nombre Rol	Id Rol (interno)	Vistas permitidas
Director	112	Proyectos, Visitas, Investigadores, Cargos, Informe financiero, Estadísticas proyectos, Estadísticas visitas, Log (Registro) y Créditos.
Administración	113	Proyectos, Visitas, Investigadores, Cargos, Informe financiero, Estadísticas proyectos, Estadísticas visitas y Créditos.
Secretaria de Investigación (SI)	114	Proyectos, Visitas, Investigadores, Cargos, Transferencias, Informe financiero, Estadísticas proyectos, Estadísticas visitas y Créditos.
Público (usuario anónimo)	115	Proyectos (público) y Créditos.
Root	116	Proyectos, Visitas, Investigadores, Cargos, Transferencias, Informe financiero, Estadísticas proyectos, Estadísticas visitas, Log (Registro) y Créditos. (Todos)
Académico	117	Proyectos, Visitas, Investigadores, Cargos, Informe financiero, Estadísticas proyectos, Estadísticas visitas y Créditos.

2.2.1 Proyectos

Al entrar en la plataforma, luego de autenticarse, un usuario público sólo puede ver y buscar proyectos de forma básica por medio del título de proyecto (véase la Figura 2). Éste no tiene acceso a las demás funcionalidades de la plataforma. Además, si dicho usuario accede a un proyecto en particular, sólo verá información limitada.



Figura 2. Interfaz de proyectos del sistema U-Proyectos (usuario público)

Para usuarios registrados con un rol en la plataforma, la vista de proyectos presenta más funcionalidades; por ejemplo, dispone de una búsqueda avanzada con filtros tales como fecha, tipo e investigadores asociados al proyecto. Esta vista permite también exportar en formato *.xlsx* o en formato *.json*, la lista actual de los proyectos que están siendo mostrados. Si el rol del usuario es Root o SI, la vista permite añadir Proyectos a la plataforma por medio de una vista extra (Fig. 3).

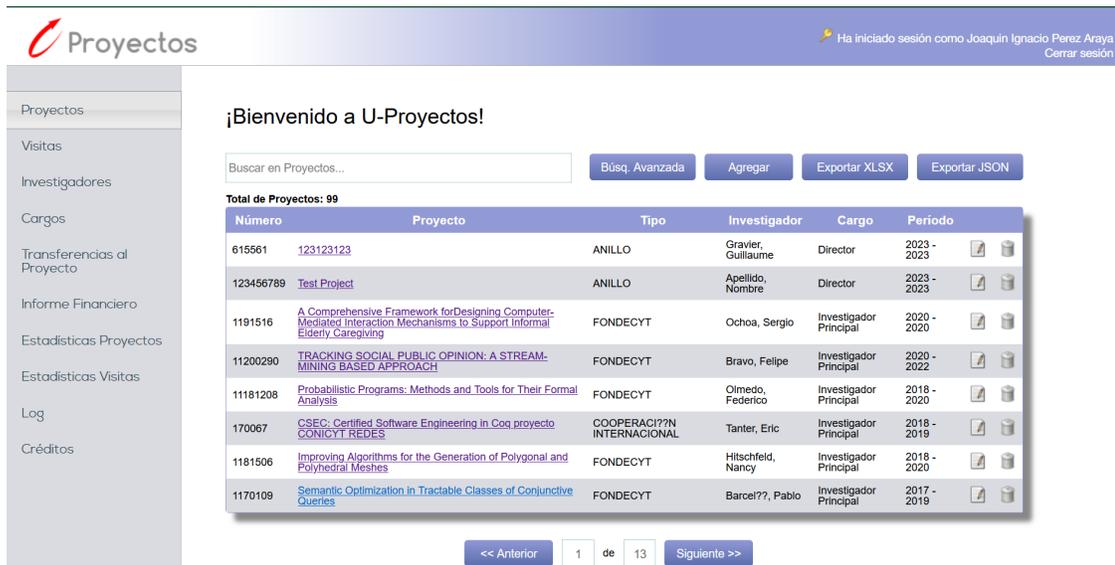
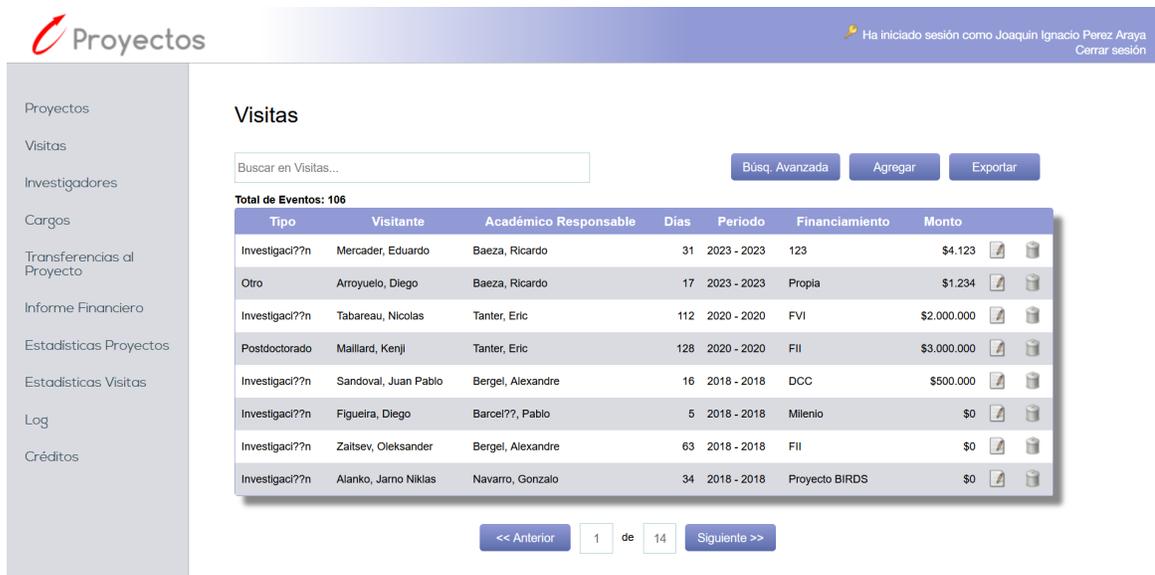


Figura 3. Interfaz de proyectos del sistema U-Proyectos (usuario root).

2.2.2 Visitas

En esta vista se pueden ver las visitas registradas que ha recibido el Departamento. Esta vista por defecto muestra el listado de todas las visitas que se han realizado, junto a la información básica asociada a ésta (ver Figura 4). Además, entrega opciones para realizar una búsqueda básica, una búsqueda avanzada, exportar la lista actual de visitas en formato .xlsx. Si el rol del usuario es Root o SI, esta vista permite además añadir Visitas nuevas a la plataforma por medio de una vista extra aparte.



Tipo	Visitante	Académico Responsable	Días	Periodo	Financiamiento	Monto
Investigaci??n	Mercader, Eduardo	Baeza, Ricardo	31	2023 - 2023	123	\$4.123
Otro	Arroyuelo, Diego	Baeza, Ricardo	17	2023 - 2023	Propia	\$1.234
Investigaci??n	Tabareau, Nicolas	Tanter, Eric	112	2020 - 2020	FVI	\$2.000.000
Postdoctorado	Maillard, Kenji	Tanter, Eric	128	2020 - 2020	FII	\$3.000.000
Investigaci??n	Sandoval, Juan Pablo	Bergel, Alexandre	16	2018 - 2018	DCC	\$500.000
Investigaci??n	Figueira, Diego	Barcel??, Pablo	5	2018 - 2018	Milenio	\$0
Investigaci??n	Zaitsev, Oleksander	Bergel, Alexandre	63	2018 - 2018	FII	\$0
Investigaci??n	Alanko, Jarno Niklas	Navarro, Gonzalo	34	2018 - 2018	Proyecto BIRDS	\$0

Figura 4. Interfaz de visitas del sistema U-Proyectos (usuario root).

2.2.3 Investigadores

Por defecto esta vista muestra todos los investigadores registrados dentro de la plataforma. Allí se indica la información básica como el nombre, correo, tipo de investigador, institución, etc. En las opciones asociadas a esta vista están, búsqueda simple/avanzada, exportar a .xlsx o .json. Si el rol del usuario es Root o SI, éste puede añadir Investigadores a la plataforma por medio de un menú pequeño que se despliega en la esquina inferior derecha (Fig. 5).

Proyectos

Visitas

Investigadores

Cargos

Transferencias al Proyecto

Informe Financiero

Estadísticas Proyectos

Estadísticas Visitas

Log

Créditos

Ha iniciado sesión como Joaquín Ignacio Pérez Araya
Cerrar sesión

Investigadores

Buscar en Investigadores...

Búsq. Avanzada Agregar Exportar XLSX Exportar JSON

Total de Investigadores: 135

Nombre	Correo	Tipo	Categoría	Institución	País	Fecha Ingreso
Antonopoulos, Timos		Externo	No tiene	Hasselt University	Bélgica	01/09/2016
Fuentes Sepúlveda, José	fs.pepe@gmail.com	Externo	No tiene	Universidad de Chile	Chile	04/10/2016
Gravier, Guillaume	guig@irisa.fr	Externo	No tiene	Rennes	Francia	13/11/2017
Leander, Nils-Gregor		Externo	No tiene	Technical University	Denmark	01/09/2016
Merino, Leonel	merino@inf.unibe.ch	Externo	No tiene	University of Bern	Suiza	09/01/2018
Viola Deambrosis, Alfredo		Externo	No tiene	Universidad de la República	Uruguay	31/08/2016
Alanko, Jarno Niklas	alanko.jarno@gmail.com	Externo	No tiene	Universidad de Helsinki	Finlandia	04/09/2018
Aldrich, Jonathan		Externo	No tiene	Carnegie Mellon University	Pensilvania	30/08/2016

<< Anterior 1 de 17 Siguiente >>

Figura 5. Interfaz de investigador del sistema U-Proyectos (usuario root).

2.2.4 Cargos en proyectos

La vista de cargos muestra los cargos posibles disponibles. Un cargo se define como un rol (por ejemplo, responsable, o director de un proyecto), junto con un tipo de proyecto (por ejemplo, ANILLO, FONDEF, etc.). Las funciones disponibles de esta vista son de búsqueda e inserción de cargos, sólo si el rol de usuario es Root o SI (Figura 6).

Proyectos

Visitas

Investigadores

Cargos

Transferencias al Proyecto

Informe Financiero

Estadísticas Proyectos

Estadísticas Visitas

Log

Créditos

Ha iniciado sesión como Joaquín Ignacio Pérez Araya
Cerrar sesión

Cargos

Buscar en Cargos...

Agregar

Total de Cargos: 49

Cargo (EN)	Cargo (ES)	Tipo de Proyecto	Prioridad
Director	Director	OTROS	1
Director	Director	EXTERNO	1
Director	Director	ANILLO	1
Director	Director	FONDEF	1
Director	Director	COOPERACIÓN INTERNACIONAL	1
Director	Director	MILENIO	1
Director	Director	FONDECYT	1
Sub Director	Sub Director	ANILLO	1

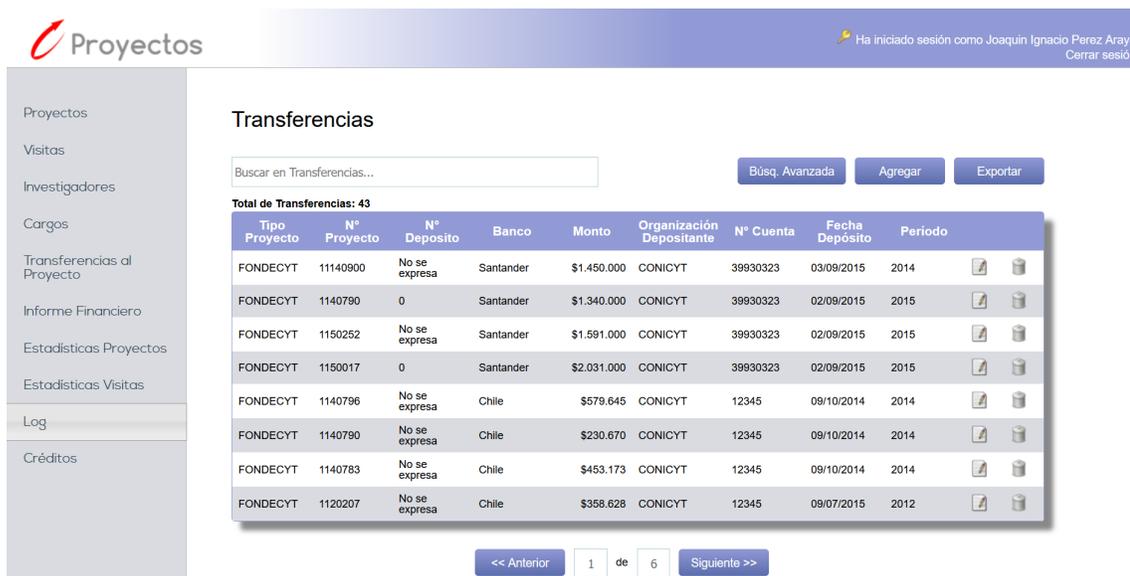
<< Anterior 1 de 7 Siguiente >>

Figura 6. Interfaz de cargos del sistema U-Proyectos (usuario root).

2.2.5 Transferencias

En esta vista, por defecto, se ven todas las transferencias asociadas a proyectos. En la barra superior de búsqueda se puede buscar transferencias por código de proyecto, mientras que la opción de búsqueda avanzada envía al usuario a otra vista que le permite buscar por tipo de proyecto, datos del depósito y fecha de éste (Figura 7).

El botón de exportar descarga la lista actual de transferencias como un documento de Excel (.xlsx). Si el usuario es Root o SI, se pueden agregar transferencias desde una visita distinta.



Projectos

Ha iniciado sesión como Joaquín Ignacio Pérez Araya
Cerrar sesión

Projectos

Visitas

Investigadores

Cargos

Transferencias al Proyecto

Informe Financiero

Estadísticas Proyectos

Estadísticas Visitas

Log

Créditos

Transferencias

Buscar en Transferencias...

Búsq. Avanzada Agregar Exportar

Total de Transferencias: 43

Tipo Proyecto	N° Proyecto	N° Depósito	Banco	Monto	Organización Depositante	N° Cuenta	Fecha Depósito	Período		
FONDECYT	11140900	No se expresa	Santander	\$1.450.000	CONICYT	39930323	03/09/2015	2014		
FONDECYT	1140790	0	Santander	\$1.340.000	CONICYT	39930323	02/09/2015	2015		
FONDECYT	1150252	No se expresa	Santander	\$1.591.000	CONICYT	39930323	02/09/2015	2015		
FONDECYT	1150017	0	Santander	\$2.031.000	CONICYT	39930323	02/09/2015	2015		
FONDECYT	1140796	No se expresa	Chile	\$579.645	CONICYT	12345	09/10/2014	2014		
FONDECYT	1140790	No se expresa	Chile	\$230.670	CONICYT	12345	09/10/2014	2014		
FONDECYT	1140783	No se expresa	Chile	\$453.173	CONICYT	12345	09/10/2014	2014		
FONDECYT	1120207	No se expresa	Chile	\$358.628	CONICYT	12345	09/07/2015	2012		

<< Anterior 1 de 6 Siguiente >>

Figura 7. Interfaz de Transferencias del Sistema U-Proyectos (usuario root).

2.2.6 Informes Financieros

La vista de informes financieros presenta dos pestañas, una de depósitos pendientes y otra de depósitos por confirmar. En depósitos pendientes, en cada depósito de esta lista está la opción de Notificar, donde se genera automáticamente un correo para ser enviado, sin embargo, esta funcionalidad no está activa de momento. Adicionalmente se puede descargar esta lista en formato Excel (.xlsx) usando el botón Exportar (Fig. 8).

Projectos Ha iniciado sesión como Joaquín Ignacio Pérez Araya
Cerrar sesión

Projectos
Visitas
Investigadores
Cargos
Transferencias al Proyecto
Informe Financiero
Estadísticas Proyectos
Estadísticas Visitas
Log
Créditos

Informe Financiero

Depósitos Pendientes **Depósitos por Confirmar**

Projectos con Depósitos Pendientes Exportar

Total de Depósitos Pendientes: 54 | Total Recaudado: \$3.085.537 | Total Pendiente: \$131.683.377 | Total Esperado: \$134.768.914

ID	Nombre del Proyecto	Periodos	Monto Recaudado	Monto Pendiente	Monto Esperado	
123456789	Test Project	2023	0	9.999	9.999	Notificar
615561	123123123	2023	0	123.123	123.123	Notificar
1234	Test	2023	0	12.315	12.315	Notificar
11200290	TRACKING SOCIAL PUBLIC OPINION: A STREAM-MINING BASED APPROACH	2020, 2021, 2022	0	0	0	Notificar
1191516	A Comprehensive Framework for Designing Computer-Mediated Interaction Mechanisms to Support Informal Elderly Caregiving	2020	0	0	0	Notificar
1181506	Improving Algorithms for the Generation of Polygonal and Polyedral Meshes	2018, 2019, 2020	0	6.629.000	6.629.000	Notificar
1150252	Reducing Social Isolation of Elderly People Using Social Media and Ambient Intelligence	2016, 2017	0	1.258.000	1.258.000	Notificar

<< Anterior 1 de 8 Siguiente >>

Figura 8. Interfaz del Informe financiero (depósitos pendientes) del sistema U-Proyectos (usuario root)

En depósitos por confirmar se muestra una lista de depósitos. Si se hace click en el nombre del proyecto, se accede a un informe detallado de los montos esperados asociado a dicho depósito, junto con la opción de ingresar justificación de dichos montos. Esta lista también puede ser descargada en formato Excel (Figura 9).

Projectos Ha iniciado sesión como Joaquín Ignacio Pérez Araya
Cerrar sesión

Projectos
Visitas
Investigadores
Cargos
Transferencias al Proyecto
Informe Financiero
Estadísticas Proyectos
Estadísticas Visitas
Log
Créditos

Informe Financiero

Depósitos Pendientes **Depósitos por Confirmar**

Projectos con Depósitos por Confirmar Exportar

Hacer click en un proyecto para ver su informe de depósitos.

Total de Cobros Pendientes: 16 | Total Confirmado: \$2.868.034 | Total Sin Confirmar: \$29.727.695 | Total Esperado: \$16.545.066

ID	Nombre del Proyecto	Periodos	Monto Confirmado	Monto Sin Confirmar	Monto Esperado	
11140900	Adoption-Centric Specification of Semantic Web Language Profiles	2014	0	1.450.000	483.333	
1150252	Reducing Social Isolation of Elderly People Using Social Media and Ambient Intelligence	2015	0	1.591.000	663.000	
1150017	Gradual Software Verification: Foundations and Applications	2015	0	2.031.000	1.189.333	
1120094	ALMA: Analyses and Tools for Monitoring and Improving Software Quality	2012, 2013, 2014	74.993	3.474.000	1.171.000	
1140790	Efficiently and Effectively Querying Linked Data on the Web	2014, 2015	230.670	2.505.000	710.000	
11110404	Data Exchange and Schema Mapping Management with Directional Constraints	2011, 2012	0	2.297.000	520.205	
1130104	Efficient Approximations of Conjunctive Queries	2013, 2014	67.568	273.000	737.750	
PE-001-2014	Ejecución de Pruebas de Stress sobre Plataforma de Postulaci2n en L'Pnea Aconcagua	2014	0	3.507.695	3.507.695	
1120054	Compressed Data Structures and Compressibility Measures	2012	97.515	0	538.500	
1110066	Compact Data Structures for Information Retrieval	2011	0	1.585.000	652.250	

<< Anterior 1 de 2 Siguiente >>

Figura 9. Interfaz del informe financiero (depósitos por confirmar) del sistema U-Proyectos (usuario root)

2.2.7 Estadísticas de Proyectos y Visitas

En estas dos vistas se puede obtener un resumen de las estadísticas asociadas a los proyectos como a las visitas realizadas por investigadores (Figura 10).

Proyectos

Ha iniciado sesión como Joaquín Ignacio Pérez Araya
Cerrar sesión

Proyectos

Estadísticas Proyectos

Proyectos por Año

Tipos de Proyecto

Estadísticas de Proyectos por Año

Académicos participantes
Deje en blanco para seleccionarlos a todos

Intervalo de fechas
Comenzando el Hasta

Tipos de proyectos

TODOS
 ANILLO COOPERACION INTERNACIONAL EXTERNO
 FONDECYT FONDEF MILENIO
 OTROS

Mostrar Exportar

Figura 10. Interfaz de estadísticas de proyectos del sistema U-Proyectos (usuario root)

En la parte superior de la vista (véase en Fig. 9 y 10) se puede escoger la forma en que serán agrupados los datos. Abajo de ésta opción, se presenta un listado de filtros para obtener información asociado a dichos parámetros, tales como académicos participantes/anfitriones, fecha, tipo de proyecto. Una vez elegida la opción, al hacer click en 'Mostrar' se desplegará una tabla y un gráfico de barras con los datos que contienen las estadísticas pertinentes (Fig. 11).

Proyectos

Ha iniciado sesión como Joaquín Ignacio Pérez Araya
Cerrar sesión

Proyectos

Estadísticas Visitas

Tipos de Visita

Visitas por Año

Visitas por Anfitrión

Estadísticas de Tipos de Visita

Académicos Anfitriones
Deje en blanco para seleccionarlos a todos

Intervalo de fechas
Comenzando el Hasta

Tipos de Visitas

TODOS
 Investigaci??n Postdoctorado Comisi??n Magister
 Comisi??n Doctorado Charla Curso
 Otro

Mostrar

Figura 11. Interfaz de estadísticas de visitas del sistema U-Proyectos (usuario root)

2.2.8 Registro de operaciones

La aplicación presenta una vista del registro de operaciones, denominado *Log*. Cada operación realizada de escritura/edición por cada usuario de la plataforma se registra, y se muestra aquí. La única opción disponible en este menú es la de exportar dicho registro en formato Excel (.xlsx). Sólo el usuario Director tiene acceso a esta vista (Fig. 12).

Usuario	Fecha	Operación	Entidad	Características	Atributo	Valor Anterior	Valor Actual
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Número	-	123456789
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Título	-	Test Project
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Título Español	-	Proyecto de Prueba
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Estado	-	Previsto
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Tipo	-	ANILLO
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Descripción	-	Comentario
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Fecha de Inicio	-	01/01/2023
Joaquín Ignacio Pérez Araya	10/01/2023	Agregar	Proyecto	ANILLO N° 123456789	Fecha Final	-	31/01/2023

Figura 12. Interfaz del informe financiero del sistema U-Proyectos (usuario root)

2.3 Arquitectura del sistema

Para ilustrar la arquitectura del sistema se utiliza el modelo de arquitectura C4 [6]. Además, se usa la herramienta PlantUML [5], junto a la extensión C4-PlantUML [7], para poder tener en código (así como también en un diagrama visual), de los distintos niveles de la arquitectura del sistema. La principal razón por la que se usa PlantUML es que es fácil de extender y re-escribir considerando futuros desarrollos, en comparación de realizar un diagrama nuevo cada vez que se modifica la arquitectura.

El código utilizado para generar los diagramas de las Figuras 13, 14 y 15 estará disponible dentro del código final asociado a este proyecto. A continuación se describen los modelos de contexto, contenedores y componentes del sistema.

2.3.1 Modelo de Contexto del Sistema

La plataforma interactúa con cuatro tipos de usuarios distintos (ver Figura 13): *Secretaría de Investigación*, *Administración*, *Académicos* y *Dirección*. Cada uno de

estos usuarios cumple un rol único con respecto a la gestión y ejecución de los Proyectos del Departamento.

La función de la *Secretaría de Investigación* es registrar los nuevos proyectos y los nuevos fondos que son asignados a estos, por lo que su interacción con la plataforma consiste en registrar proyectos nuevos con sus fondos/depósitos respectivos y obtener reportes. Es el único tipo usuario que puede ingresar nueva información al sistema.

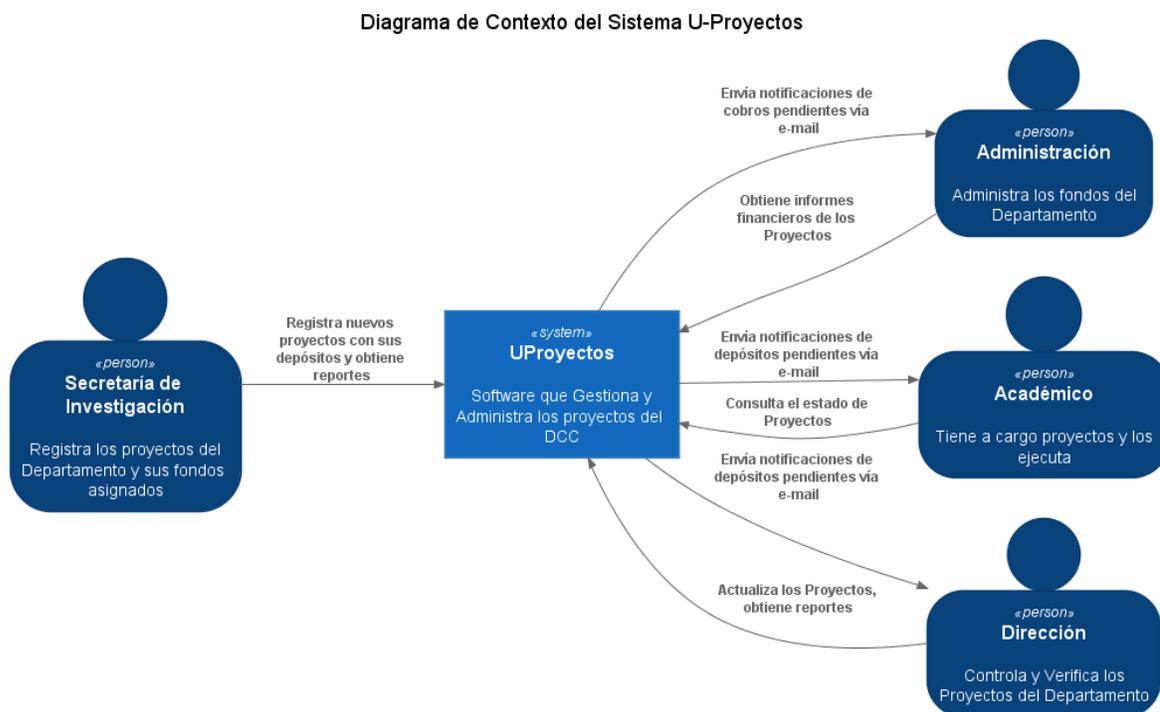


Figura 13: Modelo de contexto de la plataforma

La función de *Administración* es gestionar los recursos que son asignados al Departamento. Así, las interacciones asociadas a la plataforma consisten en obtener informes financieros y recibir notificaciones (vía correo electrónico) de los cobros pendientes.

La función de *Dirección* es controlar y verificar los proyectos del Departamento, por lo que su uso de la plataforma consiste en actualizar los proyectos activos, obtener reportes de éstos para su evaluación y recibir notificaciones (vía correo electrónico) de los depósitos pendientes.

Finalmente, la función de los *Académicos* consiste en estar a cargo de los proyectos, y que éstos se ejecuten. Por lo tanto, su interacción con la plataforma actualmente es principalmente para consultar el estado de los proyectos asociados a su persona, y obtener notificaciones de los depósitos pendientes.

2.3.2 Modelo de Contenedores

El sistema se divide en tres contenedores internos: el *servidor web*, la *aplicación* y el *esquema SQL*. El servidor web opera bajo Apache2, y es quien maneja las interacciones de los usuarios con la plataforma, dado que es éste quien se encarga de renderizar la interfaz web y manejar las peticiones de los usuarios. La aplicación, que utiliza PHP, HTML y JavaScript, es quien maneja toda la lógica de negocios del sistema y servicios de notificación vía e-mail a los usuarios (Fig. 14). Finalmente, el esquema SQL de la base de datos, se encarga de almacenar y mantener los datos que utiliza la plataforma.

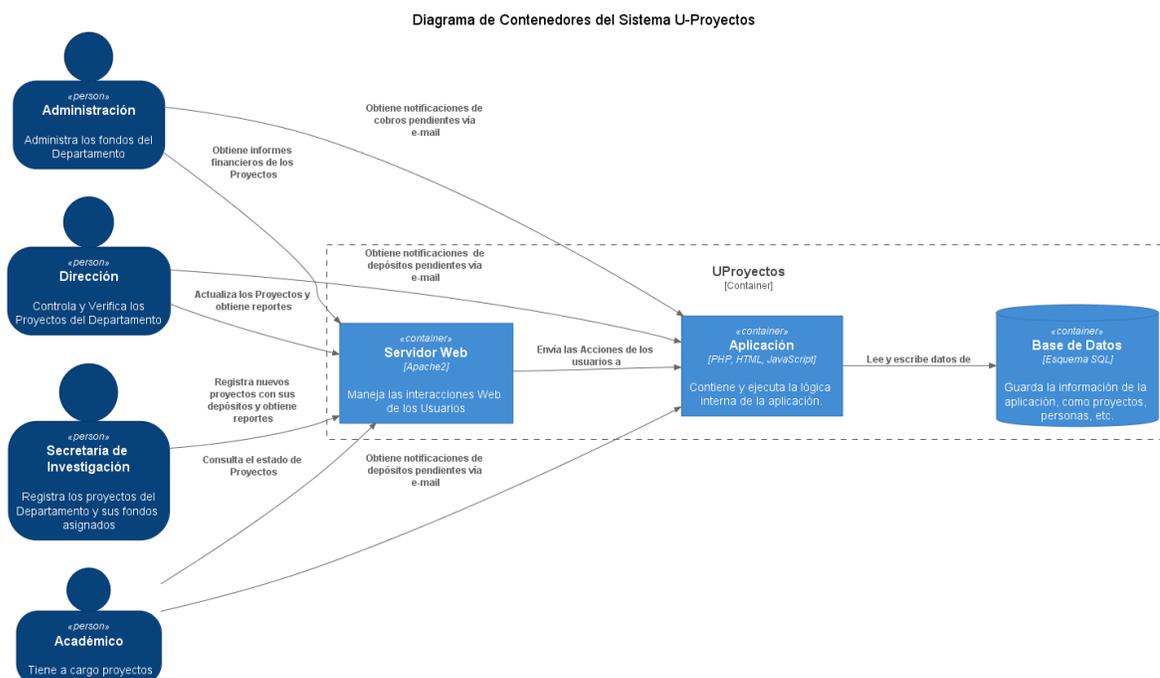


Figura 14. Modelo de contenedores de la aplicación

2.3.3 Modelo de Componentes

A nivel de componentes, el sistema implementa una arquitectura modelo-vista-controlador (Figura 15). Esto se puede observar tanto en la forma en que operan los distintos componentes de la aplicación, como su distribución a lo largo del código fuente.

El componente de modelos, contiene el código que permite la comunicación entre la aplicación y la base de datos, tanto para obtener información como para actualizarla. El componente de vistas, contiene todas las páginas que se muestran de la aplicación. El componente de controlador permite ejecutar la lógica de negocios de la aplicación.

Adicionalmente, la aplicación utiliza dos librerías externas para implementar alguna de sus funcionalidades: 1) PHP-Excel, que permite el manejo y creación de archivos Excel para poder descargar los distintos reportes dentro de la plataforma, y 2) PHP-Mailer que permite el envío de correos electrónicos para notificar de los depósitos de dinero asociados a proyectos.

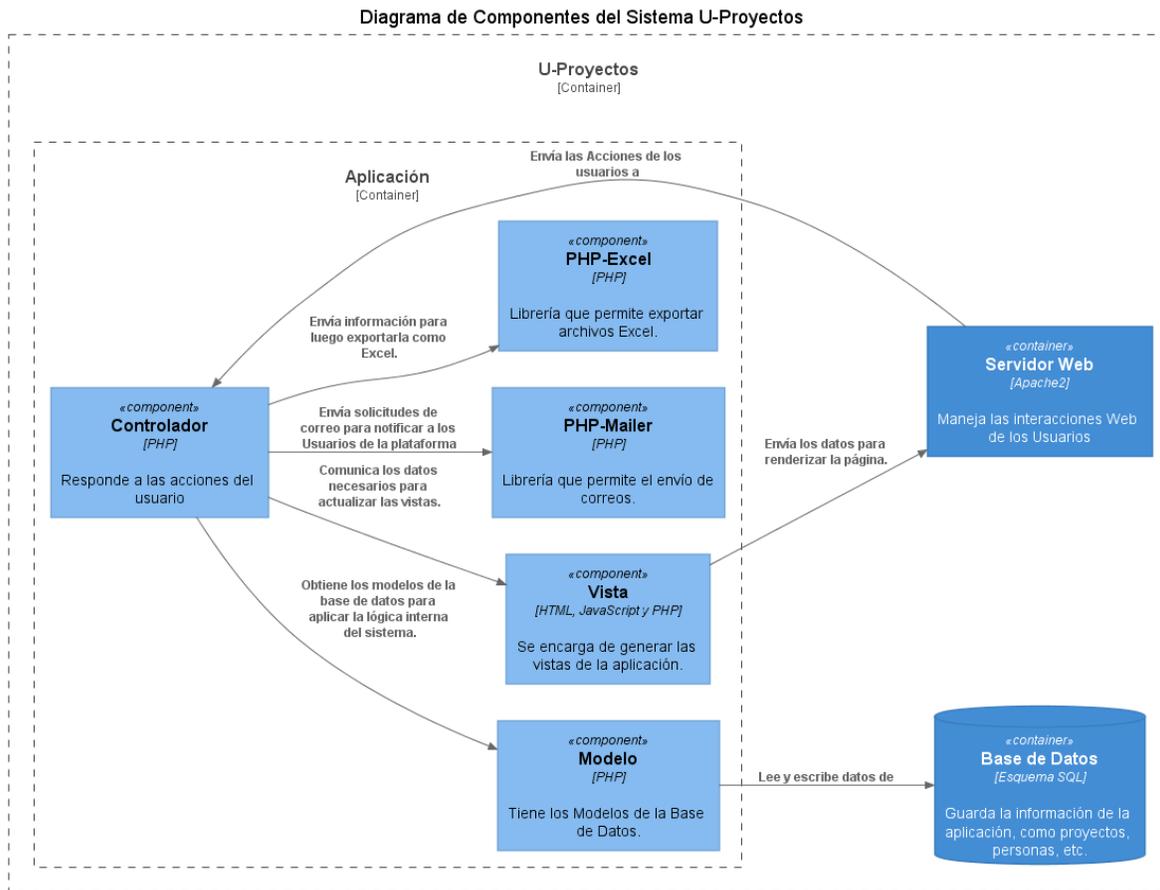


Figura 15. Modelo de componentes de la aplicación principal

2.3.4 Modelo de Datos

El modelo de datos contiene 21 entidades (Figura 16), entre las que se destacan las siguientes: *tpr_proyecto*, *tpr_persona* y *tpr_organización*. La tabla *tpr_proyecto*, como su nombre lo sugiere, contiene toda la información asociada a un proyecto, tiene 15 campos y 8 relaciones en total. La tabla *tpr_persona* contiene la información asociada a un académico o persona visitante del departamento.

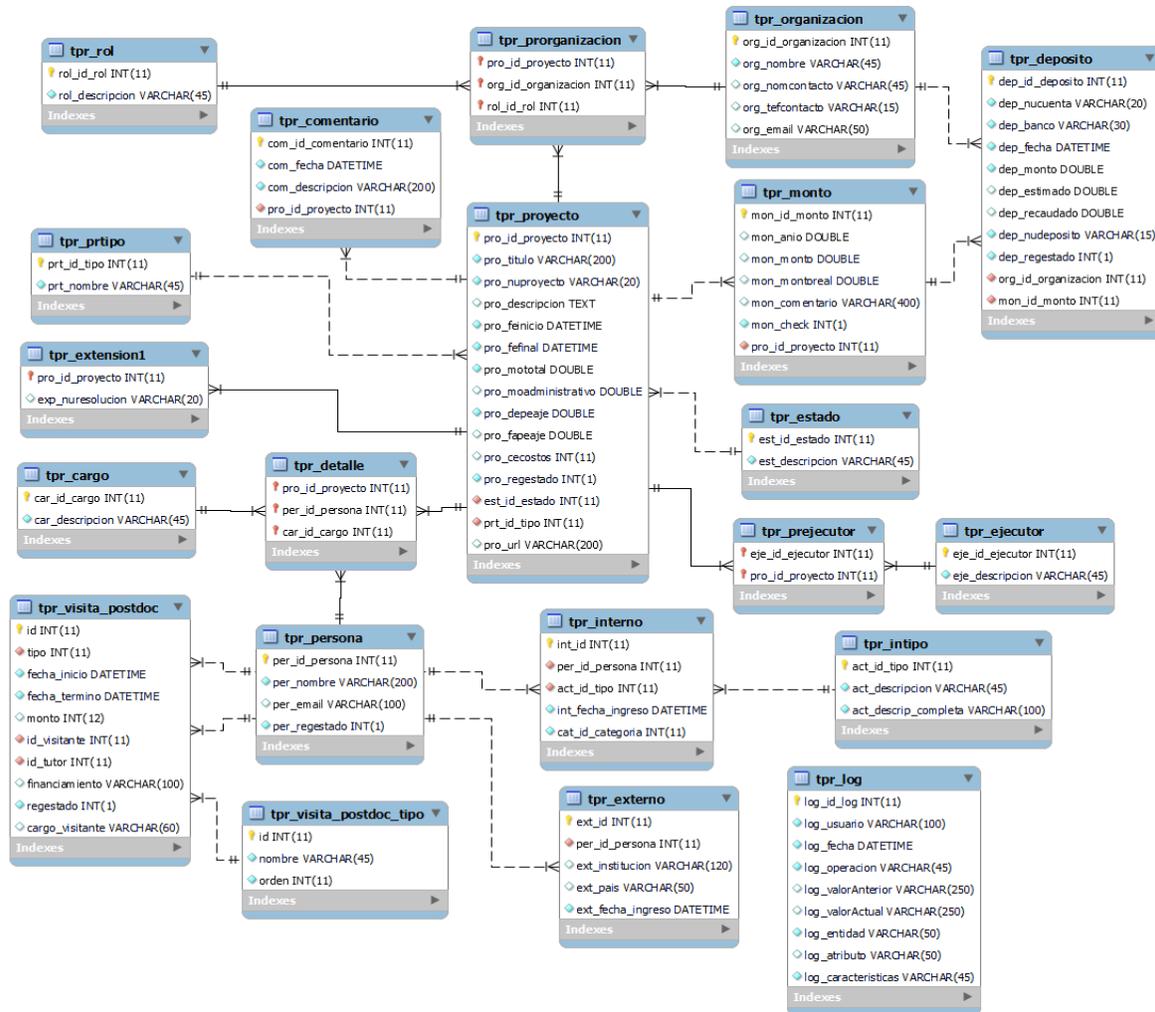


Figura 16. Modelo de datos de la plataforma

Los proyectos tienen un único estado de seis posibles, denotados por su *id_estado* y la tabla *tpr_estado*: Previsto, En Ejecución, Suspendido, Cancelado, Terminado y Activo. Por otra parte, un proyecto en particular puede tener uno o más comentarios, estos están almacenados en *tpr_comentario*, donde se guarda el contenido del comentario, la fecha donde se creó el comentario y el proyecto asociado.

Los proyectos tienen personas a cargo de estos. Esta relación se guarda en *tpr_detalle*, con el *id_proyecto* y el *id_persona* seguido de un *id* de cargo que puede ser: Director, Subdirector, Director Alterno, Coinvestigador, Responsable, Participante y Investigador Principal (estos nombres se guardan en la tabla *tpr_cargo*).

La tabla *tpr_organizaciones* guarda las distintas organizaciones que pueden estar involucradas en un proyecto. Un proyecto puede tener una o más organizaciones asociadas a éste. Estas asociaciones se guardan en *tpr_prorganizacion*, cada organización tiene un rol en particular en cada proyecto, este rol está asociado a un id de la tabla *tpr_rol*, que tiene valores tipificados tales como: Financiador, Cliente, Socio, y Beneficiario.

Cada organización puede realizar uno o más depósitos a un proyecto, con números de cuenta, bancos, estado y la cantidad de dinero asociada al depósito. Esta información se guarda en la tabla *tpr_deposito*.

2.4. Nivel de cumplimiento de los lineamientos de desarrollo del DCC

A continuación, la Tabla 2 indica el nivel de cumplimiento que tenía el sistema U-Proyectos antes de la realización de esta memoria, respecto a los lineamientos de gobernabilidad definidos para los sistemas del DCC (descritos en detalle en el Anexo I).

Tabla 2. Nivel de cumplimiento inicial del sistema U-Proyectos

Ítem Evaluado	Nivel de Cumplimiento
La arquitectura del ecosistema debe estar documentada y actualizada.	<i>No cumple.</i> No existe documentación asociada.
La arquitectura de software debe estar documentada y actualizada.	<i>Cumple parcialmente.</i> Existen diagramas parciales en el código legado de la aplicación.
El modelo de datos debe estar documentado y actualizado.	<i>Cumple parcialmente.</i> Existe documentación parcial que no está actualizada. [1]
El sistema debe contar con una API, la cual debe estar documentada y actualizada para acceder a los datos.	<i>No cumple;</i> no existe API asociada.
La autenticación de usuarios debe hacerse contra U-Pasaporte.	<i>No cumple.</i> El Login se realiza por medio de la intranet del DCC, sin

	embargo, es esta misma quien da los permisos de acceso a las vistas del software.
El código fuente debe estar apropiadamente versionado y respaldado en un repositorio.	<i>Sí cumple</i> , la aplicación posee repositorio en Github.
El stack tecnológico utilizado debe estar apropiadamente indicado.	<i>Cumple parcialmente</i> . Sólo se conoce la información del software del código legado, se desconoce qué stack tecnológico se utiliza para el ambiente de producción.
El stack tecnológico debe estar alineado con las definiciones del DCC.	<i>No cumple</i> . El stack tecnológico no forma parte del stack estándar del Área de Desarrollo.
El sistema debe contar con las indicaciones pertinentes para que un tercero pueda levantarlo en un ambiente de desarrollo.	<i>No cumple</i> . No existen instrucciones para levantar un ambiente ni código asociado para hacerlo.

2.5. Desafíos abordados para extender el U-Proyectos

Los principales desafíos a los que se vió enfrentado el desarrollador al momento de intervenir la plataforma, fueron los siguientes: 1) la falta de indicaciones para el levantamiento de un ambiente de desarrollo, 2) la falta de documentación sobre el diseño del sistema, su modelo de datos, arquitectura de sistema y su ecosistema, 3) el sistema tenía un stack tecnológico que nunca antes había sido trabajado por el desarrollador, particularmente PHP.

2.6. Extensiones al sistema U-Proyectos

La extensión realizada en el marco de este trabajo de memoria consiste en tres mejoras que van dirigidas, tanto a aumentar la seguridad de la aplicación, como a convertir a U-Proyectos en la fuente de verdad de la información que entrega. Este conjunto de

mejoras es altamente necesario, dado que la plataforma maneja datos sensibles del departamento (datos bancarios, fechas de depósitos, etc.), y esta información necesita ser obtenida de una única fuente de verdad. Por lo tanto, las mejoras realizadas son las siguientes:

- *Incorporación de credenciales Cuenta UChile [8] como mecanismo de autenticación de los usuarios.* Esto se hace mediante integración con el Portal DCC [9], con la finalidad de que el sistema cumpla los requisitos mínimos de gobernabilidad y reforzar la seguridad de la plataforma. Para ello se debió localizar el mecanismo antiguo de autenticación, y reemplazarlo con el provisto por el Portal DCC.
- *Actualización de la versión de PHP del sistema.* Particularmente, el sistema pasó de usar la versión 5.5, a la última versión estable actualmente, correspondiente a la versión 8. PHP 5.5 dejó de tener soporte el 20 Junio del 2016 [3]. Esto había dejado a la aplicación vulnerable a ataques maliciosos. [10]
- *Creación de una API para U-Proyectos de tamaño mínimo, pero de fácil extensión para futuros desarrollos.* Dado que U-Proyectos contiene toda la información de los proyectos del departamento, el contar con una API permite, por ejemplo, que la página Web del DCC consuma esa información de manera automática para poder publicarla.

3. Ajustes al sistema U-Proyectos

Es importante destacar que no se requirieron ajustes al proceso de registro y seguimiento de proyectos ya soportado por el sistema. Sin embargo, sí se realizaron modificaciones a la funcionalidad del sistema. Los ajustes a la plataforma para restaurar y mantener su operatividad fueron diversos, y principalmente se enfocaron en facilitar la creación de un ambiente de desarrollo, actualizar el código de la plataforma y restaurar funcionalidades de la aplicación que fueron mermadas por la actualización del código. A continuación se explica esto más en detalle.

3.1. Ajustes al modelo de datos

Para recuperar la operatividad de la plataforma, se tuvo que añadir tablas y/o columnas nuevas a la base de datos, dado que existían discrepancias entre las tablas del modelo recibido al inicio del proyecto, y las tablas de las consultas del código legado. A continuación se indican las tablas y columnas añadidas al modelo de datos del sistema legado, además del nuevo modelo de datos.

Tablas añadidas:

- *tpr_cargo_prtipo* es el cross join entre *tpr_cargo* y *tpr_prtipo*. Éste contiene todas las combinaciones entre cargos y tipos de proyectos. Esta tabla permite utilizar la funcionalidad de Cargos de la plataforma.
- *tpr_usuarios* es una tabla nueva que contiene los usuarios de la plataforma. Esta tabla se añade a causa del cambio del sistema de autenticación del sistema; el detalle de este cambio está en el capítulo 4.3. La tabla contiene el rut, nombre, correo electrónico y rol del usuario registrado (id interno).

Columnas añadidas:

- Rut en *tpr_persona* (*per_rut*) es la columna que indica el rut de la persona descrita en la base de datos.
- En *tpr_cargo*, se añadió la columna (*car_descripcion_en*) que representa la descripción en inglés del cargo y la columna (*car_prioridad*) indica la prioridad de dicho cargo.
- El grado de confidencialidad (*pro_confidencial*) y el título en español (*pro_titulo_es*) en *tpr_proyectos*. El primero se utiliza para saber si el proyecto es confidencial (1) o no (0), el segundo es el título del proyecto en español.

Para añadir dichos campos extras al modelo de datos se creó un script en SQL. Éste permite hacer dicho cambio de forma automática en la base de datos de producción de U-Proyectos (Figura 17).

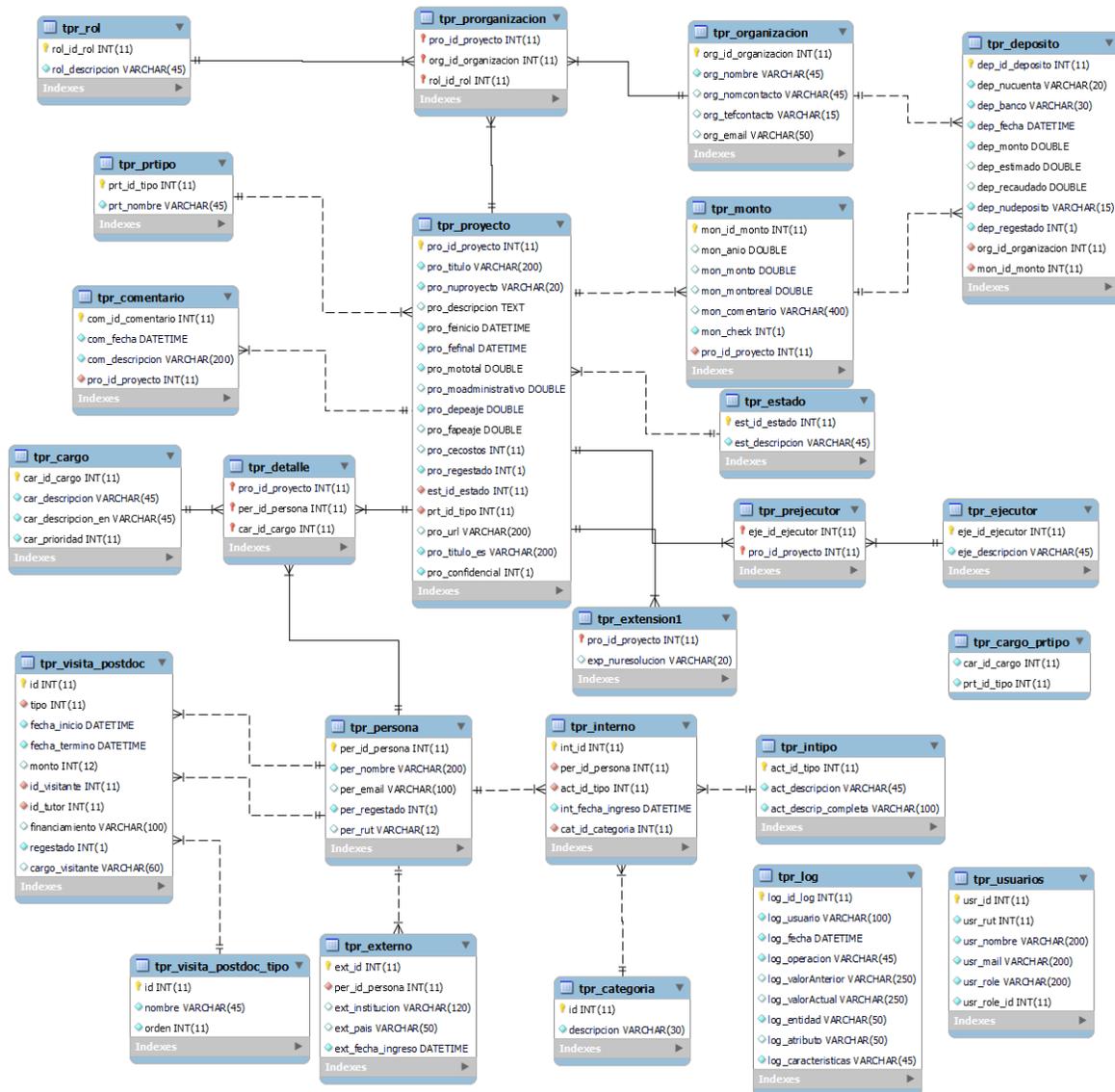


Figura 17. Nuevo modelo de datos de U-Proyectos

3.2. Ajustes al software

El proyecto fue completamente reestructurado, y para ello se crearon cuatro contenedores en Docker: 1) **web** contiene y ejecuta el código legado de la plataforma en PHP utilizando un webserver Apache, 2) **db** donde se ejecuta la base de datos de MariaDB, 3) **phpmyadmin** una herramienta del mismo nombre que permite el manejo fácil de la base de datos por medio de una interfaz web, y 4) **api** que alberga y sirve la

API creada con FastApi de Python3, cuya funcionalidad se describe en el siguiente capítulo. Los contenedores *web* y *db* contienen parte del código legado de la aplicación.

La creación de los contenedores 1) y 2) fue un proceso de ensayo y error, en el cual se realizaron ajustes de algún parámetro de construcción de las imágenes de los contenedores, y después se evaluó el impacto de dicho cambio. Este proceso se realizó para:

- Encontrar una imagen del contenedor *web* que facilitara el proceso de levantamiento del servidor en conjunto con el código, es decir que fuera un contenedor único para simplificar y delimitar su existencia.
- Conectar la base de datos *db* con la aplicación web.
- Instalar las dependencias necesarias para lograr la descarga de archivos .xlsx y .json dentro de la plataforma.

El contenedor *phpmyadmin* se creó con el objetivo de tener una forma fácil y ordenada de acceder a la base de datos sin conectarse directamente a la consola del contenedor *db*, facilitando así la visualización del modelo de datos, la visualización de estos mismos y verificar las consultas realizadas por la aplicación web (Fig. 18).

Diagrama de Contenedores del Sistema UProyectos

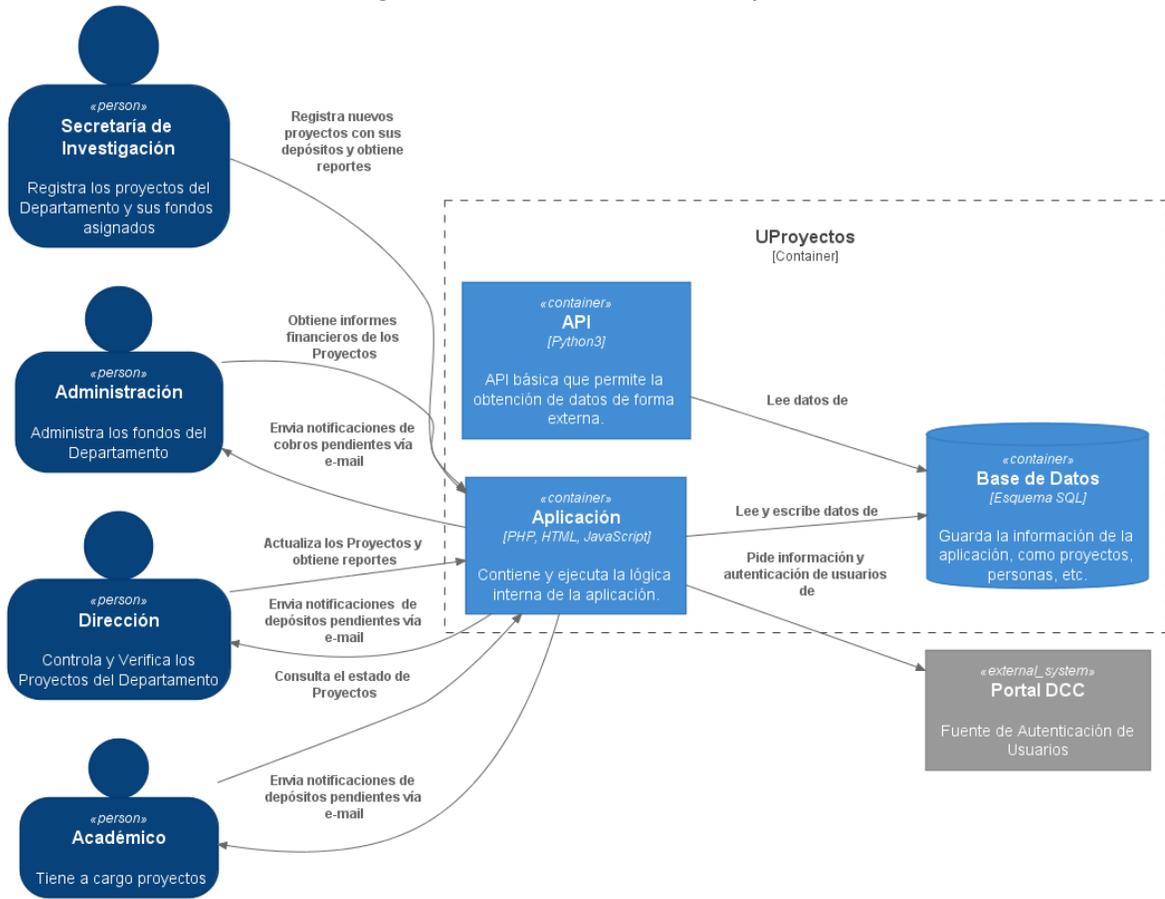


Figura 18. Modelo de contenedores actualizado con la nueva versión.

Finalmente, el contenedor *api* se creó para albergar la API nueva a desarrollar como parte de la extensión de este trabajo de memoria. La reestructuración realizada a la plataforma facilita su levantamiento y acceso, permitiendo crear un ambiente tanto productivo como de desarrollo.

A continuación, se muestra un modelo de componentes (Figura 19) actualizado considerando el cambio anteriormente descrito. Notar que como la API desarrollada es nueva, actualmente no está conectada a ningún sistema, salvo a la base de datos de U-Proyectos para poder obtener los datos. A futuro se espera que esta API sea utilizada por otros sistemas para obtener información respecto a los proyectos (y datos asociados) que están en U-Proyectos.

Diagrama de Componentes del Sistema U-Proyectos

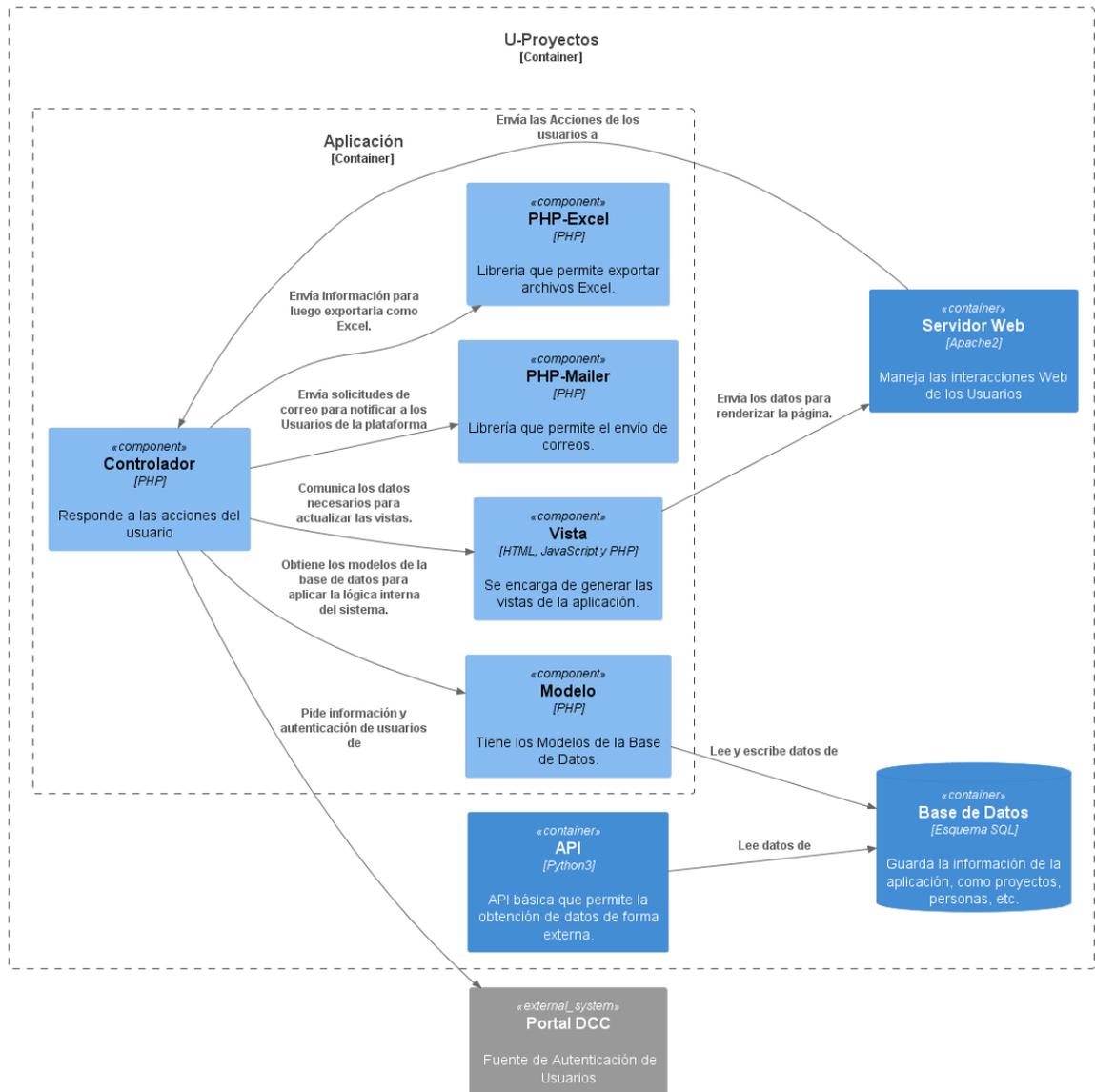


Figura 19. Modelo de componentes actualizado con la nueva versión.

Por parte de la documentación interna para la creación de un ambiente de desarrollo y/o productivo, se crearon y añadieron variables de ambiente para el ingreso con la base de datos dentro de los distintos contenedores de la plataforma. Dichas variables se describen en el archivo README.md, que se encuentra en el repositorio del código.

4. Extensión al sistema U-Proyectos

El sistema se extendió en distintos ámbitos, por ejemplo se actualizó la versión de PHP de la versión 5.5 a la versión 8.1, se añadió la autenticación de usuarios a través de U-Pasaporte, y se creó una API de tamaño mínimo pero con todas las herramientas necesarias para ser extendida a futuro. A continuación se explican cada una de estas extensiones.

4.1 Actualización de PHP

Tal como se mencionó anteriormente, el código de U-Proyectos no había sido mantenido en un periodo extenso de tiempo (desde el 2016), por lo que era de vital importancia actualizar dicho código para mantener su funcionalidad.

En términos operativos, el cambio de PHP 5 a PHP 8.1*, se realiza mediante el uso de la versión más nueva de la imagen de apache que utiliza el contenedor. Sin embargo, esto termina generando problemas dentro de la aplicación debido a que, al ser un cambio de versión abrupto, múltiples funcionalidades quedaron deprecadas y/o no operativas. Para el caso de la aplicación, existían errores de código que causaban que las vistas no funcionaran en lo absoluto, botones que no ejecutaban las consultas a la base de datos, las descargas de archivos no funcionaban, y también habían problemas para que la aplicación se hiciera conexión con la base de datos.

Para solucionar dichos problemas, se realizaron múltiples cambios de código fuente de la plataforma:

- Cambiar la sintaxis de accesos de arreglos de `{}` a `[]`, es decir, cambiar de `arreglo{14}` a `arreglo[14]`, a lo largo de todo el código.
- Añadir y/o cambiar las fechas mínima (*01/01/1970*) y máxima (*31/12/2066*) por defecto a las consultas a la base de datos, para así restaurar las funcionalidades de búsqueda de la aplicación.
- Reescribir parte del código que genera la descarga de archivos en formato excel a partir de datos de la plataforma, y añadir instrucciones dentro del archivo del contenedor de la aplicación web para permitir dicha descarga.
- Reescribir y unificar el código que se utiliza para la conexión con la base de datos, añadiendo también variables de ambiente en el proceso, para que sea de fácil configuración. Esto se hizo por medio un archivo `.env`, que se usa para el levantamiento de cualquier tipo de ambiente.

- Verificar valores nulos de variables durante diferentes operaciones dentro de los módulos del Controlador y del Modelo, dado el cambio en el manejo de dichas variables de forma interna por parte de PHP 8.1, provocando errores y pérdidas de funcionalidad.

4.2 Tablero del producto

A continuación se muestra el tablero que describe el alcance de la nueva versión del producto (objetivo y alcance), es decir, los objetivos, los requisitos funcionales y los no funcionales del sistema, luego de haber intervenido y expandido el sistema U-Proyectos (Figura 20). A continuación se explica cada sección del tablero.

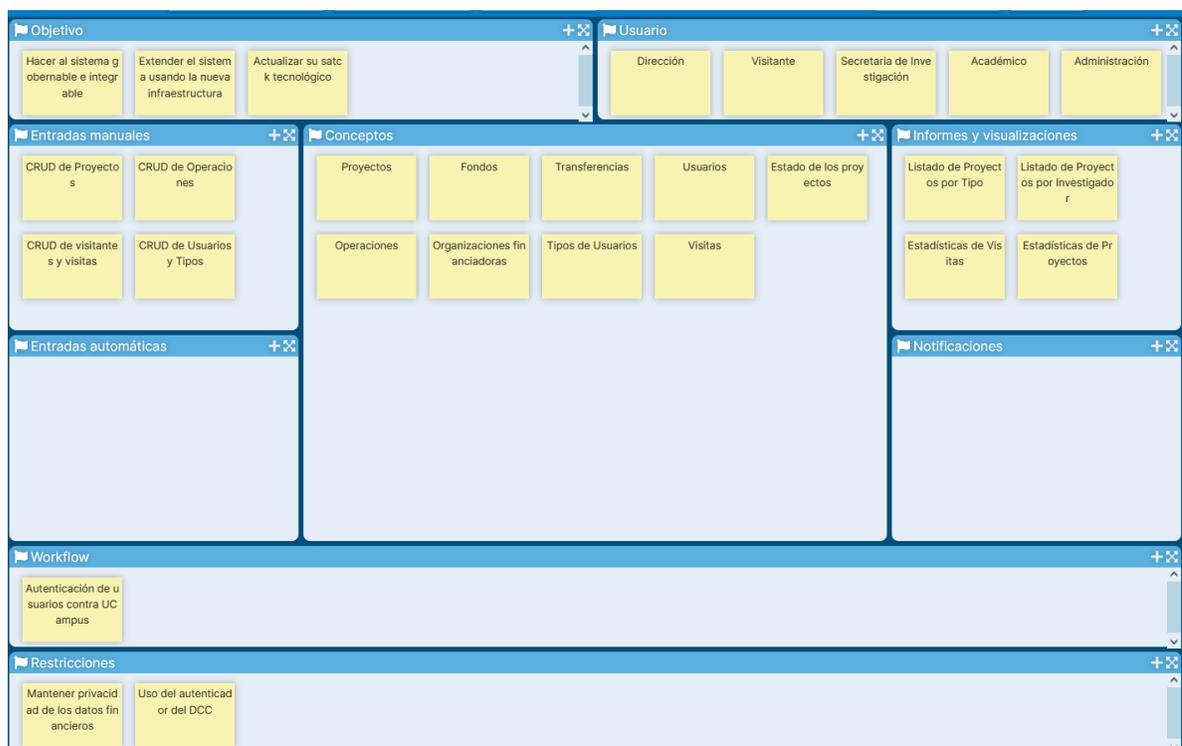


Figura 20. Tablero de producto del nuevo sistema

4.2.1 Objetivo

Los principales objetivos de la plataforma actualmente son: 1) hacer al sistema más gobernable e integrable a otros sistemas del DCC, 2) extender el sistema utilizando la nueva infraestructura, y 3) actualizar el stack tecnológico del código legado.

4.2.2 Usuarios

Los usuarios de la plataforma corresponden a los actores del tablero anterior, realizado durante el análisis de la plataforma; esto es: Investigadores/Académicos, Secretaría de Investigación, Administración y Visitantes.

4.2.3 Conceptos

Los conceptos corresponden a las entidades básicas claves que forman parte del sistema, en el caso de U-Proyectos: Proyectos, Fondos, Transferencias, Usuarios, Tipo de Usuarios, Estados de los proyectos, Operaciones, Organizaciones Financiadoras y Visitas.

4.2.4 Entradas Manuales

Las entradas manuales de la plataforma son: Proyectos, Operaciones, Visitas y Visitantes y Usuarios y sus tipos. Cada de estas entradas debe permitir un CRUD (Create, Read, Update, Delete) completo. Estas son introducidas por la secretaría de investigación.

4.2.5 Entradas Automáticas

U-Proyectos no cuenta con entradas automáticas, toda la información que entra al sistema es la información que introduce manualmente la Secretaría de Investigación.

4.2.6 Informes y visualizaciones

Los informes y estadísticas que presenta la aplicación son: Listado de Proyectos por Tipo, Listado de Proyectos por Investigador, Estadísticas de Proyectos y Estadísticas de Visitas.

4.2.7 Notificaciones

La aplicación de momento no posee Notificaciones activas de momento.

4.2.8 Workflow

Antes de que cada usuario pueda utilizar la plataforma, se debe autenticar con el Portal del DCC que utiliza UCampus.

4.2.9 Restricciones

Las restricciones de la plataforma van en torno a su seguridad y uso de información confidencial. De esta forma, la plataforma debe mantener la privacidad de sus datos financieros y debe usar la autenticación del DCC.

4.3 Inclusión del mecanismo de autenticación de usuarios con U-Pasaporte

Para incluir la Autenticación de Usuarios con U-Pasaporte, en primera instancia se evaluó qué tipo de autenticación existía. Eso se hizo con el objetivo de evaluar el nivel de extensión necesaria para ser implementada.

Antes de realizar este trabajo de memoria, la autenticación de usuarios se hacía por medio de la Intranet del DCC. Particularmente, U-Proyectos utilizaba datos provistos por ésta (por medio de la sesión de PHP), para dar permisos y acceso pertinente a los distintos tipos de usuarios de la plataforma. Por lo tanto, todas las vistas estaban configuradas para verificar qué tipo de usuario podía acceder a éstas y visitarlas.

Para integrar la nueva autenticación de usuarios, se configuró una redirección automática en caso de no estar autenticado en la plataforma, es decir, si no hay datos de sesión. La página de redirección es simple, utiliza una vista similar a las de las otras vistas de U-Proyectos, y sólo permite entrar a la página siendo redirigido al portal del DCC para ser autenticado (Figura 21).

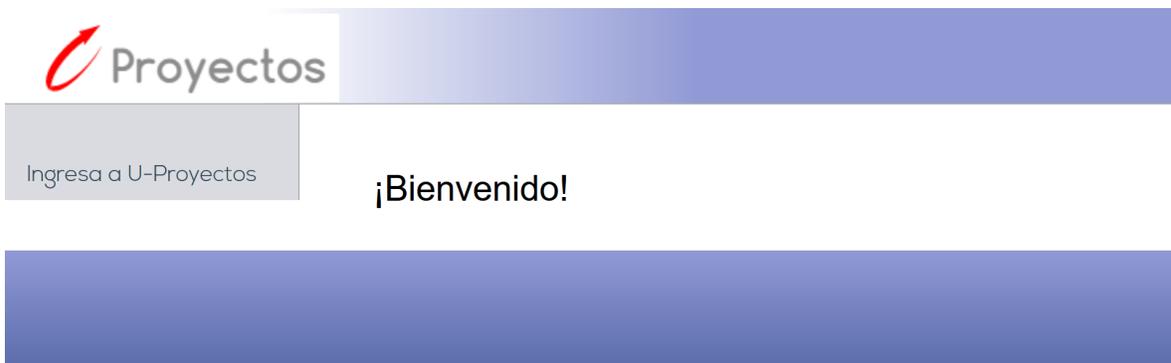


Figura 21. Nueva vista de acceso a U-Proyectos.

Una vez autenticado, el propio portal redirige al usuario de vuelta al sitio, con datos de sesión, que son utilizados para obtener el usuario de la tabla nueva *tpr_usuarios* para obtener el rol del usuario que acaba de ser autenticado. Si no existe el usuario en la base de datos, se le muestra la vista limitada de usuario público del sitio.

4.4 Descripción de la API de U-Proyectos

La API de U-Proyectos es un desarrollo completamente nuevo dado que no existía ningún elemento en el código legado que podría reutilizarse para este propósito, pues dentro de la aplicación web se realizan peticiones directas a la base de datos, según cada componente. El contar con una API permite contar con amplia libertad de diseño y de uso de tecnologías al momento de crear sistemas o servicios que interactúen con U-Proyectos.

Para implementar la API se escoge Python3, con el framework FastAPI [10], principalmente por su facilidad para crear nuevas rutas de acceso (endpoints) y la documentación de dichas rutas. Adicionalmente se elige usar en conjunto la librería SQLAlchemy [11], la cual está diseñada para ser utilizada con FastAPI para manejar las conexiones y los modelos de la base de datos. Aunque se podría utilizar directamente

una conexión a la base de datos por medio de queries, SQLAlchemy permite crear los modelos como clases, para así evitar completamente el uso del lenguaje SQL para realizar consultas.

Una de las principales razones por las cuales se escogió FastAPI, es porque ésta genera automáticamente la documentación de sus rutas de acceso, y que la creación de éstas es muy sencilla. Ambos factores satisfacen los Requisitos de Gobernabilidad e Integración (véase Anexo I) sobre la API del sistema.

La API actualmente implementa tres métodos GET, que obtienen desde la base de datos las siguientes entidades: *tpr_proyecto*, *tpr_persona* y *tpr_organizacion*. Cada método requiere de la *id* de una entidad en particular para retornar la información completa sobre ésta en la base de datos. Sólo se implementaron estos métodos de obtención de información, dado que endpoints de escritura necesitan saber las reglas de negocio de la Aplicación en general, que son manejadas internamente por U-Proyectos en su sistema principal.

La ruta para obtener datos de *tpr_proyecto*, es */proyecto*. Requiere un id, que corresponde a la id de la base de datos de la persona que se quiere obtener. Retorna la columna correspondiente a dicho id de la base de datos.

La ruta para obtener datos de *tpr_persona*, es */persona*. Este endpoint es similar al anterior; es decir, requiere un id, que corresponde a la id de la base de datos del proyecto que se quiere obtener, y retorna la columna correspondiente.

La ruta para obtener datos de *tpr_organizacion*, es */organizacion*. Este endpoint es similar a los anteriores.

A pesar de su simplicidad, esta API tiene implementada todo el modelo de datos, tanto entidades como relaciones, que pueden ser utilizados en el futuro para realizar operaciones más complejas sobre el espacio de datos manejado.

5. Evaluación del trabajo realizado

Para evaluar el incremento de gobernabilidad y la capacidad de integración de la plataforma U-Proyectos, se utilizaron las extensiones indicadas en la sección anterior como experiencias piloto. Para ello se comparó la complejidad y el esfuerzo requerido para realizar dichas extensiones en ambas versiones del sistema; es decir, en el sistema legado, y en el nuevo. Los resultados se mostrarán en una tabla comparativa, que pondrá en evidencia si el trabajo de mejora realizado sobre la plataforma (previo a

la extensión de la misma), efectivamente contribuyó a mejorar la gobernabilidad del sistema.

5.1 Proceso de evaluación seguido

El mejoramiento de la gobernabilidad y la capacidad de integración del sistema U-Proyectos se evaluó definiendo una lista de 10 tareas críticas, las cuales permiten comparar el esfuerzo necesario para llevarlas a cabo en el sistema legado y en el nuevo sistema. Con esto se espera demostrar que el nuevo sistema requiere de menos esfuerzo, y tiene menos complejidad para ser intervenido que el sistema legado.

Los resultados de estas pruebas se presentan utilizando una tabla comparativa. Las tareas que se midieron fueron las siguientes:

1. Levantar el servidor web de U-Proyectos.
2. Levantar la base de datos de U-Proyectos.
3. Realizar deployment en un ambiente productivo.
4. Obtener información del modelo de datos.
5. Obtener información sobre la arquitectura de la aplicación.
6. Obtener información sobre los usuarios de la aplicación.
7. Actualizar o cambiar el software de la aplicación.
8. Integrar una aplicación que requiera datos básicos de la aplicación.
9. Agregar una vista nueva a la aplicación.
10. Agregar un rol nuevo a la aplicación.

Las tareas 1-3 están enfocadas a la ejecución y puesta en producción del código de la aplicación, ya sea en un ambiente de desarrollo o bien en uno productivo.

Las tareas 4-6 están asociadas a conocer el contexto, el comportamiento básico y la arquitectura de la aplicación, con la finalidad de que nuevos desarrolladores puedan comprender el funcionamiento de la aplicación.

Las tareas 7-10 son operaciones de extensión y/o mantenimiento de la plataforma, además de mejorar la integración con otros sistemas.

5.2 Resultados de la evaluación

Para realizar la evaluación, se comparó el escenario anterior a esta memoria versus posterior a esta memoria, considerando las potenciales decisiones y caminos a tomar para afrontar cada caso. Este proceso ha sido realizado por el autor de este trabajo de memoria.

Para determinar el esfuerzo de realización de cada tarea, se estimó el tiempo aproximado para completarlas y su complejidad técnica asociada. La Tabla 3 muestra este análisis comparativo para las 10 actividades antes mencionadas.

Tabla 3. Evaluación de tareas críticas inicial y actual U-Proyectos

Tarea Evaluada	Nivel Inicial	Nivel Actual
Levantar el servidor web de U-Proyectos	Se debe montar desde cero un servidor para correr el código de PHP. Tiempo: Medio Complejidad: Media-Alta	Se utiliza Docker para crear una imagen que corre la aplicación web. Tiempo: Bajo Complejidad: Baja
Levantar la base de datos de U-Proyectos.	Se debe montar desde cero un servidor para ejecutar la base de datos. Tiempo: Medio Complejidad: Baja	Se utiliza Docker para crear una imagen que corre una base de datos preconfigurada. Tiempo: Bajo Complejidad: Baja
Obtener información del modelo de datos	Existen diagramas del modelo de datos, pero están desactualizados. Tiempo: Bajo Complejidad: Baja	Existe un diagrama actualizado, junto con el diccionario de datos. Tiempo: Bajo Complejidad: Baja
Obtener información sobre la arquitectura de la aplicación.	Existe un único diagramas que muestra la estructura básica del sistema. Tiempo: Bajo Complejidad: Baja	Existen diagramas utilizando el modelo C4. Cada diagrama representa un nivel distinto de abstracción del sistema. Tiempo: Bajo Complejidad: Baja

<p>Obtener información sobre los usuarios de la aplicación.</p>	<p>Existe diagrama de casos de uso.</p> <p>Tiempo: Bajo Complejidad: Baja</p>	<p>En adición al diagrama de casos de uso, está el diagrama de contexto de la aplicación.</p> <p>Tiempo: Bajo Complejidad: Baja</p>
<p>Realizar deployment en un ambiente productivo.</p>	<p>No hay instrucciones para el deployment, por lo que se hace necesario examinar el ambiente anterior para realizar deployment.</p> <p>Tiempo: Medio Complejidad: Alta</p>	<p>Se utiliza Docker para crear una imagen que corre una base de datos preconfigurada.</p> <p>Tiempo: Baja Complejidad: Baja</p>
<p>Actualizar/cambiar el software de la aplicación.</p>	<p>Se debe cambiar según el sistema que lo está ejecutando.</p> <p>Tiempo: Medio Complejidad: Media</p>	<p>Se puede cambiar la imagen de Docker, independiente del sistema que lo está ejecutando.</p> <p>Tiempo: Bajo Complejidad: Media</p>
<p>Integrar una aplicación externa que requiera datos básicos de la aplicación</p>	<p>Hay que generar una API dentro del controlador para poder acceder a los datos.</p> <p>Tiempo: Medio Complejidad: Media</p>	<p>Se creó una API básica con FastAPI de Python, que puede ser extendida fácilmente para cumplir con los requisitos establecidos por el DCC.</p> <p>Tiempo: Baja Complejidad: Media-Baja</p>
<p>Agregar una nueva vista a la aplicación.</p>	<p>Requiere modificar el código para añadir la vista y el controlador asociado.</p>	<p>Sin cambios respecto a la versión anterior.</p>

	Tiempo: Medio Complejidad: Media	Tiempo: Medio Complejidad: Media
Agregar un rol nuevo a la aplicación.	Requiere modificar extensivamente la aplicación, para agregar en cada caso, el comportamiento del nuevo rol. Tiempo: Alto Complejidad: Alta	Sin cambios respecto a la versión anterior. Tiempo: Alto Complejidad: Alta

A continuación se analiza más en detalle la evaluación de cada uno de estos ítems.

5.2.1 Levantar el servidor web de U-Proyectos

Para realizar esta tarea en el sistema legado, la solución más simple para un potencial desarrollador sería la de ejecutar el proyecto utilizando el servidor de desarrollo de PHP [13], u otra aplicación como XAMPP o nginx dependiendo de su sistema operativo. Después de esto se deben instalar las dependencias correspondientes, para poder ejecutar el código y levantar dicho ambiente.

Para realizar esta tarea con el sistema actual, un potencial desarrollador debe utilizar las imágenes de Docker, que están dentro del repositorio del código, para automáticamente descargar y crear las imágenes de los contenedores que ejecutarán la aplicación. Esto se puede hacer independientemente del sistema operativo utilice el desarrollador. Para realizar dicha tarea se necesita tener conocimientos básicos de Docker.

5.2.2 Levantar la base de datos de U-Proyectos

Para realizar esta tarea en el sistema legado, un potencial desarrollador debe instalar en su ambiente de desarrollo un servidor de MariaDB/MySQL, o bien crear una imagen de Docker para tener un contenedor de dicha base de datos.

Para realizar esta misma tarea con el sistema actual, se debe utilizar la imagen disponible de Docker, que se utiliza dentro del repositorio para levantar dicha base de datos.

5.2.3 Obtener información del modelo de datos

En el sistema legado, para obtener información respecto al modelo de datos de la aplicación había que recurrir a la memoria de Carlos Dettoni [1]. Sin embargo, este diagrama está dividido en dos partes, dado que, durante la realización de dicha memoria, el modelo de datos fue modificado. Por lo tanto, para obtener un modelo de datos actualizado, hay que realizar un análisis al volcado de la base de datos productiva de la aplicación. Considerando el ajuste necesario del modelo de datos mostrado en el Capítulo 3.1, éste tampoco estaba completo.

Actualmente, dicho modelo de datos está disponible dentro del repositorio del código de la aplicación (véase figura 17). Adicionalmente, utilizando *phpmyadmin* se puede obtener el diccionario de datos completo y actualizado.

5.2.4 Obtener información sobre la arquitectura de la aplicación

En el código del sistema legado existe un único diagrama que muestra la arquitectura lógica de la aplicación. Sin embargo, éste está desactualizado con el funcionamiento actual.

En el sistema actual están los tres diagramas del modelo de C4: el diagrama de contexto, diagrama de contenedor y diagrama de componentes. Estos diagramas corresponden a las figuras 13, 18 y 19 del presente documento.

5.2.5 Obtener información sobre los usuarios de la aplicación

En el código fuente del sistema legado existe un diagrama que muestra los roles y las acciones que pueden realizar dichas entidades en el sistema. Actualmente dicho diagrama está traspasado a un diagrama de modelo de contexto de C4 (véase figura 13).

5.2.6 Realizar deployment en un ambiente productivo

En el sistema legado no existen instrucciones para levantar dicho ambiente, ni tampoco qué otros sistemas se utilizan para hacerlo. Por lo que el proceso consistiría en mirar cómo está configurado en producción la versión de U-Proyectos que se encuentra en la Intranet del DCC.

Para realizar deployment de la nueva versión del sistema en un ambiente de producción, se deben crear los contenedores de Docker, ajustar las variables de ambiente para dejar el código corriendo.

5.2.7 Actualizar/cambiar el software de la aplicación

Para actualizar o cambiar el software de la aplicación del sistema legado se debe cambiar en el ambiente directo donde está alojada. Por lo tanto, actualizar y cambiar dicho software recae en dónde está el código en primer lugar.

En el sistema actual, se puede modificar la imagen de Docker para actualizar sus componentes. Esta es una de las principales ventajas de Docker, pues dichos cambios no dependen del sistema operativo en que esté corriendo la aplicación.

5.2.8 Integrar una aplicación externa que requiera datos básicos de la aplicación

Para realizar esta tarea en el sistema anterior se debía optar por dos posibles caminos: 1) extender el controlador de la aplicación web para soportar llamadas de API, o 2) crear una API externa que sólo se comunique con la base de datos. Al ser creada una API externa durante el desarrollo de esta memoria, esta tarea se reduce a extender dicha API para implementar la integración necesaria.

5.2.9 Agregar una nueva vista a la aplicación.

Esta tarea es una tarea relativamente de mediana complejidad. Para realizarla se debe añadir tanto la vista nueva a nivel de frontend, como su controlador correspondiente en backend.

Dada la naturaleza de los cambios realizados durante esta memoria, la complejidad de esta tarea queda sin cambiar. Sin embargo, dado a que durante este trabajo de memoria se actualizó la versión del código de PHP, modificando el código para restaurar funcionalidad, esto no será impedimento para extender la aplicación en el futuro.

5.2.10 Agregar un rol nuevo a la aplicación.

La complejidad de esta tarea es elevada debido a que para agregar un rol nuevo a la plataforma, se debe añadir dicho en cada uno de los controladores según se necesite autorizar.

Al igual que para agregar una visita a la aplicación, la complejidad de esta tarea no cambia. Sin embargo, se facilita el desarrollo dentro de la aplicación web en el código actual dado a que no se tiene que actualizar PHP.

5.3 Nivel de cumplimiento de los lineamientos de gobernabilidad e integración con otros sistemas

En relación a los lineamientos de gobernabilidad e integración definidos para los sistemas del DCC, a continuación se indica el nivel de cumplimiento que tenía U-Proyectos antes de la realización de esta memoria (Nivel Inicial), y el cumplimiento actual (Nivel Actual). Como se puede apreciar en la Tabla 3, a partir del trabajo realizado se ha mejorado el cumplimiento de los requisitos de gobernabilidad e integración definidos por el DCC.

Tabla 4. Nivel de cumplimiento inicial y actual de U-Proyectos

Ítem Evaluado	Nivel Inicial	Nivel Actual
La arquitectura de software debe estar documentada y actualizada.	<i>Cumple parcialmente.</i> Existen diagramas parciales en el código legado de la aplicación.	<i>Cumple.</i> Se especificaron los diagramas de contexto, contenedores y componentes asociados (Figuras 13, 18 y 19).
El modelo de datos debe estar documentado y actualizado.	<i>Cumple parcialmente.</i> Existe documentación parcial que no está actualizada. [1]	<i>Cumple.</i> Se creó documentación respecto al modelo de datos (Figura 17).
Debe contar con una API, la cual debe estar documentada y actualizada para acceder a los datos.	<i>No cumple.</i> No existe API asociada.	<i>Cumple.</i> Se creó una API simple con FastAPI [11], que puede ser extendida para los obtener los datos que se requieran.
La autenticación debe realizarse contra U-Pasaporte.	<i>No cumple.</i> El Login se realiza por medio de la intranet del DCC, sin embargo, es esta misma quien da los permisos de acceso a las vistas del software.	<i>Cumple.</i> La autenticación se realiza contra U-Pasaporte. Ahora los permisos están almacenados en una nueva tabla de la base de datos.
El código fuente debe estar apropiadamente versionado y respaldado en un repositorio.	<i>Cumple.</i> La aplicación posee repositorio en Github.	<i>Cumple.</i> No hay cambios respecto a la versión original.

El stack tecnológico utilizado debe estar indicado.	<i>Cumple parcialmente.</i> Sólo se conoce la información del software del código legado, se desconoce qué stack tecnológico se utiliza para el ambiente de producción.	<i>Cumple.</i> El stack tecnológico ahora es manejado por medio de imágenes de Docker. Por lo que se conoce y se puede consultar fácilmente el stack a usar.
El stack tecnológico debe estar alineado con las definiciones del DCC.	<i>No cumple.</i> El stack tecnológico no forma parte del stack estándar del Área de Desarrollo.	<i>Cumple parcialmente.</i> No es factible cambiar el stack tecnológico. Sin embargo, se implementó Docker que sí cumple con las definiciones del DCC.
El sistema debe contar con las indicaciones correspondientes para que un tercero pueda realizar el deployment del sistema.	<i>No cumple.</i> No existen instrucciones para levantar un ambiente ni código asociado para hacerlo.	<i>Cumple.</i> Existen instrucciones dentro del código vía archivo readme.md para hacer deployment del sistema.

5.3.1 La arquitectura del ecosistema debe estar documentada y actualizada

Antes de realizar este trabajo de memoria, no existía documentación que mencionara el ecosistema de la aplicación. Ahora existen los diagramas de contenedores y componentes (véase figuras 18 y 19) que muestran dicha información.

5.3.2 La arquitectura de software debe estar documentada y actualizada

Antes de este trabajo de memoria, existía documentación parcial de la arquitectura lógica de la aplicación (véase figura A4 en el Anexo III). Ahora existen los diagramas de contexto, de contenedores y componentes (véase figuras 13, 18 y 19) que muestran dicha información completa respecto a la arquitectura de la plataforma.

5.3.3 El modelo de datos debe estar documentado y actualizado

A pesar de que sí había información del modelo de datos, ésta se encontraba dispersa y desactualizada (véase sección 3.1 Cambios al Modelo de Datos). Ahora con los nuevos ajustes, el modelo de datos se encuentra actualizado y en un único diagrama.

5.3.4 Debe contar con una API, la cual debe estar documentada y actualizada para acceder a los datos

Durante este trabajo de memoria se creó una API utilizando FastAPI [11] de Python; código que tiene todo el esquema de la base de datos y es fácilmente extensible.

5.3.5 La autenticación debe realizarse contra U-Pasaporte

La autenticación se realizaba hacia la Intranet del DCC, ahora la autenticación se realiza contra U-Pasaporte, manejando los roles de los usuarios de manera interna.

5.3.6 El código fuente debe estar apropiadamente versionado y respaldado en un repositorio

El código ya estaba originalmente en un repositorio de Github ya versionado y respaldado.

5.3.7 El stack tecnológico utilizado debe estar indicado

Dentro del código legado no se indica qué software se utilizaba para ponerlo en producción. Actualmente la aplicación, al estar en contenedores de Docker, tiene una mejor documentación respecto a qué stack tecnológico se está utilizando.

5.3.8 El stack tecnológico debe estar alineado con las definiciones del DCC

El stack tecnológico actual no es parte del stack estándar del DCC. Tampoco es factible cambiar el código de PHP a Python, dado que habría que reescribir todo el código desde cero. Sin embargo, se adoptó Docker para tener un mejor control que sí está alineado con las definiciones del DCC.

5.3.9 El sistema debe contar con las indicaciones correspondientes para que un tercero pueda realizar el deployment del sistema

El código legado no disponía de instrucciones para tanto hacer deployment en un ambiente de desarrollo como uno productivo. Ahora dentro del código nuevo se encuentra un archivo readme que incluye las instrucciones para realizar deployment junto con las variables de ambiente que se tienen que ser configuradas.

6. Conclusiones y trabajo a futuro

La gobernabilidad de una plataforma es una propiedad que se va degradando en el tiempo, a medida que distintos desarrolladores (o equipos de desarrollo) van evolucionando el producto y dejando su marca en éste. U-Proyectos no es la excepción.

La intervención realizada durante este trabajo de memoria espera extender la vida útil de U-Proyectos, y así prevenir que el trabajo realizado por Carlos Dettoni [1] y otros más en el pasado no se pierda. La intervención ha sido extensa, aunque es posible que no sea percibida por los usuarios de ésta, dado a que no se añade ninguna funcionalidad en el front-end, excepto la autenticación con MiUchile. Los cambios realizados y la documentación creada durante esta memoria, estuvieron enfocados principalmente a mejorar el apoyo al Área de Desarrollo del DCC, y a los futuros desarrolladores que tengan que intervenir esta plataforma.

El aumento de la documentación y del conocimiento del funcionamiento interno de la plataforma genera que su gobernabilidad también se vea aumentada. Otro cambio importante que impacta positivamente en la gobernanza de U-Proyectos es la actualización de PHP, ya que ésta, al ser obligatoria y necesaria, le agregaba una tarea extra a cualquier desarrollador que fuese a extender y/o mantener el sistema.

Se espera a futuro que la API implementada haga que U-Proyectos sea la fuente de información principal con respecto a los proyectos, y así su integrabilidad al ecosistema del DCC también se vea mejorada.

A nivel técnico, a futuro se espera que los próximos desarrollos apunten a:

- Actualizar las versiones de las librerías utilizadas en la implementación, y utilizar un gestor para éstas.
- Restaurar el envío automático de correos electrónicos por parte de la aplicación.
- Actualizar y extender la API creada para darle un uso real, haciendo así que el sistema sea aún más integrable.

7. Bibliografía

- [1] Dettoni Toledo, C. P. (2016). *Reingeniería del Sistema de Gestión de Proyectos del Departamento de Ciencias de la Computación*. Memoria de Ingeniería Civil en Computación. Departamento de Ciencias de la Computación, FCFM, Universidad de Chile.
- [2] Ochoa, S., & Arriagada, J. P. (2022). *Trabajos de Memoria Ofrecidos - Otoño 2022*. Documento Interno, Departamento de Ciencias de la Computación de la Universidad de Chile.
- [3] *PHP 5.5*. (n.d.). PHP. Visitado el 4 de Marzo de 2023, URL: <https://prototype.php.net/versions/5.5/>
- [4] Fowler, Martin (2003). *Patterns of Enterprise Application Architecture*. Page 330. Pearson Education.
- [5] *PlantUML* (n.d.). Open-source tool that uses simple textual descriptions to draw beautiful UML diagrams. Visitado el 4 de Marzo de 2023, URL: <https://plantuml.com>
- [6] Brown, Simon. (n.d.). The C4 model for visualising software architecture. c4model.com. Visitado el 4 de Marzo de 2023, URL: <http://c4model.com>
- [7] C4-PlantUML (n.d.). C4-PlantUML combines the benefits of PlantUML and the C4 model for providing a simple way of describing and communicate software architectures. (n.d.). GitHub. Visitado el 4 de Marzo de 2023, URL: <https://github.com/plantuml-stdlib/C4-PlantUML>
- [8] Portal MiUchile (n.d.). Universidad de Chile. Visitado el 4 de Marzo de 2023. URL: <https://autenticacion.uchile.cl>
- [9] Portal Servicios DCC (n.d.). Servicios disponibles del Portal DCC. Visitado el 4 de Marzo de 2023. URL: <https://portal.dcc.uchile.cl>
- [10] PHP (n.d.). PHP: List of security vulnerabilities. Visitado el 23 de Diciembre del 2022, URL: https://www.cvedetails.com/vulnerability-list/vendor_id-74/product_id-128/year-2017/PHP-PHP.html.
- [11] FastAPI (n.d.). FastAPI framework, high performance, easy to learn, fast to code, ready for production. Visitado el 4 de Marzo de 2023, URL: <https://fastapi.tiangolo.com>.
- [12] SQLAlchemy (n.d.). SQLAlchemy, SQL databases in Python, designed for simplicity, compatibility, and robustness. Visitado el 4 de Marzo de 2023, URL: <https://sqlmodel.tiangolo.com>
- [13] PHP (n.d.) PHP: Built-in web server - Manual. Visitado el 9 de Marzo de 2023, URL: <https://www.php.net/manual/en/features.commandline.webserver.php>.

Anexo A: Requisitos de gobernabilidad e integración de sistemas del DCC

Para asegurar en gran medida la gobernabilidad de las aplicaciones y la integración entre ellas, cada sistema que se ponga en producción en el DCC debe incluir (al menos) lo siguiente:

- **Una descripción del proceso que apoya.** Esto incluye el objetivo del proceso abordado, sus actores, las macro-tareas, el workflow del mismo, los dolores actuales del proceso (problemas a resolver u oportunidades a aprovechar) y sus eventuales causas. El proceso debe describirse usando la nomenclatura BPMN (<https://www.bpmn.org/>). Además, la descripción general del proceso debe especificarse a través de un tablero digital del tipo: “exploración del problema y contexto”, como el que se muestra a continuación.

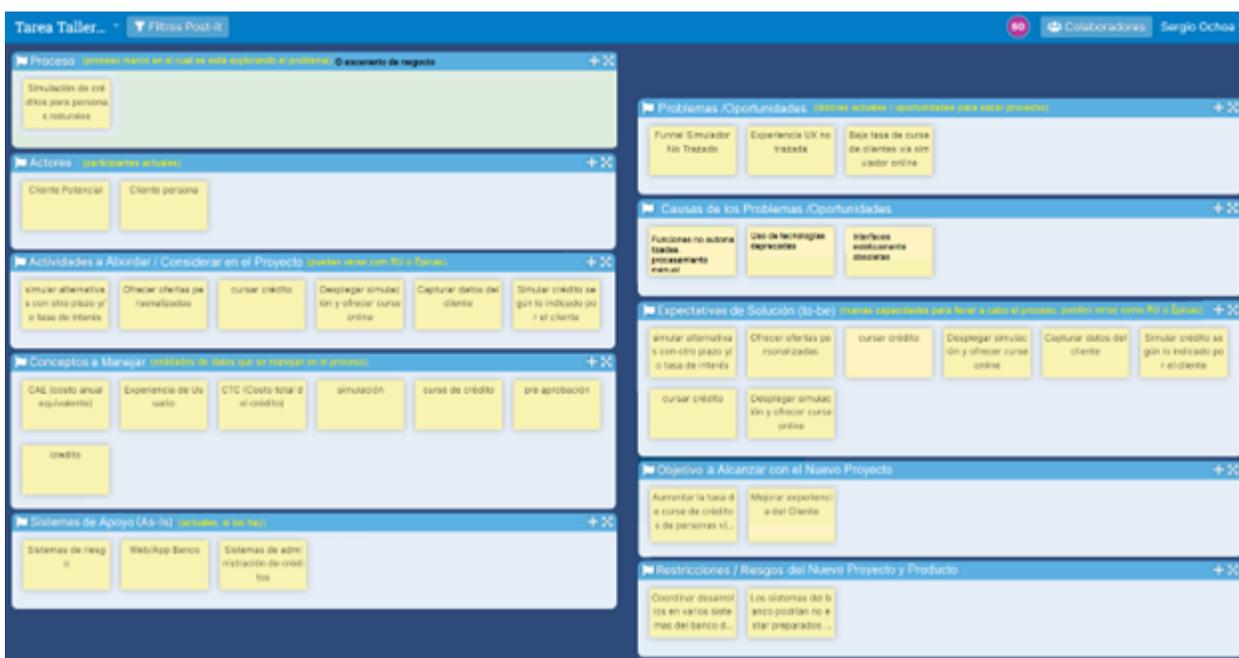


Figura A1. Ejemplo de tablero de contexto y problema a abordar

- **Objetivos y alcance del sistema.** Esto debe especificarse utilizando un tablero digital de alcance de productos (tablero de scoping), como el que se muestra a continuación.

- ***El código fuente debe estar apropiadamente versionado.*** El código fuente de la aplicación deberá estar apropiadamente versionado, estructurado y almacenado en una herramienta de control de versiones; particularmente, en github o gitlab.
- ***El stack tecnológico utilizado.*** Esto debe incluir las tecnologías y sus versiones mínimas requeridas para poder funcionar.
- ***Indicaciones para su deployment.*** Colocarlos dentro del readme.md del repositorio del proyecto, tanto indicando las variables de ambiente necesarias, comandos necesarios, etc.

Anexo B: Readme del código

El siguiente es el contenido del ReadMe del repositorio de U-Proyectos para referencia:

U-Proyectos DCC

Sistema de Administración de Proyectos DCC

Información sobre el proyecto

Stack tecnológico

- Docker
- MariaDB
- Frontend: JavaScript, HTML, CSS y jQuery
- Backend: php con PHPMailer y PHPExcel

Levantar ambiente de desarrollo

Estas son las instrucciones para levantar un ambiente de desarrollo.

Crear y configurar variables de entorno

- Crear un archivo `.env` en la carpeta `.docker`
- Este archivo debe contener los siguientes variables de entorno:

Variable de Entorno	Descripción
<code>MYSQL_ROOT_PASSWORD</code>	Contraseña del usuario root de la consola de MySQL
<code>MYSQL_DATABASE</code>	Nombre de la base de datos. Por defecto: <code>uproyectos_db</code>
<code>MYSQL_USER</code>	Nombre de usuario que utiliza el cliente para conectarse a la base de datos
<code>MYSQL_PASSWORD</code>	Contraseña de dicho usuario

MYSQL_PORT	Puerto de conexión con la base de datos. Por defecto: 3306
MYSQL_HOST	Nombre/ruta del host de la base de datos. Recomendado: mariadb
SSO_URL	URL del portal de Autenticación del DCC
SSO_APP	Nombre de la aplicación que está registrada en el portal del DCC

Para obtener las variables del SSO, comunicarse con el Área de Desarrollo del DCC.

Echar a correr las imágenes

Requisitos: Docker y/o Docker Desktop.

Para levantar la plataforma se debe ejecutar el siguiente comando en la carpeta `.docker` de este repositorio: `docker-compose up --build -d`

Esto hará que se construyan 4 contenedores:

- mariadb: Base de datos MariaDB de la aplicación, necesaria para todos los otros contenedores.
- web: Aplicación web de UProyectos guardada en `/src`.
- api: API de python con FastAPI que sirve los datos de la base de datos.
- phpmyadmin: Aplicación que permite la visualización y administración de la base de datos por medio de una interfaz.

Para un ambiente productivo se recomienda solo usar los 3 primeros contenedores.

Cargar el Dump de la base de datos

Dado un dump `dump.sql`, se puede ejecutar el siguiente comando para cargar un archivo dump a la base de datos:

```
cat dump.sql | docker exec -i uproyectos-mariadb-1 /usr/bin/mysql -u root --password=example uproyectos_db
```

Se puede realizar el proceso inverso, es decir, guardar la base de datos a un archivo `dump_uproyectos.sql` utilizando:

```
docker exec docker-mariadb-1 mysqldump --user root --password=example uproyectos_db > dump_uproyectos.sql
```

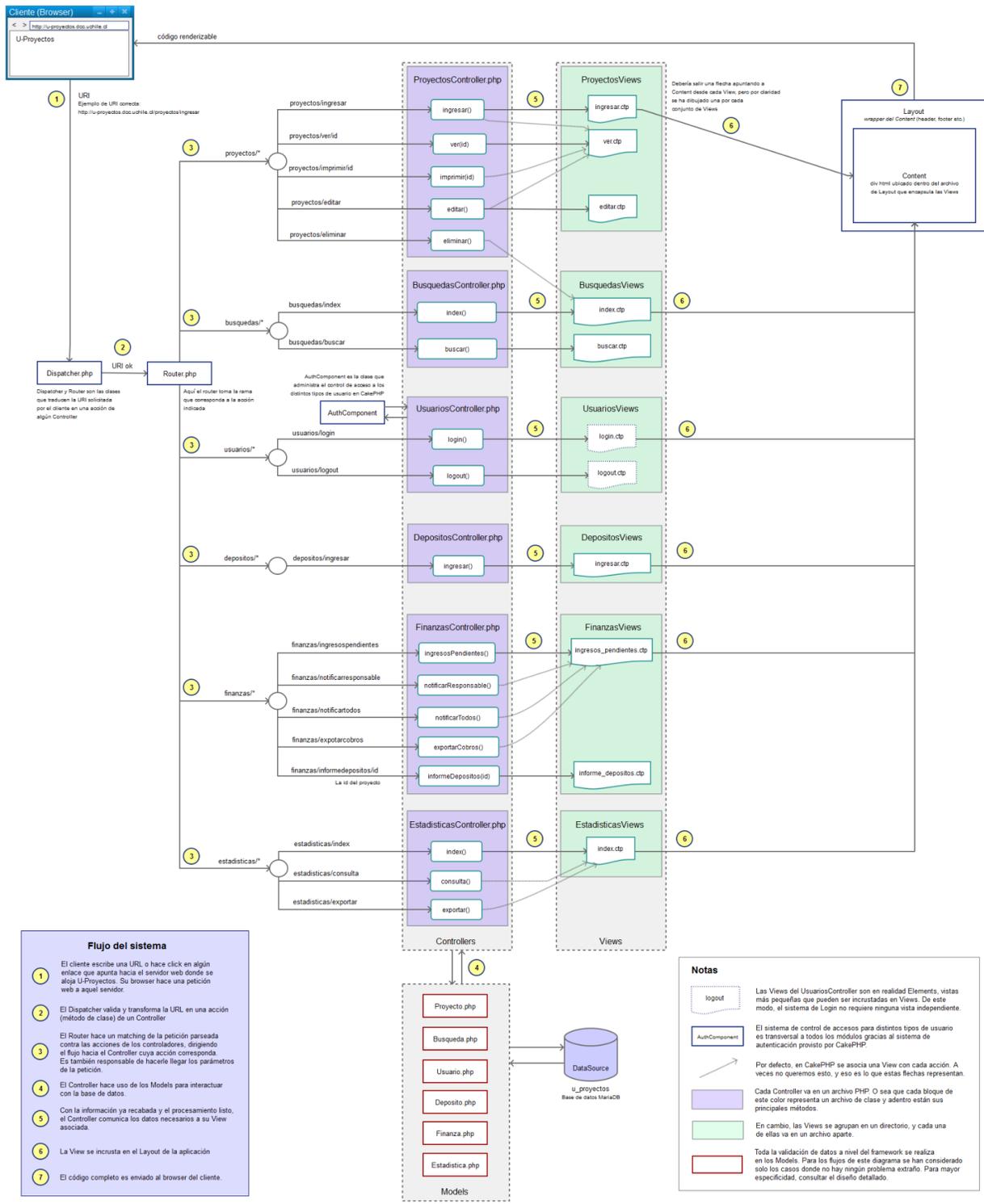



Figura A4: Arquitectura lógica de U-Proyectos. (Legado)