



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

## Reingeniería del Sistema U-Papers

### MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

Juan Guillermo Rodríguez Cruz

PROFESOR GUÍA:

Sergio Ochoa Delorenzi

PROFESOR CO-GUÍA

Juan Arrigada Cancino

MIEMBROS DE LA COMISIÓN:

Benjamín Bustos Cárdenas

Eduardo Godoy Vega

SANTIAGO DE CHILE

2024

## Resumen

Desde hace más de una década el Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile ha utilizado el sistema U-Papers para registrar las publicaciones de sus académicos y para generar información de apoyo a los procesos de acreditación de las carreras que imparte. Esta información ha servido también para monitorear la actividad científica de los investigadores del DCC.

El sistema U-Papers ha sido extendido y ajustado por muchos estudiantes distintos en diferentes instancias, por ejemplo, proyectos de cursos y memorias de ingeniería. A pesar de las mejoras realizadas a su funcionalidad y gobernabilidad, el estado actual del sistema no facilita su mantenibilidad y extensión debido principalmente a que las tecnologías involucradas no son parte del stack tecnológico del Área de Desarrollo de Software (ADS) del DCC. Particularmente, utiliza lenguaje de programación PHP y no maneja roles de usuario. Tampoco utiliza el diseño de interfaz de usuario definido para los sistemas del DCC. Corregir estos aspectos del sistema era una condición básica para que el ADS del departamento pudiera hacerse cargo de la evolución de este sistema, permitiéndole integrarse de manera armónica en su ecosistema de aplicaciones.

En este trabajo de memoria se trató la reingeniería de las funcionalidades centrales del sistema, las cuales eran viables de abordar dentro del alcance de este proyecto, para generar una nueva versión que cumpliera con los estándares actuales para los sistemas del DCC. El producto obtenido está implementado en Django (Python), ya se encuentra disponible en el ambiente de pruebas (previo a la puesta en producción), y fue evaluado por los profesores guía (como usuarios del sistema), y por un ingeniero del ADS como principal destinatario de este trabajo. Si bien los resultados obtenidos fueron buenos, mostrando que se lograron alcanzar los objetivos inicialmente planteados.

El paso siguiente es completar el proceso de pruebas exhaustivas, y hacer los ajustes necesarios antes de que el software se ponga en producción. Como trabajo a futuro se podría agregar la funcionalidad para que genere reportes e indicadores de forma automática, y se los envíe a los académicos según corresponda. De esa manera, se espera que el sistema facilite la supervisión propia de la actividad científica de los académicos, y del Departamento en su conjunto, transformándose en una pieza clave en los procesos de acreditación de las carreras de postgrado.

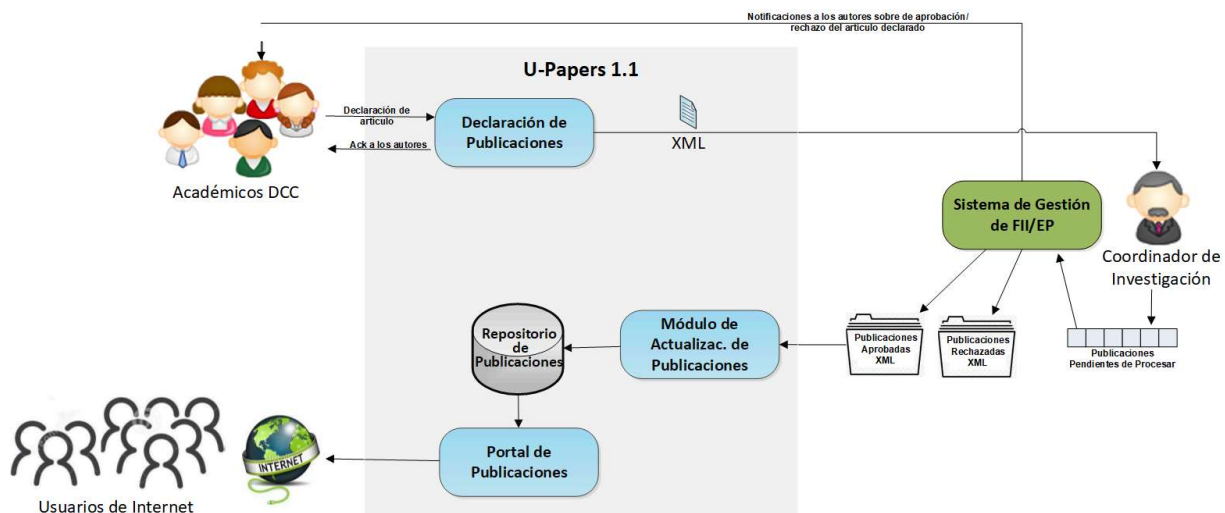
## **Tabla de contenido**

<b>1. Introducción</b>	<b>1</b>
1.1. Problema abordado	2
1.2. Objetivos de la memoria	3
1.3. Solución propuesta	3
1.4. Estrategia de evaluación de la solución	5
1.5. Estructura del documento	5
<b>2. Análisis del Sistema Legado</b>	<b>6</b>
2.1. Arquitectura de U-Papers	6
2.2. Modelo de datos del sistema	7
2.3. Interfaz de usuario del sistema	9
2.4. Tecnologías utilizadas por el sistema legado	10
2.5. Tecnologías a utilizar en el nuevo sistema	10
2.6. Decisiones y requisitos para el desarrollo de la nueva solución	11
<b>3. Concepción de la Solución</b>	<b>13</b>
3.1. Principales Requisitos de la Solución	13
3.2. Diseño de la Arquitectura	14
3.3. Acceso a servicios de la plataforma	15
3.4. Diseño de interfaces para dispositivos móviles	17
<b>4. Implementación de la Solución</b>	<b>20</b>
4.1. Cambio de Lenguaje de Programación	22
4.3. Interfaces para el usuario no autenticado	23
4.4. Interfaces para el Académico	24
4.5. Interfaces para el Administrador	26
4.6. Dificultades encontradas en la implementación	27
<b>5. Evaluación de la Solución</b>	<b>29</b>
5.1. Evaluación del trabajo de reingeniería del sistema	29
5.1.1 Mantenibilidad del código fuente	31
5.1.2 Usabilidad y utilidad del sistema	32
5.1.3 Utilidad del API pública	32
5.1.4 Verificación del funcionamiento	32
5.2. Evaluación de las nuevas interfaces de usuario	33
<b>6. Conclusiones y Trabajo a Futuro</b>	<b>34</b>
<b>Bibliografía</b>	<b>36</b>
<b>Anexo A: Checklist para la Entrega de Sistemas Desarrollados por Estudiantes</b>	<b>37</b>
Nivel básico	37
Nivel intermedio	38
Nivel avanzado	39

# 1. Introducción

A lo largo de los años, en el Departamento de Ciencias de la Computación (DCC), se han puesto en producción distintos sistemas de software, con el fin de solucionar problemáticas internas del Departamento. Estas aplicaciones han surgido como iniciativas personales de académicos del DCC. En el caso de esta memoria se abordará la mejora de una de esas aplicaciones, el sistema de gestión de publicaciones llamado *U-Papers*. Esta aplicación web consta de un repositorio digital de publicaciones científicas, con el fin de concentrar y dar visibilidad al trabajo de investigación realizado por académicos y alumnos del DCC además de asistir a los responsables de los procesos de acreditación.

El sistema U-Papers implementa un workflow en el cual intervienen diversos actores, y que interactúa con un sistema de software llamado *Sistema de Gestión de FII/EP*. El acoplamiento entre estos dos sistemas es por datos. Particularmente, hay un intercambio de archivos XML en ambos sentidos (entre los dos sistemas), donde cada archivo representa la declaración de una o más publicaciones. La Figura 1 muestra un esquema básico que permite ver el escenario de trabajo de U-Papers, los principales actores involucrados y su interacción con el *Sistema de Gestión de FII/EP* y con *Internet*.



**Figura 1.** Escenario de trabajo de U-Papers (versión legada)

El proceso de registro de información comienza cuando un académico del DCC declara una publicación a través de un formulario web que ofrece la versión legada de U-Papers. Al enviar el formulario, el sistema genera un documento XML con toda la información de la publicación. Dicho archivo es enviado por correo electrónico al Coordinador de Investigación, y también se adjunta como parte del mensaje de

confirmación de la declaración que reciben los autores por email. Una vez al mes el Coordinador de Investigación junta todas las declaraciones pendientes de procesar, y alimenta con esa información al sistema de gestión de FII/EP.

El sistema de gestión de FII/EP le permite al Coordinador de Investigación determinar qué declaraciones de papers están aprobadas, y cuáles están rechazadas. Estas últimas en general se rechazan por falta de información, o bien porque contienen información incorrecta, pero pueden volver a declararse con la información precisa en una nueva instancia.

Las publicaciones aprobadas son utilizadas para alimentar la base de datos de U-Papers, y una vez que han pasado seis meses desde que la publicación fue aceptada como válida por el Coordinador de Investigación, éstas se vuelven visibles para la comunidad (a través de Internet). Los Usuarios internos y externos pueden acceder directamente al sistema que está vinculado a la sección de Investigación del Sitio Web del DCC.

### **1.1. Problema abordado**

Desde su concepción en 2012, el proyecto ha sido extendido por diversas personas de forma poco sincronizada, dificultando su mantención, evolución e integración en el ecosistema de aplicaciones del departamento. A partir de su última actualización (descrita en [1]), se puede apreciar que U-Papers cumple con la mayoría de los lineamientos de gobernabilidad establecidos por el área de desarrollo del DCC. Sin embargo, su backend y frontend debían ser mejorados considerablemente. Respecto al backend, la versión legada utiliza el framework PHP CodeIgniter 3, tecnología que debía ser actualizada para minimizar la vulnerabilidad del sistema ante amenazas externas como intentos de acceso no autorizados, pérdida o modificación maliciosa de los datos; además, este framework no se encuentra dentro del stack tecnológico del ADS lo cual dificulta aún más su mantención.

Por otra parte, la plataforma no posee un diseño responsivo dado que fue ideada para pantallas de escritorio, por lo que se requiere de una actualización visual para que los componentes del frontend cumplan con esta cualidad, haciendo a la aplicación usable desde diferentes dispositivos y tamaños de pantallas. Además, era necesaria llevar a cabo una reestructuración del aspecto gráfico de U-Papers, debido a que el estilo presente en la plataforma no corresponde a la estética de las aplicaciones contemporáneas; desde los iconos utilizados, hasta la paleta de colores repleta de grises.

El trabajo se enfocó en mayor medida en la extensión de las funcionalidades de U-Papers, puesto que al ser una plataforma legada, hay servicios que se debían

agregar, eliminar y modificar, con el fin de mejorar la experiencia de los usuarios y la comunicación interna con el ecosistema de servicios del DCC. Particularmente, se debía mejorar la integración con el Sitio Web del DCC, y la entrega de reportes a académicos, al coordinador de investigación y a las autoridades del Departamento.

Por otra parte, respecto a lo que se debía agregar, estaban los nuevos criterios de búsqueda de los artículos, utilizando áreas y subáreas de investigación; esto permite agrupar documentos que abarquen temas similares. Además, se debían generar estadísticas que fueran importantes, tanto para los académicos como para los usuarios externos, ya que a la fecha solo es posible visualizar información acerca de las visitas a la plataforma, y el desempeño de los autores del DCC con respecto a sus pares. Esto último muestra evidencia de la evolución en la cantidad de publicaciones por año que registra cada autor.

## **1.2. Objetivos de la memoria**

El objetivo general de esta memoria es hacer a U-Papers un sistema funcional, que se pueda integrar de manera fluida y coherente al ecosistema de software del DCC, y que adhiera a los lineamientos definidos para todos los sistemas del Departamento. Para lograr este objetivo general, se definieron los siguientes objetivos específicos:

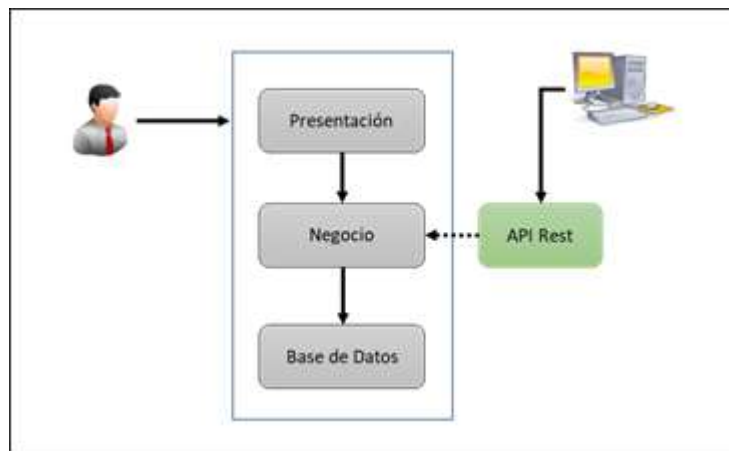
1. Realizar una migración del sistema, del framework CodeIgniter en el lenguaje PHP, al framework Django en el lenguaje Python. Esta migración de servicios, que en realidad es una reingeniería, debe mantener operativa toda la funcionalidad del sistema legado.
2. Mejorar la usabilidad de las interfaces del sistema actual, para que adhiera al look-and-feel definido para los sistemas del DCC, manteniendo la coherencia gráfica en la experiencia del usuario entre los diversos sistemas del departamento y mejorar su usabilidad, permitiendo el acceso a los servicios de U-Papers a través de diversos dispositivos computacionales.
3. Implementar servicios con los que no cuenta el sistema legado, particularmente aquellos que requieren ser accedidos a través de la API del sistema, y completar la adherencia de requisitos para los sistemas del DCC, para eventualmente migrar completamente a la nueva versión de U-Papers.

## **1.3. Solución propuesta**

La estructura básica del nuevo sistema (U-Papers 2.0) se muestra en la Figura 2. Entre las cosas importantes éste cuenta con una API que permite a otros usuarios (y

sistemas del DCC) interactuar con la información que éste posee. La implementación del nuevo U-Papers continuará utilizando una arquitectura de tres capas, que se explica brevemente a continuación:

1. *Presentación*: Capa a través de la cual los usuarios pueden acceder al sistema utilizando un navegador Web y una conexión a Internet. Esta capa sólo se comunica con la capa de Negocio.
2. *Negocio*: Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos para solicitar al gestor de base de datos el almacenar o recuperar datos de él. En la nueva versión del sistema, ésta se encuentra implementada en el lenguaje de programación Python y el framework de desarrollo Django<sup>1</sup>.
3. *Base de datos*: En esta capa es donde residen los datos, y es la encargada de acceder a los mismos. Ésta está formada por un administrador de base de datos, donde se reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. El motor de base de datos utilizado es MySQL<sup>2</sup>.



**Figura 2.** Arquitectura de tres capas de U-papers 2.0

El patrón de arquitectura de software utilizado para la construcción de este sistema está basado en el Modelo-Vista-Controlador, dada la facilidad que entrega el framework Python seleccionado para implementar el sistema; es decir, Django. Los componentes de este framework (Models, Views, Templates) permiten emular el patrón MVC de la versión legada. Por otra parte, la API Rest se encuentra implementada en Python a través del framework Django REST, el cual provee una

<sup>1</sup> <https://www.djangoproject.com/>

<sup>2</sup> <https://www.mysql.com/>

gran escalabilidad y rapidez.

#### **1.4. Estrategia de evaluación de la solución**

Dado que el sistema legado ya estaba en producción, y que era considerado suficientemente usable y útil por sus usuarios, la evaluación de las interfaces del nuevo sistema se limitó a comparar la usabilidad y utilidad de las nuevas, contra las del sistema legado. Además de que la data mostrada ya se encuentra validada por el propio uso del sistema legado, no es necesario agregarla a la evaluación. En dicha evaluación participaron tres académicos del DCC.

La funcionalidad disponible para el administrador no sufrió un cambio significativo(en el caso legado se accede mediante el motor de la base de datos, mientras que en la nueva implementación se maneja mediante Django Admin), por lo tanto ésta no fue evaluada por el actual coordinador de investigación.

#### **1.5. Estructura del documento**

En el siguiente capítulo se presenta, en una primera instancia, un análisis del sistema legado y todo lo implementado hasta el momento de esta reingeniería. En la descripción de dicho sistema se tomó en cuenta la arquitectura del sistema, el modelo de datos implementado, y las tecnologías utilizadas en el trabajo reportado en [1]. Luego, en el capítulo 3 se procede a detallar: 1) el proceso de concepción de la solución, incluyendo los requisitos con los que debe cumplir la aplicación para los distintos tipos de usuarios, la arquitectura de la nueva pagina web, así como también se enumeran los principales casos de uso, para luego mostrar las interfaces tipo propuestas para dispositivos pequeños, principalmente teléfonos inteligentes.

En el capítulo 4 se muestran las principales interfaces para administradores, académicos y usuarios no autenticados. En el capítulo 5 se describen los procesos de evaluación realizados y los resultados obtenidos. Finalmente, en el capítulo 6 se presentan las conclusiones y el trabajo a futuro.



## 2. Análisis del Sistema Legado

En este apartado se presenta el análisis y descripción detallada del sistema legado abordado en esta memoria, y de las diferentes áreas en las que se enfocó el trabajo, destacando los desafíos enfrentados. A continuación se analiza la versión legada de la plataforma U-Papers.

### 2.1. Arquitectura de U-Papers

A continuación se presenta la arquitectura del ecosistema, y los macrocomponentes de U-Papers luego de haber sido extendido. Dicha arquitectura corresponde a un modelo de contexto expresado en el lenguaje C4<sup>3</sup> [2].

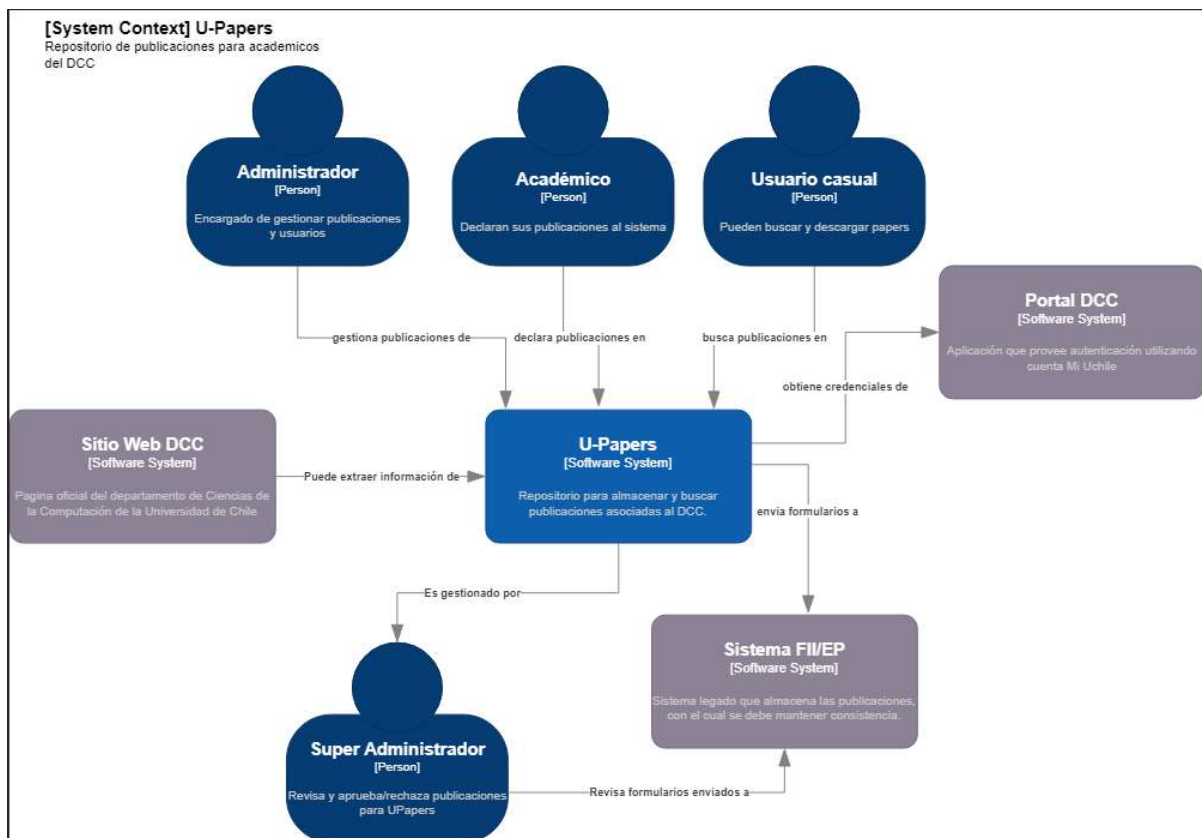


Figura 3. Modelo de contexto del sistema (obtenida de [1])

<sup>3</sup> <https://c4model.com/>

Luego de las últimas modificaciones, realizadas en la memoria de Roberto Aguilera [1], el sistema tuvo la capacidad de generar nuevas interacciones con otros componentes de su ecosistema. Por un lado, gracias a la API de entrega de datos implementada, ahora los sistemas externos (como por ejemplo, el sitio web del DCC u otros sistemas y personas) pueden obtener información sobre las publicaciones directamente desde la base de datos. Esto facilita el flujo de información al permitir la extracción directa de datos cuando se cuenta con los permisos pertinentes. Asimismo, la plataforma U-Papers es capaz de autenticar usuarios, dirigiéndolos al Portal DCC donde deben ingresar con sus credenciales MiUChile<sup>4</sup>.

## 2.2. Modelo de datos del sistema

U-Papers utiliza una base de datos relacional para almacenar toda la información referente a las publicaciones, autores, usuarios, configuraciones del sistema, estadísticas y tablas temporales. En la última instancia de modificación (reportada en [1]), se agregaron nuevas tablas y nuevos campos al modelo ya definido, como por ejemplo: se agrego la tabla “areas”, “publication\_has\_area” y “author\_has\_areas”, así como el campo “pasaporte” a la tabla “users”.

El modelo de datos, representado en la Figura 4, muestra los seis grupos principales de información en el sistema: *publicaciones*, *autores*, *usuarios*, *configuraciones*, *estadísticas* y *tablas temporales*.

---

<sup>4</sup> [www.MiUchile.cl](http://www.MiUchile.cl)

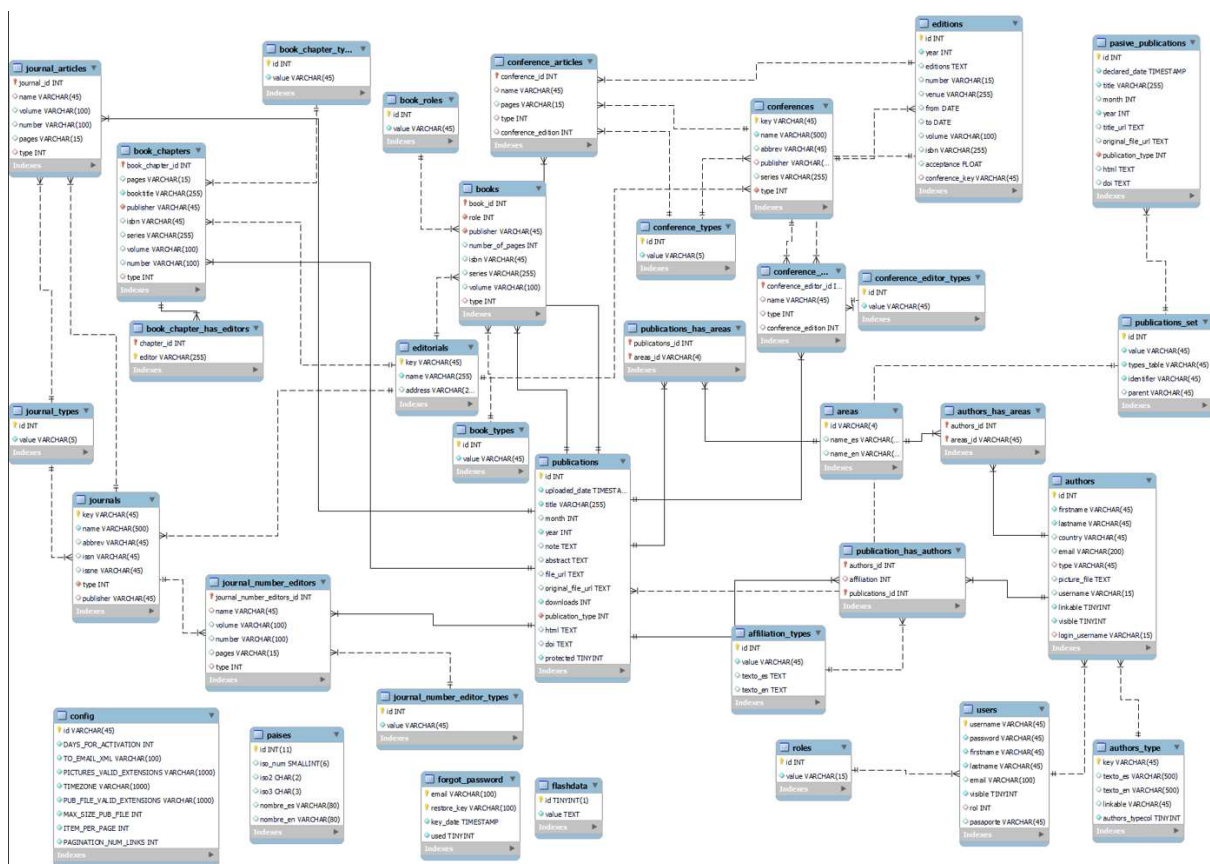


Figura 4. Modelo de datos del sistema U-Papers (obtenido de [1])

La tabla "publications" contiene información general sobre todas las publicaciones, y está relacionada con otras tablas que dividen las publicaciones en seis subgrupos: artículos de revistas, artículos de conferencias, libros, capítulos de libros, editores de conferencias y editores de números de revistas. Estas divisiones se realizan a través de tablas que indican el tipo de publicación, la editorial que la publicó y la edición correspondiente.

La tabla "authors" almacena los datos de todas las personas que han subido papers al sistema. Esta tabla está relacionada con la tabla "publications" a través de una tabla intermedia para evitar la repetición de tuplas con autores existentes. Además, se relaciona con la tabla "users", que contiene información general sobre los usuarios del sistema. Cada usuario puede tener uno de tres roles posibles (*superadmin*, *admin* o *autor*), lo que determina su grado de autoridad en el sistema. Además hay una vista pública del sistema para usuarios no autenticados.

El grupo restante incluye la tabla "config", que almacena información necesaria para mantener la coherencia del sistema, como la zona horaria utilizada y el tipo de extensión de documentos aceptada. Las tablas "forgot\_password" y "flashdata"

contienen información temporal que se utiliza para confirmar ciertas acciones de los usuarios. Por último, la tabla "países" y otras tablas no incluidas en el modelo de la Figura 4 se utilizan para mantener un registro de estadísticas en el sistema.

### **2.3. Interfaz de usuario del sistema**

La aplicación web que se encuentra en producción en estos momentos consta de una página principal, la que contiene una *navbar* con el icono de la aplicación y botones que redireccionan a otras vistas (por ejemplo, a la vista de estadísticas, al inicio de sesión, etc.) o a otros sitios (como el portal institucional del DCC).

En el lado izquierdo tiene una barra de navegación en donde se encuentran 3 buscadores: un buscador general de publicaciones, un buscador por fecha de publicación, y un buscador por autor. Además, se encuentra un componente que muestra las visitas a la página por país. Finalmente, el componente principal de la página consta de una barra de búsqueda de publicaciones, y más abajo, un listado con las últimas publicaciones. Todo esto se puede ver en la Figura 5, que es una interfaz definida únicamente para ser usada en computadores desktop y laptop.

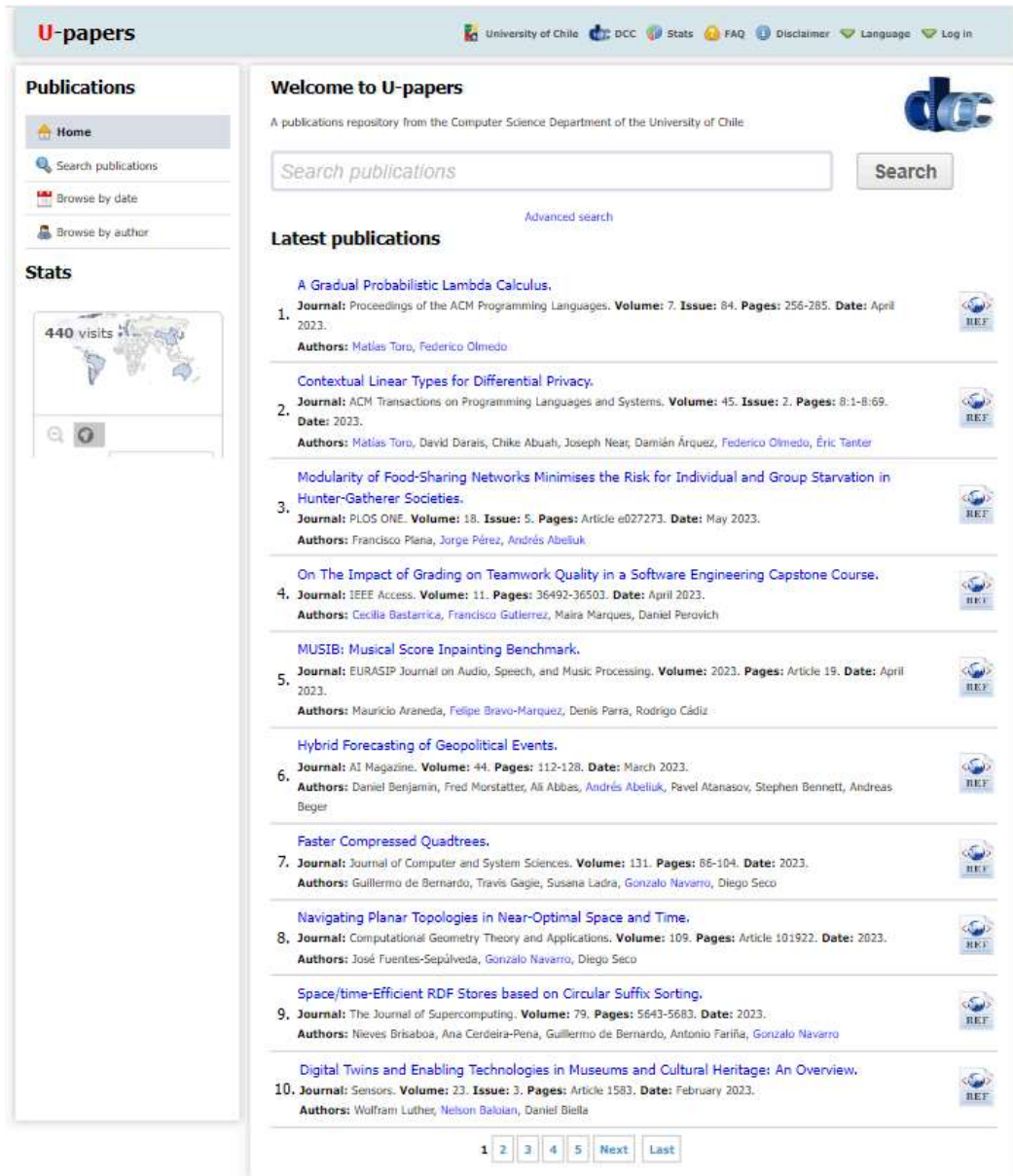


Figura 5. Interfaz principal del sistema U-Papers (versión legada)

## 2.4. Tecnologías utilizadas por el sistema legado

En este momento el proyecto se encuentra montado sobre un servidor web Apache HTTP Server, mientras que el backend se encuentra desarrollado en el framework CodeIgniter 3.1.13 del lenguaje PHP, y la base de datos es MySQL. Finalmente, la API del sistema se encuentra desarrollada con el framework FastAPI de Python. FastAPI utiliza *uvicorn*, que es una herramienta que ayuda a desplegar y ejecutar aplicaciones web escritas en Python.

## 2.5. Tecnologías a utilizar en el nuevo sistema

En esta iteración se migró todo el código de CodeIgniter a Django, debido a la escalabilidad que este último proporciona, además de contar con una comunidad activa que proporciona una gran cantidad de recursos disponibles que facilitan el aprendizaje y resolución de problemas. Por otra parte, el framework Django fue escogido como “tecnología core” por parte del Área de Desarrollo de Sistemas (ADS) del DCC, por lo que el memorista debía adherir a este requisito tecnológico. Por otra parte, la base de datos seguirá siendo relacional pero se migra al lenguaje PostgreSQL que es el que utiliza Django por defecto. La API también fue migrada dentro de la misma aplicación utilizando Django REST Framework.

## **2.6. Decisiones y requisitos para el desarrollo de la nueva solución**

Como ya se indicó, se tomó la decisión de migrar el sistema de CodeIgniter 3 a Django debido a que esto es ahora un requisito impuesto por el ADS. Esto debido a que, además de las razones antes expuestas, se ha observado una tendencia hacia el uso de arquitecturas de software más flexibles en el backend. Los frameworks populares, como Django y Ruby on Rails, han simplificado el proceso de desarrollo, permitiendo una mayor eficiencia en la implementación de nuevas funcionalidades. Estos frameworks también han facilitado la integración de APIs externas y servicios web, lo que ha maximizado las capacidades de las plataformas web.

Por otra parte, en los últimos años, el desarrollo y la mejora continua de las plataformas web han sido de gran importancia para proporcionar mejores experiencias de usuario. En este contexto, numerosos avances y tendencias han surgido con el objetivo de mejorar la funcionalidad y usabilidad de dichas plataformas.

Respecto a la usabilidad del sistema, Jakob Nielsen [5] definió esta característica que, lejos de ser unidimensional, debe cumplir con los siguientes cinco aspectos: aprendible, eficiente, intuitivo, asertivo y fácil/agradable de usar. En primer lugar, se requiere que el sistema sea intuitivo y fácil de usar, lo que permite al usuario familiarizarse rápidamente con el software (aprendizaje). Además, debe permitir al usuario mejorar su productividad cuando lo utiliza (eficiencia). Otro aspecto importante es que su funcionamiento sea fácil de recordar incluso después de períodos prolongados sin utilizarlo. Esto facilitará al usuario retomar el uso del sistema sin dificultades (intuitivo).

Por otra parte, con el software se busca minimizar la tasa de errores. El sistema debe ser diseñado de tal manera que el usuario tenga menos probabilidades de cometer errores, y en caso de que ocurran, sea fácil corregirlos sin complicaciones (asertivo).

Finalmente, es esencial que el sistema agrade al usuario, a través de un diseño estético y una interfaz visualmente atractiva que brinde una sensación positiva al interactuar con el sistema (fácil/agradable de usar).

Una de las áreas que ha experimentado un gran crecimiento, es la implementación de interfaces de usuario más intuitivas y amigables. En este sentido, se han introducido técnicas de diseño centradas en el usuario y con un enfoque responsivo, lo que permite una adaptación fluida de la interfaz a diferentes dispositivos y tamaños de pantalla. Cuanto más pequeña es la pantalla, más desafiante es el diseño a realizar y la estrategia de escalamiento de las interfaces.

Actualmente se destacan tres grandes estrategias para apoyar el desarrollo de interfaces web mobile: el diseño responsive, el diseño adaptativo y el desarrollo nativo. El primero busca adaptar los componentes de la interfaz basado en el tamaño disponible de la pantalla. El diseño adaptativo, necesita de componentes similares, pero distintos tamaños de los cuales se muestra el componente más acorde según las capacidades del dispositivo. Finalmente, la tercera estrategia consiste en separar la aplicación en dos o más sitios, uno para la visualización de escritorio, y otros adicionales para la visualización en dispositivos móviles (smartphones, smartwatch, etc.).[6]

En base a esto, se decidió implementar un diseño responsivo, el cual adapte los componentes al espacio disponible en las pantallas, de esta manera, por ejemplo, esconder las barras laterales haciéndolas desplegadas, cuando haya poco espacio disponible acomodando de mejor manera los componentes principales mejorando su visualización. Por otra parte, eliminar los widgets que no muestran información relevante y no son capaces de adaptarse a la pantalla, como es el caso del gráfico de estadísticas de visitas por país. Todo esto acompañado del hecho de que los componentes ajusten su tamaño de acuerdo al espacio total disponible en cada dispositivo.

## 3. Concepción de la Solución

En este capítulo se describen las principales actividades realizadas para el desarrollo de la nueva solución, la cual debía estar alineada con los principios de gobernabilidad de sistemas definidos por el ADS del DCC. A continuación se presentan los principales requisitos involucrados en dicha implementación, los ajustes a la arquitectura y el abordaje de diversos aspectos de diseño de las interfaces de usuario.

### 3.1. Principales Requisitos de la Solución

Entre los principales requisitos de usuario involucrados en este trabajo están los siguientes:

- *Autenticación de usuarios con credenciales MiUchile*: Los usuarios deben poder autenticarse en la aplicación con sus credenciales de MiUchile.
- *Búsqueda de publicaciones*: Los usuarios, tanto autenticados como no autenticados, serán capaces de buscar publicaciones mediante *keywords*.
- *Búsqueda de autores*: Los usuarios, tanto autenticados como no autenticados, serán capaces de buscar publicaciones según sus autores (por su nombre y/o apellido).
- *Creación de publicaciones*: Los usuarios autenticados serán capaces de crear nuevas publicaciones, ya sean estas artículos de revista, libros, conferencias, etc. Esto se podrá realizar siempre y cuando el usuario que declara sea autor o co-autor de la publicación.
- *Comunicación con la infraestructura de sistemas del DCC*: U-Papers será capaz de alimentar con información a otras aplicaciones del DCC (por ejemplo a la página Web institucional), mediante el uso de su API. La información entregada por la API dependerá de las credenciales del usuario o del sistema solicitante.
- *Descarga de publicaciones*: Los usuarios, tanto autenticados como no autenticados, serán capaces de descargar los papers directamente desde U-Papers, siempre que estos tengan un link de descarga disponible.
- *Estética del sistema*: Las interfaces del sistema deberán adherir a la estética establecida por el ADS del Departamento, el cual corresponde a lo que muestra la página institucional.

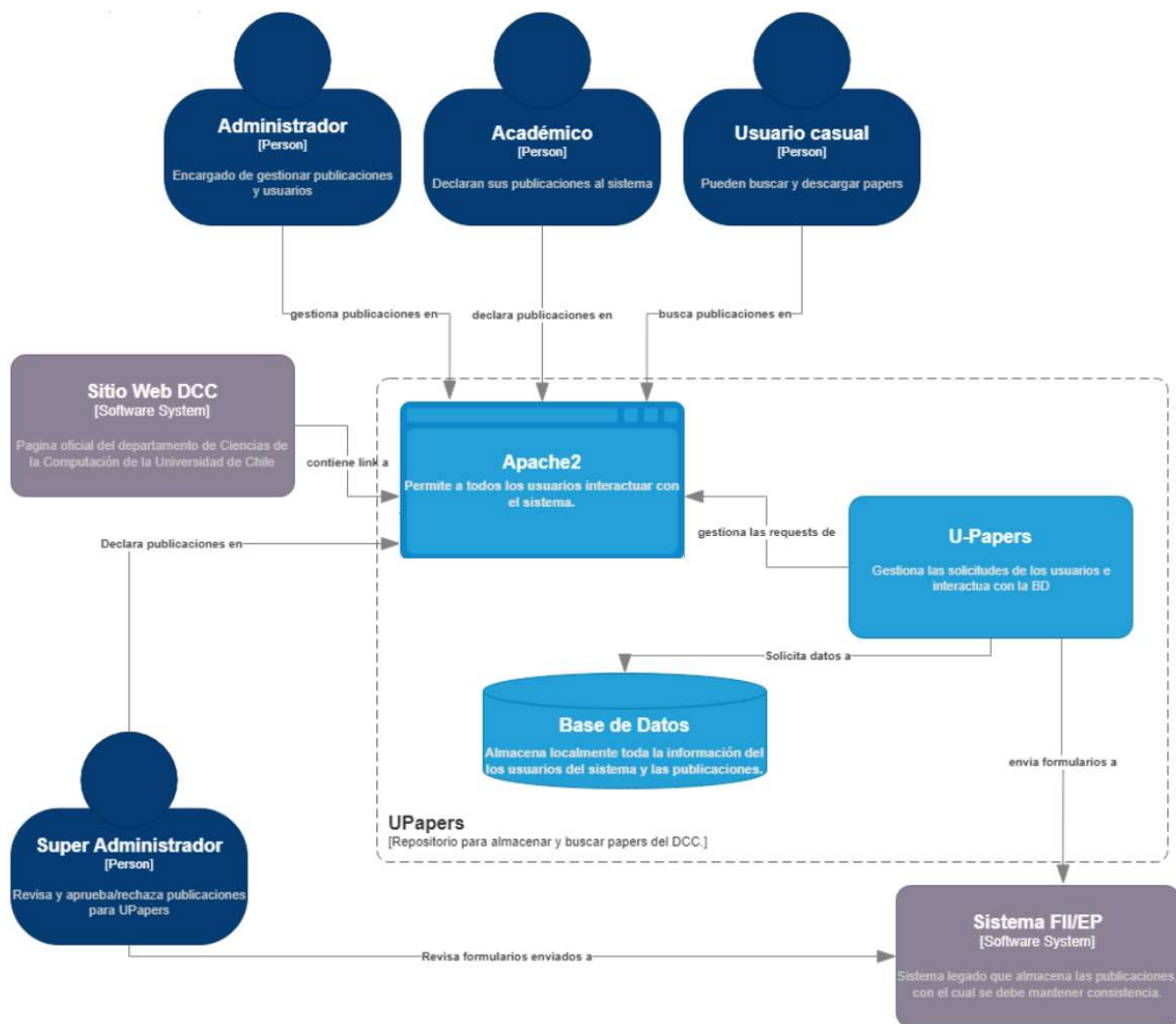


- *Responsividad de las interfaces de usuario*: Las interfaces de usuario deberán ser responsive, para que los usuarios puedan usar cómodamente el sistema tanto en computadoras (desktop o laptops), como en teléfonos inteligentes.

### **3.2. Diseño de la Arquitectura**

La Figura 6 muestra los contenedores del nuevo sistema, utilizando la vista de contenedores del lenguaje C4 [2]. Ahí se puede apreciar los tres componentes principales (o contenedores): la aplicación web, el controlador y la base de datos. Cada uno de estos refleja una capa de la arquitectura Modelo-Vista-Controlador (MVC).

La aplicación web representa la vista, que es la que recibe las solicitudes de los usuarios o de otros sistemas del ecosistema de servicios del DCC. Este componente reenvía las solicitudes al controlador, el cual las procesa y las hace llegar a la base de datos. Este último componente extrae la información pertinente y se la devuelve al controlador.



**Figura 6.** Mapa de contenedores del sistema en lenguaje C4 [2] (adaptado de [1])

Además de estos tres contenedores, U-Papers cuenta con una API que permite el consumo de datos a través de consultas enviadas por usuarios o sistemas externos debidamente autenticados. Para el acceso a la información la API expone un conjunto de endpoints que fueron explicados en [1].

### 3.3. Acceso a servicios de la plataforma

Tal como se mencionó antes, el sistema tiene tres tipos de usuario: el usuario anónimo (no autenticado), el académico (profesores del DCC) y el administrador (coordinador de investigación del DCC). Los permisos de acceso a la información son incrementales; es decir, el académico puede acceder a los servicios del usuario anónimo, y además a otros particulares para él. Lo mismo ocurre con el usuario administrador, que extiende los permisos del usuario académico. La siguiente tabla

indica los principales servicios disponibles para los usuarios, y quiénes tienen acceso a los mismos.

Tabla 1. Principales servicios disponibles para los usuarios

Servicio	Anónimo	Académico	Admin.
Búsqueda de artículos en el repositorio a través de la interfaz Web (por keyword y por autor)	X	X	X
Búsqueda de autores (por nombre y/o apellido)	X	X	X
Descargar publicación	X	X	X
Crear una publicación		X	X
Crear un autor			X
Crear una conferencia, revista, libro o similar (venue)			X
Modificar/eliminar publicaciones, autores, conferencias u otro concepto de datos			X
Crear/modificar un usuario y asignarle un rol (perfil)			X
Descargar publicaciones (archivo XML)			X

A continuación se describen algunos casos de uso, y se indican los pasos que debe seguir el usuario respectivo.

#### **Caso de uso 1:** *Descargar una publicación*

**Perfil de usuario:** No autenticado

**Pasos a seguir:** Al acceder a la página el usuario se encontrará con una vista con las últimas publicaciones cargadas en la aplicación. Si es que la publicación deseada no se encuentra dentro de éstas, el usuario será capaz de acceder a dos pestañas de búsqueda que se encuentran en el header de la página web. Por ejemplo, si el usuario conoce el título (o palabras claves) de la publicación, puede acceder al buscador de publicaciones haciendo click en la pestaña *publicaciones*. Allí le aparecerá un listado con todas las publicaciones registradas y aprobadas por el administrador, y podrá filtrarlas por *keywords* dentro del título de esta misma. Al identificar la publicación, éste podrá clicar el símbolo al costado del título en donde puede descargarla.

En el caso de que el usuario no conozca el título de la publicación, pero sí conoce al autor, puede seleccionar la pestaña *autores*, en donde tendrá un listado de los

autores donde podrá filtrar por nombre y/o apellido. Una vez encontrado el autor, el usuario deberá clickear sobre el nombre de éste, para así acceder al listado de sus publicaciones; repitiendo el proceso anterior podrá descargar la publicación.

### **Caso de uso 2:** *Crear una publicación*

**Perfil de usuario:** Académico

**Pasos a seguir:** Una vez que el usuario ha accedido a la página principal, éste deberá iniciar sesión ingresando sus credenciales MiUchile; para esto deberá presionar el icono en la parte superior derecha de la aplicación. Luego de autenticarse, el académico deberá dirigirse a la vista de publicaciones haciendo click en la pestaña *publicaciones* presente en el header. Adicionalmente, el usuario deberá presionar el botón *crear publicación* que se encuentra al costado derecho de la barra de búsqueda, para luego elegir el tipo de publicación que desea crear, siendo redirigido al formulario de creación donde deberá completar con la información pertinente. Finalmente, una vez completado el formulario de ingreso de datos, el usuario deberá presionar el botón de “creación” ubicado al final del formulario, luego de lo cual será redirigido a la página de inicio.

### **Caso de uso 3:** *Crear una conferencia*

**Perfil de usuario:** Administrador

**Pasos a seguir:** El administrador deberá dirigirse a la página principal, para luego iniciar sesión con sus credenciales UPasaporte. Una vez autenticado, podrá presionar en la pestaña de usuario que reemplazó al icono de inicio de sesión, en donde ahora se desplegará un menú de opciones dentro donde se encuentra el acceso al panel de administración.

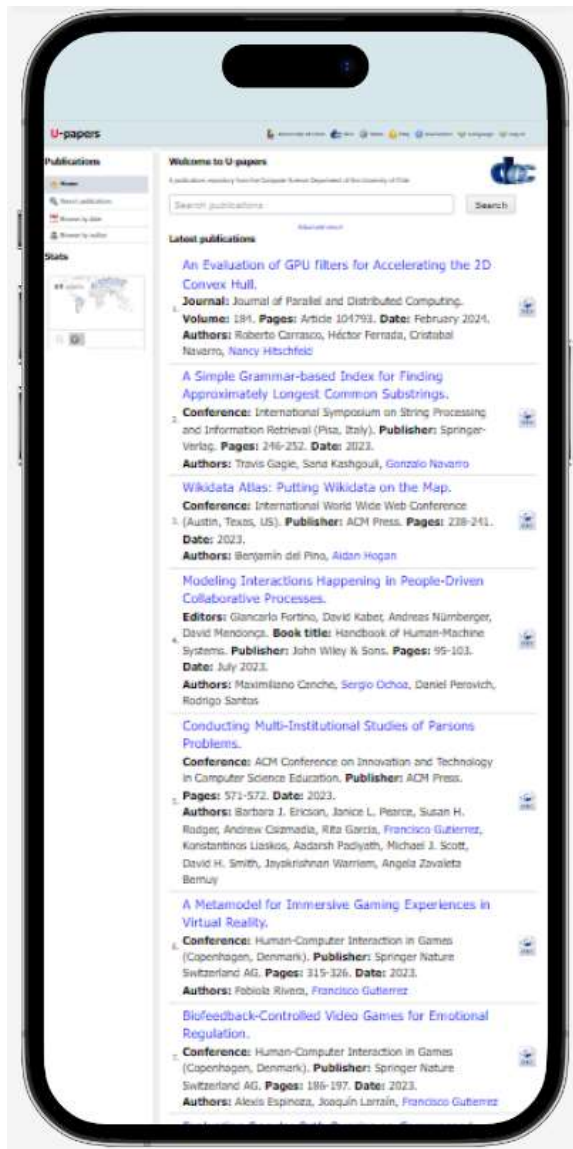
Ya en el panel de administración, el usuario deberá buscar el modelo asociado a los datos que desea modificar, en este caso, el modelo de *Conferencia*. Luego el administrador deberá presionar en la fila de este modelo, el botón *añadir* para ser redirigido a un formulario de creación de nuevas instancias. Una vez completado el formulario podrá guardar la información de la instancia siendo ahora visible para la creación de nuevos documentos por parte de los Académicos. Es importante destacar que existe otro mecanismo, accesible a través del sitio web, para que tanto el administrador como un académico puedan crear una conferencia. Sin embargo, dicho mecanismo es limitado y no permite modificar o eliminar una conferencia, cosa que sí permite el servicio aquí descrito. Estos dos servicios son complementarios y cumplen distintos roles.

## **3.4. Diseño de interfaces para dispositivos móviles**

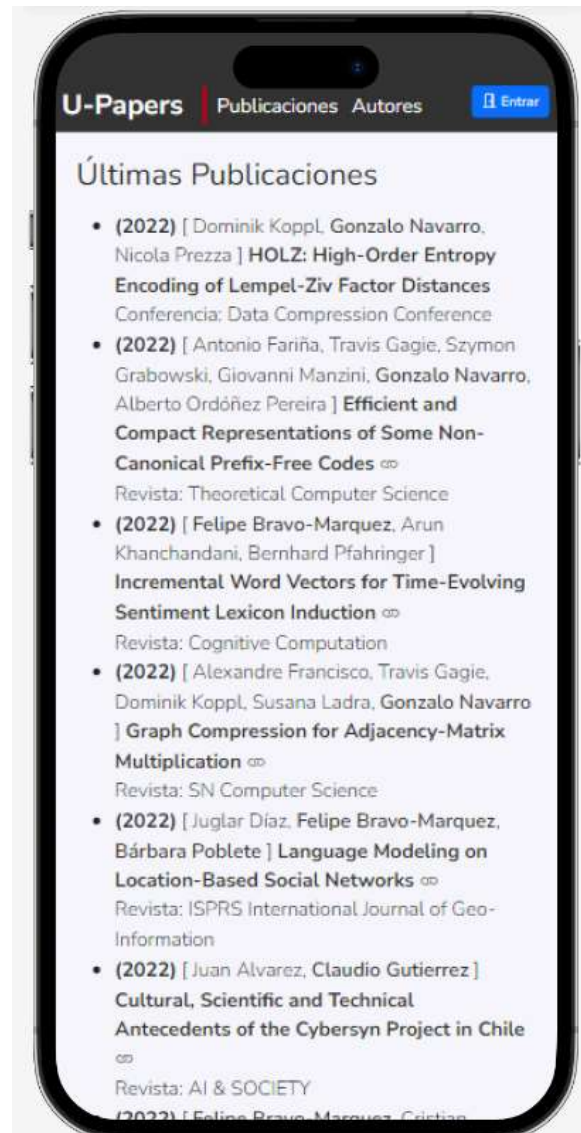
A continuación se muestra el diseño estructural de las interfaces de U-Papers para hacerlas responsivas. El diseño final se ajusta a los aspectos estéticos definidos por

el Área de Desarrollo de Software (font, colores, iconografía etc.) para todos los sistemas del Departamento. En la Figura 6 se presenta la vista principal de la página web, en donde se puede notar que el icono de la aplicación se mueve para que la *navbar* tenga más espacio.

La Figura 7.a muestra este diseño usando el look-and feel anterior de la aplicación, y en la Figura 7.b se muestra la interfaz final.



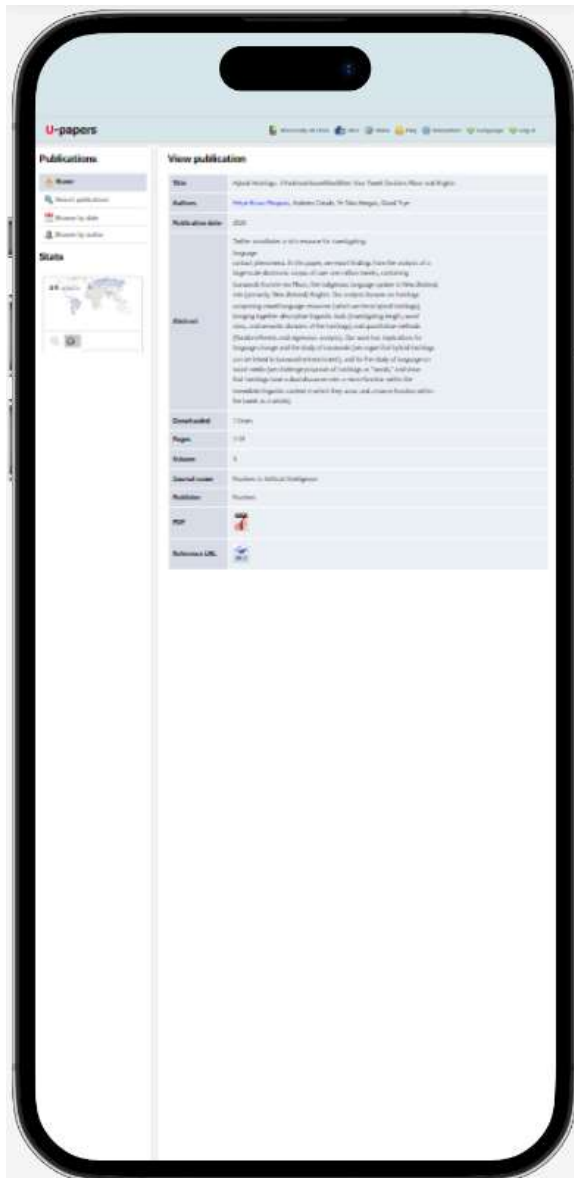
(a) Look-and-feel legado



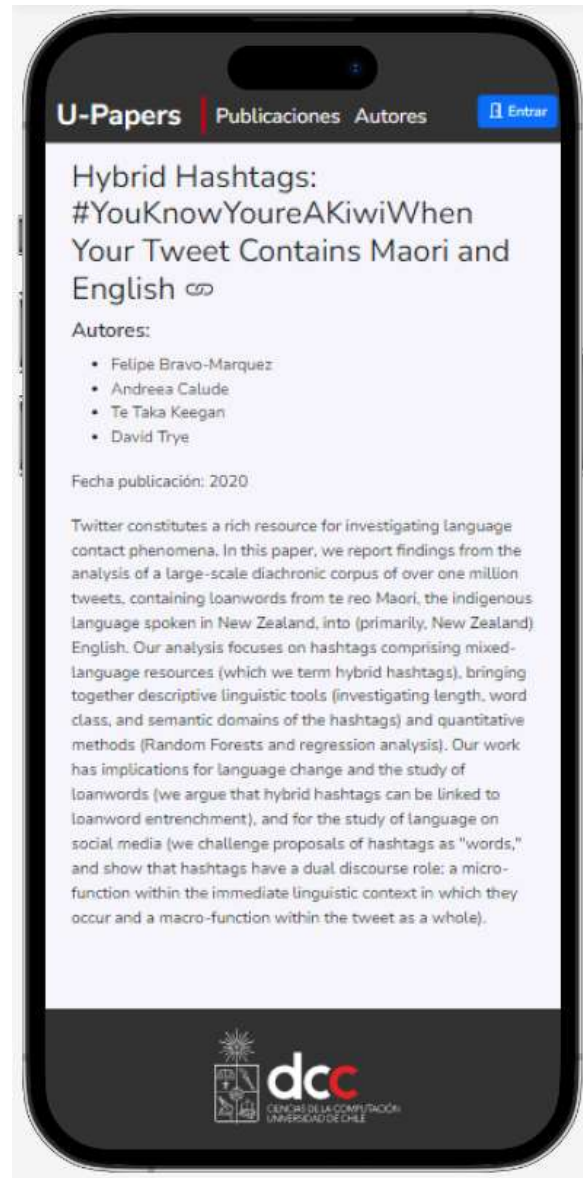
(b) Nuevo look-and-feel

**Figura 7.** Vista principal (estructural) del sistema desplegado en un dispositivo móvil

Similar al caso anterior, en la Figura 8 se muestra el contenido de una publicación sobre la interfaz de un dispositivo móvil, utilizando la interfaz legada y la nueva.



(a) Look-and-feel legado



(b) Nuevo look-and-feel

**Figura 8.** Vista de una publicación particular desplegada sobre la pantalla de un dispositivo móvil

## 4. Implementación de la Solución

A continuación se indican los principales desafíos abordados debido al cambio de lenguaje de programación y de framework. Luego se presentan las principales interfaces de usuario del sistema.

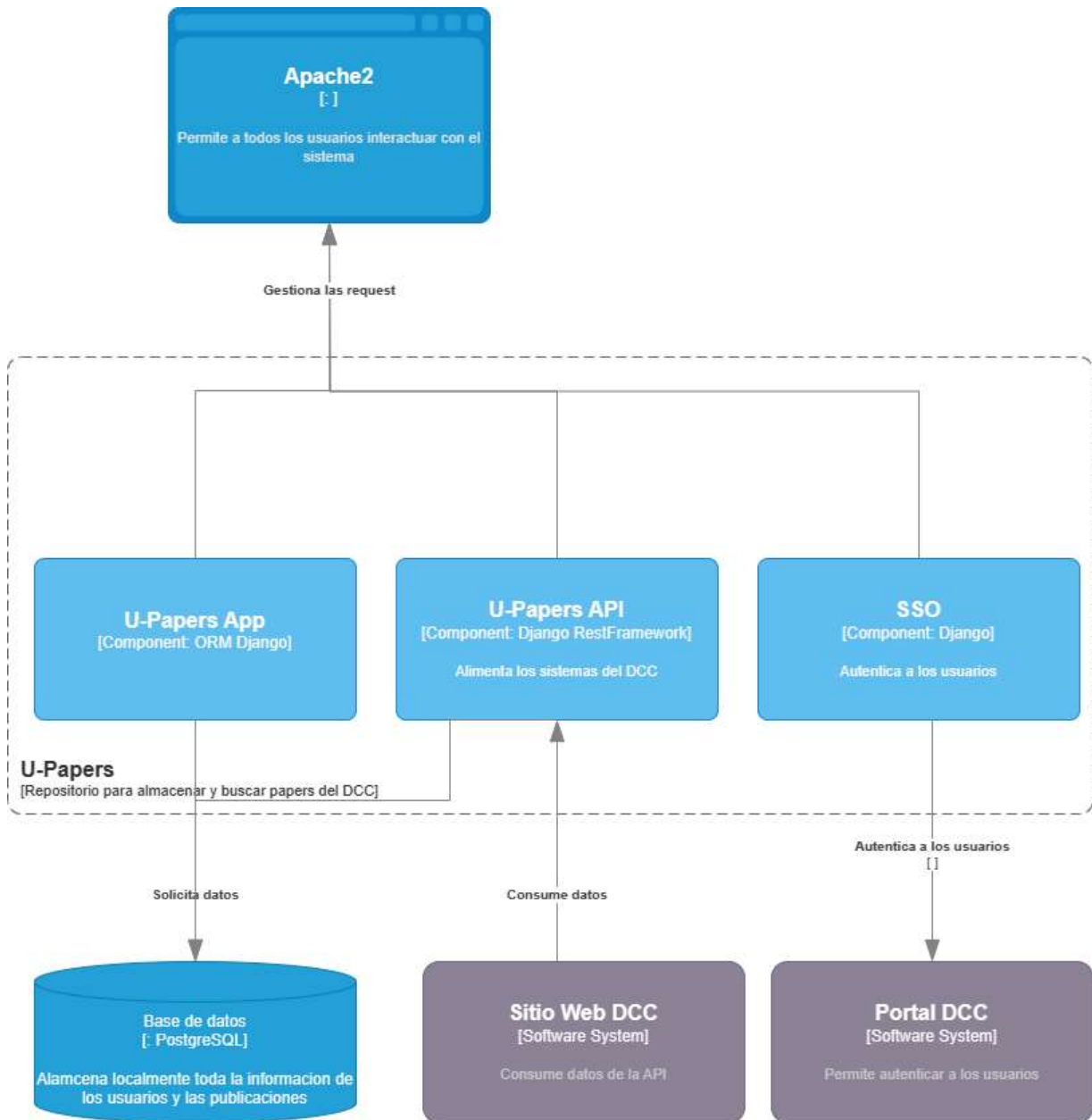


Figura 9. Diagrama de componentes del nuevo sistema

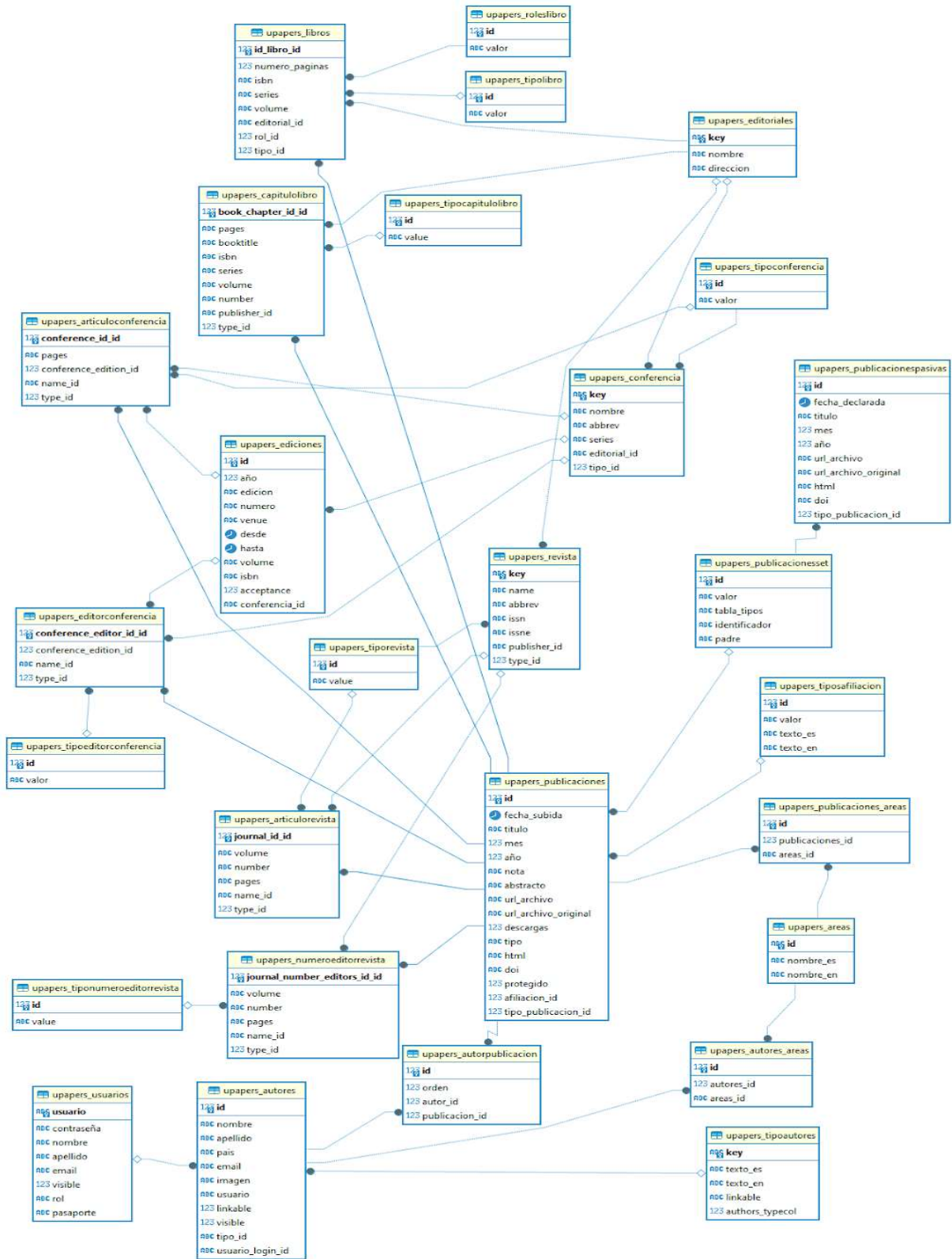


Figura 10. Modelo de datos del nuevo sistema



## 4.1. Cambio de Lenguaje de Programación

Para realizar la reimplementación del código desde Codeigniter 3 (PHP) a Django (Python) en una primera instancia se realizó un levantamiento de los requisitos con que debía cumplir la nueva aplicación para poder reemplazar al sistema legado. Estos requisitos fueron explicitados en el capítulo anterior.

Luego, para comenzar este trabajo, el estudiante utilizó un boilerplate de aplicaciones en Django provisto por el Área de Desarrollo de Software (ADS). La implementación partió con la emulación de las tablas del sistema legado, usando modelos de django, eliminando las tablas intermedias como “publication\_has\_area” y “publication\_has\_authors”, y las tablas de configuración como “config” y “flashdata”. En el caso de las primeras, es decir, las tablas intermedias, se eliminaron porque Django de manera implícita crea mediante las relaciones definidas entre tablas. En el caso de las segundas, se eliminaron porque estas configuraciones se definen dentro de la aplicación de Django.

Después de eso, se procedió a la creación de las urls y las distintas vistas que se mostrarán en la página. Éstas se implementan mediante vistas basadas en clases, y en este punto se toma la decisión de desarrollar la API dentro de la misma aplicación de Django. Para ello se utilizó Django REST Framework, reemplazando la API anterior desarrollada con FastAPI.

Una vez creadas las vistas y urls, se procede con la migración de los datos. Para esto se descargan los datos de la base de datos anterior en formato .JSON, los cuales son procesados por un script (el cual se ejecuta mediante django commands), creando las instancias de sus respectivos modelos en conjunto con sus relaciones. Dicho proceso tomó varias horas de desarrollo, puesto que la creación de instancias debe seguir un cierto orden para la correcta ejecución de las relaciones entre modelos. En este caso, las relaciones *many-to-many* fueron las más desafiantes puesto que es necesario que las instancias de ambos modelos a enlazar estén creadas de antemano.

En este punto la aplicación ya contaba con lo mínimo para su funcionamiento, búsqueda y visualización de publicaciones, por lo que se procede a la creación del formulario para la creación de nuevas publicaciones. Dentro de las librerías que están incluidas en el boilerplate se encuentra crispy forms, ésta facilitar la implementación de formularios y los hacer estéticamente agradables. Sin embargo, debido a los requerimientos que debe cumplir este formulario, no fue posible su utilización, teniendo que ser implementado de manera más estándar para permitir realizar ajustes finos a los formularios.

Finalmente, una vez creado el formulario, se procede a abordar errores menores y mejoras de diseño, como por ejemplo, el aumento de tamaño de las fuentes para hacer la interfaz más legible, o el cambio de colores en ciertos componentes que se muestran al pasar el cursor sobre ellos.

## 4.2. Principales Interfaces de Usuario

Dado que en el capítulo anterior se introdujeron los templates de las interfaces para dispositivos móviles, en este capítulo se muestran principalmente las interfaces de usuario para computadores de escritorio o laptops. La Figura 9 muestra la nueva home page de U-Papers para dichos dispositivos. Luego se muestra la funcionalidad que tiene disponible cada uno de los tipos de usuario antes indicados.

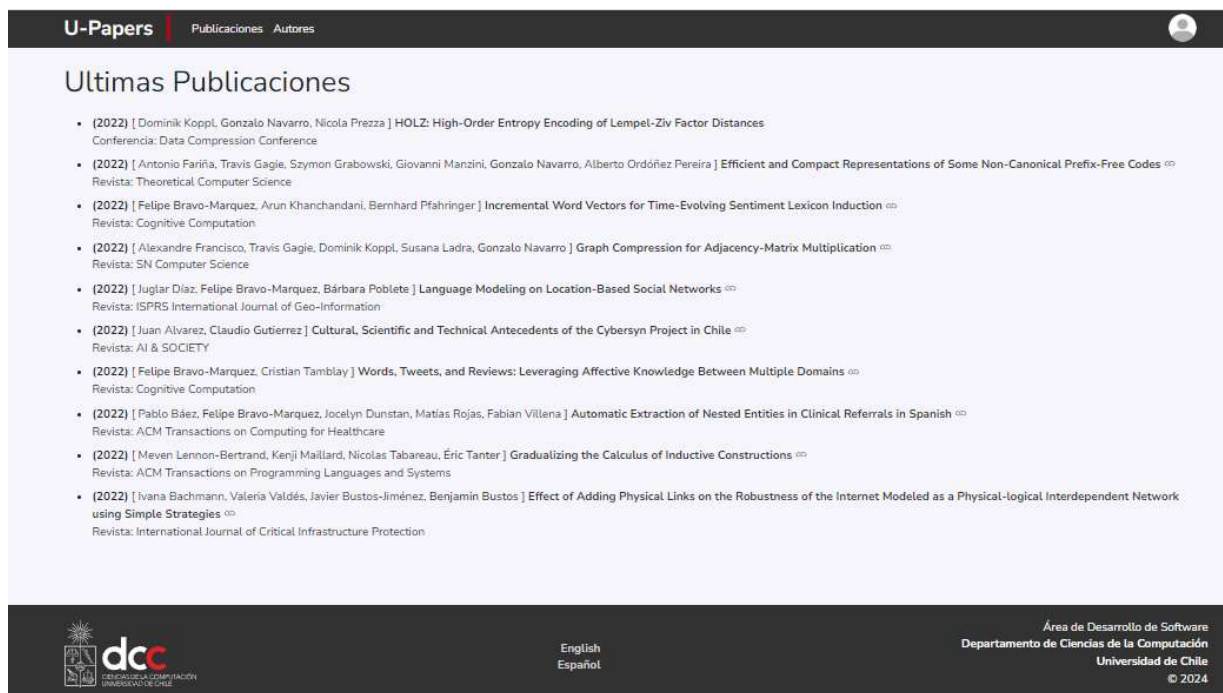


Figura 11. Homepage del nuevo sistema

## 4.3. Interfaces para el usuario no autenticado

El usuario no autenticado tiene la posibilidad de buscar publicaciones de autores, tanto buscando por su nombre, como por su apellido. De esta manera, se puede también acceder al perfil del autor, en donde se muestran todas las publicaciones (Figura 10).

U-Papers | Publicaciones Autores Entrar

## Felipe Bravo-Marquez

- (2022) [ Felipe Bravo-Marquez, Cristian Tamblay ] Words, Tweets, and Reviews: Leveraging Affective Knowledge Between Multiple Domains [↗](#)  
Revista: Cognitive Computation
- (2022) [ Juglar Díaz, Felipe Bravo-Marquez, Bárbara Poblete ] Language Modeling on Location-Based Social Networks [↗](#)  
Revista: ISPRS International Journal of Geo-Information
- (2019) [ David Trye, Andreea Calude, Felipe Bravo-Marquez, Te Taka Keegan ] Maori Loanwords: A Corpus of New Zealand English Tweets [↗](#)  
Conferencia: Conference of the Association for Computational Linguistics: Student Research Workshop
- (2020) [ Pablo Badilla, Felipe Bravo-Marquez, Jorge Pérez ] WEFE: The Word Embeddings Fairness Evaluation Framework [↗](#)  
Conferencia: International Joint Conferences on Artificial Intelligence
- (2019) [ Felipe Bravo-Marquez, Steven Lang, Eibe Frank, Christopher Beckham, Mark Hall ] WekaDeepLearning4j: a Deep Learning Package for Weka based on DeepLearning4j [↗](#)  
Revista: Knowledge-Based Systems
- (2019) [ Felipe Bravo-Marquez, Bernhard Pfahringer, Alan Ansell ] An ELMo-inspired Approach to SemDeep-5's Word-in-Context Task [↗](#)  
Conferencia: Workshop on Semantic Deep Learning (SemDeep-5)
- (2022) [ Felipe Bravo-Marquez, Arun Khanchandani, Bernhard Pfahringer ] Incremental Word Vectors for Time-Evolving Sentiment Lexicon Induction [↗](#)  
Revista: Cognitive Computation
- (2020) [ Frank D. Zamora-Reina, Felipe Bravo-Marquez ] DCC-Uchile at SemEval-2020 Task 1: Temporal Referencing Word Embeddings [↗](#)  
Conferencia: Workshop on Semantic Evaluation
- (2021) [ Alan Ansell, Felipe Bravo-Marquez, Bernhard Pfahringer ] PolyLM: Learning about Polysemy through Language Modeling [↗](#)  
Conferencia: European Chapter of the Association for Computational Linguistics
- (2019) [ Felipe Bravo-Marquez, Eibe Frank, Bernhard Pfahringer, Saif M. Mohammad ] AffectiveTweets: a Weka Package for Analyzing Affect in Tweets [↗](#)  
Revista: Journal of Machine Learning Research

« primera anterior 1 siguiente última »

**Figura 12.** Vista de perfil de autor

Además, el usuario sin autenticar será capaz de buscar publicaciones mediante keywords y/o autor (usando nombre y/o apellido). Una vez recuperadas las publicaciones, es posible descargarlas o acceder a la vista con información más detallada como fecha de publicación, resumen, etc. Siendo posible también descargar la publicación desde esa vista (Figura 11).

U-Papers | Publicaciones Autores Entrar

## Words, Tweets, and Reviews: Leveraging Affective Knowledge Between Multiple Domains [↗](#)

**Autores:**

- Felipe Bravo-Marquez
- Cristian Tamblay

Fecha publicación: 2022

Three popular application domains of sentiment and emotion analysis are: 1) the automatic rating of movie reviews, 2) extracting opinions and emotions on Twitter, and 3) inferring sentiment and emotion associations of words. The textual elements of these domains differ in their length, i.e., movie reviews are usually longer than tweets and words are obviously shorter than tweets, but they also share the property that they can be plausibly annotated according to the same affective categories (e.g., positive, negative, anger, joy). Moreover, state-of-the-art models for these domains are all based on the approach of training supervised machine learning models on manually annotated examples. This approach suffers from an important bottleneck: Manually annotated examples are expensive and time-consuming to obtain and not always available. In this paper, we propose a method for transferring affective knowledge between words, tweets, and movie reviews using two representation techniques: Word2Vec static embeddings and BERT contextualized embeddings. We build compatible representations for movie reviews, tweets, and words, using these techniques, and train and evaluate supervised models on all combinations of source and target domains. Our experimental results show that affective knowledge can be successfully transferred between our three domains, that contextualized embeddings tend to outperform their static counterparts, and that better transfer learning results are obtained when the source domain has longer textual units than the target domain.

**Figura 13.** Vista de detalle de una publicación

## 4.4. Interfaces para el Académico

Además de las funcionalidades anteriormente mencionadas, el académico es capaz de crear nuevas publicaciones de distintos tipos (libros, conferencias, y journal, entre

otras) mediante un formulario web (Figura 12). Dicho formulario le solicitará la información específica, dependiendo del tipo de publicación a crear, teniendo en común información como el título, autores, áreas de investigación, etc. La figura 13 muestra el formulario que permite crear (dar de alta) a un artículo de revista; allí se pueden ver los campos específicos que el académico debe indicar sobre su publicación.

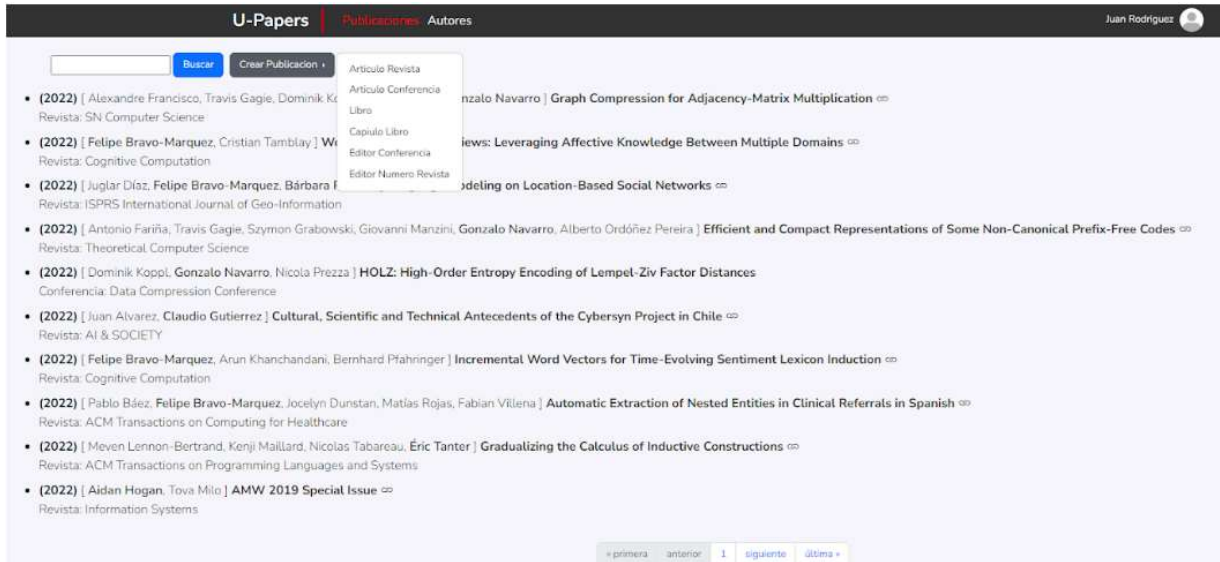


Figura 14. Panel de creación de una publicación según su tipo

Figura 15. Vista de formulario de creación de un artículo de revista

## 4.5. Interfaces para el Administrador

El administrador tiene acceso a la vista de administración, en donde es capaz de crear nuevas instancias que servirán para agregar nuevas publicaciones y componentes de éstas, como por ejemplo: editoriales, autores, conferencias, etc. Usando esta vista también tiene la capacidad de editar información de las instancias ya creadas de publicaciones y autores, entre otras entidades (Figura 14). Por otro lado, el administrador puede gestionar los grupos de usuarios, y la creación/actualización de estos cuando se trata de académicos y/o administradores.

### Administración de Django

Sitio administrativo

AUTENTICACIÓN Y AUTORIZACIÓN		
Grupos	+ Añadir	Modificar
Usuarios	+ Añadir	Modificar

UPAPERS		
Areass	+ Añadir	Modificar
Articulo conferencias	+ Añadir	Modificar
Articulo revistas	+ Añadir	Modificar
Autor publicaciones	+ Añadir	Modificar
Autoress	+ Añadir	Modificar
Capitulo libros	+ Añadir	Modificar
Conferencias	+ Añadir	Modificar
Edicioness	+ Añadir	Modificar
Editor conferencias	+ Añadir	Modificar
Editoriales	+ Añadir	Modificar
Libross	+ Añadir	Modificar
Numero editor revistas	+ Añadir	Modificar
Publicacion	+ Añadir	Modificar
Publicaciones pasivass	+ Añadir	Modificar
Publicaciones sets	+ Añadir	Modificar
Revistas	+ Añadir	Modificar
Roles libros	+ Añadir	Modificar
Roless	+ Añadir	Modificar
Tipo autoress	+ Añadir	Modificar

Acciones recientes

---

**Mis acciones**

Ninguno disponible

**Figura 16.** Vista de administración del sistema (accesible sólo por los usuarios administradores)

## 4.6. Dificultades encontradas en la implementación

Dentro de la implementación de la solución se presentaron diversas dificultades, entre las cuales se pueden destacar las siguientes:

- *Relaciones m2m al importar los datos:* Como se mencionó al inicio del capítulo, la importación de las relaciones many-to-many resultó ser un desafío, puesto que se necesitaba la existencia de ambas instancias a enlazar para poder realizar dicha importación. Para esto se tuvo que utilizar la opción “commit=false” al momento de guardar una instancia, para así evadir las restricciones al momento de crear una instancia que no tiene todos los campos con datos.
- *Orden de los campos de formulario:* La información de las publicaciones se reparte en 2 partes: 1) “Publications” que guarda la información general de las publicaciones, y 2) “JournalPublication”, “ConferencePublication”, etc. que almacenan la información que depende del tipo de publicación. Por motivos de usabilidad, se solicitó que el orden del formulario de creación de publicaciones fuera idéntico a la versión anterior de U-Papers, por lo que fue necesario modificar el formulario intercalando los campos de la tabla general, con los campos de las tablas específicas. Luego se reordenaron los campos en la lógica de lograr guardar la información de la publicación utilizando el Form correspondiente a su tabla (Modelo). Debido a esta personalización, no fue posible la utilización de crispy forms en el formulario de creación de publicaciones.
- *Orden de autores en una publicación:* El orden en que aparecen los autores en las publicaciones es un dato importante que se había pasado por alto al momento del levantamiento inicial, por lo que fue necesario la creación de una tabla intermedia explícita entre la publicación y los autores la cual cuenta con el campo adicional “orden” sobre el cual se especifica el orden en el que deben aparecer los autores al mostrarse la información de una publicación. De esta manera se importaron los autores de las publicaciones respetando el orden de aparición en la publicación. Por otro lado, se tuvo que implementar una forma de guardar el orden en que se agregan los autores en la creación de publicaciones, lo que se logró utilizando JavaScript en el template del formulario.
- *Áreas y subáreas en formulario:* Para poder diferenciar áreas y subáreas en el formulario de creación de publicaciones, se utilizó JavaScript para resaltar las

“áreas de la publicación” en negrita, y en el caso de las “subáreas”, éstas se indentan bajo el área a la cual pertenecen.

## 5. Evaluación de la Solución

Antes de explicar el proceso de evaluación realizado, es importante destacar que este trabajo de memoria estuvo más relacionado con rehacer el back-end de la aplicación, y hacer a la aplicación más integrable al ecosistema del DCC, que con redefinir sus servicios e interfaces de usuario. Si bien esto último se realizó en cierta medida, el trabajo sobre el back-end fue lo más destacable. Por lo tanto, se evaluó principalmente el trabajo sobre el back-end, y en menor medida el trabajo sobre las interfaces de usuario. A continuación se describen ambas evaluaciones.

### 5.1. Evaluación del trabajo de reingeniería del sistema

Se solicitó al Área de Desarrollo de Software (ADS) la evaluación de la nueva versión del sistema U-Papers. Esta evaluación tiene como objetivo determinar el estado actual del sistema, y su capacidad para cumplir con los requisitos actuales y futuros (abordables en el corto plazo). La evaluación se realizó en base al “Checklist de Entrega para Sistemas Desarrollados por Estudiantes” (ver Anexo 1), que evalúa el nivel de madurez de los sistemas construidos por memoristas y tesisistas del Departamento.

La siguiente tabla muestran los ítems evaluados y su cumplimiento:

Ítem	Evaluación	Cumplimiento
B1	La descripción del problema y/o proceso se encuentra presente parcialmente en ésta memoria y más en detalle en las anteriores.	Cumple
B2	Los objetivos y alcance del sistema se encuentra presente parcialmente en ésta memoria y más en detalle en las anteriores.	Cumple
B3	La arquitectura del sistema se encuentra documentada utilizando el C4 model.	Cumple
B4	El stack tecnológico se encuentra documentado y está basado en la plantilla creada por el ADS.	Cumple
B5	El modelo de datos se encuentra actualizado en esta memoria.	Cumple



B6	Las indicaciones para el deployment se encuentran presentes en la forma de archivos Dockerfile y un Docker Compose.	Cumple
B7	La funcionalidad fue validada por Sergio Ochoa, académico jornada completa y guía de este trabajo; además fue validada por Juan Arriagada, profesor experto externo y co-guía de este trabajo.	Cumple
B8*	Opcional. Existe documentación de las APIs que dispone este sistema.	Cumple
Nivel básico alcanzado.		
I1	El código se encuentra formateado utilizando <i>black</i> , una librería de python que adhiere al estándar PEP8.	Cumple
I2	El código fue evaluado utilizando <i>flake8</i> , una librería de python que evalúa la salud del código.	Cumple
I3	Incluye título y descripción del proyecto en README.md	Cumple
I4	Las indicaciones para el deployment se encuentran presentes en la forma de archivos Dockerfile y un Docker Compose.	Cumple
I5	Existe un archivo que contiene todas las variables de entorno necesarias para que el sistema funcione correctamente.	Cumple
I6	El sistema fue evaluado en el ambiente de test del ADS.	Cumple
I7	Se realizaron pruebas con el memorista respecto de las funcionalidades del sistema.	Cumple
I8*	Opcional. El código se encuentra en un VCS propiedad del ADS.	Cumple
I9*	Opcional. El sistema fue evaluado en el ambiente de test del ADS.	Cumple
Nivel intermedio alcanzado.		
A1	La arquitectura del ecosistema se encuentra documentada utilizando el C4 model.	Cumple
A2	No incluye documentos en el CVS que describan qué hace el sistema, y para quién está destinado.	Cumple

A3	No incluye documentos en el VCS que indiquen mejoras futuras.	Cumple
A4	Existe un archivo que contiene todas las variables de entorno necesarias para que el sistema funcione correctamente.	Cumple
A5	El proyecto utiliza Docker. Teniendo Docker, Visual Studio Code y GitHub Desktop en el equipo, fue posible tener el sistema funcionando en menos de 5 minutos.	Cumple
A6*	Opcional. El sistema disponibiliza APIs que permiten la integración con otros sistemas en el ecosistema.	Cumple
A7*	Opcional. El sistema fue evaluado en el ambiente de test del ADS.	Cumple
Nivel avanzado alcanzado.		

**Tabla 2.** Checklist de Nivel de Madurez de Sistemas del DCC

En general la evaluación encontró que la nueva versión de U-Papers es un sistema robusto y escalable que alcanza un nivel de madurez avanzado según el checklist provisto por el ADS del DCC, esto quiere decir que cumple con todos los requisitos y lineamientos del ADS. La verificación del funcionamiento fue realizada en conjunto entre un ingeniero del ADS y el memorista. El detalle de lo que involucra cada ítem evaluado puede ser revisado en el anexo correspondiente.

No obstante lo anterior, se identificó algunas áreas que podrían mejorarse, como el seguimiento de los lineamientos propios de Django, que fueron definidos para la estructuración del código. En las siguientes subsecciones se destacan los aspectos más relevantes de la evaluación.

### 5.1.1 Mantenibilidad del código fuente

El proyecto está basado en la plantilla definida para el DCC, lo que garantiza una organización y estructura estandarizadas, en conformidad con los lineamientos del departamento; además se aprecia el esfuerzo por seguir buenas prácticas en el desarrollo. Con el objetivo de aumentar aún más la compatibilidad con el estilo de desarrollo actual de sistemas del DCC, como trabajo a futuro se sugirió apearse más estrictamente a los lineamientos establecidos por el framework, particularmente para la estructuración del código y para enriquecer la información en el README.md, pensando en los desarrolladores que retomen el proyecto en el futuro.

### **5.1.2 Usabilidad y utilidad del sistema**

Respecto a la usabilidad, el sistema sigue los lineamientos de estilo y despliegue de información propios del DCC. La información es fácil de navegar pudiendo accederse usando no más de tres clicks. Respecto a la utilidad, el sistema brinda un acceso rápido a la información de las publicaciones.

En opinión del ADS, esta herramienta agrega un valor significativo al permitir la visualización de forma responsiva, permitiendo una experiencia de usuario homogénea, a través de los diferentes sistemas públicos del Departamento. Si bien las vistas muestran toda la información relevante, se recomienda, como trabajo futuro, evaluar la usabilidad percibida por los usuarios, por ejemplo utilizando la escala SUS (System Usability Scale) [7] o USE (Usefulness, Satisfaction, and Ease of use) [8]. En base a los resultados obtenidos se determinarán las eventuales mejoras a las interfaces de usuario.

### **5.1.3 Utilidad del API pública**

La API del sistema permite acceder a las mismas funcionalidades de consulta que ofrece U-Papers, pero de manera programática. Esto la convierte en una herramienta útil para permitir que otros sistemas consulten su información, permitiendo desplegar información relevante de manera integrada en otros sistemas del ecosistema del DCC, como por ejemplo, el Sitio Web del departamento o generar estadísticas para las personas o procesos que la requieran. En general se cumple con el formato y los estándares visuales utilizados por el ADS.

### **5.1.4 Verificación del funcionamiento**

Para verificar el correcto funcionamiento del servicio de registro de publicaciones, se cargaron dos publicaciones, por cada uno de los tipos de publicaciones soportadas por el sistema. Eso se realizó utilizando el nuevo formulario web que los académicos utilizan para reportar sus artículos. Los artículos habían sido ya reportados en el sistema legado.

Luego se revisó la información guardada en el nuevo sistema, y se la comparó con aquella registrada en el sistema legado. Como resultado se verificó que el registro de información era igual en ambas fuentes de datos. Por lo tanto, el archivo XML que extrae el coordinador de investigaciones para alimentar el sistema FII/EP (que se muestra en la Figura 1), es equivalente en ambos casos. Eso muestra que el nuevo sistema puede integrarse con otros sistemas a nivel de datos, tal como lo venía realizando el sistema legado.

## 5.2. Evaluación de las nuevas interfaces de usuario

Las interfaces del sistema legado fueron oportunamente probadas por sus usuarios, particularmente por los académicos al momento de su puesta en producción, quienes las consideraban usables. En términos generales se puede mencionar que el proceso de reporte de una nueva publicación en el sistema, involucra cuatro pasos cuando se utiliza el sistema antiguo, y un paso cuando se utiliza el nuevo sistema. Además, el sistema legado no cuenta con interfaces de usuario responsive, mientras que el nuevo sí.

Respecto a los formularios utilizados para declarar una publicación, los del nuevo sistema no sufrieron mayores cambios respecto de los anteriores. Las cosas que se cambiaron fueron principalmente estéticas (adherencia a los fonts y colores institucionales del DCC), y se realizaron simplificaciones menores en los formularios.

Para verificar que el nuevo sistema era al menos similar al anterior en términos de usabilidad, se le pidió a tres académicos revisar y comparar la funcionalidad y estética de tres formularios del sistema, considerando la versión nueva y la versión legada. Los formularios fueron los siguientes: 1) búsqueda de publicaciones, 2) consulta del detalle de una publicación, y 3) declaración de una publicación. Al final de la revisión los evaluadores entregaron sus comentarios.

En términos generales encontraron a ambas versiones parecidas, pero apreciaron que el nuevo sistema adhiriera al patrón estético del DCC. Respecto de los formularios, la búsqueda de artículos en el nuevo sistema es un poco más potente que en el sistema anterior. Respecto a la consulta del detalle de una publicación, ambas versiones fueron percibidas como iguales. Lo mismo pasó con el formulario de carga de una publicación; sin embargo, los usuarios apreciaron no tener que realizar cuatro pasos para declarar un artículo, tal como ocurre en el sistema legado. Por lo tanto, el nuevo sistema es percibido como igual de útil que el anterior, y un poco más usable (agradable al uso).

## 6. Conclusiones y Trabajo a Futuro

Este trabajo de memoria abordó la reingeniería del sistema U-Papers, el cual se ha venido usando como fuente oficial de publicaciones del Departamento durante las últimas dos décadas. Este sistema ha sido intervenido en múltiples ocasiones por estudiantes del DCC, tanto en proyectos de cursos, como en memorias de ingeniería; la aplicación se encuentra en producción y actualmente cumple con su cometido.

En el año 2022 el DCC crea su Área de Desarrollo de Software (ADS) que se encarga de administrar y extender los sistemas del Departamento, incluyendo U-Papers. Para ello, el ADS definió un conjunto de requisitos con los que debe cumplir cada aplicación, para que pueda ser admitida como un sistema Departamental, y a partir de ahí, que pueda ser administrado por dicha unidad. En el 2022 y 2023 se avanzó en esa dirección, en el marco de la memoria reportada en [1]. Sin embargo, quedaron varios requisitos sin cumplir. El principal de ellos fue que el stack tecnológico del sistema legado (basado principalmente en PHP), no se correspondía con lo definido por el ADS (basado en Python). El cambio de tecnologías de back-end, junto con varios otros requisitos, llevó a realizar la reingeniería del sistema.

En el desarrollo del nuevo sistema sólo se reutilizó el conocimiento, los conceptos y parte de los diseños del sistema legado, y fue implementado en su totalidad siguiendo los lineamientos del ADS. El producto resultante es la aplicación reportada en este documento, la cual se encuentra actualmente en el ambiente de testing del ADS. La adherencia de la aplicación a los lineamientos del ADS fue evaluada por un ingeniero de dicha unidad, quien determinó que el producto está listo para ser adoptado y gobernado por esa unidad. Por otra parte, la funcionalidad de la aplicación fue evaluada por tres académicos, quienes determinaron que el nuevo sistema tiene una funcionalidad equivalente al anterior.

Estos resultados indican que los objetivos inicialmente definidos fueron alcanzados, y por lo tanto el sistema podría ponerse en producción en breve. Sin embargo, U-Papers tiene mucha funcionalidad que espera ser agregada, ya sea en futuras memorias o en proyectos internos del ADS. En ese sentido, es importante destacar que a partir de ahora toda extensión que se haga podrá aportar al producto, debido a que la aplicación base ya cumple con los lineamientos del ADS, y además están claras las condiciones bajo las cuales desarrollar dichas extensiones. En ese sentido, el trabajo realizado en esta memoria representa un paso adelante en la dirección que el DCC ha fijado para la evolución de sus sistemas.

Como se mencionó anteriormente, el sistema ya puede ser puesto en producción, pero hay elementos que podrían evaluarse para su futura implementación, y de esa forma darle más valor al servicio entregado. Algunas de las funcionalidades que se podrían mejorar son las siguientes:

- *Manejo de publicaciones desde la plataforma:* Que los autores de una publicación sean capaces de editar la información de ésta, desde dentro de la aplicación, sin la necesidad de ponerse en contacto con los administradores o el coordinador de investigación.
- *Implementación de analítica:* Que el sistema sea capaz de obtener información de las publicaciones más vistas, descargadas, buscadas, etc.
- *Agregar nuevos endpoints:* Estudiar la implementación de nuevos endpoints en la API con el fin de mejorar su funcionalidad y convivencia con el ecosistema de aplicaciones del DCC.

# Bibliografía

- [1] Aguilera, R. (2023). Mejora de la Gobernabilidad de U-Papers y de su Capacidad de Integración al Ecosistema de Software del DCC. Memoria de Ingeniería Civil en Computación. DCC, FCFM, Universidad de Chile.
- [2] Brown, S. (2023). The C4 model for visualising software architecture. LeanPub Publishing.
- [3] Brooke, J. (1995). SUS: A quick and dirty usability scale. Usability Eval. Ind. 189.
- [4] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13 (3): 319-340, doi:10.2307/249008.
- [5] Nielsen, J (2012). Usability 101: Introduction to Usability, <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. Última visita: 30 de Abril de 2024.
- [6] Badiu, R (2016) Mobile Websites: Mobile-Dedicated, Responsive, Adaptive, or Desktop Site?, <https://www.nngroup.com/articles/mobile-vs-responsive/>. Última visita: 30 de Abril de 2024.
- [7] Brooke, J. (1995). SUS: A quick and dirty usability scale. Usability Eval. Ind. 189.
- [8] Gao, M., Kortum, P., & Oswald, F. (2018). Psychometric Evaluation of the USE (Usefulness, Satisfaction, and Ease of use) Questionnaire for Reliability and Validity. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 62(1), 1414-1418.

# Anexo A: Checklist para la Entrega de Sistemas Desarrollados por Estudiantes

En el DCC es común que se desarrolle software en el contexto de trabajos de memoria o tesis de estudiantes. Estos sistemas, cuando son para el DCC, generalmente buscan resolver o comprender problemáticas complejas o procesos con la finalidad de encontrar una primera alternativa para darles soporte.

En este contexto, comprender el nivel de madurez de estos sistemas desde una perspectiva de Gobernabilidad se vuelve fundamental. Para esto se usa como base el documento de [Requisitos de Gobernabilidad e Integración de Sistemas del DCC](#).

Se han definido los siguientes niveles de madurez: [básico](#), [intermedio](#) y [avanzado](#). Un sistema puede cumplir indistintamente puntos relacionados con diferentes niveles de madurez no obstante, para alcanzar un nivel de madurez superior, debe cumplir con todos los requisitos obligatorios del nivel específico.

Para la evaluación de madurez o evolución de la misma se entrega un checklist que el/la memorista o tesista puede incluir en su informe a modo de evaluación de su trabajo o comparar el estado de madurez actual de un sistema respecto de su estado actual. Esta evaluación puede ser realizada por parte del ADS o como trabajo de la persona responsable.

## Nivel básico

En un nivel básico, el sistema debe poseer de manera obligatoria:

1. [Descripción del problema](#) o proceso que busca resolver/abordar. Se recomienda utilizar la herramienta de Tablero Digital (IS2), BPMN o al menos una descripción en texto plano.
2. [Objetivo y alcance del sistema](#). Se recomienda utilizar la herramienta de Tablero Digital (IS2) o al menos una descripción en texto plano.
3. [Arquitectura del software](#). Se recomienda utilizar un framework de especificación de arquitectura como C4model o al menos una descripción clara en texto plano acompañada por diagramas que faciliten su comprensión).



4. [Stack tecnológico](#). Debe incluir lenguajes de programación, plataformas, frameworks, sus versiones y configuraciones necesarias.
5. [Modelo de datos](#). Se recomienda utilizar una herramienta de modelado como MySQL Workbench que permita revisar la coherencia de las relaciones.
6. [Indicaciones para el deployment](#). Debe incluir los pasos y requisitos necesarios para que un tercero pueda levantar un ambiente de desarrollo y recomendaciones para su puesta en producción.
7. Existe evidencia de validación de la funcionalidad por parte de un tercero. Usualmente, la validación del profesor guía es suficiente.

De manera opcional:

8. [APIs que disponibiliza](#), en caso de que posea.

## Nivel intermedio

En un nivel intermedio, el sistema debe poseer de manera obligatoria:

1. El código sigue y se ajusta a algún estándar de codificación (por ejemplo, PEP8).
2. El código utiliza y cumple con las recomendaciones de un linter (por ejemplo, flake8).
3. El README.md incluye: título y una descripción del proyecto.
4. El README.md u otro archivo/documento en el repositorio contiene información detallada sobre cómo desplegar el sistema de forma local.
5. El README.md u otro archivo/documento en el repositorio incluye instrucciones sobre la configuración necesaria para el sistema, incluyendo las variables de entorno necesarias, dependencias y servicios externos necesarios y cómo instalar cualquier dependencia externa.
6. El sistema fue probado en un ambiente de testing con características similares a las de producción.
7. Existe evidencia de que se han realizado pruebas exhaustivas de todas las funcionalidades del sistema.

De manera opcional:

8. El código se encuentra disponible en un VCS de propiedad del ADS.

9. El sistema fue probado en un ambiente de testing provisto por el ADS.

## Nivel avanzado

1. [Arquitectura del ecosistema](#). Se recomienda utilizar un framework de especificación de arquitectura como el C4 model, o al menos una descripción clara en texto plano acompañada por diagramas que faciliten su comprensión).
2. El README.md u otro archivo/documento en el repositorio incluye un resumen conciso que describa qué hace el programa y para quién está destinado.
3. El README.md u otro archivo/documento en el repositorio incluye una sección sobre mejoras futuras tales como nuevas integraciones de API, optimizaciones de rendimiento y/o mejoras de interfaz, destacando también ideas no implementadas en la versión actual pero valiosas para desarrollos futuros.
4. El README.md u otro archivo/documento en el repositorio contiene indicaciones sobre archivos de configuración, variables de entorno y/o servicios externos que deben ser configurados previo a su puesta en test/producción.
5. Es posible levantar un ambiente de desarrollo desde cero en menos de 30 minutos.

De manera opcional:

6. El sistema disponibiliza [APIs](#) que permiten a otros sistemas integrarse con él.
7. El sistema fue probado en un ambiente de testing provisto por el ADS.