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

MEJORA DE LA GOBERNABILIDAD DE U-PAPERS Y DE SU CAPACIDAD DE INTEGRACIÓN AL ECOSISTEMA DE SOFTWARE DEL DCC

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

ROBERTO ARTURO AGUILERA DONOSO

PROFESOR GUÍA:
Sergio Ochoa Delorenzi
PROFESOR CO-GUÍA:
Juan Arriagada Cancino

MIEMBROS DE LA COMISIÓN:
Éric Tanter
Hugo Mora Riquelme

SANTIAGO DE CHILE
2023

MEJORA DE LA GOBERNABILIDAD DE U-PAPERS Y DE SU CAPACIDAD DE INTEGRACIÓN AL ECOSISTEMA DE SOFTWARE DEL DCC

El Departamento de Ciencias de la Computación (DCC) está llevando a cabo un proceso de homogeneización de sus sistemas de software. Para ello, el Área de Desarrollo de Software (ADS) se ha encargado de definir los requerimientos de gobernabilidad que deben cumplir todos los sistemas internos del DCC.

En ese contexto, el Sistema de Gestión de Publicaciones (conocido como U-Papers), es una de las aplicaciones que requiere ser intervenida y mejorada, pues no cumple con ninguno de los requerimientos establecidos por el ADS. Por ejemplo, no posee ningún tipo de documentación, por lo que su funcionamiento interno sólo puede ser conocido revisando el código fuente. Tampoco cuenta con acceso programático a sus datos, lo cual no le permite integrarse con otros sistemas que componen el ecosistema de aplicaciones del DCC. Finalmente, el sistema implementado está usando tecnología actualmente obsoleta.

El objetivo principal de este trabajo de memoria es convertir U-Papers en un sistema gobernable y capaz de integrarse al ecosistema del DCC, a través del cumplimiento de los lineamientos de gobernabilidad establecidos por el ADS.

Para cumplir este objetivo, el primer paso fue revisar el sistema legado, y generar documentación que permite entender el flujo de datos del proceso actual y su alcance. Luego, se procedió a ajustar el sistema para que sea gobernable. Por ejemplo, se actualizó el código fuente para que pudiera operar con versiones más nuevas del framework usado en su implementación. Además, se reemplazó el análisis de datos interno como las estadísticas de visitas por Google Analytics, se incluyó la autenticación de usuarios utilizando UPasaporte y se terminaron de implementar funcionalidades inconclusas en anteriores intervenciones al sistema. Finalmente, se automatizó la actualización de las fotos de perfil de los académicos, utilizando imágenes obtenidas desde el sitio web oficial del DCC.

Para corroborar que las mejoras mencionadas en el párrafo anterior, efectivamente mejoraron su gobernabilidad y su capacidad de integración al ecosistema del DCC, se realizó una comparación entre el estado inicial (anterior a la intervención) y el estado actual del sistema, considerando los requerimientos de gobernabilidad establecidos por el ADS. Los resultados obtenidos muestran que se lograron los objetivos inicialmente planteados en esta memoria; no obstante, aún queda trabajo por realizar dentro del sistema U-Papers. La interfaz de usuario aún no es muy amigable ni es "responsive". Además, los autores deberían poder visualizar y actualizar sus áreas de trabajo, lo cual hoy solo puede hacerlo el administrador del sistema. También, es necesario seguir actualizando las tecnologías usadas en la implementación del sistema, pues algunas de ellas ya no cuentan con soporte oficial, lo que dejaría a U-Papers vulnerable ante ataques informáticos. Por otra parte, la versión del framework actualmente en uso en el sistema aún recibe mantención y actualizaciones de seguridad, pero ya existe una versión nueva en desarrollo, por lo que avanzar a esa línea sería pertinente.

A mi querida familia,
y a mi querida gatita.

Roberto

Agradecimientos

Esta memoria representa el final de un camino que no podría haber recorrido solo. Tuve la suerte de tener la compañía y cariño de buenas personas que dejaron su marca en mi.

Gracias a mi papá, mamá y hermanos que me dieron su incondicional apoyo a lo largo de estos años. Gracias por su paciencia y consejos. Siempre han sido y serán un pilar en corazón.

Gracias a mi tía Teresa, tío Alfonso y mi primos que me dieron un lugar en su hogar este último año sin pedir nada a cambio. Nunca podré terminar de agradecerles todo lo que hicieron por mi.

Gracias también a todos mis amigos y amigas con los que pasé muy buenos momentos que me hicieron disfrutar los tiempos difíciles de la universidad. Y en especial, gracias a Sebastián, Javiera, Nataly y Diego. Todas las tardes que pasé trabajando con ellos fueron un pequeño paso en este proyecto y sin ellos no habría sido lo mismo. Me alegro de haberlos conocido y poder llamarlos mis amigos.

Finalmente, agradezco a mi profesores guía, Sergio Ochoa y Juan Pablo Arriagada quienes siempre estuvieron ahí para ayudarme a encaminar este trabajo y aconsejarme.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

En el Departamento de Ciencias de la Computación, en adelante el DCC, se han desarrollado a través de los años diversos sistemas de software, con el fin de resolver problemas internos de éste. Estos sistemas han surgido gracias a la iniciativa de académicos del departamento, y uno de estos es el Sistema de Gestión de Publicaciones, llamado *U-Papers* [1]. Esta es una aplicación Web que contiene un repositorio digital de artículos científicos, concebido con el propósito de dar visibilidad a las publicaciones de académicos y alumnos del DCC. El proyecto comenzó en 2012 como un trabajo de memoria[1], y, desde entonces, ha sido extendido por diversas personas.

Al no ser un desarrollo formal del DCC, U-Papers no fue inicialmente concebido para funcionar en conjunto con otros sistemas del departamento. Esto actualmente dificulta el libre intercambio de información entre U-Papers y otros sistemas (por ejemplo, la página Web del DCC que consume información de U-Papers), limitando el funcionamiento interno de varias actividades del DCC. Actualmente U-Papers está incluido en la forma de un enlace en la página web debido a que no es posible extraer información; contar con los datos de U-Papers permitiría mostrar esta información en el mismo contexto de la página web del departamento, así como también poder extender la información en los perfiles de los académicos.

Junto con lo anterior, y por el hecho de haber sido intervenido por distintos desarrolladores, U-Papers no cuenta con una documentación oficial, ni actualizada. Su código fuente se encuentra en diferentes estados en varias ramas de su repositorio, pero ninguno es contado como la versión oficial. Por último, no existe una definición formal sobre la estructura que debe respetar ni de los servicios que debe usar para integrarse con otros productos del DCC.

Teniendo en cuenta lo anterior, y que U-Papers no es el único sistema que presenta esos problemas, en el 2021 se creó formalmente el Área de Desarrollo del Software del DCC. Ésta tiene por objetivo dar mantenimiento y capacidad de evolución a sistemas legados (como U-Papers), así como también, desarrollar nuevos productos para apoyar en la mejora de los procesos administrativos y de docencia del departamento, y la integración de datos y servicios con unidades de la Facultad y la Universidad.

Con eso en mente, el área ha definido ciertos criterios mínimos que los sistemas de DCC

[1] Felipe Chacón, 2012. U-Papers Memoria de Ingeniería Civil en Computación, FCFM. Universidad de Chile.

deben cumplir para garantizar su gobernabilidad y facilitar su integración. Estos criterios se indican en la sección de anexo del documento.

En esta memoria se pretende dar a conocer cómo se intervino el sistema U-Papers para lograr que éste evolucione hasta un punto en que, como mínimo, cumpla con todos los criterios antes mencionados, y, por lo tanto, pase a formar parte de los sistemas formales del DCC. Así, beneficiando al departamento y la universidad, pues mejoraría la actual infraestructura de software y permitiría desarrollar a U-Papers de manera razonable de cara al futuro.

1.1. Situación Inicial

La versión legada del sistema U-Papers cuenta con funcionalidades que permiten la búsqueda y recuperación de publicaciones científicas del departamento, así como también la gestión de las publicaciones y usuarios del sistema. De esa manera, brinda la posibilidad a los académicos de dar a conocer sus trabajos de manera directa. Aunque el sistema es útil, éste tiene diversas limitaciones en términos de escalabilidad y mantenibilidad.

Por otra parte, y tal como se mencionó anteriormente, el DCC creó el Área de Desarrollo de Software (ADS), para desarrollar nuevos sistemas e integrarlos al ecosistema de software del departamento. El ADS también se encargará de monitorear el cumplimiento de las pautas de gobernanza e integración, por parte de los sistemas legados de la organización; uno de estos sistemas es U-Papers.

U-Papers no cuenta con una documentación oficial, ni tampoco está correctamente versionado, lo que desemboca en una deuda técnica que dificulta su escalabilidad y adecuada mantenibilidad. Tampoco se conoce el alcance del producto en términos de funcionalidad, por lo tanto, no se tiene una idea clara de cómo seguir evolucionando este. Finalmente, la versión legada del sistema no cuenta con una manera de conectarse con los demás sistemas del DCC, es decir, no tiene forma de compartir sus datos con áreas del departamento que podrían necesitarlas, ralentizando el flujo de la información entre ellas.

Estas problemáticas representan desafíos para el ADS, los cuales necesitan ser resueltos para que U-Papers pueda convertirse en un proyecto formal del DCC. Con estos motivos se origina este proyecto de memoria, el cual busca mejorar la gobernabilidad de U-Papers, dándole sostenibilidad y escalabilidad en el tiempo, y capacidad de integrarse a los demás sistemas del DCC.

1.2. Objetivos de la Memoria

El objetivo general de la presente memoria es convertir a U-Papers en un sistema gobernable y capaz de integrarse al resto del ecosistema del DCC; por lo tanto, debe mejorarse para poder cumplir con todos los criterios establecidos anteriormente. Para lograr este objetivo general, se definieron los siguientes objetivos específicos:

1. Revisar el sistema legado y, a partir de ello, generar la documentación indicada como parte de los criterios mínimos a cumplir y, además, crear un repositorio de código fuente gobernable. La revisión del sistema podría concluir que la mejor opción es reimplementar desde cero, en vez de modificarlo. En ese caso, se deberá documentar el "as-is" y el "to-be" del sistema.

2. Modificar el sistema legado para cumplir con las capacidades mínimas de integración entre sistemas y gobernabilidad de este.

3. Implementar un servicio que extienda la funcionalidad actual del sistema, para demostrar que efectivamente U-Papers ha mejorado su gobernabilidad y su capacidad de integración al ecosistema de software del DCC.

1.3. Solución Inicial

La solución propuesta busca analizar y extender el sistema legado (U-Papers) con el objetivo de integrarlo correctamente junto a los demás sistemas internos del DCC. Se considera mantener las tecnologías ya usadas en U-Papers (MySQL, PHP, JQuery y el framework CodeIgniter), modificando el código para usar sus versiones actuales con soporte oficial e integrando otras nuevas según sea necesario. Para que U-Papers alcance los niveles de gobernabilidad e integración requeridos, estos son los requisitos que se deben cumplir:

- **Integración de datos:** Agregar una capa extra a la arquitectura, incorporando una API REST, a través de la cual se compartirán los datos de U-Papers con otros sistemas del ecosistema del DCC. De esa manera, se conseguirá una apropiada integración entre éstas, que permita el libre flujo de la información.

- **Gobernabilidad:** Analizar el código fuente del sistema, para comprender sus dolencias y alcance, con tal de crear una documentación actualizada y adecuada. Además, se debe crear un repositorio para el código fuente, en el cual el U-Papers esté correctamente versionado, para facilitar futuras extensiones.

- **Implementación de un nuevo servicio:** Éste debía ser una extensión al legado, y demostrar que luego de su mejora, se había logrado cumplir con los criterios mínimos de gobernanza e integración definidos (ver Anexo).

1.4. Estructura del documento de memoria

En el próximo capítulo se muestra un análisis de la aplicación legada, considerando los requisitos definidos por el DCC para mantener un buen nivel de gobernabilidad e integración de sus sistemas. Para esto se construye un tablero de exploración de problema y contexto, con el cual se busca plantear, por ejemplo, quiénes participan del proceso a mejorar, cuáles son los dolores que tiene la organización debido a esos problemas, y cómo podrían solucionarse/mitigarse. Además, se analizan los riesgos que podrían encontrarse en el desarrollo de la solución.

Por otra parte, se define el diagrama de contexto del sistema y del ecosistema usando la nomenclatura del modelo de contextos C4 [2], así como también un modelo de la base de datos de U-Papers, con el fin de documentar el estado actual del sistema.

En el capítulo 2 también se analizan las interfaces de usuario del sistema, en busca de posibles aspectos a mejorar. Teniendo estos antecedentes a la vista, se determina el cumplimiento de los requisitos definidos por el DCC, y también se plantean los desafíos que supone modificar el sistema, según lo encontrado en el análisis.

En el capítulo 3 se describen los ajustes realizados al sistema U-Papers. El primer ajuste es la actualización de la versión del framework (CodeIgniter) sobre el cual está construido el sistema. Éste se encontraba en su versión 2.0.2, y por lo tanto se hizo un *upgrade* hasta la versión 3.1.13, la cual es compatible con versiones de PHP que cuentan con soporte estable; es decir, al menos para PHP 7.0. La siguiente modificación es el empaquetamiento del sistema en Docker, con el objetivo de facilitar la implementación de las extensiones y simplificar el *deployment* de la aplicación.

En el capítulo 4 se explican las extensiones realizadas al sistema con el objetivo de cumplir los objetivos planteados en esta memoria. Una de estas extensiones tiene que ver con la inclusión de un método de *log in* para los usuarios que hace uso de la autenticación mediante las credenciales de MiUchile. Además, se implementa una API para consultas a la base de datos, la cual permite obtener la lista de académicos, junto con sus IDs de usuario, y las publicaciones de un autor en particular ordenadas por fecha. Por último, se realizó la automatización de la actualización de las fotografías de perfil de los usuarios de rol “Académico”.

En el capítulo 5 se describe el proceso de evaluación realizado y se muestran los resultados obtenidos. Finalmente, el capítulo 6 presenta las conclusiones y el trabajo a futuro.

Capítulo 2

Análisis del Escenario a Intervenir

A continuación, se presentará el trabajo adelantado durante el semestre. Éste consistió, principalmente, en un análisis inicial del sistema intervenido y la generación de una parte inicial de la documentación requerida.

2.1. Identificación del problema y contexto abordado

La Figura 2.1 presenta el tablero de exploración del problema y contexto abordado por la versión legada de la aplicación U-Papers. Luego se explican los distintos componentes (o secciones) del tablero.

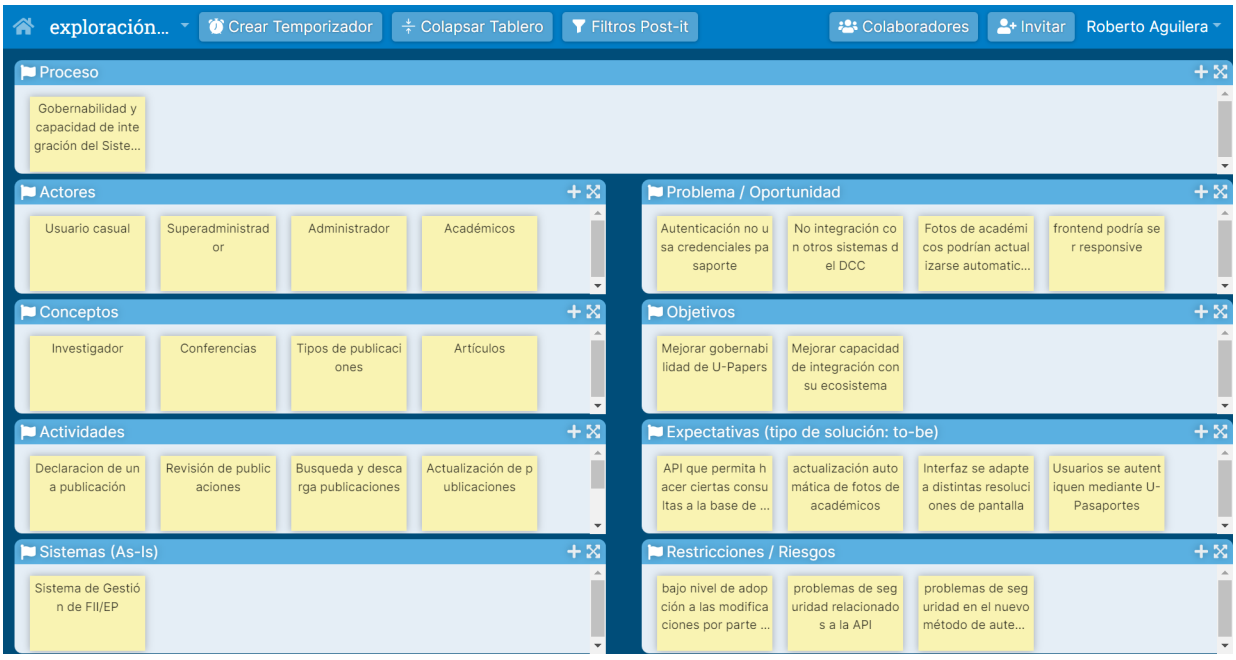

Figura 2.1: Tablero de exploración del problema y contexto abordado por el sistema U-Papers

En la sección **proceso**, se indica que lo que se está analizando es el workflow de U-Papers; además, se está abordando cómo mejorar su gobernabilidad y capacidad de integración a las demás aplicaciones de software del ecosistema del DCC. Para esto se toman en cuenta varios aspectos; el primero de ello, como se muestra en la Fig. 1, son los **actores** que participan del

proceso, y que juegan un rol importante en él.

Un primer tipo de actor son los *usuarios casuales* (visitantes) que no poseen un rol definido, por lo que no pueden autenticarse. Otro de los actores es el *superadministrador*, que está encargado de revisar y aprobar las publicaciones que se añaden al sistema. Por su parte, el *administrador* gestiona publicaciones. Finalmente, están los académicos, que son quienes declaran las publicaciones y son autores de éstas.

Respecto de los **conceptos** que maneja el proceso, estos están asociados a las entidades de datos involucradas, como, por ejemplo, los investigadores, los artículos, los tipos de publicaciones, y las conferencias y revistas (journals) donde se publican dichos artículos.

Las **actividades** se refieren a las macro actividades que componen el proceso. En el caso de U-Papers, estas corresponden a las declaraciones de publicaciones por parte de los académicos, las cuales son enviadas al superadministrador, quien se encarga de aprobarlas (o rechazarlas). Las publicaciones pueden ser buscadas y descargadas para ser difundidas, el cual es el principal propósito del proceso de U-Papers.

La sección **sistemas** (es decir, el as-is) hace referencia a los sistemas que apoyan la operación del proceso en estudio. En este caso, el único sistema que apoya la ejecución del workflow de U-Papers es el Sistema de Gestión FII/EP. Éste es un sistema legado que se alimenta de U-Papers, pues las declaraciones de publicaciones son enviadas a éste, y luego pasan a la base de datos oficial de dicha aplicación.

Respecto a la sección **problemas/oportunidades** indicadas en el tablero, en ésta se indican las limitaciones del proceso actual, las cuales deben ser solucionadas (en el caso de los problemas) o aprovechadas (en el caso de las oportunidades). Entre los principales problemas de integración de U-Papers está el hecho de que la autenticación de los usuarios no hace uso de las credenciales de MiUchile de UPasaporte, y que el sistema no cuenta con una funcionalidad que le permita hacer fluir su información dentro del ecosistema del DCC.

Otro problema que tiene U-Papers, pero que tiene menos urgencia, es que la interfaz del sistema no es adaptable a otras resoluciones de pantalla. Por otra parte, dentro de las oportunidades de mejora se incluye la posibilidad de que las fotografías de los académicos puedan actualizarse automáticamente usando las que se muestran en el sitio Web del DCC.

Los **objetivos** por alcanzar dentro del proyecto son los mencionados en la sección 1.2; se indican como los objetivos específicos de la memoria. Por su parte, la sección de **expectativas** indicadas en el tablero corresponde a los servicios que la nueva solución debe aportar al proceso. Tomando en cuenta los problemas principales descritos anteriormente, los nuevos servicios a implementar son los siguientes: 1) una API que permite hacer consultas a la base de datos del sistema para mejorar la integración, y 2) incluir en el *log in* de U-Papers la autenticación mediante el servicio de credenciales de UPasaporte.

Finalmente, en la sección **riesgos** se muestran aquellos que se pueden visualizar a priori para el desarrollo de la solución. Estos riesgos son la potencial baja adopción de las modificaciones por parte de los usuarios, y los eventuales problemas con la seguridad que podría

provocar la implementación de nuevos servicios.

2.2. Análisis del Sistema U-Papers

A continuación, se presentará el modelo de datos de U-Papers [1], su arquitectura, algunas interfaces y las extensiones comprometidas, más allá de todos los cambios que se requiere realizar al sistema para que pueda cumplir con los lineamientos establecidos por el área de desarrollo del DCC.

2.2.1. Arquitectura del Sistema

Para especificar la arquitectura del sistema en distintos niveles de abstracción se utiliza el modelo C4 [2]. A continuación, se presenta la arquitectura de ecosistema (contexto) y de los macro componentes (contenedores) del sistema de U-Papers.

2.2.1.1. Diagrama de contexto del ecosistema

Como se aprecia en la Figura 2.2, dentro del ecosistema de U-Papers se encuentran distintos tipos de usuarios; cada uno tiene sus interacciones particulares con el sistema. Los usuarios *Casuales* (o anónimos) son los de menor jerarquía (no tienen un rol dentro del sistema) teniendo permitido solo buscar y descargar las publicaciones almacenadas en el repositorio. Después vienen los *Académicos*, *Administradores* y *Superadministradores* quienes poseen más atribuciones. Estos pueden, por ejemplo, solicitar la inclusión de una publicación al sistema (Académico), hasta gestionar los usuarios y publicaciones presentes en el sistema (Administrador y Superadministrador).

También hay otros sistemas que interactúan de manera directa e indirecta con U-Papers; por ejemplo, está el Sistema FII/EP, el cual es un sistema legado que sirve de fuente de información para U-Papers, por lo que la información en sus respectivas bases de datos debe ser consistente. Para esto, las solicitudes de nuevas publicaciones son enviadas al Sistema FII/EP, donde son revisadas por el superadministrador, y una vez aprobadas, la publicación es almacenada en ambos sistemas. Por otro lado, con la página oficial del DCC se tiene una relación indirecta, pues contiene un enlace en una de sus interfaces que lleva directamente a U-Papers.

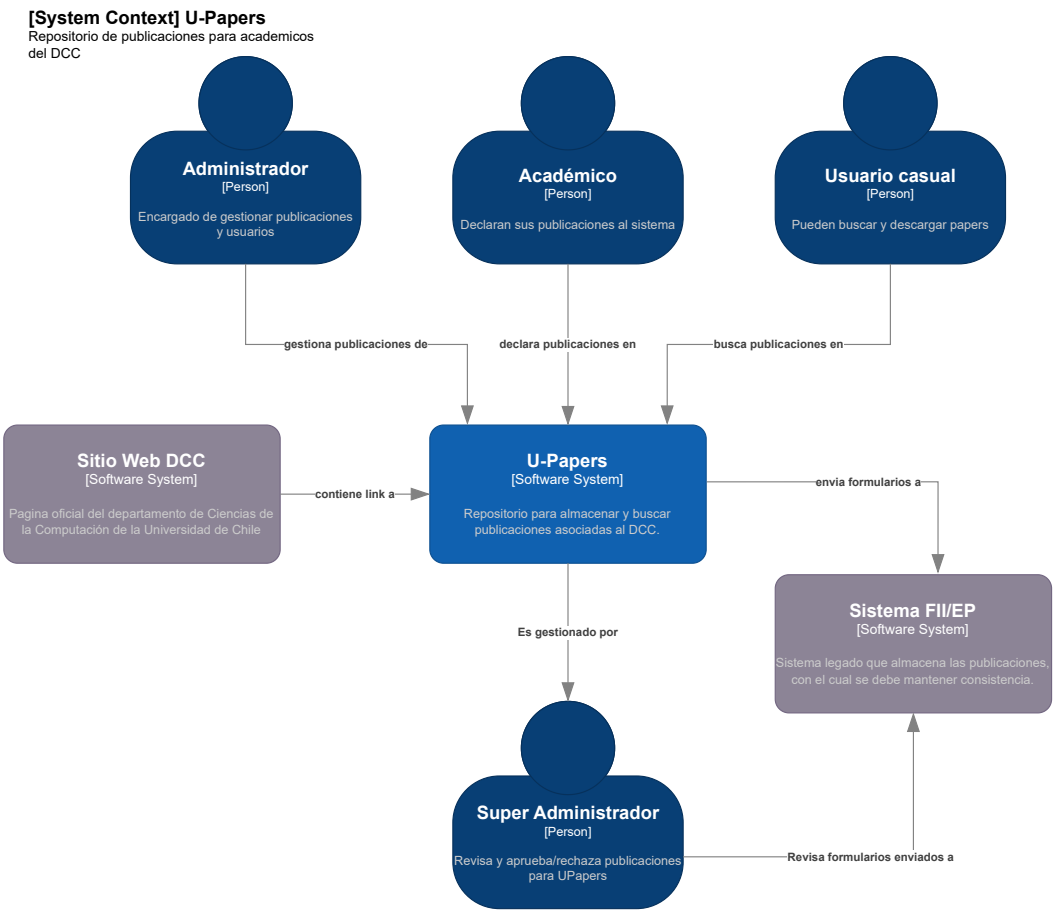

Figura 2.2: Modelo de contexto del sistema en C4

2.2.1.2. Diagrama de containers del sistema

Pasando al interior del sistema, en la Figura 2.3 se aprecia que éste está compuesto por tres componentes principales: una aplicación web, un controlador y una base de datos. Cada uno de estos refleja una capa de la arquitectura Modelo-Vista-Controlador (MVC).

La aplicación web representa la vista, pues es la que recibe las interacciones directas de los usuarios del sistema y las envía al controlador, el cual las procesa y las hace llegar a la base de datos. Esta última representa la capa de datos de la arquitectura MVC, y se encarga de extraer los datos necesarios y devolverlos al controlador.

La aplicación web está montada sobre un servidor web Apache HTTP Server [3], mientras que el controlador funciona sobre el framework de PHP, CodeIgniter [4]. Finalmente, el motor de la base de datos es MySQL [5].

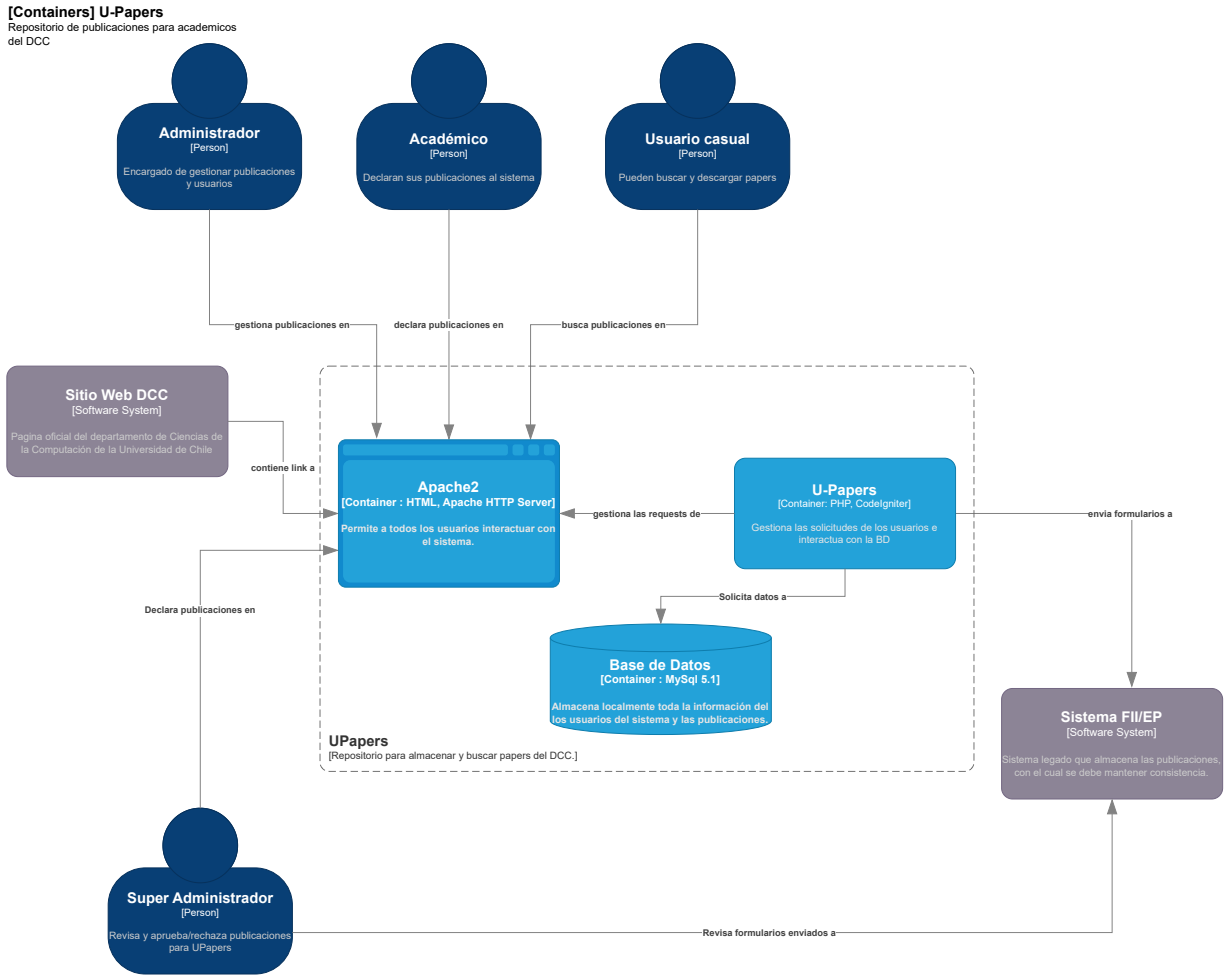

Figura 2.3: Modelo de contenedores del sistema en C4

2.2.2. Modelo de Datos

La base de datos de U-Papers es relacional, y almacena toda la información referente a las publicaciones, autores, usuarios, configuraciones del sistema, entre otras (Figura 2.4).

Actualmente, la información del sistema se divide en seis grupos: *publicaciones, autores, usuarios, configuraciones, estadísticas y tablas temporales*. La tabla llamada "publications", es la que contiene la información general de todos los papers(publicaciones). Se relaciona a otras tablas que dividen las publicaciones en 6 subgrupos: artículos de journal, artículos de conferencia, libros, capítulos de libro, editores de conferencias y editores de número de journal. Todos estos se forman a través de una serie de tablas que indican su tipo, editorial que lo publicó y edición de la publicación.

La tabla "authors" contiene los datos de todas las personas que han subido papers al sistema. Se relaciona a las publicaciones, mediante una tabla intermedia, para evitar la repetición de tuplas con autores ya existentes. Además, se relaciona a la tabla "users", la cual almacena la información general de los usuarios del sistema, donde cada uno puede tener uno de tres roles posibles, que le otorgan distintos grados de autoridad en el sistema (superadministrador, administrador y autor).

Luego, de los tres grupos restantes, la tabla "config" guarda información necesaria para la coherencia del sistema, como la zona horaria que usa o el tipo de extensión que se acepta en los documentos. Las tablas "forgot_password" y "flashdata" contienen información temporal útil para confirmar ciertas acciones de los usuarios. Finalmente, la tabla "países", junto con muchas otras tablas que no se agregaron al modelo de la Figura 2.4, sirven para mantener un registro de estadísticas en el sistema.

El modelo de datos que se muestra en la Figura 2.4 corresponde a una versión anterior, por lo que faltan algunas entidades como "forgot_password", "flashdata", y "países", entre otras. Estas extensiones y la revisión exhaustiva de las relaciones entre las tablas es parte del trabajo a realizar en este trabajo de memoria, para así cumplir uno de los criterios mínimos de gobernabilidad indicados en el Anexo.

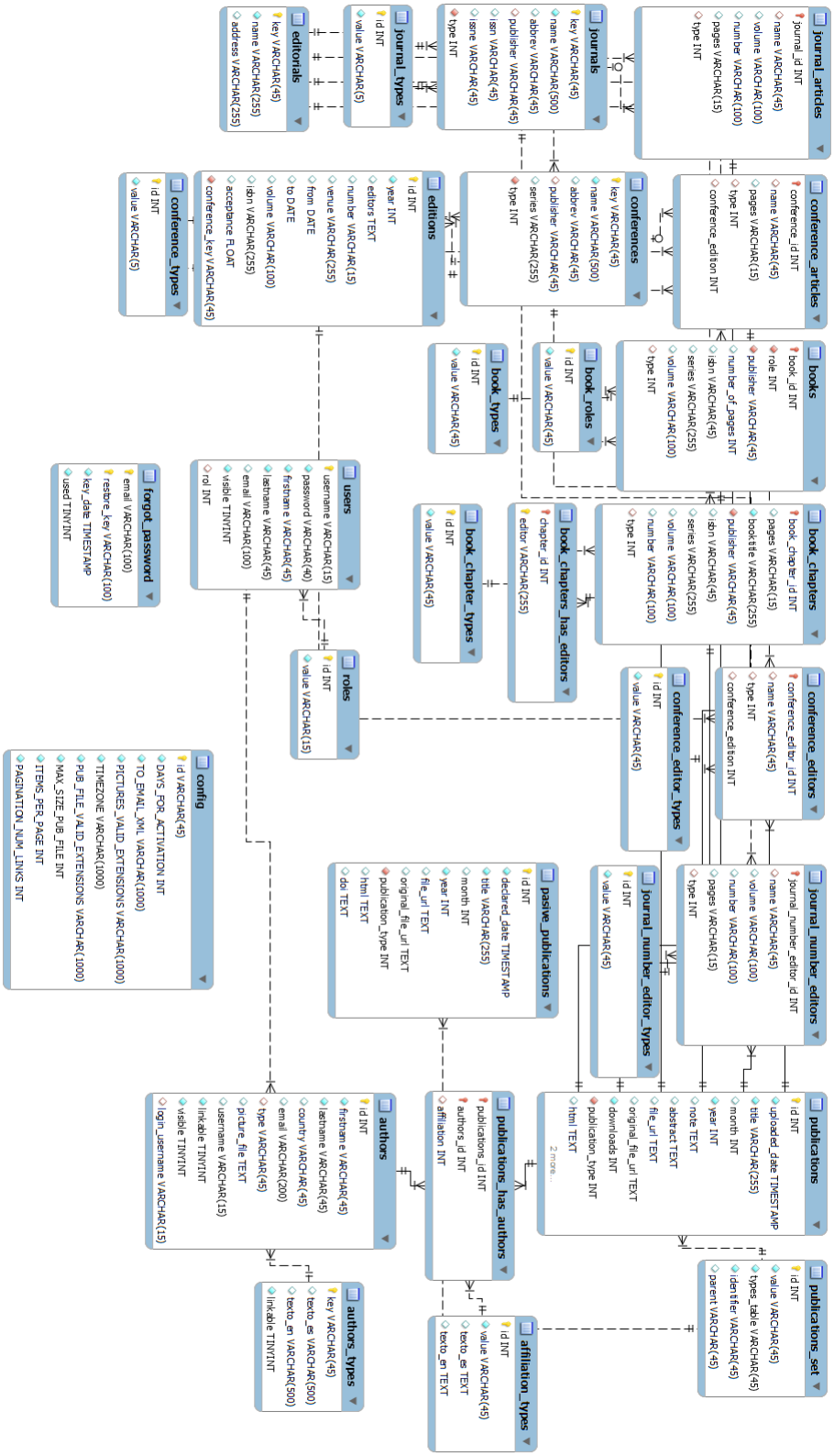

Figura 2.4: Modelo de datos del sistema U-Papers[1]

2.2.3. Interfaces Actuales

Las vistas del sistema U-Papers (versión legada) cuentan con tres secciones principales (Fig.2.5), un *header*, un menú lateral y el cuadro principal donde se despliega el contenido solicitado por el usuario. En el caso del header, éste es el mismo para todas las vistas. Éste está compuesto por el logo de U-Papers, un enlace al sitio web de la Universidad de Chile[2], otro enlace al sitio web del DCC[3], un enlace a una vista con estadísticas del sitio (Stats), un enlace a otra vista de preguntas frecuentes (FAQ), un enlace al disclaimer, un botón para elegir el idioma de la página (español o inglés), y un botón que despliega un formulario para iniciar sesión.

El menú lateral cuenta con dos secciones, una pública que permite elegir distintas formas de acceder a las publicaciones (buscador avanzado, por fecha y por autor), y una privada, disponible sólo para los usuarios registrados con un rol definido (académico, administrador y superadministrador).

La Figura 2.5 muestra una vista de la página principal. Esta cuenta con las dos secciones descritas anteriormente (común a todas las vistas), mientras que su cuadro principal contiene una breve explicación del alcance de U-Papers, un buscador sencillo y uno avanzado, junto con un listado de las últimas publicaciones añadidas al repositorio.

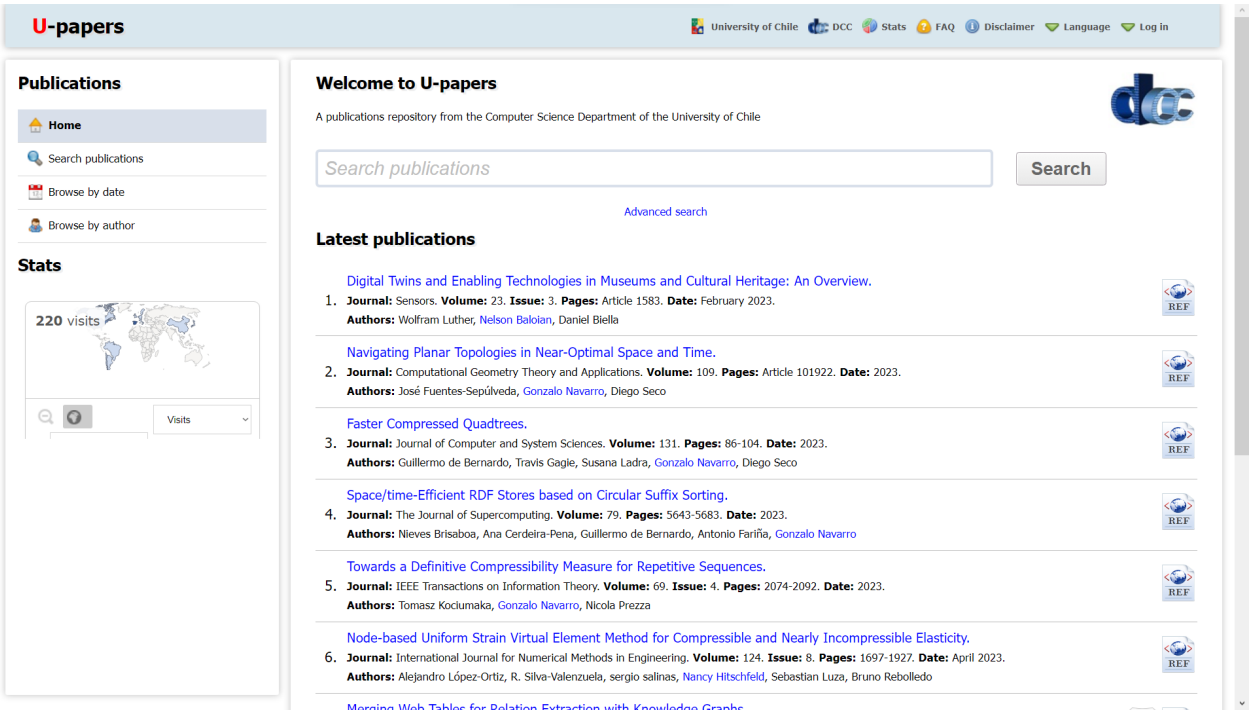

Figura 2.5: Página principal de U-Papers

[2] https://www.uchile.cl
[3] https://www.dcc.uchile.cl

2.3. Nivel de Cumplimiento con los Requerimientos Mínimos Establecidos por el DCC

A continuación, se indica el nivel de cumplimiento que tenía U-Papers antes de la realización de esta memoria, respecto a los requisitos mínimos de gobernabilidad e integración definidos para los sistemas del DCC (descritos en detalle en el Anexo).

Tabla 2.1: Nivel de cumplimiento inicial de U-Papers

Ítem Evaluado	Nivel de Cumplimiento
La arquitectura del ecosistema debe estar documentada y actualizada.	No cumple
La arquitectura de software debe estar documentada y actualizada.	Cumple parcialmente
El modelo de datos debe estar documentado y actualizado.	Cumple parcialmente
El sistema debe contar con una API, la cual debe estar documentada y actualizada para acceder a los datos.	No cumple
La autenticación de usuarios debe hacerse contra UPasaporte.	No cumple
El código fuente debe estar apropiadamente versionado y respaldado en un repositorio.	No cumple
El stack tecnológico utilizado debe estar apropiadamente indicado.	No cumple
El stack tecnológico debe estar alineado con las definiciones del DCC.	No cumple
El sistema debe contar con las indicaciones pertinentes para que un tercero pueda levantarlo en un ambiente de desarrollo.	No cumple

2.4. Desafíos para Extender el U-Papers

A lo largo de este proyecto se encontraron variados desafíos que debieron abordarse antes de extender el sistema. Estos afectan la adecuada gobernanza y funcionalidad de U-Papers, pues están relacionados directamente con el código fuente. Los desafíos fueron apareciendo a medida que se hacía una exhaustiva revisión inicial del código.

En primer lugar se nota que la versión en uso del framework CodeIgniter es la 2.0.2, la cual ni siquiera se puede descargar desde la página oficial, pues la versión más antigua que se puede encontrar es la 2.0.3. A priori, se intenta correr el código en local usando el servidor web (Apache) y sistema de gestión de bases de datos (MySQL) que provee el paquete de software libre XAMPP, obteniendo una gran cantidad de errores entre los que se destacan problemas para establecer conexión a la base de datos y dificultades para abrir la página de inicio del sitio. Por lo tanto, se llega a la conclusión de que la versión del framework fue

dada de baja hace tanto tiempo que ya no soporta el uso de versiones más modernas de PHP y MySQL, por lo que se decidió actualizar la a la versión más reciente; específicamente a CodeIgniter 3.1.13.

Otro desafío encontrado es que las versiones del sistema que se encontraban en producción, y en la rama principal del GitHub del ADS, eran distintas. Siendo más específico, en la versión que se encuentra en GitHub hay modelos y funcionalidades incompletas, que no aparecen en el sitio en producción. Éstas apuntaban a la búsqueda de publicaciones por "áreas de trabajo" de cada autor del DCC, y estadísticas sobre las publicaciones en dichas áreas. Dada esta situación, se considera que la mejor opción es completar estas funcionalidades, y agregarlas a la nueva versión de U-Papers como otra de sus extensiones.

2.5. Extensiones al Sistema U-Papers

Las extensiones formalmente comprometidas y realizadas al sistema U-Papers son las siguientes:

1. *Desarrollo de una API, con endpoints correspondientes, que permita solicitar publicaciones por autor y fecha.* Con esta API se espera facilitar el uso de U-Papers por parte de otros sistemas, es decir, mejorar su integración con el ecosistema de software del DCC. Por ejemplo, el sitio web oficial del DCC cuenta con una sección dedicada a sus académicos donde, entre otras cosas, se muestran publicaciones de cada uno. Haciendo uso de esta API, se podrían mantener actualizadas de manera automática las publicaciones de cada docente. Esta extensión no presenta un gran impacto para el sistema actual de U-Papers, pues la API sería una adición externa y se haría uso de funcionalidades ya existentes para hacerla funcionar y conectarla.

2. *Actualización automática de fotografías de académicos.* El sitio web oficial del DCC cuenta con fotografías para todos los académicos del departamento, las cuales se van actualizando manualmente cada cierto tiempo. Las fotografías cuentan con un link único que no varía, aunque se haga un cambio de imagen. Por lo tanto, haciendo uso de este mismo enlace se podrían mantener actualizadas las fotografías de los académicos en U-Papers, sin mucho impacto en el código ya existente, simplemente cambiando las URL actuales de las fotografías por estos links permanentes.

3. *Autenticación de usuarios mediante credenciales de MiUChile.* Con esto se estaría cumpliendo uno de los requisitos de gobernabilidad e integración definidos por el ADS, pues se estaría pasando a utilizar el servicio de autenticación de usuarios interno de la universidad, además del proporcionado por el framework CodeIgniter. Esta extensión implica un impacto moderado en el sistema, pues probablemente se requiere intervenir el modelo de datos del sistema para poder integrar los usuarios de UPasaporte, al sistema de U-Papers.

Capítulo 3

Ajustes al Sistema U-Papers

Con el objetivo de mejorar la gobernabilidad y capacidad de integración de U-Papers, se realizaron ajustes al sistema legado para garantizar el adecuado desempeño de las futuras extensiones. Estos ajustes suponen la actualización y adaptación del código fuente del sistema.

3.1. Actualización del framework

El primer ajuste realizado es la actualización del framework del código fuente. Como se mencionó en la sección anterior, este se encuentra en la versión 2.0.2, pero CodeIgniter 2 ya no cuenta con soporte oficial y trabaja con versiones de PHP y MySQL que tampoco lo tienen. Para actualizarlo se recurre a la documentación del *framework*[4]. La versión estable más actual a la fecha de realización de este proyecto es la 3.1.13, por lo que se determina llegar hasta esta en el proceso de *upgrading*.

El primer paso para pasar de una versión a la siguiente es reemplazar la carpeta "*system*" por la de la nueva versión. Esta carpeta contiene los archivos que controlan aspectos generales del sistema que no cambian mucho de un proyecto a otro. En el caso de U-Papers, esta carpeta no tenía modificaciones previas, por lo que se reemplazaron todos sus archivos por los nuevos obtenidos de la página oficial.

Las siguientes modificaciones se pueden agrupar en las siguientes categorías: **cambios de notación, mover archivos entre carpetas, modificación en el uso de librerías y reemplazo de librerías**. Para la primera categoría se modifican los nombres de los archivos de clases, donde todas las librerías, controladores y modelos de datos pasan a tener nuevas formas de escribir sus nombres. También hay cambios en los nombres de algunas variables, como "*active_record*" (que se encarga de dar permiso de hacer modificaciones a la base de datos), que pasa a llamarse "*query_builder*". Finalmente, la notación para las llaves de los diccionarios de idioma de la librería "Form Validation" es modificada agregando el prefijo "*form_validation_*" a todas las llaves.

La siguiente categoría considera todos los archivos que deben ser cambiados de directorios por disposición del *framework*. Entre ellos están los *overrides* y extensiones de la clase "Log", que pasan del directorio "*system/core*" a "*application/core*", así como también, los *templates* de error pasan del directorio "*application/errors*" a "*application/views/errors*".

Para las modificaciones en el uso de librerías, los cambios son más extensos, por lo que solo se mencionan los más relevantes. Entre ellos están: las modificaciones a la configuración de opciones de la librería "Session" (encargada de mantener el estado de los usuarios y seguir su actividad mientras navegan por el sitio), donde algunos ajustes fueron redefinidos y otros removidos por quedar obsoletos. También se hacen arreglos en algunas funciones de distintas clases (Config, Input, Session, URI) y librerías (Input, Cookie helper, Array helper) que cambiaron su retorno de *False* a *Null*.

Finalmente, varias librerías fueron reemplazadas o removidas por quedar obsoletas en CodeIgniter 3. Algunas son la librería "SHA1" cuyo uso es cubierto por la función nativa de PHP "sha1()". La librería "Encrypt" que es reemplazada por otra librería, "Encrytion". También, funciones de algunas librerías como "File Helper", "String Helper", "Form Helper" o "Email helper" fueron reemplazadas por funciones nativas de PHP que cumplen el mismo propósito.

3.2. Reemplazo de Matomo

El siguiente ajuste realizado es el reemplazo de Matomo[6] (antes llamado Piwik Analytics) por Google Analytics[7]. Ambas son plataformas de análisis web que entregan informes visuales sobre el tránsito de usuarios por el sitio web. Si bien, Matomo es un buen servicio ampliamente usado, la versión que se encuentra dentro de U-Papers en producción está desactualizada, tanto que tiene el nombre de Piwik, el cuál fue cambiado a Matomo en 2018.

El paso más obvio sería actualizar el servicio, pero esto implica hacer varios cambios a la base de datos que podrían perjudicar los mismos, por lo que antes de hacerlo se estudia la opción de Google Analytics, teniendo presente que al menos este debe ofrecer las mismas características de Matomo que se usan dentro de U-Papers. Estas son visualizaciones del número de visitas por país, gráficos del número de visitas mensuales por un rango de meses y un resumen con los promedios de estadísticas principales como el número de visitas, la duración de las visitas, acciones por visita, porcentaje de visitas que dejan el sitio luego de ver una página, entre algunas otras.

Se llega a la conclusión de que Google Analytics cumple con estas características mínimas. Además de esto, su instalación requiere una intervención mínima del sitio, limitándose a copiar y pegar unas líneas de código en la cabecera del sitio (junto con la activación de algunos servicios que ofrece Google, pero que no vale la pena detallar), no ocupa espacio en el servidor de U-Papers y no requiere mantención, pues es un servicio externo.

El único contra es que, a diferencia de Matomo, las estadísticas no pueden ser visualizadas dentro de la interfaz de U-Papers, pero estas gráficas no son relevantes para los usuarios casuales o usuarios autores, y basta con que el administrador tenga acceso a estas dentro de Google Analytics, por lo que quitarlas de U-Papers no supone un problema para el proceso soportado por el sistema.

Finalmente, se quitan las estadísticas entregadas por Matomo del sitio, se elimina la carpeta que lo contiene junto las tablas que ocupa dentro de la base de datos y se integra Google

Analytics al sistema. En resumen, Matomo requiere una actualización con una fuerte intervención en los datos del sitio, ocupa espacio en el servidor y necesita mantención, mientras que Google Analytics requiere una instalación simple, no ocupa espacio en el servidor, no requiere mantención y por sobre todo cumple las mismas características relevantes de Matomo, por lo que el reemplazo es una opción más que viable.

3.3. Eliminación de búsqueda por mes

Otro ajuste realizado al sistema es la eliminación del mes de publicación de un *paper* como filtro de búsqueda. Tanto el formulario de búsqueda avanzada, como el de búsqueda por fecha, utilizaban el atributo de mes de publicación como un filtro para las consultas a la base de datos.

A simple vista parece ser una opción natural, pero al crear una publicación, el mes en que esta fue publicada no es un campo obligatorio, lo que resulta en que muchas publicaciones dentro de la base de datos tienen su mes como "*Null*". Esto produce fallas en los resultados de las búsquedas, pues, por cómo se definieron las consultas en el modelo de datos de las publicaciones, siempre va a buscar dentro de un rango de fechas, pero como hay publicaciones que no cuentan con el mes para comparar, no las toma en cuenta en la búsqueda, incluso cuando no se especifica un mes para buscar.

Este problema fue identificado mientras se extendía una funcionalidad de búsqueda por área de investigación. Se intentaba que la búsqueda arrojase una publicación en particular filtrando por el campo de "área de investigación", pero no se mostraba ningún resultado. Al intentar resolver este problema se da con este *bug*, pues la publicación que se buscaba era justo una que tenía su campo de "mes" como "*Null*". La consulta a la base de datos arrojaba un resultado vacío, pues no había una publicación que calzara dentro del rango de meses establecido.

Debido a lo anterior, se decide eliminar el mes de publicación de los filtros de búsqueda, pues es la solución que requiere menos impacto en el código fuente, y se evita la aparición de posibles nuevos errores, pues usar un campo no obligatorio como filtro de búsqueda no es una buena práctica.

3.4. Empaquetamiento del sistema en Docker

El siguiente ajuste realizado al sistema es empaquetarlo usando Docker. El objetivo principal de este ajuste es simplificar los pasos necesarios para levantar el sistema en un ambiente de desarrollo y así facilitar futuras implementaciones. Con esto también es más directo cumplir uno de los lineamientos establecidos por la ADS, el cual indica que el *deployment* del sistema debe estar descrito en función de los pasos y requerimientos que se deben seguir para levantarlo.

La estructura del sistema dentro de Docker se compone de contenedores que albergan las distintas imágenes de cada componente del sistema. En un principio se definen dos contenedores: la aplicación U-Papers y la base de datos. Cada imagen se levanta por separado y sus contenedores se conectan dentro de Docker para compartir información entre los controlado-

res, vistas y la base de datos del sistema. Cada una cuenta con su respectivo "*Dockerfile*" que contiene las instrucciones de instalación para las tecnologías necesarias, así como un archivo de "*docker-compose*" dentro del cual se unifican las instrucciones para construir todas las imágenes.

Más adelante, al momento de implementar las extensiones, se agrega un tercer contenedor para la API de datos, el cual cuenta con su "*Dockerfile*" y también se conecta a la base de datos, para que la API pueda extraer la información que necesite.

Como Docker es un ambiente aislado dentro del servidor, cuenta con un archivo "*.env*" donde se definen todas las variables de ambiente que el sistema utiliza, como las credenciales de la base de datos o los identificadores que la API necesita para conectarse con el Portal DCC.

Lamentablemente, al momento en que se escribe este documento, aún no es posible levantar el sistema de forma integral dentro de Docker. Si bien, la estructura necesaria para crear las imágenes y levantar los contenedores está correctamente definida, existe un error que impide que el contenedor de U-Papers y el de la base de datos se conecten (esto está actualmente en camino a una solución). Este error hace que el sistema completo no se termine de levantar adecuadamente, aun cuando los contenedores por separado se levantan y funcionan correctamente. A pesar de esto, se considera necesario mencionar los ajustes realizados en esta línea, pues una vez resuelto este error, el sistema debería funcionar correctamente dentro de Docker, y así cumplir con uno de los requisitos definidos por el ADS.

Capítulo 4

Extensión al Sistema U-Papers

Además de los ajustes antes indicados para mejorar la gobernabilidad y capacidad de integración de U-Papers, y a posteriori de dichos ajustes, se extendió el sistema con el propósito de poder verificar si efectivamente se habían mejorado dichos aspectos. Por otra parte, es importante destacar que no se requirieron ajustes al proceso de trabajo (workflow) soportado por U-Papers.

4.1. Tablero de Producto

A continuación, se muestra el tablero de la nueva versión del producto (objetivo y alcance), es decir, el sistema luego de haber sido intervenido y extendido.

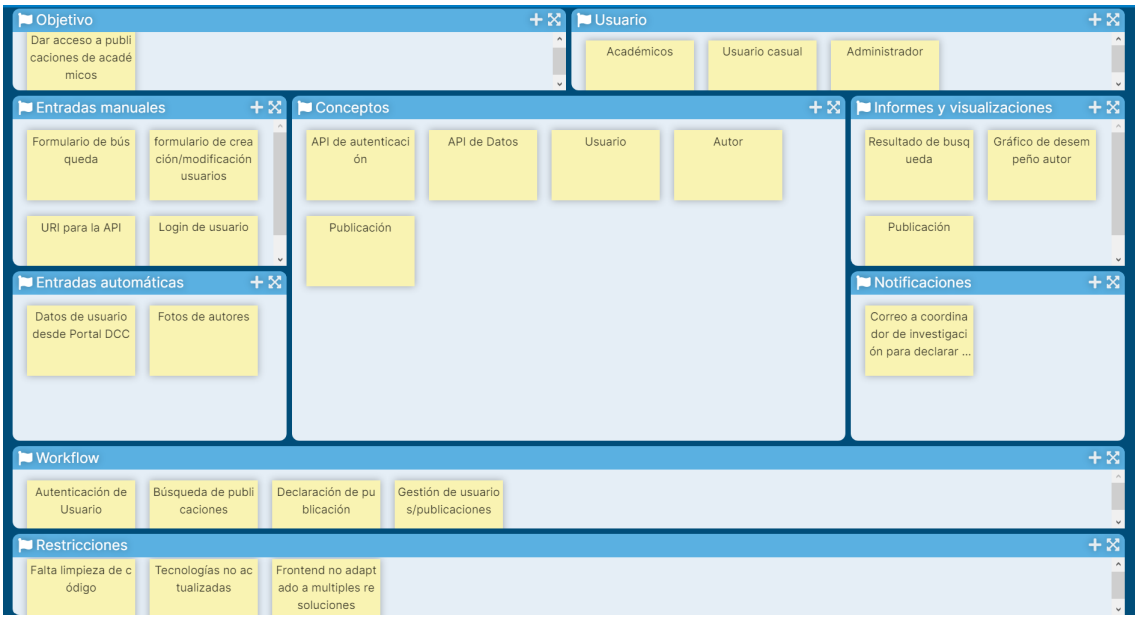

Figura 4.1: Tablero de producto del nuevo sistema

Dentro del tablero, el **objetivo** presenta el propósito fundamental de U-Papers, el cual es dar acceso de forma pública y gratuita a las publicaciones de académicos hechas dentro del contexto del DCC, pues en su concepción, el sistema fue pensado como una forma de centralizar el conocimiento e investigaciones desarrolladas por académicos del DCC. Por lo mismo, sus **usuarios** son los académicos que aportan el contenido al sitio, los denominados

"usuarios casuales" que no tienen una cuenta registrada, buscan fuentes y conocimiento, así como también los administradores (superadministrador y administrador), encargados de gestionar el sitio, sus usuarios y publicaciones.

Las **entradas manuales** corresponden a los datos necesarios para dar sentido a los conceptos y son capturados con intervención humana. En este caso, los formularios de búsqueda, declaración y creación/modificación de publicaciones dan sentido a estas. También están los formularios de creación y modificación de usuarios que usa el superadministrador. Para acceder a otras funcionalidades del sitio, se tiene el formulario de *log in* que ya existía en el sistema legado con el cual los usuarios pueden acceder a sus roles. Finalmente, el ingreso de las URI para los endpoints de la API de datos también es una entrada manual.

Respecto de las **entradas automáticas**, estas corresponden a los datos que requiere un concepto, pero que son obtenidos de forma automática desde un servicio externo o sistema dentro de la organización (DCC). Para U-Papers estos son los datos de usuario que se obtienen desde el Portal DCC para hacer la autenticación de usuario con credenciales MiUchile, los cuales son enviados desde el Portal DCC a través de una *request*. También se puede destacar las fotografías de los académicos que se obtienen directamente del sitio oficial del DCC, con el objetivo de mantenerlas actualizadas.

Los ya mencionados **conceptos** son la unidad básica a través de la que se construye el sitio. Los usuarios y publicaciones son los objetos en torno a los que gira todo el sistema, mientras que autor es un concepto que los enlaza, pues autor es un usuario que tiene publicaciones a su nombre. Las API son nuevas unidades que funcionan de forma paralela al sistema legado, que permiten interactuar de forma indirecta con la información del sitio.

Los **informes y visualizaciones** son formas de explorar los conceptos para tomar decisiones informadas respecto de estos. Con los resultados de búsqueda de publicaciones es posible categorizar estas en función a distintos factores como autor, fecha, tipo y área de investigación. Al ingresar a la vista por publicación, se puede saber cuántas veces ha sido descargada, mientras que al ingresar a la vista del perfil de un autor se puede ver un gráfico sobre la cantidad de publicaciones que ha declarado por año.

Las **notificaciones** son eventos que se disparan cuando ocurre algún evento relevante para el sistema que debe ser notado rápidamente. En este caso, cuando un autor declara una publicación que desea subir a U-Papers, llega un correo al encargado de revisarlas (superadministrador) con la información entregada por el autor.

El anterior es uno de los procesos que forman parte del **workflow**, el cual hace referencia a una representación de la automatización de los procesos que se desarrollan dentro del sistema. Para U-Papers el flujo parte cuando el usuario entra al sitio, este puede o no autenticarse y podrá buscar publicaciones. Si decide iniciar sesión, dependiendo de su rol dentro del sistema puede declarar publicaciones (autor) o gestionar publicaciones y usuarios (administrador).

Finalmente, las **restricciones** son aspectos del sistema o su entorno que pueden afectar la automatización de los procesos. En el caso de este sistema el front-end no se adapta a distintas resoluciones de pantalla, en especial para dispositivos móviles, lo que puede complicar la

navegación y, por ende, dificultar el workflow del mismo. Por otro lado, el código fuente no está en las condiciones óptimas de limpieza y refactorización, por lo que trabajar en él puede ser un desafío, y peor aún, las tecnologías que usa el sistema no son las más actuales y pueden llegar a estar deprecadas lo que puede generar problemas de compatibilidad.

4.2. Arquitectura del Sistema Extendido

A continuación se presenta el estado de la arquitectura del sistema y el modelo de datos después de haber sido extendido.

4.2.1. Arquitectura del Sistema

Al igual que en el análisis de la sección 2, se hace uso del modelo C4 para especificar distintos niveles de abstracción en la arquitectura. A continuación, se presenta la arquitectura de ecosistema y de los macro componentes de U-Papers luego de haber sido extendido.

4.2.1.1. Diagrama de contexto del ecosistema

Como se aprecia en la Figura 4.2, el sistema obtuvo la capacidad de generar nuevas interacciones con elementos de su ecosistema. Por un lado, gracias a la API de datos ahora sistemas externos como el sitio web del DCC u otros sistemas y personas pueden extraer información sobre publicaciones directamente desde la base de datos, lo que facilita el flujo de información.

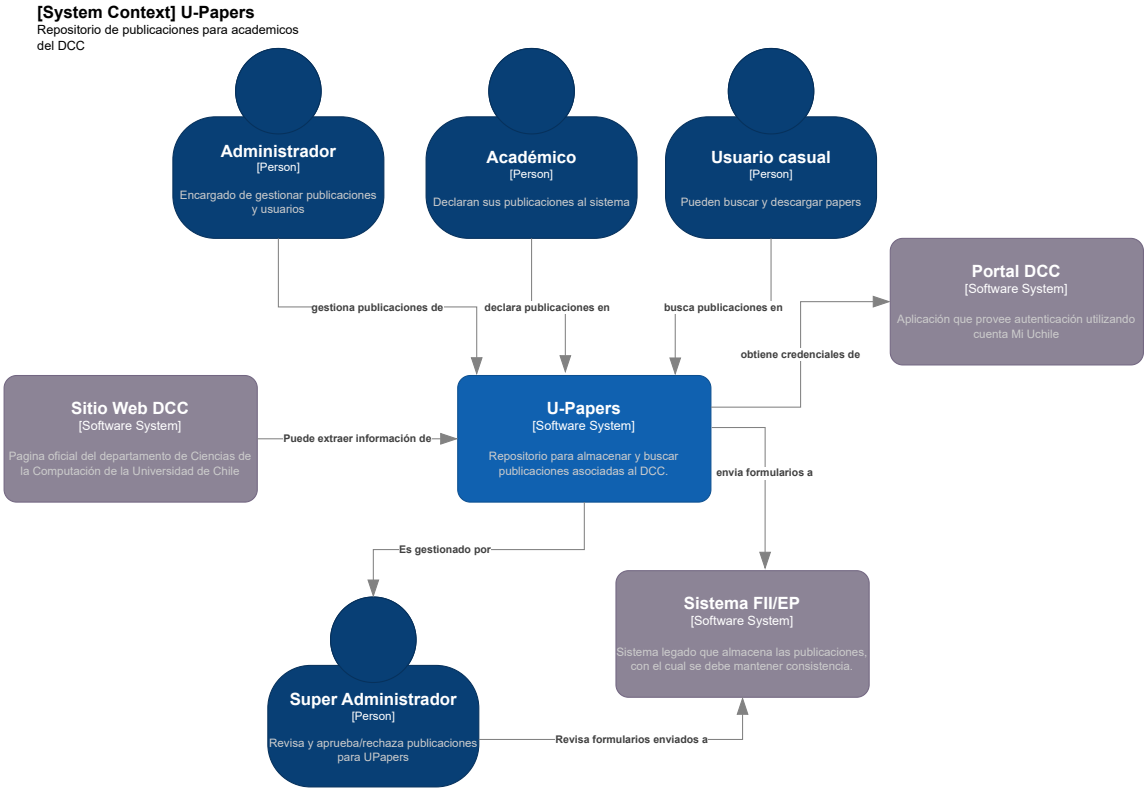

Figura 4.2: Nuevo modelo de contexto del sistema en C4

A su vez, U-Papers ahora puede redirigir a sus usuarios al Portal DCC, donde ingresando sus credenciales MiUchile pueden autenticarse e iniciar su sesión. Más allá de los cambios

y mejoras antes mencionados, la arquitectura del ecosistema se mantuvo intacta como se explicó en la sección 2 de este documento.

4.2.1.2. Diagrama de containers del sistema

Para el interior del sistema U-Papers se puede notar que además de la aplicación web, controlador y base de datos que se explicaron en la sección 2, ahora cuenta con una API desarrollada con el *framework* FastAPI[8] de Python (Fig. 4.3). Ésta interactúa directamente con la base de datos haciendo consultas personalizables desde sus endpoints, lo que permite que otros sistemas extraigan información.

Además de esto, el controlador de U-Papers es capaz de interactuar con el Portal DCC para obtener autenticaciones de usuario con credenciales MiUchile y así iniciar sesión dentro de U-Papers para usuarios registrados.

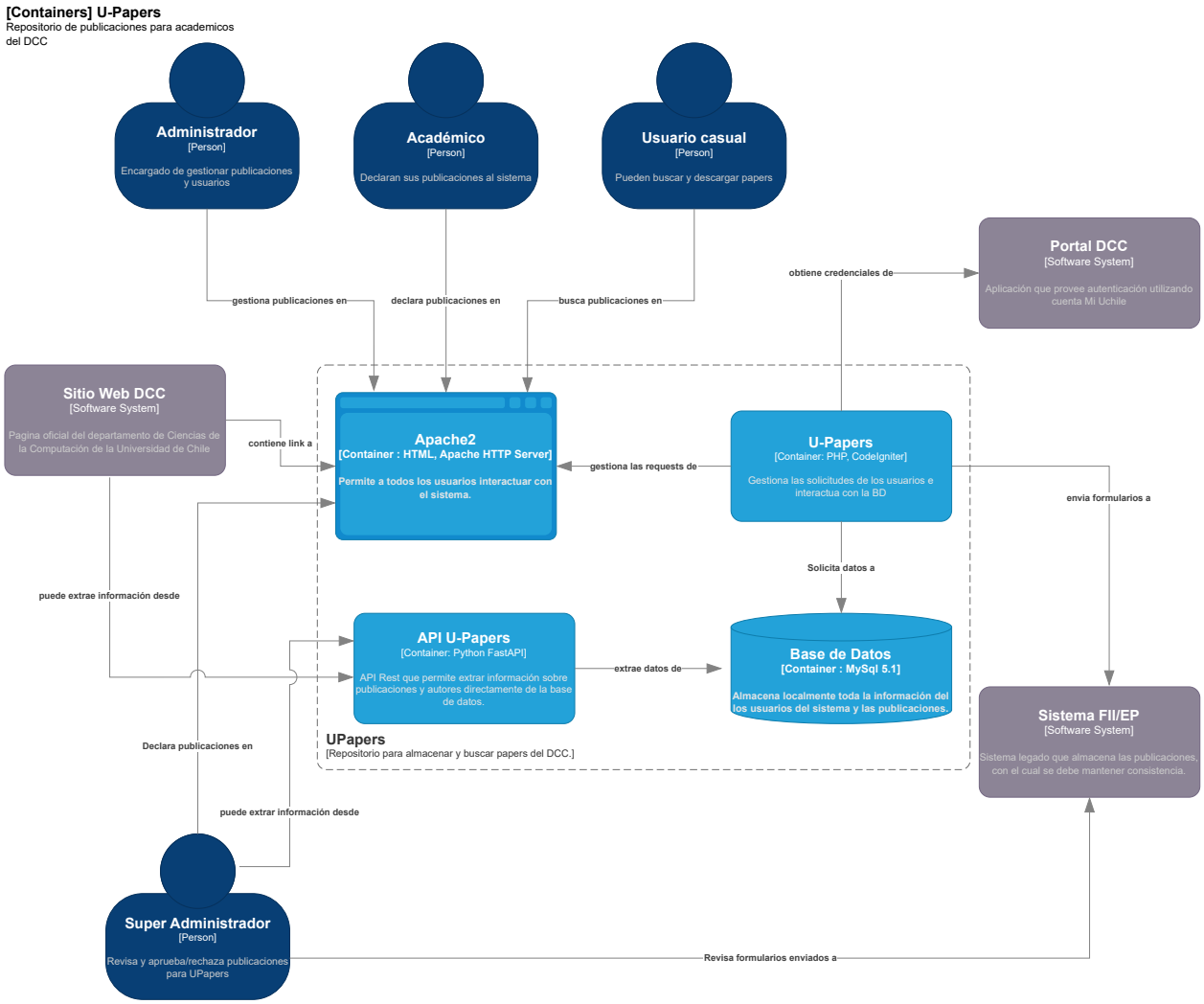

Figura 4.3: Nuevo modelo de contenedores del sistema en C4

Sin contar los cambios mencionados anteriormente, la arquitectura de los contenedores de U-Papers se mantuvo idéntica a lo explicado en la sección 2 de este documento.

4.2.2. Modelo de Datos

Al igual que con la arquitectura, el modelo de datos se mantuvo intacto salvo por la inclusión de nuevas tablas necesarias para la integración de algunas extensiones. Estas tablas son “areas”, que contiene una lista con las áreas de investigación que se desarrollan dentro del DCC (Fig. 4.4).

Tanto las publicaciones como los autores pueden tener áreas de investigación asociadas, por lo que se crearon las tablas “publications_has_areas” y “authors_has_areas” como nexo entre la tabla “areas” con las tablas “publications” y “authors” respectivamente. Esto fue incluido para dar coherencia a algunas funcionalidades que se encontraban incompletas que serán explicadas más adelante.

Además de estas tablas nuevas, se añadió el campo “pasaporte” a la tabla “users” como uno de los pasos necesarios para implementar la autenticación a través del Portal DCC. Este campo contiene el nombre de usuario de las credenciales MiUChile de los usuarios de U-Papers.

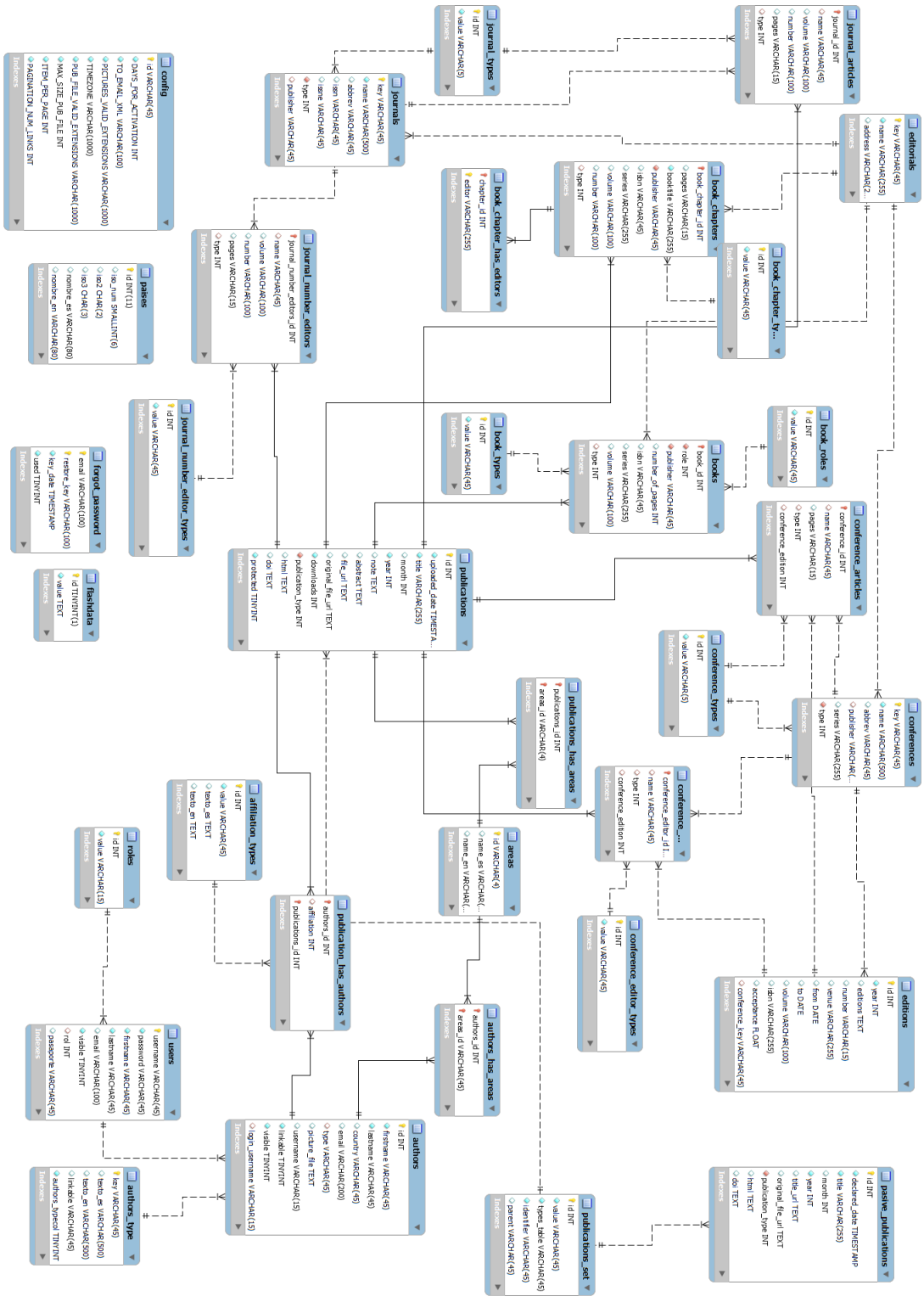

Figura 4.4: Nuevo modelo de datos del sistema U-Papers.

4.3. Inclusión de Autenticación de Usuarios con UPasaporte

Esta extensión es uno de los requerimientos fijados por el Área de desarrollo de Software del DCC, donde se especifica que "Debe utilizar los servicios que provee el DCC. Se debe implementar un esquema propio de roles y permisos.". Para cumplir la primera parte del requerimiento se usa el servicio que provee el Portal DCC para autenticar usuarios con las credenciales de sus cuentas MiUchile. Mientras que, para la segunda parte, se usa el esquema de roles y permisos ya existente dentro de U-Papers, por lo que cuenta como uno propio.

Con el objetivo de implementar esta nueva autenticación de usuarios en U-Papers, los pasos fueron los siguientes:

1. **Crear un nuevo botón de Log in.** Se crea un nuevo botón justo encima del antiguo Log in de U-Papers. La función de este es redireccionar al usuario al Portal DCC. Para lograr esto al presionar el botón en realidad el usuario es redireccionado a un controlador creado específicamente para esta nueva funcionalidad, el cual es llamado "sso".

2. **Generar redirección a Portal DCC.** Dentro del controlador "sso", su función "**index()**" es llamada automáticamente. La única misión de esta es generar el verdadero redireccionamiento al Portal DCC, usando la función de PHP "redirect(URL)".

3. **Capturar respuesta de Portal DCC.** La razón por la que el controlador es llamado "sso" es que el Portal DCC siempre busca el endpoint "*https://<url/ip>/**sso/login***" dentro del sistema que hizo contacto con él. Por ende, luego de que el usuario ingrese sus credenciales MiUchile dentro del Portal este será redireccionado nuevamente al sitio de origen (U-Papers en este caso) junto con un diccionario compuesto por su nombre de usuario (username) y una variable llamada "secret". Para poder capturar esta respuesta, se define la función "**login()**" dentro del controlador "sso", la que además se encarga de validar los datos recibidos.

4. **Validar datos de usuario.** Una vez dentro de la función "login()" esta toma los datos recibidos (username y secret) y consulta al Portal DCC si los datos recibidos son válidos, es decir, si las credenciales ingresadas anteriormente se corresponden con un usuario de MiUchile. Si la respuesta es no, entonces la función no inicia ninguna sesión y muestra un mensaje al usuario indicando que no ingresó correctamente sus credenciales. Por otro lado, si las credenciales son válidas, entonces la función comprueba que dentro de la tabla "users" en la base de datos de U-Papes existe algún usuario con el *username* enviado desde el Portal DCC. Si existe entonces inicia su sesión. En caso contrario no inicia sesión y vuelve a mostrar un mensaje indicando que el usuario no ingresó credenciales válidas.

5. **Modificar modelo de datos.** Finalmente, para que todo lo anterior tenga sentido dentro del modelo de datos de U-Papers, se añadió el campo "pasaporte" a la tabla "users" el cual debe contener el "username" de las credenciales MiUchile del usuario o "Null" en caso de no poseer dichas credenciales. Además, se define la función "**get_user_by_pasaporte**" dentro del archivo "users_model" (el modelo de datos de la tabla "users") la cual extrae de la base de datos todos los datos del usuario cuyo campo "pasaporte" coincida con el username buscado.

Para mantener esta nueva lógica en el futuro, es necesario modificar algunos formularios dentro de U-Papers. Estos son los formularios de creación y modificación de usuarios, a los cuales se les agrega el campo "**Usuario MiUchile**". Con esto se consigue que para futuras inserciones de usuarios los administradores puedan ingresar de inmediato el "username" de las credenciales MiUChile en caso de que el nuevo usuario las posea.

Es importante destacar que los usuarios pueden usar el antiguo Log in para autenticarse en el sistema, ya que este no fue tocado durante este proceso, pues como se menciona al principio de esta sección, no se hicieron cambios a procesos ya soportados por U-Papers.

4.4. Descripción de la API de U-Papers

Esta API es otro de los requerimientos definidos por el ADS, el cual especifica que "Debe implementarse una API a través de la cual otros sistemas puedan acceder a sus servicios o datos. Esta debe estar apropiadamente documentada en Swagger".

El paso más obvio para cumplir este requisito fue encontrar algún framework para APIs con el cual construirla y se barajaron 2 opciones: buscar algún framework asociado a CodeIgniter para mantener las tecnologías en la misma línea que venían o usar FastAPI, un conocido framework para desarrollar REST APIs basado en Python. Si bien usar uno en PHP asociado a CodeIgniter parece un camino más natural, intervenir el código fuente directamente incluye riesgos de seguridad que no es necesario tomar en este caso.

FastAPI es ampliamente usado y especializado en el desarrollo de APIs, lo que tiene garantías de seguridad que otras opciones no. Además, no se tendría que intervenir el código, pues trabajaría en paralelo a los procesos de U-Papers, lo cual es más seguro. Por último, se tomó en cuenta que el requerimiento deja claro que la API debe ser documentada en Swagger y FastAPI cuenta con documentación automática en este formato, por lo que termina siendo la mejor opción para este desarrollo.

Entregando las credenciales correctas, la API permite una conexión directa con la base de datos de U-Papers, mientras que corre dentro de su propio servidor local llamado *Uvicorn*[9], el cual es un servidor web implementado en Python.

Tomando en cuenta los datos presentes dentro de U-Papers, que serían publicaciones y sus autores, se concluye que los usuarios de la API deberían poder extraer información sobre autores para conseguir sus "id", la lista de publicaciones asociadas a un autor y la información de una publicación en particular. Por esto se definen 3 endpoints para la API:

1. **/api/authors:** Este endpoint no recibe parámetros. Su función es entregar la lista de autores dentro del sistema. Mostrando solamente los valores necesarios para que el usuario los identifique (nombre y apellido), junto a su "id", el cual puede ser usado en el siguiente endpoint. La consulta es simple:

```
1    authors.select(
2   ).order_by(authors.c.linkable.desc())
3
```

Con esto selecciona todas las columnas dentro de la tabla “authors” y retorna una lista de los autores ordenados por el campo “linkable”, el cual es un atributo binario que indica si el autor tiene un perfil al cual acceder. Esto es así para darle prioridad dentro de la lista a los autores que son académicos del DCC. Más adelante, haciendo uso de una función auxiliar, se filtran los datos que se entregarán como respuesta del endpoint, siendo estos “id”, nombre (firstname) y apellido (lastname).

2. **/api/author/{author_id}:** El siguiente endpoint recibe como parámetro {author_id}, que corresponde al id de un autor. Su función es entregar la lista de publicaciones asociadas a un autor, donde se muestran todos los datos asociados a cada publicación dentro de la tabla “publications”. La consulta es la siguiente:

```
    select([publications]).where(
      publications.c.id == publications_has_author.c.publications_id
      ).where(
         publications_has_author.c.authors_id == author_id
      ).order_by(
         publications.c.year.desc()
      ).order_by(
         publications.c.month.desc())

```

Para entender esta consulta se debe partir desde el final. Primero, se ordena la tabla “publications” por año y por mes de publicación. Luego, dentro de la tabla “publications_has_author” filtra todas aquellas filas en las que el valor de la columna “authors_id” sea igual al valor de “author_id” (input del endpoint). Finalmente, teniendo las filas filtradas, muestra todas las columnas de la tabla “publications” y selecciona solo las filas en las que el “id” sea igual al valor de alguna fila de la columna “publications_id” de la tabla “publications_has_authors”. Con esto se consigue una lista con todas las publicaciones de un autor.

3. **/api/publication/{pub_id}:** Este último endpoint recibe como parámetro el identificador de alguna publicación {pub_id}. Su función es entregar todos los datos asociados a una única publicación dentro de la tabla “publications”. Su consulta también es simple:

```
      publications.select(
      ).where(publications.c.id == pub_id)

```

Esta consulta selecciona todas las columnas de la tabla “publications”, en cuya fila el “id” sea igual a pub_id (input del endpoint).

Para dar sentido a las consultas necesarias para la API, el modelo de datos fue basado en 3 tablas de la base de datos de U-Papers, estas son “publications”, que contienen todas las publicaciones del sistema; “authors”, que contiene todos los autores del sistema y “publications_has_authors” que asocia cada publicación con uno o más autores.

4.5. Incorporación de funcionalidades "legadas"

Como se mencionó en la sección 2.4, la versión del código fuente de U-Papers presente en el repositorio oficial del Área de Desarrollo de Software es la que se usó para trabajar este proyecto, la cual es distinta a la versión que se encuentra en uso en producción. Esto se debe a que otros estudiantes fueron modificando el código fuente en sus respectivos trabajos de título, agregando cambios y funcionalidades poco a poco.

Estos cambios quedaron grabados en distintas ramas del repositorios, por lo que para simplificar el trabajo de elegir cual usar, se opta por usar como base la que está por defecto como la rama principal. Está rama contiene algunas funcionalidades incompletas; a continuación se profundizará en las que fueron completadas y agregadas como parte de las extensiones al sistema.

1. **Búsqueda de publicaciones por área de investigación.**

   Esta fue la primera funcionalidad en ser completada. Su modelo de datos estaba completo, pues ya existía el modelo para la tabla "areas", la cual debería contener todas las áreas de investigación del DCC y también estaba el modelo para la tabla "publications_has_areas" la cual asocia las áreas de investigación con las publicaciones.

   Las tablas asociadas a cada modelo no existían, por lo que se procede a crearlas y agregarlas a la base de datos. La tabla "areas" contiene 3 columnas: "id" (identificador único de cada elemento), "name_es" (nombre de cada área en español) y "name_en" (nombre de cada área en inglés). Con estas 3 columnas, las funciones definidas en su modelo de datos cobran sentido. Para la tabla "publications_has_areas" se definen 2 columnas: "publications_id" (identificador de la publicación) y "areas_id" (identificador del área asociada).

   En lo que respecta a la funcionalidad de búsqueda, esta se implementa en el formulario de búsqueda avanzada. El campo ya definido en la vista, pero se encontraba comentado, por lo que se descomenta. Al comienzo se prueba buscando algunas publicaciones por su área de investigación y la búsqueda no arroja resultados cuando sí debería haberlo hecho. Sin embargo, como se explicó en la sección 3.3 existía un *bug* relacionado a la búsqueda por mes de publicación, el cual era la verdadera fuente del problema con esta nueva funcionalidad. Una vez resuelto el *bug*, las búsquedas por área de investigación comienzan a arrojar los resultados esperados.

2. **Agregar área de investigación a autores.**

   Al igual que la funcionalidad anterior, los modelos de datos necesarios estaban creados: "areas" y "authors_has_areas", pero la tabla "authors_has_areas" no existía, por lo que se procede a agregarla. Esta tabla contiene 2 columnas: "authors_id" (identificador de cada autor) y "areas_id" (identificador de cada área de investigación). Con respecto al modelo de datos, este fue pensado para que cada autor tenga solo un área de investigación asociada, pero se considera que este no es un acercamiento acertado, pues en el sitio oficial del DCC se puede ver que cada académico es afín a 2 o más áreas de investigación.

Por esto, se modifica el modelo de "authors_has_areas" para que cada autor pueda tener más de un área asociada.

La vista para los formularios de crear y modificar autores usa un solo formulario como molde, lo cual simplifica el trabajo. En este formulario el campo de "área de investigación" se encuentra creado, pero es una caja de selección única, por lo que se procede a modificarla para que permita seleccionar múltiples áreas a la vez. Esto es muy similar a la funcionalidad anterior, solo que el procesamiento posterior de los datos seleccionados es distinto, pero la vista es la misma, por lo que se decide usar como molde la caja de selección múltiple que existe en el formulario de búsqueda avanzada. Con esto se consigue al crear o modificar un autor, se puedan seleccionar varías áreas para asignarle.

Ahora solo falta modificar la forma en cómo se procesan los datos que se ingresen en este nuevo campo del formulario. Antes se enviaba un solo dato que contenía el "id" del área de investigación seleccionado, pero como ahora se puede elegir más de una, el input obtenido desde la vista es una lista de "id". Además de esto, se debe tener en cuenta que no solo se pueden agregar nuevas áreas, pues al modificar un autor ya existente, el usuario podría eliminar un área ya asignada o cambiarlas todas. Por esto, cuando se recibe la lista de inputs desde la vista, ésta se compara con la lista de áreas asociadas que tenga el autor que está siendo modificado (obtenida desde la base de datos). Si la lista de la base de datos está vacía, entonces directamente se añaden todas las áreas presentes en la lista obtenida de la vista. En caso contrario, se deben revisar 2 casos:

a) *que se haya eliminado alguna de las áreas que el autor tenía asociadas*

b) *que se haya agregado algún área que antes no estaba asociada.*

Para lograr esto, se van comparando ambas listas. En el primer caso se recorre la lista obtenida desde la base de datos. Si hay algún área en esta que no esté en la lista obtenida desde la vista, implica que el usuario quiere eliminar estas áreas, por lo que se procede a eliminar directamente de la tabla "authors_has_areas" las filas que contengan al autor que se está modificando y las áreas que se desean eliminar. Para el segundo caso, se recorre la lista obtenida desde la vista. Si esta tiene áreas que no están en la lista obtenida de la base datos, implica que el usuario desea asociar estas nuevas áreas al autor, por lo que se agregan nuevas filas a tabla "areas_has_authors" que asocien al autor con estas áreas. Una vez recorridas ambas listas, se añadieron todas las nuevas áreas seleccionadas y se quitaron todas las áreas que se hayan deseleccionado en la vista.

3. **Gráfico de desempeño para autores DCC.**

Esta es la única funcionalidad agregada que se considera estaba completa sin necesidad de intervención. Consiste en gráficos que muestran la evolución en la cantidad de publicaciones que cada autor registra por año. Son interactivos pues se puede modificar el tipo de publicación que se tiene en cuenta para graficar y también los años que se consideran. La evolución de cada autor se compara con la evolución del promedio de todos los autores. Esta funcionalidad aplica solo para autores que son académicos del DCC y

se mantiene pues se considera que puede ser una herramienta útil para los autores. A continuación, se muestra un ejemplo del perfil de uno de los autores.

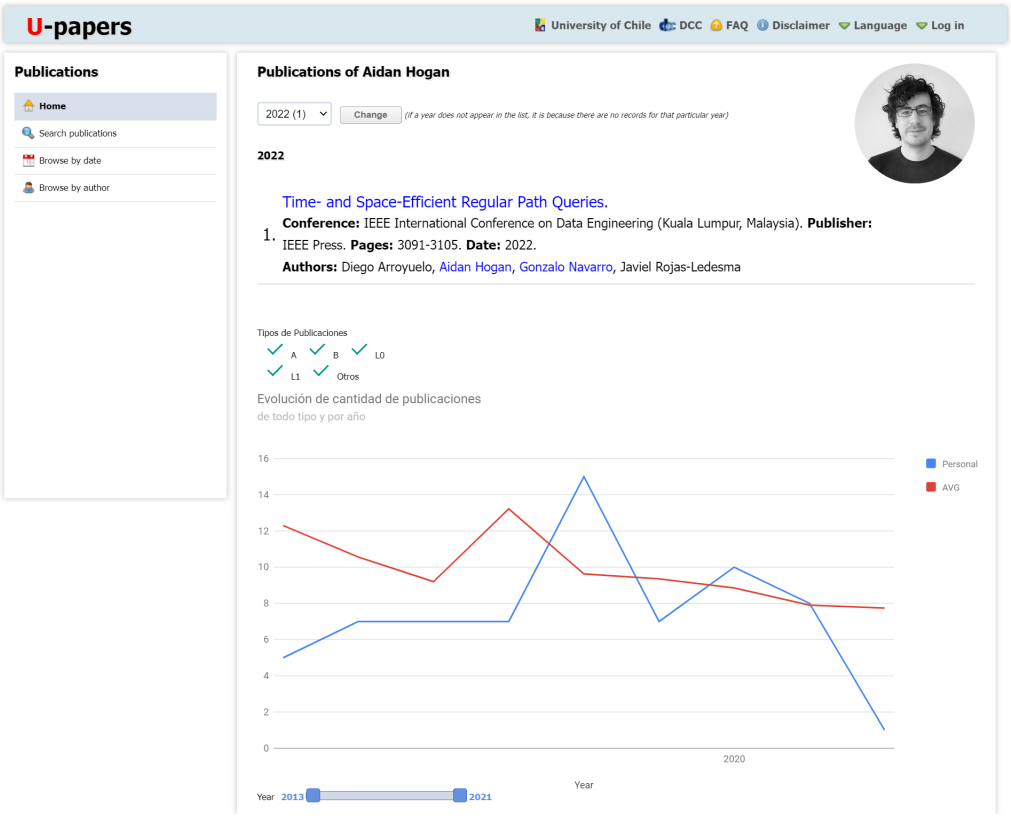

Figura 4.5: Perfil de publicaciones de un académico comparado con el promedio del DCC.

4.6. Actualización automática de fotografías de autores DCC

La idea de esta extensión nace del hecho que las fotografías usadas en U-Papers para los académicos del DCC no sólo están desactualizadas, sino que además nadie se encarga de mantenerlas actualizadas. Dentro del repositorio de papers, éstas no han sido cambiadas en al menos 7 años. Una posible razón para esto es la implementación que se hizo en su momento para añadirlas al sitio. Éstas se encuentran almacenadas dentro del sistema en el servidor (no dentro de la base de datos) dentro de una carpeta llamada "pictures".

Cada imagen es un archivo jpg que tiene por nombre las iniciales del académico al que corresponde. En el archivo "config.php" del sistema se encuentra definida una variable global llamada "**PICTURES_FOLDER**" la cual almacena un string con la ruta a la carpeta "pictures". Además, dentro de la tabla "authors" en la base de datos existe el campo "**picture_file**", para académicos del DCC contiene el nombre de archivo de la imagen que le corresponde. Con todo esto, cuando se necesita usar la imagen en alguna vista, se llama a la variable "PICTURES_FOLDER" para obtener la ruta a la carpeta de las imágenes y se extrae el nombre del archivo desde la tabla "authors" para completar la ruta hasta la imagen solicitada.

Por todo esto, actualizar las imágenes implica cambiarlas dentro de la del servidor y asegurarse de que su nombre coincida con el que aparece en la base de datos. Son muchos pasos para obtener la imagen, pero se decidió usar esta estructura como base para implementar la extensión, y ahí evitar modificar innecesariamente el código fuente.

El primer paso de la extensión es descartar la carpeta "pictures", pues se usarán directamente las fotografías de académicos obtenidas desde el sitio web oficial del DCC. Las imágenes usadas dentro de este sitio tienen una URL única que no cambia aún si cambian la fotografía usada, por lo que son el elemento central de esta extensión. Como se usa la misma estructura de antes, la variable "PICTURES_FOLDER" seguirá siendo usada, para no quitarla en todas las líneas en que ya es usada, pero pasa a ser un string vacío "". Mientras que los URLs antes mencionados son ingresados a la tabla "authors" en el campo "picture_file" reemplazando el nombre del archivo en la carpeta "pictures".

De esta forma, desde ahora cuando se llama a la variable "PICTURES_FOLDER" y al valor de "picture_file" para crear la ruta a la imagen a usar, esta será el resultado de la concatenación de un string vacío y el URL a la imagen dentro del sitio web del DCC, obteniendo así la imagen deseada.

El resto de las modificaciones que se realizaron para esta extensión fueron pequeños cambios en el frontend con el objetivo de que las nuevas imágenes tuvieran el tamaño adecuado para cada vista.

4.7. Levantar ambiente de desarrollo

Como se mencionó en la sección 3.4, levantar un ambiente de desarrollo para U-Papers en Docker aún no es posible, si se logra levantar uno usando el paquete de software libre XAMPP[10]. Esto representa un gran paso para mejorar la gobernabilidad del sistema, pues disminuye considerablemente la cantidad de horas que se tendrían que dedicar a resolver un error o agregar una nueva funcionalidad.

Antes de definir los pasos para levantar este ambiente de desarrollo, modificar U-Papers era una tarea que requería una gran inversión de tiempo solo para comenzar a trabajar, pues no existía claridad sobre cómo poner en marcha el sistema. Ahora basta con cumplir los requerimientos tecnológicos y seguir los pasos preestablecidos.

El ambiente de desarrollo se sostiene sobre XAMPP y el uso de variables de entorno para conectar el sistema a la base de datos y al Portal DCC. Debido a que el *framework* no soporta el uso de archivos ".env" por defecto, se agrega esta funcionalidad cargando las librerías necesarias. Luego se crea el archivo ".env" y se define un conjunto de variables de ambiente (predeterminadas) que permiten levantar el ambiente de desarrollo casi automáticamente. Las instrucciones para realizar este *deployment* están disponibles en el README del repositorio de Github del proyecto.

A su vez, se definieron los pasos para levantar la API de acceso a datos dentro de este ambiente de desarrollo. Si bien, ésta se encuentra dentro del mismo servidor que U-Papers, funciona de forma independiente, a través de un servidor local que provee Python. Por lo que para levantarla se requiere crear un ambiente virtual, dentro del cual se instalan las librerías que la API necesita y se corre el servidor Uvicorn. Las instrucciones para el *deployment* de la API también se encuentran en el repositorio.

Capítulo 5

Evaluación del Trabajo Realizado

En relación al cumplimiento de los lineamientos de gobernabilidad e integración definidos para los sistemas del DCC, a continuación se indica el nivel asociado al sistema legado (Nivel Inicial), y el nivel de cumplimiento actual (Nivel Actual). Como se puede apreciar en la Tabla 5.1, a partir del trabajo realizado se ha mejorado la adherencia a los lineamientos definidos.

En cada fila de la Tabla 5.1 se puede apreciar cómo la intervención al sistema U-Papers generó un cambio con respecto a la versión legada. Para el punto 1, existe un diagrama en la memoria de Felipe Chacón[1] sobre cómo el sistema legado interactúa con el sistema FII/EP. Sin embargo, éste no considera todo el ecosistema del DCC, ni tampoco sigue las especificaciones del modelo C4. En la documentación actual del sistema se puede encontrar un diagrama de contexto del sistema en C4, el cual muestra las distintas interacciones entre U-Papers y otros sistemas dentro del ecosistema del DCC.

Con respecto al ítem 2 de la tabla, dentro de la misma memoria [1] existe un diagrama simple de la arquitectura del sistema legado que no está en C4, ni tampoco considera todos los posibles usuarios del sistema y sus interacciones. Por otro lado, en la versión actual se cuenta con un diagrama de componentes C4, que considera la arquitectura del sistema y cómo las distintas partes de esta (modelo-vista-controlador) interactúan con usuarios y otros sistemas.

En cuanto al modelo de datos (ítem 3), la documentación antigua cuenta con un diagrama de la primera versión de U-Papers, pero éste ya se encuentra desactualizado, pues no considera modificaciones e incorporaciones (nuevas tablas) que se han hecho al sistema durante los últimos años. Esta información sí se puede apreciar en el modelo actualizado.

El ítem 4 no se cumple en la versión legada, pues ésta no cuenta con una API que permita a otros sistemas obtener datos directamente de la base de datos de U-Papers. Mientras que la versión actual, el sistema cuenta con una API desarrollada en FastAPI, documentada en Swagger, que permite el acceso a los datos sobre publicaciones y autores dentro de U-Papers. Lo mismo ocurre con el ítem 5, donde en su estado inicial U-Papers no ofrece a sus usuarios la opción de autenticarse vía UPasaporte, y solo cuenta con un Login interno con sus propias credenciales. Mientras que, en su nueva versión, el sistema mantiene el Login interno y al mismo tiempo permite a sus usuarios iniciar sesión validando sus credenciales MiUchile contra UPasaporte dentro del Portal DCC.

Con respecto al ítem 6, el estado inicial no cumple, pues si bien hay un repositorio en Github con el código fuente, no se sabe si alguna de las ramas contiene la última versión oficial del sistema en producción, pues la rama que se encuentra definida como principal es distinta a la que está en producción. Por otro lado, el nuevo sistema cumple parcialmente con este ítem, pues al momento en que se escribe este documento, se sigue trabajando en la rama que pasará a ser la principal del repositorio.

El stack tecnológico (ítem 7) de la versión legada de U-Papers no está mencionado en ninguna documentación, y en su mayoría se encuentra obsoleto. Mientras que las tecnologías de la versión actualizada se encuentran indicadas dentro de la documentación del repositorio, y además fue actualizadas con respecto a sus versiones originales.

Con respecto al ítem 8, al ser U-Papers un sistema legado, el stack tecnológico usado no se puede alinear completamente con las definiciones impuestas por el DCC, pues no usa Python, ni Django en su implementación. Aun así, se cumple parcialmente con este requerimiento, pues la API está implementada con un framework de Python y también se sentaron las bases para levantar un ambiente de desarrollo en Docker.

Finalmente, con respecto al ítem 9, en su estado inicial U-Papers no cuenta con instrucciones sobre cómo levantar el sistema dentro de un ambiente de desarrollo, lo cual representó un verdadero desafío al inicio de este proyecto. Por otro lado, la versión actual cumple este requisito, ya que define y documenta las instrucciones y requerimientos tecnológicos necesarios para levantar el sistema en un ambiente de desarrollo en poco tiempo.

Tabla 5.1: Nivel de cumplimiento inicial y actual del sistema U-Papers, según los requerimientos de gobernabilidad definidos por el ADS

Ítem Evaluado	**Nivel inicial**	**Nivel Actual**
1. La arquitectura del ecosistema debe estar documentada y actualizada.	No cumple	Cumple
2. La arquitectura de software debe estar documentada y actualizada.	Cumple parcialmente	Cumple
3. El modelo de datos debe estar documentado y actualizado.	Cumple parcialmente	Cumple
4. El sistema debe contar con una API, la cual debe estar documentada y actualizada para acceder a los datos.	No cumple	Cumple
5. La autenticación de usuarios debe hacerse contra UPasaporte.	No cumple	Cumple
6. El código fuente debe estar apropiadamente versionado y respaldado en un repositorio.	No cumple	Cumple parcialmente
7. El stack tecnológico utilizado debe estar apropiadamente indicado.	No cumple	Cumple
8. El stack tecnológico debe estar alineado con las definiciones del DCC.	No cumple	Cumple parcialmente
9. El sistema debe contar con las indicaciones pertinentes para que un tercero pueda levantarlo en un ambiente de desarrollo.	No cumple	Cumple

Capítulo 6

Conclusiones y Trabajo a Futuro

Antes de comenzar este trabajo de memoria el sistema U-Papers utilizaba tecnologías deprecadas, y no contaba con información debidamente que facilitara su mantenibilidad o extensión sin tener que ir a analizar el código fuente. Además, el sistema no cumplía con gran parte de los lineamientos definidos por el DCC para homogeneizar aspectos claves de sus sistemas internos. Por ejemplo, el sistema no contaba con una interfaz de acceso a datos, la cual permitiría a otros sistemas consultar la información de U-Papers en línea. Tampoco contaba con un mecanismo de autenticación unificado de usuarios; sólo por mencionar algunos de los lineamientos con los que no cumplía el sistema.

Luego de actualizar la tecnología usada en la implementación del sistema, y de haber corroborado de que el sistema mantenía la funcionalidad original, se dio el segundo paso. Éste consistió en agregar algunos servicios (como los antes mencionados), y documentar piezas claves del software que son muy útiles de cara a la evolución del sistema; por ejemplo, su arquitectura o su nuevo modelo de datos.

Haciendo el análisis del nivel de cumplimiento de los lineamientos definidos por el DCC, podemos decir que la nueva versión de U-Papers mejora considerablemente el nivel de gobernabilidad y capacidad de integración del sistema, comparado con la versión legada. Por lo tanto, se cumplen los objetivos definidos.

A pesar de estos resultados, aún queda trabajo por hacer en U-Papers. Por ejemplo, la interfaz de usuario debería ser capaz de adaptarse a distintas resoluciones de pantalla, pero actualmente solo soporta pantallas de escritorio, por lo que la navegación en dispositivos móviles es incómoda para el usuario. También se considera que una nueva actualización de tecnologías será necesaria dentro de pocos años, pues las actuales ya no cuentan con soporte oficial, lo que deja vulnerable al sistema contra ataques informáticos, como es el caso del lenguaje usado (PHP). O pronto dejarán de tener soporte y mantención como es el caso del framework (CodeIgniter), para el cual ya existe una nueva versión en desarrollo constante. Por último, una funcionalidad que podría ser muy útil para los autores es darles la facultad de modificar directamente sus áreas de investigación y desplegarlas dentro de sus perfiles (vista del autor) como información sobre ellos.

Bibliografía

[1] Chacón, F., "Desarrollo de un repositorio de artículos científicos. memoria de ingeniería civil en computación. departamento de ciencias de la computación, fcfm, universidad de chile.," 2012.

[2] "C4: Context, containers, components and code," Última visita: 01/03/2023., https://c4model.com/.

[3] "Apache http server project," Última visita: 01/03/2023., https://httpd.apache.org.

[4] "Codeigniter," Última visita: 01/03/2023., https://www.codeigniter.es.

[5] "Mysql," Última visita: 01/03/2023., https://www.mysql.com.

[6] "Matomo," Última visita: 01/03/2023., https://matomo.org.

[7] "Google analytics," Última visita: 01/03/2023., https://support.google.com/analytics/answer/9304153.

[8] "Fastapi," Última visita: 01/03/2023., https://fastapi.tiangolo.com.

[9] "Uvicorn," Última visita: 01/03/2023., https://www.uvicorn.org.

[10] "Xampp," Última visita: 01/03/2023., https://www.apachefriends.org/es/index.html.

[11] "Business process model and notation," Última visita: 01/03/2023., https://www.bpmn.org/.

ANEXO

A continuación, se detallan los principales requisitos con los que debe cumplir cualquier sistema implantado en el DCC, con el objetivo de cumplir con un nivel mínimo de gobernabilidad.

- **Descripción del proceso que apoya:** Incluye objetivo, actores, macro tareas, workflow y dolores del proceso y sus posibles causas. El proceso debe describirse usando la nomenclatura BPMN[11].
- **Objetivo y alcance del sistema:** Debe especificarse utilizando un tablero de alcance de productos.
- **Arquitectura del ecosistema:** El diagrama del ecosistema debe indicar la relación entre sistema descrito y otros con los que interactúa. La especificación de la arquitectura debe hacerse utilizando modelos de contexto en C4[2].
- **Arquitectura del software:** Esta arquitectura debe especificarse usando contenedores y componentes de C4.
- **Modelo de datos:** El modelo de datos debe contar con todos los elementos mandatorios de uno. Debe estar basado en tablas donde se especifique las relaciones y cardinalidad entre ellas, así como también señalar su llave principal y foráneas (si es que las hay).
- **API para acceder a servicios o datos que ofrece el sistema:** Debe implementarse una API a través de la cual otros sistemas puedan acceder a sus servicios o datos. Esta debe estar apropiadamente documentada en Swagger.
- **Autenticación de Usuarios:** Debe utilizar los servicios que provee el DCC. Se debe implementar un esquema propio de roles y permisos.
- **Código fuente debe estar apropiadamente versionado:** Debe utilizarse una herramienta de control de versiones (Git).
- **Stack tecnológico utilizado:** Incluir tecnologías y sus versiones mínimas requeridas para su funcionamiento.
- **Indicaciones para el deployment:** Las indicaciones deben contener los pasos y requerimientos necesarios para que un tercero pueda poner en producción al software.