



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

U-COMUNIDAD: SISTEMA AUTOMATIZADO QUE CENTRALIZA LA GESTIÓN DE
ROLES ORGANIZACIONALES DE LA COMUNIDAD DEL DEPARTAMENTO DE
CIENCIAS DE LA COMPUTACIÓN

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

NICOLÁS SEBASTIÁN DELGADO VEGA

PROFESOR GUÍA:
DANIEL PEROVICH GEROSA

MIEMBROS DE LA COMISIÓN:
PATRICIO POBLETE OLIVARES
CÉSAR GUERRERO SALDIVIA

SANTIAGO DE CHILE
2018

Resumen

La comunidad del Departamento de Ciencias de la Computación de la Universidad de Chile (DCC) está compuesta por muchas personas que cumplen diferentes roles. Esta información se encontraba en diversas fuentes, varias difíciles de consultar de forma automática, algunas con información incompleta y otras con información contradictoria entre ellas.

La información de los miembros de la comunidad y los roles que cumplen se hace necesaria de tener centralizada y disponible para automatizar procesos como autorización y generación de contenidos dirigidos en la nueva plataforma de microservicios del DCC.

El objetivo de este trabajo es proveer al DCC del sistema *U-Comunidad*, software que centraliza la información actualizada tanto de los miembros de la comunidad del departamento como sus roles organizacionales, permitiendo su consulta por los componentes de la plataforma de aplicaciones.

Por lo anterior, se realiza un levantamiento de información sobre los roles de la comunidad, encontrando las fuentes que se deberán consultar y la necesidad de un microservicio, llamado *Gestión de Personal*, que gestione los roles sin fuente.

Finalmente se entrega al DCC el resultado del levantamiento de información de los roles y sus fuentes, la implementación de *U-Comunidad* y el prototipo del microservicio de *Gestión de Personal*.

Por todo su esfuerzo, a mis padres.

Agradecimientos

Gracias a tod@s por quienes pude llegar hasta aquí.

Gracias a mis padres por darme todo lo necesario para dedicarme a estudiar.

Gracias *Bagus* por ser mi guía espiritual y quien está cuando nadie más puede.

Gracias a mi ex-compañera por haberme dado el apoyo para no rendirme.

Gracias Mila, por haber llegado en el peor momento de mi vida, mantenerme a flote y tirarme para arriba con su cariño y conversas :3.

Gracias a Varitas por mantener la esperanza de que la U aún puede cambiar y podemos mejorar nuestra profesión.

Gracias a Jorgi3, Ardilom, Gabo y tod@s l@s que me recibieron en su hogar cuando lo necesité.

Gracias al *culto* por todo el apoyo recibido en los últimos años y por tantos buenos momentos.

Gracias a mis compañeros de la Mente Popular por mantener el trabajo de hormiga tratando de mejorar nuestro Pudahuel.

Gracias a Daniel Perovich, mi profesor guía, quien siempre estuvo pendiente de que este trabajo saliera adelante ofreciendo su ayuda en lo que necesitara y comprendiendo los malos momentos por los que pasé el 2017.

Finalmente, muchas gracias a tod@s l@s que compartí una conversación en la salita, una cerveza, pizza o baile por ahí, los que me dieron ánimo cuando estaba estresado y los que queremos cambiar este mundo por uno mejor para tod@s.

Tabla de contenido

1. Introducción	1
1.1. Problema	2
1.2. Objetivos	4
1.3. Metodología	5
1.4. Estructura del documento	5
2. Conceptualización de U-Comunidad	6
2.1. Visión de <i>U-Comunidad</i>	6
2.2. Escenarios de uso	8
2.2.1. Seguridad	8
2.2.2. Contenidos Dirigidos	9
2.2.3. Notificaciones Dirigidas	10
2.2.4. Alertas de inconsistencia	10
2.3. Requisitos Funcionales	11
2.4. Análisis de los roles organizacionales	11
2.4.1. Detección de roles	12
2.4.2. Homologación de roles	13
2.4.3. Elección de fuentes	15
3. Diseño de U-Comunidad	19
3.1. Contexto del sistema	19
3.2. Arquitectura lógica del sistema	20
3.3. API del servicio	22
3.4. Modelo de Información	24
3.5. Escenarios de uso	25
3.5.1. Seguridad y contenidos dirigidos	25
3.5.2. Notificaciones dirigidas	26
3.5.3. Alerta de inconsistencia	28
4. Implementación y Validación	29
4.1. Implementación del sistema <i>U-Comunidad</i>	30
4.1.1. Microservicio de <i>U-Comunidad</i>	31
4.1.2. Agente de U-Campus	33
4.1.3. Gestión de Roles	35
4.2. Validación	36
4.2.1. Servicio de Gestión del Personal	36

5. Conclusiones y trabajo a futuro	42
Bibliografía	43
A. Roles	45

Índice de tablas

2.1.	Resumen de las fuentes y cantidad de roles encontrados.	12
2.2.	Roles Homologados I.	15
2.3.	Roles Homologados II.	16
2.4.	Roles y Fuentes agrupadas.	17
2.5.	Fuentes escogidas para los grupos de personas.	18
3.1.	API de Miembros de la Comunidad, <i>people</i>	23
3.2.	API de Roles, <i>roles</i>	23
3.3.	API de las organizaciones, <i>organizations</i>	23
3.4.	API de los perfiles de los miembros de la Comunidad, <i>profiles</i>	24
3.5.	API de las relaciones entre Miembros de la Comunidad y sus roles, <i>has_role</i>	24
3.6.	API de consultas útiles	24
4.1.	Códigos de Carreras en <i>U-Campus</i> correspondientes a pregrado, magíster y doctorado.	33
A.1.	Lista de todos los roles homologados I.	45
A.2.	Lista de todos los roles homologados II.	46
A.3.	Lista de todos los roles homologados III.	47
A.4.	Roles obtenidos desde entrevista a Jefa Administrativa.	48
A.5.	Roles obtenidos desde el directorio telefónico de la página del DCC, pedido a <i>Sistemas</i>	49
A.6.	Roles obtenidos desde la página del DCC I.	50
A.7.	Roles obtenidos desde la página del DCC II.	51
A.8.	Roles obtenidos desde la página del DCC III.	52
A.9.	Roles obtenidos de las páginas blancas de la universidad.	52
A.10.	Roles obtenidos de las listas de correos actuales del DCC.	52
A.11.	Roles adicionales obtenidos del Área de Gestión Estratégica.	53

Índice de ilustraciones

2.1. Esquema de la arquitectura de microservicios del DCC	7
2.2. Estructura Organizacional del DCC.	14
2.3. Parte del árbol de jerarquías de los roles	17
3.1. U-Comunidad como microservicio	20
3.2. Justificación de la independencia de los agentes	21
3.3. Arquitectura Lógica de Referencia.	21
3.4. Arquitectura concreta.	22
3.5. Modelo de información.	25
3.6. Autenticación en el portal del DCC.	26
3.7. Representación de Contenidos dirigidos en el portal del DCC.	26
3.8. Representación de Autorizaciones distintas en el portal del DCC.	26
3.9. Notificación de un evento o noticia DCC automatizado	27
3.10. Notificación de un servicio	27
3.11. Alerta de inconsistencia	28
4.1. Implementación para validación de <i>U-Comunidad</i> junto con el prototipo de Servicio de <i>Gestión del Personal</i>	30
4.2. Modelo de la Base de Datos de <i>U-Comunidad</i>	32
4.3. Vista del login del prototipo	37
4.4. Vista de la redirección a login de U-Pasaporte	37
4.5. Vista de bienvenida a un estudiante	38
4.6. Vista de bienvenida a un funcionario	38
4.7. Vista del perfil a un estudiante	39
4.8. Vista del perfil a un funcionario con poder de edición	39
4.9. Vista del perfil completo de un estudiante	40
4.10. Vista del detalle de un rol, se puede ver la lista de ruts asociados.	41

Capítulo 1

Introducción

El Departamento de Ciencias de la Computación (DCC) de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile tiene como misión ser un centro de excelencia en docencia, investigación y extensión para Chile y el extranjero en diversas áreas de las Ciencias de la Computación. El cumplimiento de esta misión está a cargo del equipo de académicos y personal de colaboración del Departamento, cuya principal responsabilidad es la operación. La operación involucra tres pilares: docencia, investigación y extensión. En docencia, el DCC está a cargo de cursos de pregrado del plan común, de la carrera de pregrado Ingeniería Civil en Computación, y de tres programas de postgrado: dos magíster y un doctorado. En investigación, el DCC tiene una gran trayectoria y reconocimiento en publicaciones científicas, y ejecuta proyectos de investigación y cooperación, tanto nacionales como internacionales, con financiamiento concursable provenientes de diversas fuentes. En extensión, el DCC cuenta con un área de comunicaciones responsable de la difusión de las actividades del departamento por diversos medios digitales, con un área de educación continua a cargo de los diplomas de actualización profesional, y con un área de vinculación externa responsable del vínculo con egresados y con la industria, contexto en el cual se realizan también proyectos como asesorías y peritajes, entre otros. Además, el DCC cuenta con áreas de gestión y servicios, responsables de la dirección y administración del Departamento.

La comunidad del DCC está conformada por más actores que académicos y personal de colaboración. En ella participan también los estudiantes, quienes tienen un rol fundamental en docencia, ya sea como alumnos de los cursos, como auxiliares en el dictado de algunos cursos, y en la organización del estudiantado en el Centro de Alumnos del DCC. Los estudiantes juegan un rol importante en investigación, principalmente los estudiantes de postgrado, participando en la co-autoría de artículos científicos con académicos del Departamento. Los egresados también forman parte de la comunidad, quienes participan en diversas actividades del Departamento, como ser comisiones evaluadoras, cursos de diplomados, y la autoevaluación de las carreras en los procesos de acreditación, entre otros. La comunidad incluye también a académicos e investigadores de otros departamentos e instituciones que son convocados, visitan o colaboran con el DCC en variadas actividades.

Así, cada miembro de la comunidad del DCC aporta a éste desde al menos un *rol*. Conocer qué personas comprenden la comunidad del Departamento, y qué *roles* juegan en ella,

es de gran relevancia para el Departamento. Conocer a la comunidad habilita y apoya a un conjunto importante de actividades que se dan en la operación. Un ejemplo son los procesos de acreditación de carreras a cargo de las áreas de docencia. En estos procesos es necesario conocer qué estudiantes, académicos y egresados se vieron involucrados en el último período, no solo porque deben listarse en los informes de la acreditación, sino porque deben ser encuestados para conseguir retroalimentación e identificar oportunidades de mejora de las carreras. Otro ejemplo es el apoyo a la extensión, principalmente a la difusión y la educación continua, las cuales deben llegar con notificaciones y contenidos dirigidos a ciertos grupos de la comunidad, como ser noticias y eventos relevantes del DCC, el inicio de períodos de inscripción a cursos de educación continua, etc.

En el Departamento, no era posible conocer fácilmente quiénes eran los miembros de la comunidad del DCC ni qué roles cumplían en ella, mucho menos la historia de los miembros y sus roles. El conocimiento de la conformación de la comunidad existía, pero estaba disperso en diversos medios digitales o físicos, e incluso, en algunos casos, este conocimiento solo residía en las personas. Un ejemplo de esto último es el rol de ex-director del Departamento. Mientras que el director actual se encuentra anunciado en la página web del DCC, los directores anteriores no lo están, y no es tarea sencilla intentar reconstruir la historia de quiénes fueron directores.

En lo referente a medios digitales, el DCC cuenta con un conjunto de aplicaciones de apoyo a la operación. Algunas son provistas por la Universidad o la Facultad, como *U-Campus* que apoya la gestión curricular de los estudiantes y el funcionamiento administrativo de la Universidad, y *U-Cursos* que gestiona la dinámica del dictado de los cursos. Otros son provistos por terceros, como *PG Scpio*, que mantiene seguimiento detallado de los estudiantes y egresados de las carreras de postgrado. Otros han sido desarrollados internamente en el DCC, como *U-Proyectos*, que gestiona el registro de los proyectos de investigación y extensión, *U-Papers*, que mantiene el registro de las publicaciones con afiliación DCC, *U-Fondos*, que apoya la administración de los fondos de investigación de los académicos, *U-Productividades*, que ayuda en la gestión de las productividades de académicos y personal de colaboración, *U-Vinculación*, que administra la información de seguimiento de los egresados, entre otros. A su vez, los sitios web de la Universidad, Facultad y del propio DCC proveen información respecto a algunos *roles organizacionales* y quiénes los cumplen, como quién es el Director, el Subdirector, el Jefe Docente o el Coordinador de Postgrado. El DCC cuenta, además, con wikis en donde se indican también otros *roles organizacionales*, como la conformación de los comités académicos y de los claustros de las carreras de postgrado.

1.1. Problema

Un análisis de estas aplicaciones, sitios web y wikis, permitió identificar diversos problemas. Primero, la información está descentralizada, por lo que no es posible consultar una única fuente que provea información de la comunidad completa. Segundo, las diversas fuentes tienen información redundante, ya que múltiples aplicaciones refieren a miembros de la comunidad y cada una de ellas guarda información de estos miembros. Por ejemplo, tanto *U-Proyectos* como *U-Papers* tiene la lista de académicos del DCC. Otro ejemplo es que tanto *U-Campus*

como *PG Scopio* tienen la lista de estudiantes de postgrado. Tercero, las diversas fuentes tienen información inconsistente, tanto en los nombres de los *roles* como en qué personas los cumplen. Incluso se detectaron fuentes que reportan *roles* obsoletos, es decir, que en la actualidad ya no es considerado un *rol organizacional*.

Así, el Departamento se enfrenta a los siguientes problemas:

- Los sitios web de la Universidad, Facultad y del Departamento tienen información potencialmente inconsistente, ya que estos sitios se alimentan de fuentes de información diferentes, en algunos casos dependiendo de la edición manual de los sitios web.
- La presencia de la misma información en múltiples sistemas requiere que los académicos y/o el personal de colaboración ingrese y actualice manualmente la información múltiples veces, lo cual ocurre frecuentemente ante cada cambio en quienes cumplen los roles de gestión.
- Se requiere de un esfuerzo considerable para ensamblar listados de miembros de la comunidad con roles específicos, debido a la diversidad e independencia de las múltiples fuentes en las que se encuentra la información. Esto ocurre cada vez que se necesita difundir una notificación a un grupo específico de la comunidad, pues las listas generadas previamente deben ser revisadas ya que se construyen manualmente.
- La diversidad de las fuentes no permite que otras aplicaciones del DCC hagan uso de la información de la comunidad, sino que, en cambio, deban proveer un mecanismo propio de registro de la información relevante para cada aplicación. Esto ha sucedido en el desarrollo de todas las aplicaciones internas del DCC.

El contar con un sistema que centralice la información actualizada y la exponga a otras aplicaciones del DCC, facilitaría y agilizaría escenarios que, de ocurrir hoy en la operación se solucionarían con un esfuerzo manual significativo. En particular, un sistema de estas características permitiría:

- **Mejora de la seguridad de las aplicaciones.** La mayoría de las aplicaciones del DCC basan la autorización en términos de roles del sistema, esto es, llevan un registro de qué personas cumplen qué rol del sistema, y en función de eso, las aplicaciones otorgan los permisos correspondientes a cada usuario. El usar roles del sistema obliga a re-configurar la seguridad en las aplicaciones cada vez que hay un cambio en el cumplimiento de un *rol organizacional*. Por ejemplo, ante cada cambio de Director o de los miembros Consejo del Departamento, cada aplicación que utilice esta información debe re-configurarse para otorgar los permisos correspondientes a las nuevas personas. Contar con un sistema centralizado que provea información de qué *roles organizacionales* cumple cada persona le permite a las aplicaciones basar la autorización en términos de *roles organizacionales*, en lugar de hacerlo en términos de roles del sistema, evitando así la re-configuración manual y replicación de esfuerzos en cada aplicación.
- **Generación de contenidos dirigidos.** Entregar a cada miembro de la comunidad los contenidos y funciones específicas a sus *roles* en la organización. Como ejemplo, un portal web disponible en la intranet del Departamento puede utilizar la información de los roles del usuario para proveerle aquellos contenidos o funciones que son relevantes a sus *roles*. Así, mientras a un miembro del Consejo del Departamento se le proveerían las actas de las sesiones de consejos anteriores, a un coordinador de un programa de post-

grado se le proveerían las próximas defensas de tesis y a un estudiante se le proveerían los cursos de su carrera y los horarios de clase, etc.

- **Generación de notificaciones dirigidas.** Enviar a grupos particulares de miembros de la comunidad, por un medio digital como el correo electrónico, notificaciones específicas dependiendo de los *roles* que cumplan en la organización. Como ejemplo, un sistema de notificaciones podría informar a los académicos acerca de las futuras charlas de investigación que se impartirán en el DCC. Hoy, estas notificaciones las envía el área de Comunicaciones utilizando listas de correo que gestiona manualmente el área de Sistemas del DCC.
- **Alerta de inconsistencias en la información externa.** Los sitios web de la Universidad y Facultad reportan información acerca de quiénes cumplen algunos roles organizacionales en el DCC. Estos sitios web no son gestionados por el DCC directamente, y cualquier cambio en esta información debe ser solicitado siguiendo un procedimiento administrativo. El seguir este procedimiento administrativo depende de que un académico o personal de colaboración recuerde hacerlo. Contando con la información centralizada, habilitaría la detección automática de inconsistencias entre la información del sistema y los sitios web externos, recordando a quién corresponda corregir la información en estos últimos.

En este trabajo de memoria se definió la arquitectura del sistema *U-Comunidad* que actúa como guía en la construcción del sistema y en la evolución de éste al momento de integrar nuevas *fuentes* de información. A su vez, se implementó una primera versión del sistema con la capacidad de centralizar la información y exponerla a otras aplicaciones mediante un API REST [2]. Esta primera versión extrae automáticamente la información de estudiantes y egresados de *U-Campus* y permite, a través de un nuevo servicio, gestionar los *roles* del personal del Departamento, ya que no se contaba con ninguna *fuentes* de información que cumpliera con dicha función.

1.2. Objetivos

El objetivo general de este trabajo de memoria es proveer al DCC de una fuente centralizada de información acerca de los miembros de su comunidad y los *roles organizacionales* que ellos cumplen, mediante un servicio de información que capture de forma automática y periódica la información de otras aplicaciones consideradas como *fuentes* válidas.

Para lograr lo anterior, se establecen los siguientes objetivos específicos:

- Definir los *roles organizacionales* e identificar *fuentes* válidas de consulta para extraer los miembros que cumplen cada rol.
- Definir la arquitectura del servicio de información que dé los lineamientos para el desarrollo y la evolución del servicio teniendo en cuenta el contexto provisto por el Área de Aplicaciones del DCC.
- Implementar el servicio de información que actúe como repositorio centralizado de los miembros de la comunidad del DCC y de los *roles organizacionales* que ellos cumplen, exponiendo la información como un servicio de consulta por otras aplicaciones.

- Implementar un agente que extraiga de forma automática y periódica de *U-Campus* los miembros de la comunidad correspondientes a estudiantes y a egresados de las carreras del DCC.
- Construir una aplicación que permita administrar los roles directivos y de gestión del DCC, y que actúe como *fuentes* de información para el servicio de información.

1.3. Metodología

1. Investigar y determinar cuáles son todos los *roles organizacionales* existentes en el DCC.
2. Investigar y determinar cuáles son las aplicaciones existentes que pueden actuar como *fuentes* de información para ser consultadas automáticamente por el sistema.
3. Determinar la correspondencia entre los *roles organizacionales* y las *fuentes de información*, seleccionar cuáles de estas *fuentes* son recomendables para utilizar, e identificar qué *roles organizacionales* requerirán de una nueva aplicación para gestionarlos y actuar como su *fuentes* de información.
4. Priorizar los *roles organizacionales* según su urgencia y utilidad en las *aplicaciones* existentes en el DCC y determinar cuáles serán incluidas en el alcance de este trabajo.
5. Diseñar la arquitectura del servicio de información que dé los lineamientos para el desarrollo y la evolución del servicio.
6. Implementar el repositorio de información del servicio y exponerlo mediante una API REST a otras aplicaciones.
7. Implementar un *agente* para extraer la información de los *roles organizacionales* prioritarios desde la *fuentes* más relevante.
8. Implementar una aplicación que permita administrar los roles del personal del DCC, ya que no se cuenta con una, que actúe como *fuentes* de información para el servicio *U-Comunidad*.
9. Validar la arquitectura y la implementación de *U-Comunidad*, evaluando la información cargada automáticamente en el servicio de información a través del *agente* implementado.
10. Escribir el documento de memoria.

1.4. Estructura del documento

Este informe de memoria tiene la siguiente estructura. El Capítulo 2 presenta la conceptualización del servicio de información *U-Comunidad*, definiendo su visión, escenarios de uso, y el análisis de los *roles organizacionales* del DCC. El Capítulo 3 describe el diseño del servicio de información mediante la definición de su arquitectura lógica y la resolución de los escenarios de uso. El Capítulo 4 explica la implementación y validación del servicio de información y sus principales componentes. Finalmente, el Capítulo 5 discute las conclusiones y lecciones aprendidas, e identifica posibles líneas de trabajo futuro.

Capítulo 2

Conceptualización de U-Comunidad

En este capítulo se presenta la conceptualización de *U-Comunidad*, el cuál se insertará en la plataforma de aplicaciones del DCC. Primero se describe la visión, cuya característica principal es actuar como un servicio de información de los *roles organizacionales* del departamento y los miembros de la comunidad que los cumplen. Luego se discuten los escenarios de uso de *U-Comunidad*, que son los motivadores de las funcionalidades concretas que éste ofrece, definiendo así, los principales requisitos funcionales del sistema. Por último, se presenta el análisis que se realizó para identificar los *roles organizacionales* del DCC, su estructura y dependencia, y para determinar cuáles son las *fuentes* de información confiables para la recuperación automática de la información que realiza el sistema.

2.1. Visión de *U-Comunidad*

El Área de Aplicaciones del DCC está migrando la plataforma de aplicaciones que dan el soporte a la operación y gestión del departamento, desde un conjunto de aplicaciones monolíticas independientes a un conjunto de microservicios que favorecen la interoperabilidad, la componentización y el reuso, minimizando la información repetida y las inconsistencias. Con esta nueva arquitectura, se busca brindar a los miembros de la comunidad del DCC un portal integrado que les provea los contenidos que ellos necesitan y que les asista en las tareas que ellos realizan en su operación diaria. El portal es responsable de la interfaz de usuario de la plataforma, y delega a un conjunto de microservicios la gestión y procesamiento de la información. Cada microservicio es responsable de una entidad de negocio relevante en el DCC, como son las publicaciones, los proyectos, las noticias, los eventos, los fondos de investigación, las notificaciones automatizadas, etc.

U-Comunidad representa un nuevo microservicio de esta plataforma. La Figura 2.1 ilustra cómo *U-Comunidad* se integrará con otros microservicios de la plataforma, los cuáles son responsabilidad del Área de aplicaciones del DCC. En la figura, los servicios del DCC aparecen en gris, mientras que los externos en blanco. En línea punteada se pueden ver los elementos implementados en este trabajo. En esta plataforma, un miembro de la comunidad utilizará el portal, el cuál brindará acceso a la información y a las funciones de negocio. A

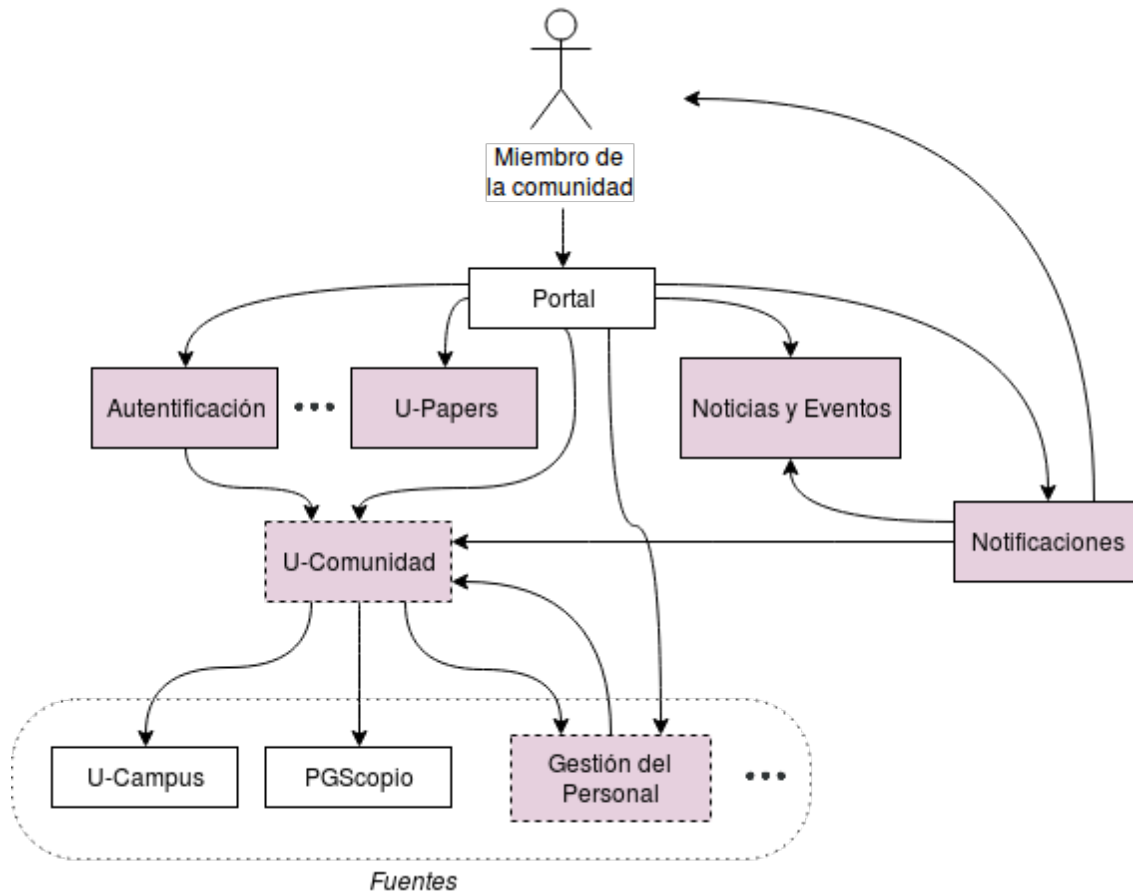


Figura 2.1: Esquema de la arquitectura de microservicios del DCC

su vez, el portal, interactúa con los microservicios para acceder, procesar y almacenar información. Por ejemplo, *U-papers*, *Gestión de Personal* o *Notificaciones y Eventos*. Otros como *Autenticación* y *U-Comunidad* tienen otras funcionalidades más allá de manipular información.

A continuación se describirá cómo otros microservicios se vincularán con el nuevo.

La seguridad es un aspecto fundamental para el portal dado que deberá permitir la entrada solo a las personas que pertenecen a la comunidad y autorizar los servicios según el rol que cumple la persona que accede. Para validar quién puede ingresar, el portal se comunicará con un sistema encargado de la autenticación, como *U-Pasaporte*, el cual junto con *U-Comunidad* entregará la información necesaria para que el portal despliegue la información y el acceso a los servicios de acuerdo a los roles organizacionales que el usuario cumpla en la comunidad.

El sistema de *Noticias y Eventos* tiene por función ayudar al Área de Comunicaciones del DCC en la difusión de los acontecimientos importantes en el Departamento. La información manejada por este sistema se utiliza para su publicación en las pantallas instaladas en los corredores del DCC y puede utilizarse también para su publicación en el sitio web del Departamento.

El sistema de *Notificaciones* busca enviar avisos y alertas de forma dirigida a ciertos miembros de la comunidad a través de correo electrónico. Dentro de los temas a comunicar se encuentran las noticias y eventos gestionados por el sistema anteriormente nombrado, pero teniendo en cuenta la diversidad de la comunidad, se necesita discriminar el público objetivo de la información. Una de las formas es discriminar según el rol organizacional del miembro a comunicar. Por ejemplo, mientras a los académicos se les recuerda la fecha de las próximas charlas de académicos visitantes, a los estudiantes se les puede recordar la fecha de la próxima asamblea.

Por otro lado, el detalle de la información interna de los roles del DCC se concentra en el sistema de *Gestión de Personal*, el cuál se retroalimenta con *U-Comunidad* al ser *fuentes* y un consumidor a la vez. Este sistema fue concebido durante el trabajo de esta memoria al encontrar que no existe una *fuentes* que contenga esta información disponible para los demás servicios.

Como se ve, *U-Comunidad* será un pilar de todos los servicios descritos anteriormente y a la vez, cada uno de ellos le será útil a otros como el servicio de *Notificaciones* al de *Noticias y Eventos* o al de *Monitoreo de Contenido Externo*.

2.2. Escenarios de uso

En el Capítulo 1 se indicaron cuatro beneficios principales de contar con un sistema o servicio como *U-Comunidad*: proveer *contenidos dirigidos*, enviar *notificaciones dirigidas*, *mejorar la seguridad* y permitir comparar información para emitir *alertas de inconsistencia*. Estos beneficios representan los principales escenarios de uso del sistema.

2.2.1. Seguridad

En términos de seguridad, las aplicaciones del DCC se enfocan en dos aspectos principales: *autenticación* y *autorización*.

Para la autenticación, las aplicaciones del DCC utilizan uno de los siguientes mecanismos para validar las credenciales de un usuario: (1) mecanismo propio de la aplicación, almacenamiento y gestión de una lista de usuarios y credenciales independiente del resto de la plataforma (se usa en *U-Papers* y *U-Fondos*), (2) *Sistema de Administración de Usuarios (SAU)*, gestión manual de usuarios y credenciales a cargo del Área de Sistemas (se usa en *U-Proyectos*, *U-Productividades* y *U-Vinculación*), o (3) delegar a *U-Pasaporte*¹ (se utiliza en *Entregas de MTI* o *U-Dashboard*). En los dos primeros se requiere una gestión manual, lo que requiere tiempo y es propenso a errores. Aunque el último es automático y no necesita gestiones de parte del DCC, actualmente no permite distinguir entre quiénes son miembros de la comunidad del DCC ni se pueden registrar miembros temporales de la comunidad como profesores visitantes o proveedores. Tampoco se pueden registrar roles temporales como la

¹<https://www.u-cursos.cl/upasaporte/index>

directiva de los estudiantes.

Para la autorización, las aplicaciones del DCC utilizan uno de los siguientes mecanismos para determinar a qué funciones o contenidos tiene acceso un usuario: Mecanismo propio del sistema para asociar usuarios a roles de sistema o *SAU*, que tiene registradas algunas de las aplicaciones del DCC, sus roles de sistema (como por ejemplo *Administrador*, *Visitante*) y los usuarios que tienen cada *rol*. En ambos casos la administración es manual y los roles de sistema cambian constantemente junto con los cargos de las personas, lo que se traduce en posibles errores y costos de gestión.

Las aplicaciones *Entregas de MTI* y *U-Fondos*[4] son algunas de las pocas aplicaciones del DCC que utilizan *roles organizacionales* para autorizar a usuarios las funcionalidades del sistema, en lugar de utilizar roles de sistema (que no están asociados a los *roles organizacionales*). Sin embargo, estas aplicaciones deben registrar en sus propias bases de datos las listas de usuarios que tienen cada *rol organizacional* relevante para la aplicación.

Contar con *U-Comunidad* permite a las aplicaciones del DCC consultar qué *roles organizacionales* posee el usuario que se está autenticando, sin necesidad de mantener la lista de usuarios en la propia aplicación, lo que obliga al Área de Sistemas del DCC a mantenerlas actualizadas manualmente. Contar con *U-Comunidad* permite y promueve, además, que otras aplicaciones puedan utilizar un modelo de autorización basado en *roles organizacionales*.

2.2.2. Contenidos Dirigidos

Actualmente los académicos y funcionarios del DCC utilizan un sistema llamado Intranet² que consiste en un portal que lista las aplicaciones del DCC. El portal le presenta a cada usuario solamente aquellas aplicaciones a las que él tiene acceso. La configuración del acceso a las aplicaciones se obtiene del Sistema de Administración de Usuarios (SAU)³. Este sistema tiene dos problemas principales. Por un lado, dado que la lista de aplicaciones está incompleta, el portal no presenta todas las aplicaciones utilizadas por los usuarios. Por otro lado, la configuración de acceso que registra el SAU no considera los *roles organizacionales* de los usuarios. Como consecuencia de esto, cualquier cambio en quién tiene un *rol organizacional* (por ejemplo cuando cambia el Director del Departamento), requiere que los accesos sean reconfigurados manualmente.

Contar con *U-Comunidad* permite ofrecer *contenidos dirigidos* a los usuarios de acuerdo a su *rol organizacional*. El portal puede utilizar *U-Comunidad* para determinar qué roles corresponden al usuario que lo está usando. Así, el portal brindaría a cada usuario acceso a los servicios y contenidos correspondientes a sus *roles organizacionales*. Lo anterior permite eliminar la necesidad de gestionar usuarios y sus permisos en cada aplicación al delegarlo al servicio centralizado de *U-Comunidad*, lo cual dejaría obsoleto al *SAU*. A su vez, en la nueva versión de la plataforma, la Intranet evolucionará al portal

²<https://intranet.dcc.uchile.cl/>

³<https://intranet.dcc.uchile.cl/saudcc/>

2.2.3. Notificaciones Dirigidas

El DCC cuenta con un servicio que permite registrar todas las noticias y eventos del Departamento⁴, el cuál está siendo migrado a la plataforma de microservicios. Se espera que este servicio de *noticias y eventos* alimente de información al sitio web, a los televisores informativos y al portal del DCC. Además, el Área de Aplicaciones del DCC construyendo el servicio de Notificaciones que permite enviar notificaciones de eventos a los miembros de la comunidad del DCC. El Sistema de Notificaciones extraerá los eventos del servicio de Noticias y Eventos, y cuenta con reglas de distribución y suscripción para determinar a quién notificar de los eventos vía correo electrónico. Las reglas de distribución están basadas en los *roles organizacionales* de los miembros de la comunidad del DCC.

Contar con *U-Comunidad* permitirá al Sistema de Notificaciones determinar cuáles son los destinatarios concretos para cada tipo de notificación. Dado un nuevo evento en la Aplicación de Noticias y Eventos, el Sistema de Notificaciones aplicará las reglas que tiene configuradas para determinar los *roles organizacionales* que deben ser notificados, y utilizará *U-Comunidad* para determinar a qué destinatarios enviar la notificación.

2.2.4. Alertas de inconsistencia

Como se describe en la Sección 2.4.1 y se muestra en Apéndice A, son muchas las páginas web en las que se comunica la información de cuáles son los miembros de la comunidad del DCC que tienen los distintos *roles organizacionales*. Muchas veces la información es inconsistente, incompleta o incorrecta. Por ejemplo, la página del Magíster en TI en el sitio web de la Universidad tiene una versión en español⁵ y una versión en inglés⁶. A la fecha de escritura de este informe, la versión en español está actualizada, indicando a Cecilia Bastarrica como *Coordinadora del Programa*. Sin embargo, la versión en inglés aún no está actualizada, la cual indica a José Pino como *Coordinador*. Para entender el problema, hay que considerar que tal cambio se realizó a mediados del 2016 y este informe se escribió a fine del 2017, es decir, el sitio lleva más de un año mostrando información equivocada.

Contar con *U-Comunidad* permitirá construir servicios que procesen la información del DCC publicada en los sitios web de la Facultad y de la Universidad, y que alerten automáticamente en caso de que haya una discrepancia en la información. Esto le permitirá al Área de Administración del DCC solicitar la actualización de las páginas web o actualizar la información en la fuente de información que le proveyó la información incorrecta a *U-Comunidad*.

⁴ <https://comunicaciones.dcc.uchile.cl/>

⁵ <http://uchile.cl/u50703>

⁶ <http://uchile.cl/u98673>

2.3. Requisitos Funcionales

Dado los escenarios de uso de *U-Comunidad* descritos en la Sección 2.2, se identifican los siguientes requisitos funcionales:

1. **Consulta de los roles organizacionales de una persona.** Este servicio recibe un identificador de un miembro de la comunidad (ya sea su RUT, correo electrónico o uno generado por el sistema), y retorna una lista con los *roles organizacionales* que tiene la persona en el Departamento.
2. **Consulta de las personas que tienen un rol organizacional.** Este servicio recibe un *rol organizacional* del Departamento y entrega una lista con las personas que actualmente cumplen dicho rol en la comunidad.
3. **Consulta del perfil de una persona.** Este servicio recibe un identificador (mismo del punto 1) y entrega la información de perfil de esa persona (como su nombre, RUT, contacto, etc).
4. **Descubrimiento de las personas que tienen los roles organizacionales desde fuentes externas.** Esta función requiere la consulta frecuente de las fuentes de información identificadas, la extracción de la información de las personas asociadas a los *roles organizacionales*, la comparación con la información de *U-Comunidad* y, cuando corresponda, la actualización de los datos en el sistema.

Los tres primeros requisitos deben estar disponibles a través de una API REST para otros servicios de la plataforma de microservicios del departamento. Mediante esta API se concretan los escenarios de uso discutidos en la Sección 2.2. El cuarto requisito es una función interna del sistema que mantiene la base de datos actualizada.

2.4. Análisis de los roles organizacionales

Antes de poder utilizar los roles de la comunidad, se hizo una revisión preliminar de ellos para conocerlos. Tras algún tiempo de investigación, se detectaron varios problemas: no había una fuente única de consulta, las *fuentes* se contradecían o usaban términos distintos para referirse a lo mismo, entre otros detalles. Debido a lo anterior se decidió formalizar el análisis para poder determinar los roles organizacionales del departamento, para lo cual se siguieron 4 pasos. En primer lugar, se **detectaron los roles** consultando variadas fuentes en busca de los *roles* existentes en el departamento. La lista completa se puede ver en las Tablas A.4, A.5, A.6, A.7, A.9, A.8, A.10 y A.11 del Apéndice A. En esta etapa se encontraron varios problemas en los nombres de los roles como inconsistencias en las personas a cargo de dichos roles, e incluso roles que se encuentran obsoletos. En segundo lugar, al haber encontrado inconsistencias en los roles detectados, se **homologaron** construyendo una lista unificada. Para ello se siguió la estructura organizacional del DCC definida por el Área de Gestión Estratégica (ver Figura 2.2), ordenando todos los roles encontrados en el primer paso. En tercer lugar, tras la homologación, se encontraron roles que no tenían cabida en la estructura organizacional del DCC, por lo que se **construyó una lista de recomendaciones** para refinar dicha estructura. Finalmente, se **escogieron las fuentes de información** que se

Cantidad de Roles	Fuente
73	Sitio web del DCC
25	Jefa Administrativa
16	Directorio telefónico en el sitio web del DCC
9	Área de Gestión Estratégica
7	Listas de Correos
7	Páginas Blancas de la Universidad

Tabla 2.1: Resumen de las fuentes y cantidad de roles encontrados.

considerarán oficiales para *U-Comunidad* de entre todas las revisadas para extraer los *roles organizacionales*.

2.4.1. Detección de roles

Para la detección de roles organizacionales se contó con el apoyo de la Jefa Administrativa del DCC, del Área de Gestión Estratégica y del Área de Sistemas, quienes dieron información sobre los roles del departamento. Además, se consultaron los sitios web de la Universidad, de la Facultad y del Departamento. Como resumen, en la Tabla 2.1 se muestran los medios y personas consultadas con la cantidad de *roles* que se encontraron en cada una.

Analizando los *roles* encontrados, se pueden observar varios problemas:

- **Gestión descentralizada**

No existe un único lugar para consultar por todos los *roles* existentes en la comunidad del DCC y hay trabas técnicas para recuperar toda la información disponible de algunas fuentes. Por ejemplo, en las *páginas blancas de la Universidad*⁷, solo se puede consultar por funcionarios y tiene un límite de 15 resultados por búsqueda. También, *U-Campus* posee mucha información, incluyendo egresados, estudiantes y roles administrativos, pero falta información importante para el Departamento como saber que Alexandre Bergel es el *Coordinador de las Prácticas Profesionales*⁸ o que Sergio Ochoa es *Coordinador del PEC*⁹.

- **Inconsistencias de nombre**

Se utilizan palabras distintas para referirse al mismo *rol*. Por ejemplo, el *sitio web del DCC*¹⁰ y la *guía telefónica de U-Campus*¹¹, se refieren a Magna Bornand como *Secretaria de Dirección*, mientras que el *directorio telefónico del sitio web del DCC*¹² se refiere a ella como *Asistente Dirección*.

- **Diferencias entre instituciones**

Para una misma persona, el Departamento, la Facultad y la Universidad muestran *roles*

⁷<http://www.blancas.uchile.cl/index.php>

⁸Ver Tabla A.1, rol 30

⁹Ver Tabla A.2, rol 36

¹⁰<https://www.dcc.uchile.cl/coordinaciones>

¹¹https://ucampus.uchile.cl/m/guia_telefonica/?q=Magna+Bornand+Cepeda

¹²<https://www.dcc.uchile.cl/guia>

distintos. Por ejemplo, Agustín Villena aparece en las *páginas blancas de la Universidad*¹³ como *Instructor Adjunto*, mientras que en la *guía telefónica de U-Campus*¹⁴ se encuentra como *Académico, Profesor Adjunto* y en el *sitio web del DCC*¹⁵ se muestra como *Profesor Jornada Parcial*.

- **Inconsistencia en las personas que cumplen los roles.**

Para un mismo rol, se encontraron lugares donde la información estaba desactualizada. Por ejemplo, en la *página de postgrado del sitio web del DCC*¹⁶ aparece José Pino como *Coordinador Académico del Programa de Magíster en Tecnologías de la Información*, siendo que dejó de serlo hace un año. Actualmente Cecilia Bastarrica es la coordinadora del programa.

- **Roles obsoletos**

Se encontraron *roles* que ya no existen en el DCC. Por ejemplo, ningún integrante de la comunidad posee el cargo de *Coordinador de Sistemas*, que se indica en el *sitio web del DCC*¹⁷ o el de *Coordinadora IEPI* que se encuentra en otra página del mismo sitio¹⁸.

2.4.2. Homologación de roles

En el 2014, el Área de Gestión Estratégica del DCC capturó la estructura organizacional del Departamento mostrado en la Figura 2.2. Desde entonces, el Área la ha refinado en sucesivas instancias. El relevamiento que se realizó para el sistema *U-Comunidad* (Sección 2.4.1 y Apéndice A) le brinda la oportunidad al área de refinar la estructura organizacional y de homologar los *roles* detectados a ésta.

En las Tablas 2.2 y 2.3 se presenta la homologación de *roles* considerando las fuentes originales de información mostradas en el Apéndice A. Se hace notar que en esta primera versión, el *estado* del rol se mantiene guardado en el rol mismo. Por ejemplo, un estudiante de pregrado puede estar “congelado” o “regular”, entre otros. Para efectos de esta memoria se considerarán roles distintos: “Estudiante Pregrado” y “Congelado Pregrado”

Roles no clasificados. La gran parte de los *roles* encontrados se pudo ajustar a la estructura organizacional del DCC. Sin embargo, dado que quedaron fuera algunos *roles*, es necesario que el modelo se refine. Para ello, se provee a continuación una lista de recomendaciones de mejora a la estructura organizacional, de forma tal que considere todos los *roles organizacionales* detectados.

- Académicos de Jornada Completa: Podría ir en una nueva sección de Recursos Humanos.
- Académicos de Jornada Parcial: Podría ir en una nueva sección de Recursos Humanos.

¹³<http://www.blancas.uchile.cl/resultado.php>

¹⁴https://ucampus.uchile.cl/m/guia_telefonica/?q=agustin+villena

¹⁵<https://www.dcc.uchile.cl/ajp>

¹⁶https://www.dcc.uchile.cl/contactos-programa-de-postgrado_1218

¹⁷<http://www.dcc.uchile.cl/sistemas>

¹⁸<https://www.dcc.uchile.cl/ingenieria-de-ejecucion-en-procesamiento-de-la-informacion>



Figura 2.2: Estructura Organizacional del DCC.

- Profesional: Podría ir en nueva sección de Recursos Humanos.
- Traductor de inglés: Podría ir en Servicios, Traducciones.
- Encargado del Museo: Podría ir en Extensión, Museo.
- Directiva CaDCC: Podrían ir en Pregrado, CaDCC: Presidente, Vicepresidente, Director de Secretaría y Comunicaciones, Director de Finanzas, Director de Docencia.

Roles Agregados. Para facilitar el trabajo de las aplicaciones externas que usen *U-Comunidad*, se detectaron una serie de roles *agregados*, éstos no son más que la unión de otros roles “hijos”. Por ejemplo: el rol “Estudiante del Magíster”, corresponde a la unión de los “Estudiantes de Magíster en Ciencias” y los “Estudiantes de Magíster TI”. Lo anterior se puede ver en la Figura 2.3, donde se muestra el árbol de jerarquía para los estudiantes regulares.

Estudiante	Pregrado	Carrera	Jefe Docente
Comité Docente		Prácticas Profesionales	Coordinador
Jefe de Estudios		CC1001	Coordinador
Secretaria Docente		CC1002	Coordinador
Egresado		Proyecto de Software	Coordinador
Comité Técnico Docente (CTD)		Especialidades	Coordinador Académico Área Gestión Informática
			Coordinador Académico Área Ciencia e Ingeniería de Datos
			Coordinador Académico Área Informática Educativa
Coordinador Académico Área Seguridad Computacional			
Coordinadora Académica Área Ingeniería y Calidad de Software			
	Titulación	Coordinador	
Estudiante	Posgrado	Magíster TI	Claustro
Comité			Comité Académico
Coordinador			Coordinador
Secretario			Estudiante
Jefe de Estudios			Profesor Colaborador
			Profesor Visitante
		Magíster CS	Claustro
			Comité Académico
			Coordinador
			Estudiante
Profesor Colaborador			
Profesor Visitante			
Doctorado		Claustro	
		Comité Académico	
		Coordinador	
		Estudiante	
		Profesor Colaborador	
		Profesor Visitante	

Tabla 2.2: Roles Homologados I.

2.4.3. Elección de fuentes

A grandes rasgos, los *roles* se pueden organizar en cinco grandes grupos: estudiantes, egresados, personal de colaboración, académicos y cargos reconocidos internamente en la comunidad del DCC. A su vez, estos cinco grupos se pueden clasificar en tres categorías de acuerdo a la relación que mantienen con las *fuentes* de información. En la Tabla 2.4 se puede ver el resumen de los grupos, su clasificación y sus *fuentes*.

	Extensión	Proyectos Externos	Coordinador
		PEC	Coordinador de Consultorías
			Asistente de Coordinación
			Coordinador
			Ejecutivo en ventas
			Ejecutivo programas corporativos
			Jefe
		Profesor	
Vinculación Externa	Coordinador		
Coordinador	Investigación	Publicaciones	.
Secretario		Redes Académicas	.
		Proyectos de investigación	.
		Postdoctorado	Postdoctor
	Administración	Jefe Administrativo	
		Administrativo	
		Secretaria de Administración	
	Dirección	Consejo departamental	
		Director	
		Secretaria de Dirección	
		Subdirector	
	Gestión Estratégica	Asistente	
Coordinador			
Acreditación	.		
Servicios	Comunicaciones	Periodista	
		Coordinador	
		Coordinador Página Web	
		Coordinador Revista Bits	
	Aplicaciones	Coordinador	
	Sistemas (Plataforma)	Jefe	
		Subjefe	
Técnico			
Infraestructura	Auxiliar		

Tabla 2.3: Roles Homologados II.

La *Categoría A* incluye a los estudiantes y egresados. Las fuentes de dónde se puede extraer la información para esta categoría son *U-Campus*, que tiene información de los estudiantes y egresados de todas las carreras, y *PG Scopio*, que solo tiene información de las carreras de postgrado. Estas fuentes permiten consultar y extraer el universo completo de las personas de la categoría.

La *Categoría B* refiere a los nombramientos de quienes trabajan en el DCC, incluyendo al personal de colaboración y a los académicos. La guía telefónica del DCC en *U-Campus*¹⁹ provee una lista del personal del DCC. Sin embargo, esta lista está incompleta ya que, por

¹⁹[https://ucampus.uchile.cl/m/guia_telefonica/?q=gru_id:"fcfm.ciencias_computacion"](https://ucampus.uchile.cl/m/guia_telefonica/?q=gru_id:)

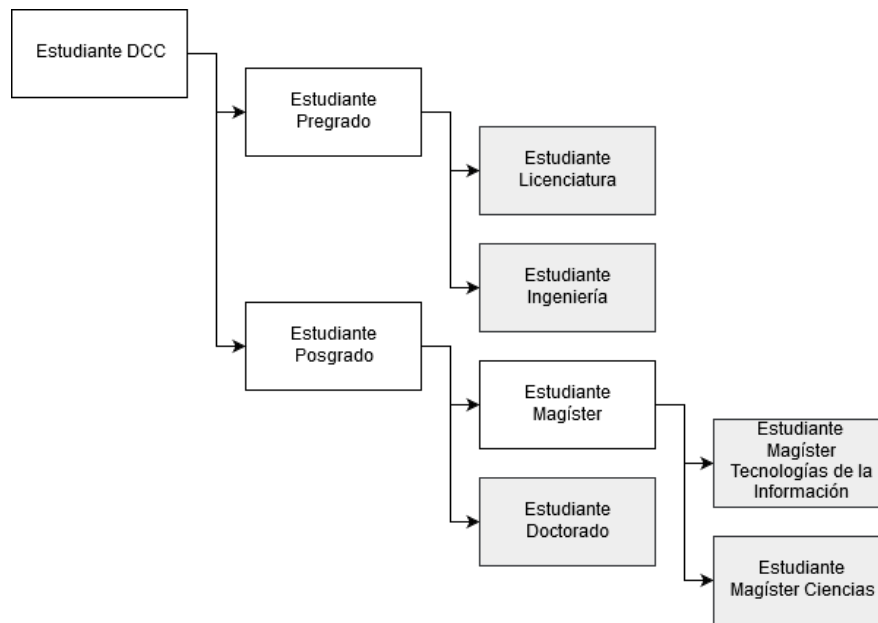


Figura 2.3: Parte del árbol de jerarquías de los roles

Categoría	Grupo	Fuente
A	Estudiantes	U-Campus: Todos
	Egresados	PG Scopio: Solo posgrado
B	Personal de colaboración	Páginas blancas y de transparencia de la Universidad
	Académicos	
C	Cargos del DCC	No hay fuente única completa

Tabla 2.4: Roles y Fuentes agrupadas.

ejemplo, los Auxiliares del DCC no están registrados. La página de transparencia de la Universidad²⁰ tiene la información de los nombramientos del personal. Sin embargo, esta página refiere a todos los nombramientos de la Universidad y no se puede filtrar para obtener los nombramientos del DCC directamente, a lo sumo se puede filtrar para obtener los nombramientos de la Facultad. Por lo tanto, esta página permitiría conocer el nombramiento de cada persona, pero no permitiría conocer la lista de todas las personas. Por otro lado, las páginas blancas de la Universidad²¹ sí permiten filtrar por las personas contratadas en el DCC, pero brinda un límite de 15 resultados como máximo por búsqueda, por lo que tampoco permite construir la lista completa de forma directa.

La *Categoría C* incluye a todos los cargos o responsabilidades internas del Departamento, los que principalmente se encuentran mencionados en el sitio web del DCC. Sin embargo, este sitio, a pesar de haber servido de fuente de información para la detección de roles, no servirá de fuente de información para el sistema *U-Comunidad*, sino que, por el contrario, se quiere que sea *U-Comunidad* quien le brinde al sitio web del DCC la información de los miembros de la comunidad y sus cargos. Manteniendo la información centralizada en *U-Comunidad*, todas las aplicaciones del departamento incluyendo el sitio web utilizarán la misma información.

²⁰<http://www.uchile.cl/portal/presentacion/informacion-publica/dotacion-de-personal>

²¹<http://www.blancas.uchile.cl/>

Categoría	Fuente	Grupo
A	U-Campus: Todos PG Scopio: Claustro de Posgrado	Estudiantes
		Egresados
B y C	Servicio de Gestión del Personal (SGP)	Personal de colaboración
		Académicos
		Cargos del DCC

Tabla 2.5: Fuentes escogidas para los grupos de personas.

Otras fuentes de información fueron descartadas por no aportar información vigente, completa o suficientemente detallada. Por ejemplo, el sitio web de la Facultad donde se listan los académicos del DCC²² contiene información equivalente a la que entrega las *páginas blancas de la Universidad* y permite saber quiénes son los Académicos de Jornada Parcial y de Jornada Completa. Sin embargo, esta página no indica los nombramientos específicos de cada académico, es decir, si es Profesor Adjunto, Instructor Adjunto, Profesor Asistente, Profesor Asociado, Profesor Titular o Experto Externo.

Por lo tanto, para las *Categorías B y C* no se cuenta con una fuente de información completa, confiable y actualizada. En consecuencia, se necesitó crear una nueva *fuentes* que se llamó “*servicio de gestión del personal*” (SGP), que permite al Área de Administración del Departamento gestionar esta información y que la provea a *U-Comunidad*. Para esta nueva fuente se usa la lógica definida para el sistema *U-Dashboard* [3], el cual, entre otras funcionalidades, fue diseñada para la gestión de cargos y *roles* del personal del DCC.

Finalmente, las *fuentes* escogidas para los grupos de personas se muestran en la Tabla 2.5.

²²<http://uchile.cl/i92399>

Capítulo 3

Diseño de U-Comunidad

Como se explicó en el Capítulo 2, el sistema es parte de la plataforma de microservicios del DCC, y fue concebido para permitir la realización de los escenarios de uso identificados y discutidos en la Sección 2.2, manteniendo un repositorio con la información actualizada y centralizada de los miembros de la comunidad del DCC y sus roles organizacionales.

En este capítulo se describe el diseño del sistema. Primero, se explica el contexto del sistema y los principales servicios y sistemas con los que interactúa. Segundo, se describe la arquitectura del sistema, incluyendo su estructura modular y la API ofrecida. Se detalla también el modelo de información, que provee una abstracción de la información que gestiona el sistema. Por último, se explica la interacción del sistema con los actores externos para cada escenario de uso identificados.

3.1. Contexto del sistema

En la visión de *U-Comunidad* presentada en la Sección 2.1, se lo concibe como un servicio central a la plataforma de microservicios del Departamento. Su función principal es ser la única fuente en la plataforma en lo referente al perfil de los miembros de la comunidad del DCC y a los roles organizacionales que ellos cumplen, proveyendo esta información al resto de los servicios de la plataforma a través de una API. Para cumplir con esta funcionalidad, *U-Comunidad* debe mantener actualizada la información de los perfiles y los roles organizacionales extrayendo la información periódicamente de *fuentes* externas. La Figura 3.1 muestra el contexto operacional del servicio *U-Comunidad* y su interacción con otros sistemas.

Describiendo el uso del servicio, por el lado de los consumidores de *U-Comunidad*, se tienen todos los servicios de la plataforma descrita en el la Sección 2.1. Para comunicarse con el sistema, deberán hacerlo a través de la API, la cual responde a tres peticiones: información de perfil de un miembro de la comunidad, los roles que cumple un miembro y los miembros asociados a cierto rol.

De forma paralela e independiente de lo anterior, *U-Comunidad* debe mantener actualizada

la información que provee, por lo que, para cada *fente* definida, tiene un proceso particular que extrae esa información. Estos procesos, conocidos como *agentes*, están especializados en la fuente que le corresponde consultar y no interfieren entre sí. De este modo, mientras algunos deben procesar HTML, otros se comunicarán a través de APIs de otros sistemas.

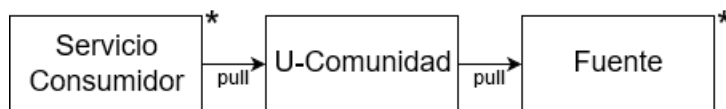


Figura 3.1: U-Comunidad como microservicio

3.2. Arquitectura lógica del sistema

La arquitectura del sistema *U-Comunidad* se decidió basarlo en el patrón de arquitectura *Blackboard* [1]. Este patrón se aplica en sistemas en que la información que este maneja debe evolucionar y enriquecerse en el tiempo, y esta evolución depende de un conjunto de conocimientos expertos específicos. Este patrón organiza un sistema en tres tipos de componentes: (1) un único componente **Blackboard** que tiene la responsabilidad de gestionar y almacenar la información del sistema, (2) un conjunto de componentes **Knowledge Source** (KS, fuentes de conocimiento en inglés), que tienen la capacidad de utilizar y actualizar el Blackboard con un fin específico, ya sea para extraer información o para refinarla según el conocimiento que materializa cada KS, y (3) el **Control**, que representa la responsabilidad de determinar cuándo cada KS debe ejecutarse. Hay varias formas de definir el control: (a) realizarlo como un componente independiente que decide cuándo debe ser ejecutada cada KS y las ejecuta, usualmente en base a la información que el control visualiza en el Blackboard, (b) realizarlo en el Blackboard, el cual ante cambios en la información que gestiona invoca a las KS que deben actuar en consecuencia, (c) realizarlo en el Blackboard y en las KS, donde el Blackboard dispara un evento del cambio ocurrido en su información y las KS reaccionan ante tales eventos, y (d) realizarlo en cada KS, cuando son éstas las que tienen el conocimiento de cuándo necesitan ejecutarse.

Para el sistema *U-Comunidad*, se decidió que el *Microservicio U-Comunidad* cumpla la responsabilidad del **Blackboard**, y, por lo tanto, gestione la información de los miembros de la comunidad y sus roles organizacionales. Además, se definen dos tipos de KS, uno correspondiente a los *Servicios Consumidores* que utilizar a *U-Comunidad* para consultar información, y otro correspondiente a los *agentes* cuya responsabilidad es extraer la información de las fuentes externas y poblar el blackboard. El primer tipo corresponde a sistemas externos al sistema *U-Comunidad*, a los cuales se les provee una API que ofrece la funcionalidad que éstos requieren. Dado que la plataforma de microservicios interopera mediante solicitudes web, se decidió proveer el API de *U-Comunidad* como un API REST. El segundo tipo corresponde a componentes internos a *U-Comunidad*. Los *agentes* son parte esencial del sistema con la responsabilidad de mantener actualizada la información del mismo. Para la realización de los agentes se consideraron dos alternativas: (i) incluirlos dentro del propio microservicio, y (ii) tratarlos como microservicios independientes. Dado que el sistema *U-Comunidad* evolucionará en el tiempo y se prevé la inclusión de más agentes (por ejemplo, extrayendo información

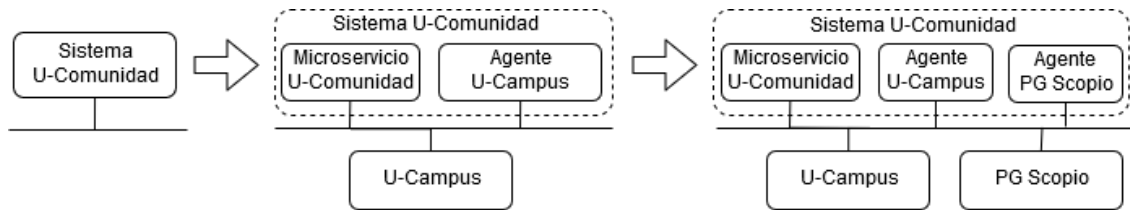


Figura 3.2: Justificación de la independencia de los agentes

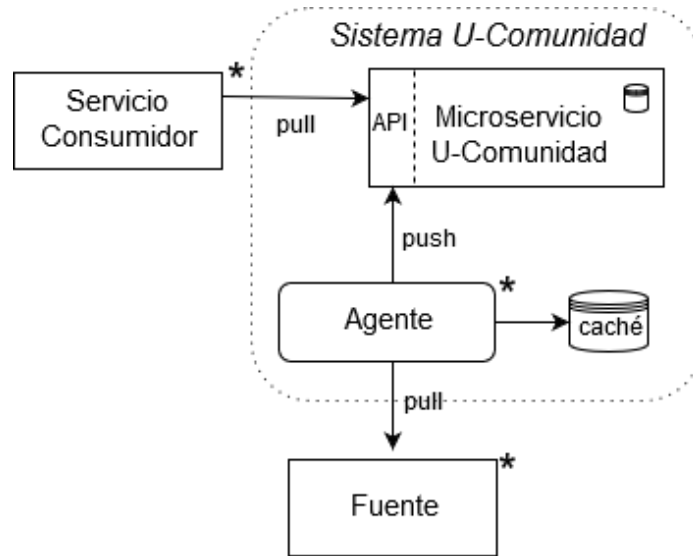


Figura 3.3: Arquitectura Lógica de Referencia.

de otras fuentes para la definición de roles más específicos como el haber participado en proyectos, que puede obtenerse de u-proyectos), se decidió que los agentes sean microservicios independientes. De esta forma, cada vez que se incluya construya nuevo agente, éste no implica ningún cambio en la implementación ni en la instalación del *microservicio U-Comunidad*. La Figura 3.2 ilustra este paso de evolución. Dado que los agentes son microservicios independientes, el API del *microservicio U-Comunidad* debe proveer además las funciones necesarias para manipular la información que gestiona. Esto está alineado con lo que recomienda el patrón blackboard, en el que éste gestiona la información y las KS le solicitan actualizarla reflejando el conocimiento que representan.

Para el **Control** se decidió utilizar el mecanismo en que cada KS determina cuándo le corresponde ejecutar. Para el caso de las KS correspondientes a *Servicios Consumidores*, son éstos que deciden cuándo requieren la información del blackboard, por lo que la decisión es apropiada. Para el caso de las KS correspondientes a *agentes*, son éstos quienes tienen el conocimiento de cuándo la información que saben extraer está obsoleta y, por lo tanto, debe ser actualizada. Los agentes no necesitan la información del blackboard para determinar cuándo deben ejecutarse, sino que ellos mismos tienen configurada la periodicidad con la que deben re-procesar la información.

La arquitectura de referencia del sistema *U-Comunidad* se ilustra en la Figura 3.3. Es una arquitectura de referencia ya que no refiere a ningún agente o fuente de información en concreto. Esta arquitectura se usa como referencia en la evolución del sistema, guiando cómo

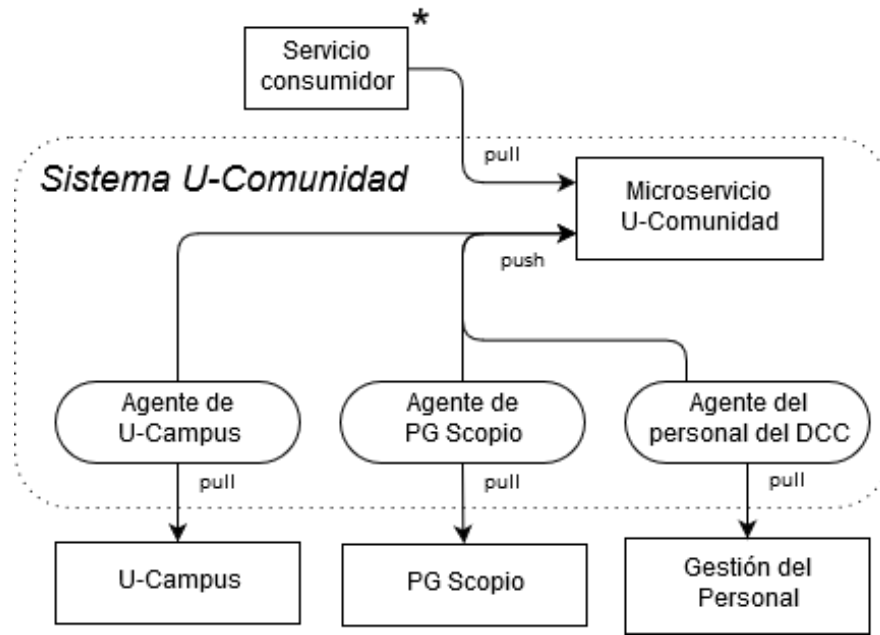


Figura 3.4: Arquitectura concreta.

pueden incorporarse nuevas fuentes.

Arquitectura concreta

Siguiendo la Arquitectura de referencia mostrada en la Figura 3.3 y, teniendo en cuenta las *fuentes* detectadas en la Sección 2.4.3 y resumidas en la Tabla 2.5, la arquitectura concreta se puede observar en la Figura 3.4. Las *fuentes* son *U-Campus*, *PG Scopio* y *Servicio de Gestión del Personal (SGP)*. Los roles que se extraen de cada uno están enumerados en la Tabla 2.5, desde *U-Campus* se extraen los roles de los estudiantes: *estudiante*, *congelado*, *desertor* y *egresado*; desde *PG Scopio* se extrae la información de los claustros de posgrado, mientras que los roles del personal están en la nueva *fuentes*, *SGP*.

3.3. API del servicio

Para comunicarse con el resto de la arquitectura de microservicios, *U-Comunidad* dispone una API de lectura, con la que se cubren los tres primeros requisitos de la sección 2.3. La API también habilita la escritura para que los *agentes* puedan modificar el *blackboard*, las consultas de lectura más importantes están descritas a continuación., mientras que las de escritura se pueden ver en las Tablas 3.1, 3.2, 3.3, 3.4, 3.5 y 3.6.

- **Rol → Lista de Identificadores de miembros de la comunidad**

GET */roles/id*, entrega el rol que tiene el *id* dado junto con la lista de identificadores de los miembros de la comunidad que cumplen dicho rol. El identificador de un rol se

estructura de acuerdo al área de la organización al que pertenece, por ejemplo, el rol *secretario* del área de Gestión será “*gestion-secretario*”.

- **Identificador de miembro de la comunidad → Detalles del perfil y Lista de Roles**

GET */people/id*, entrega el perfil de la persona y la lista de identificadores de roles que cumple. El *id* puede ser el RUT, el correo o el identificador asignado por el sistema.

A continuación se listan las tablas y sus APIS correspondientes: Con los *endpoints* de la Tabla 3.1 se gestiona la información de los miembros de la comunidad, con los de la Tabla 3.2, los roles y con los de la Tabla 3.3, las organizaciones. En la Tabla 3.4 se muestran las acciones posibles sobre la relación entre miembros de la comunidad y sus perfiles mientras que en la Tabla 3.5 se muestra la API sobre la relación de miembros y sus roles. Finalmente, en la Tabla 3.6 se dejan las consultas a todos los roles, todos los usuarios y todas las relaciones entre ellos.

Método	Path	Descripción
GET	<i>/people/id</i>	Retorna perfil del miembro de la comunidad correspondiente a id junto con una lista de los roles que cumple en la comunidad
POST	<i>/people/id</i>	Crea miembro de la comunidad correspondiente a id
PUT	<i>/people/id</i>	Actualiza miembro de la comunidad correspondiente a id
DELETE	<i>/people/id</i>	Elimina miembro de la comunidad correspondiente a id

Tabla 3.1: API de Miembros de la Comunidad, *people*

Método	Path	Descripción
GET	<i>/roles/id</i>	Retorna rol correspondiente a id, junto con una lista de todos los identificadores de miembros de la comunidad que cumplen el rol.
POST	<i>/roles/id</i>	Crea rol correspondiente a id
PUT	<i>/roles/id</i>	Actualiza rol correspondiente a id
DELETE	<i>/roles/id</i>	Elimina rol correspondiente a id

Tabla 3.2: API de Roles, *roles*

Método	Path	Descripción
GET	<i>/organizations/id</i>	Retorna organización correspondiente a id
POST	<i>/organizations/id</i>	Crea organización correspondiente a id
PUT	<i>/organizations/id</i>	Actualiza organización correspondiente a id
DELETE	<i>/organizations/id</i>	Elimina organización correspondiente a id

Tabla 3.3: API de las organizaciones, *organizations*

Método	Path	Descripción
POST	/profiles/id	Crea perfil de miembro correspondiente a id
PUT	/profiles/id	Actualiza perfil de miembro correspondiente a id
DELETE	/profiles/id	Elimina perfil de miembro correspondiente a id

Tabla 3.4: API de los perfiles de los miembros de la Comunidad, *profiles*

Método	Path	Descripción
POST	/has_role?role=id&people=id	Asocia al miembro de la comunidad con el rol
DELETE	/has_role?role=id&people=id	Elimina asociación entre miembro de la comunidad con el rol

Tabla 3.5: API de las relaciones entre Miembros de la Comunidad y sus roles, *has_role*

Método	Path	Descripción
GET	/roles/	Retorna la lista de todos los roles
GET	/people/	Retorna la lista de todos los miembros de la comunidad
GET	/has_role/	Retorna la lista de todas las relaciones de miembros con roles

Tabla 3.6: API de consultas útiles

3.4. Modelo de Información

El modelo de información provee una abstracción de la información del sistema como se observa en la Figura 3.5. El modelo está conformado por:

- **Organizaciones.** Contiene las divisiones de las áreas y subáreas del departamento. Por ejemplo: *Pregrado*, *Titulación*, *Postgrado* o *Doctorado*. Una organización puede ser parte de otra como *Titulación* lo es de *Pregrado*. De esta forma es posible modelar y capturar la estructura organizacional del DCC presentada en la Figura 2.2.
- **Roles.** Contiene los *roles* del departamento, los cuáles podrán ser de dos tipos: *Simples* o *Agregados*. Cada rol pertenece a una organización.
- **Roles Simples.** Contiene los *roles* directos de la tabla homologada (ver Figuras A.1, A.2 y A.3).
- **Roles Agregados.** Son los *roles* que representan la unión de otros *roles*. Por ejemplo los *egresados* de postgrado son los egresados de los dos Magíster y del Doctorado, mientras que los *egresados* del DCC son los egresados de pregrado y de postgrado. Esto evita tener que registrar a los usuarios a múltiples *roles* debido a los *roles agregados*. Por ejemplo, es suficiente registrar a un usuario como *egresado* de *Doctorado* para que sea considerado también como *egresado* de postgrado y del DCC.
- **Miembros de la comunidad.** Contiene la información mínima para identificar a los miembros que cumplen los *roles organizacionales*. El identificador puede ser el RUT, el correo o el identificador que el sistema le da automáticamente.
- **Perfil del miembro.** Contiene la información detallada de los miembros de la comunidad por cada fuente consultada en el sistema, como por ejemplo: RUT, nombres, apellidos, foto, correo, dirección, número telefónico, etc.

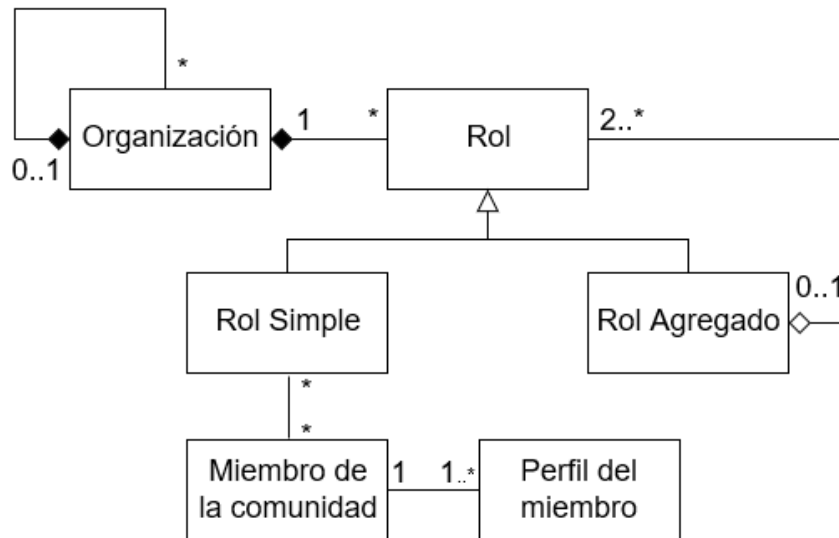


Figura 3.5: Modelo de información.

3.5. Escenarios de uso

A continuación se muestran ejemplos de cómo se integra *U-Comunidad* con otros servicios para cumplir con los escenarios de uso descritos en la Sección 2.2.

3.5.1. Seguridad y contenidos dirigidos

En este escenario se muestra el ejemplo de uso de ambas características al entrar al portal del DCC. Cuando se habla de Seguridad, se abordan solo dos aspectos de ella: la *autenticación* y la *autorización*. La primera significa que el usuario asegura que es quien dice ser, mientras que la segunda se refiere a los permisos que tendrá la persona dentro de la aplicación. Es aquí donde se involucra *U-Comunidad*.

Los contenidos dirigidos se refieren a la capacidad de filtrar los contenidos para ciertos usuarios dependiendo de los roles que cumpla. En cierta forma, es equivalente a la autorización.

Tanto la seguridad como los contenidos dirigidos son evidenciados en el escenario en que un usuario entra al portal del DCC mostrado en la Figura 3.6.

Para poder acceder al portal, primero es necesario autenticarse. Como se vió en la Sección 2.1, esto lo hace un servicio especializado, el que puede ser (I) interno del departamento como (II) externo, por ejemplo *U-Pasaporte* o cualquier sistema que use *oauth*, como google. En el caso (I), en donde se autentifica con un servicio del DCC, el portal muestra un formulario de login, en el que el usuario ingresa sus credenciales, si el sistema lo valida, consulta a *U-Comunidad* por sus roles. Con ello arma un *token* que manda al portal, el cual redirige al usuario. Mientras en el caso (II), ilustrado en la Figura 3.6, autenticación con servicio externo como *U-Pasaporte*, tras llegar al portal (1), se redirige al usuario a la página de

U-Pasaporte (2), donde coloca sus credenciales y el sistema junto con validarlo, consume un *endpoint* del portal para validar los permisos del usuario (3). En este *endpoint* el portal obtiene los roles desde *U-Comunidad* (4) y construye el *token* que envía a *U-Pasaporte* (5), donde el usuario es redirigido a una URL que posee el *token* en su interior (6). Ya dentro del portal, éste sabe quién es y qué roles cumple el usuario (7).

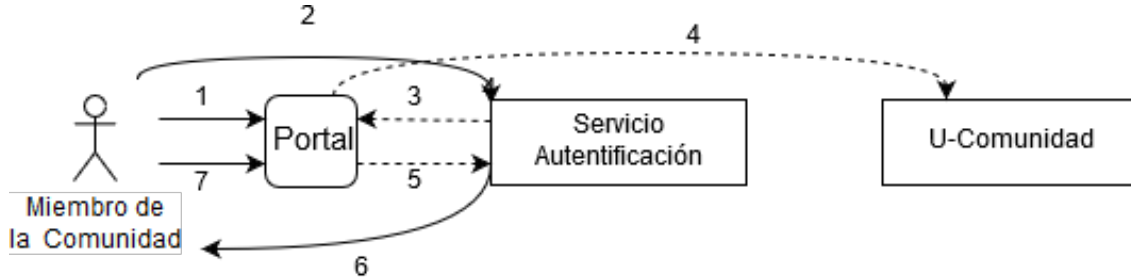


Figura 3.6: Autenticación en el portal del DCC.

Una vez autenticado, ocurren los dos casos explícitos de contenidos dirigidos y autorización. Usando el *token* construido, el portal sabe los roles que cumple el usuario sin necesidad de consultar nuevamente a *U-Comunidad*. El usuario verá su página de inicio distinta según los roles que cumpla como ilustra la Figura 3.7. Por ejemplo, las noticias para un estudiante de pregrado puede aparecer la fecha de la próxima inscripción académica, mientras que en las noticias de un académico la fecha de la próxima evaluación. Por otro lado, dentro de las mismas vistas, los permisos pueden ser distintos como se ilustra en la Figura 3.8. Por ejemplo: suponiendo que el estudiante y el ayudante pueden ver las notas de una persona en particular, el ayudante puede borrar, mientras que el estudiante solo ver.

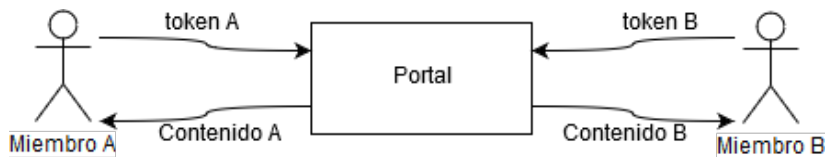


Figura 3.7: Representación de Contenidos dirigidos en el portal del DCC.

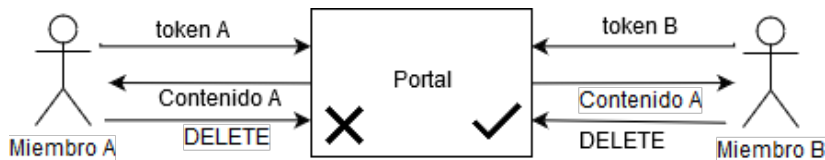


Figura 3.8: Representación de Autorizaciones distintas en el portal del DCC.

El caso anteriormente descrito ocurre en todos los escenarios en que el usuario necesita autenticarse en alguna aplicación del DCC. La autenticación y la autorización van de la mano con un filtro en los contenidos gracias a *U-Comunidad*.

3.5.2. Notificaciones dirigidas

El último de los escenarios planteados en la Sección 2.2.3, consiste en apoyar el servicio de *Notificaciones*. Hay dos casos de uso (también mostrados en la Figura 2.1. En el primero

(ilustrado en la Figura 3.9) el servicio de *Notificaciones* monitorea constantemente el servicio de *Noticias y Eventos* (1), y cuando se detecte un nuevo evento o noticia, se debe consultar a *U-Comunidad* (2) para obtener la lista de personas que cumplen los roles interesados en al menos un evento o noticia, junto con la información de contacto, y (3) contacta a los miembros de la comunidad enviándoles la información a las personas con el rol “A”, obteniendo la información de contacto de esos miembros, finalmente envía una notificación a quienes cumplen el rol “A”. En el segundo (ilustrado en la Figura 3.10), un servicio le solicita a *notificaciones* que realice una notificación a un miembro o un grupo de ellos que comparten al menos un rol “B” (1), *Notificaciones* consulta a *U-Comunidad* por el o los contactos necesarios (2) y envía la notificación correspondiente (3).

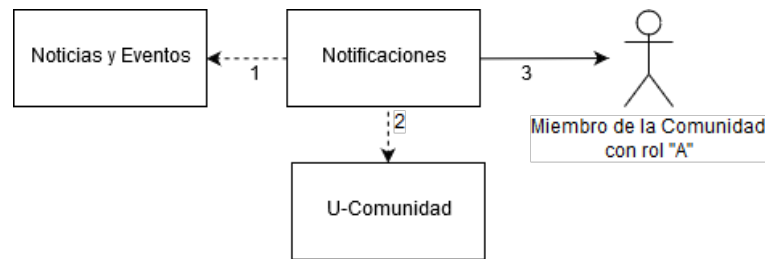


Figura 3.9: Notificación de un evento o noticia DCC automatizado

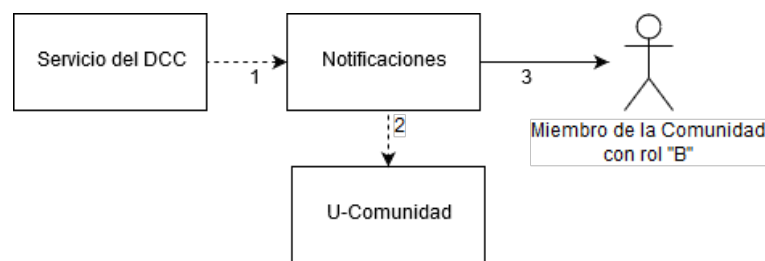


Figura 3.10: Notificación de un servicio

3.5.3. Alerta de inconsistencia

Dentro de los escenarios planteados en la Sección 2.2.3 y como se puede ver en la Figura 2.1, se contempla un servicio que revise frecuentemente si la información de *U-Comunidad* se alinea con la que muestran los sistemas web que no están bajo el control del Departamento, como la web de la Facultad o de la Universidad. Como muestra la ilustración de la Figura 3.11, este servicio revisa constantemente alguna web externa (1), compara la información con lo que tiene *U-Comunidad* (2) y, si encuentra alguna inconsistencia, solicita al servicio de *Notificaciones* que alerte a quien corresponda (3).

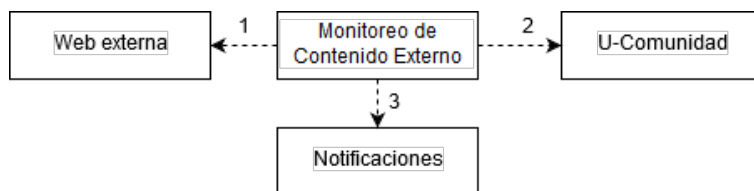


Figura 3.11: Alerta de inconsistencia

Capítulo 4

Implementación y Validación

En este capítulo se describe la implementación del sistema *U-Comunidad*, de acuerdo al diseño definido en el Capítulo 3. Primero, se detalla el desarrollo del microservicio a cargo de la gestión de la información de miembros de la comunidad y sus *roles organizacionales*, y de proveer acceso a esta información mediante una API REST. Este microservicio es suficiente para conseguir la implementación de los escenarios de uso descritos en la Sección 2.2. Segundo, se documenta el desarrollo de un agente, específicamente del microservicio a cargo de la extracción de la información de *U-Campus*. Con este agente se logra poblar y mantener actualizada automáticamente la información relativa a todos los estudiantes y egresados de las carreras que imparte (y ha impartido) el departamento. Dado que la API de U-Campus no ofrece directamente el servicio de obtener los estudiantes y/o egresados de la carrera, la implementación del agente debió adecuarse a las funciones que sí provee la API de U-Campus para lograr la extracción.

Se necesitarán nuevos agentes para agregar *fuentes* de información. La implementación de otros agentes es análoga a la del agente de *U-Campus*. Esto significa que se debe implementar un microservicio que periódicamente consulte el sistema externo que representa a la fuente de información, compara la información extraída con la ya registrada en *U-Comunidad*, y posteriormente la actualiza si hay cambios detectados. La complejidad de la implementación de cada agente depende de la disponibilidad de la información en la *f fuente*, la cual podría ser desde un servicio con una API a una simple página estática de la que se debe trabajar procesando el contenido en HTML. Ahora bien, si se modifica la API o la estructura de la página *f fuente*, se debe modificar la implementación del agente correspondiente.

Al momento de la escritura de este informe de memoria, la API del sistema *PG Scpio* proveía solo acceso a las tesis de los estudiantes y graduados, sin permitir extraer información del claustro académico ni profesores colaboradores (roles que son conocidos únicamente por ese sistema). Dado que *PG Scpio* no es un sistema propio del DCC, no se tuvo acceso a extender dicha API. No se construyó el agente pues, la información que *PG Scpio* provee actualmente ya es extraída de *U-Campus*. Por otro lado, en este trabajo de memoria se identificó la relevancia de contar con un servicio de *Gestión del Personal* del departamento (*SGP*), que actúe como la única fuente de verdad respecto a la información de los cargos de académicos y personal de colaboración del DCC. Este servicio servirá de *f fuente* de información

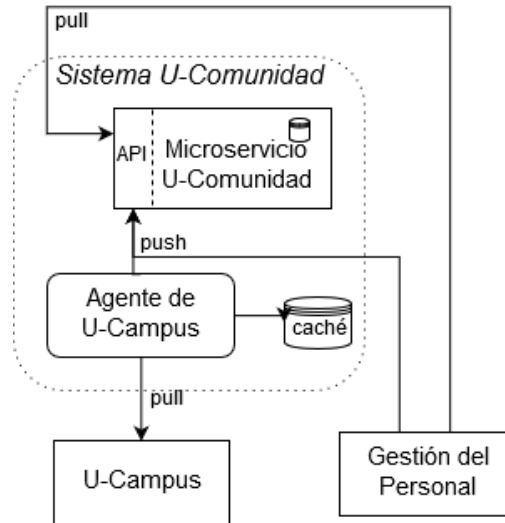


Figura 4.1: Implementación para validación de *U-Comunidad* junto con el prototipo de Servicio de *Gestión del Personal*

de *U-Comunidad* para lo que habrá que construir el agente correspondiente. En este trabajo de memoria se implementó un prototipo de este sistema, principalmente con el propósito de validar los escenarios de uso identificados para *U-Comunidad*.

Este capítulo finaliza describiendo la validación del servicio *U-Comunidad*. Dado que éste es un microservicio que no atiende directamente a usuarios, sino que provee información y funciones a otros microservicios, es que la validación se realizó mediante la implementación de un prototipo de portal, que actúa como servicio consumidor y que permite a un usuario acceder a la información y funciones de *U-Comunidad*.

4.1. Implementación del sistema *U-Comunidad*

Como se describe en el Capítulo 3 y como se puede observar en la Figura 4.1, se implementó el sistema *U-Comunidad* basándose en el patrón *blackboard*, desarrollando el microservicio *U-Comunidad*, el agente de *U-Campus* y la prueba de concepto del servicio de *Gestión del Personal* del DCC para validar algunos escenarios de uso. El prototipo consiste en una aplicación web con atribuciones temporales para escribir directamente en la API de *U-Comunidad*, es decir, funciona como su propio *agente* y usa el microservicio de *U-Comunidad* como si fuera su propio repositorio de información. Esto permite mostrar las funcionalidades de *U-Comunidad* sin tener que construir otro sistema completo que lo use o depender de la actualización de uno ya creado.

A continuación se detalla cada componente observable en la Figura 4.1.

4.1.1. Microservicio de *U-Comunidad*

El microservicio *U-Comunidad* se implementó en la plataforma *Java* utilizando el framework *Spring* ¹. *Spring* facilita la construcción del servicio al proveer librerías que ahorran líneas de código, permitiendo que el trabajo se concentre en resolver el problema del negocio que en aspectos tecnológicos. Para facilitar la ejecución del microservicio, se utilizó específicamente *Spring Boot*, tecnología de *Spring* que permite embeber en la aplicación un servidor web (como Tomcat²), haciendo a la aplicación auto contenida e instalable mediante la ejecución de un archivo *.jar* en un servidor que tenga soporte de la máquina virtual de *Java*.

El microservicio está organizado de acuerdo al patrón MVC. *Spring* da soporte directo a la implementación de este patrón. La implementación de los **controladores** requiere de una clase cuyos métodos incluyen anotaciones que describen las solicitudes HTTP que el controlador ofrece. El framework se encarga de convertir tales solicitudes en la invocación a estos métodos, incluso de la conversión desde los parámetros y la serialización del objeto retornado. Por ejemplo, el controlador del *endpoint* `/roles/id`, que entrega un rol dado su *id*, se vería como se muestra a continuación:

```
@RequestMapping(path = "/roles/{id}", method = RequestMethod.GET)
public RoleEntity getRole(@PathVariable String id){...}
```

La anotación *RequestMapping* gestiona el *endpoint*, en *path* se parametriza la URL donde esperará y en *method* el tipo de petición HTTP. Finalmente, el método *getRole* será el que se ejecute cuando se haga la petición *GET* al *endpoint* `/role/{id}` con *id* concreto.

Dado que el servicio ofrece un API REST que responde un archivo de estructura JSON, las **vistas** corresponden a la serialización en JSON de los objetos retornados por los métodos del **controlador**, de las cuales también se hace cargo *Spring*. Por último, el **modelo** corresponde a la abstracción que provee el framework para el acceso a los datos. *Spring*, a través de *Java Persistence API* (JPA), que, usando *Object Relational Mapping* (ORM), permite la conversión de objetos en memoria a registros en la base de datos, tanto al escribir como al leer. Esta tecnología simplifica el acceso a los datos y abstrae la implementación del sistema de las particularidades de la base de datos concreta facilitando el trabajo. Por ejemplo, la Tabla *roles* se ve de la siguiente forma:

```
@Entity
@Table(name = "roles")
public class RoleEntity{...}
```

La clase debe contener las columnas de la tabla como atributos, y los métodos *getters* y *setters* correspondientes a estos atributos.

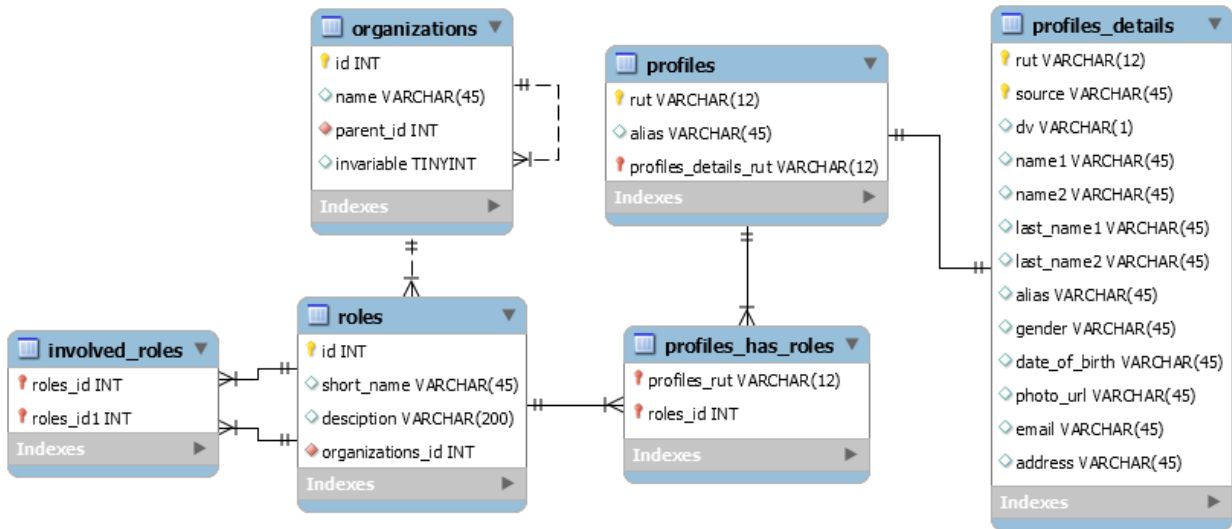


Figura 4.2: Modelo de la Base de Datos de *U-Comunidad*

Base de Datos

La base de datos fue implementada en PostgreSQL siguiendo el modelo relacional mostrado en la Figura 4.2. Con esas tablas se logra implementar el modelo de información de la Sección 3.4, donde se tienen las *Organizaciones*, *Roles simples* y *Roles agregados*, *Miembros de la comunidad* y sus *Perfiles*.

Las *Organizaciones* son representadas por la tabla *organizations*, manteniendo la idea de que hay organizaciones hijas de otra.

Los *Roles* son representados con la tabla de igual nombre en el modelo, donde están ligadas a una *Organización*. Para obtener los *Roles agregados*, cada Rol (simple o agregado) tiene una lista de *Roles agregados* con los que se relaciona (en la tabla *involved_roles*). Por ejemplo, el rol *Estudiante de Licenciatura (simple)* tendrá a *Estudiante de Pregrado()* en su lista y, a su vez, *Estudiante de Pregrado* tendrá a *Estudiante del DCC*. Así, en la base de datos los *Roles simples* se diferencian de los *agregados* en que ningún *Rol simple* estará en la lista de *involved_roles* de otro.

Los *Miembros de la comunidad* están representados en la tabla *profiles* y su interacción con los roles está presente en *profiles_has_roles*.

El *Perfil* de cada miembro está asociado con una *fuentes* de información en la tabla *profiles_details*, y contiene el RUT, dígito verificador, nombres, apellidos, dirección, fecha de nacimiento, teléfono y correo.

¹Framework de Java, <https://spring.io/>

²Servidor Web, <http://tomcat.apache.org/>

Código en U-Campus	Carrera de Interés	Vigente
3	Bachiller en Ciencias, Mención Computación	No
26	Ingeniería de Ejecución en Procesamiento de la Información	No
30	Licenciatura en Ciencias de la Ingeniería, Mención Computación	No
41	Ingeniería Civil en Computación	Sí
44	Magíster en Ciencias, Mención Computación	Sí
118	Programador de Computadores	No
127	Doctorado en Ciencias, Mención Computación	Sí
139	Magíster en Tecnologías de la Información	Sí
140	Magíster en Comunicaciones Digitales	Sí
11030694	Doctorado en Computación	Sí

Tabla 4.1: Códigos de Carreras en *U-Campus* correspondientes a pregrado, magíster y doctorado.

4.1.2. Agente de U-Campus

Solo se creó el *Agente* que extrae la información de la *fuentes* con mayor cantidad de información: *U-Campus*. *U-Campus* contiene toda la información acerca de los estudiantes como carreras cursadas, fechas de inicio y fin y estados en cada carrera de cada uno de las personas que han pasado por la facultad.

Extracción de la información

U-Campus entrega información de los estados de las carreras como “Activo” o “Titulado”, entre otros, y las fechas en que esas carreras inician o terminan. Con esta información se puede saber el rol que cumple el estudiante: *titulado*, *egresado*, *desertor*, etc. Pero estos datos se encuentran disponibles por un período determinado (un semestre), es decir, hay que consultar por todos los períodos para obtener la información de todos los estudiantes que han estado y están en el DCC. Por ejemplo, si se pregunta por el período 2017.1, que corresponde al primer semestre del 2017, no aparecerá listada gente que egresó el 2016, por lo que se hace necesario escanear todos los períodos.

Para poder extraer los *roles* de los estudiantes, necesitamos primero saber los períodos válidos que deberán ser buscados. Luego, para cada uno de ellos, preguntar por los estudiantes de ese período y filtrar las carreras que importan al DCC. La Tabla 4.1 lista los códigos en *U-Campus* de las carreras del DCC. Para cada uno de los RUTs de esas personas, se pregunta por sus carreras actuales. Allí se analiza qué roles cumple y con cuáles están enlazados. Como ejemplo se ve en la Figura 2.3 la jerarquía de los *roles agregados*, partiendo por *Estudiantes DCC* y siguiendo con *Estudiantes Pregrado*, *Estudiantes Posgrado*, que se divide en un rol agregado, *Estudiante Magíster* y uno simple *Estudiante Doctorado*. Finalmente se llega hasta los roles simples (coloreados en el dibujo): *Estudiante de licenciatura*, *ingeniería*, *doctorado*, *magíster en ciencias y tecnologías de la información*. Mientras, en otro proceso, con la misma lista de RUTs, se busca para cada uno de ellos el perfil que guarda *U-Campus*.

Dado que hay mucha información que debe actualizarse constantemente y a tiempo, se tiene un caché, el cual fue implementado como un hash calculado en dos ocasiones: al buscar la información de perfil y al buscar las carreras asociadas a una persona. Este hash se extrae desde la respuesta de la API, si no han habido cambios se continúa con la siguiente persona, si no, se actualiza la información y se guarda el nuevo hash.

Implementación

Este componente también está implementado en el *Spring* con *Spring Boot*, con lo que se puede ejecutar como aplicación independiente instalable desde un *.jar*. Utiliza la capacidad de *scheduling* del framework para disparar la ejecución de la extracción en forma periódica. El *agente* utiliza *rest template*, capacidad de *Spring* para consumir fácilmente un servicio REST.

A continuación se detallan los cuatro pasos nombrados en la sección anterior:

- **Actualización de períodos**

Se consulta la API de *U-Campus*, *"/periodos"*, para conocer la lista de períodos que son válidos hasta la actualidad. Estos períodos son guardados en una tabla intermedia que sirve a modo de caché al tener un campo en que guarda la última vez que se actualizó esta información, el cuál, según configuración previa, dirá cuando es necesario revisar los estudiantes de ese período en particular.

- **Actualización de RUTs por período**

Para conocer la lista de ruts que corresponden a estudiantes que estuvieron o están asociados a *carreras* en el DCC, por cada período que deba revisarse según la tabla construida o actualizada en el paso anterior, se consulta a *U-Campus*, *"/carreras_alumnos?periodo={period}"*, y se guarda, en otra tabla intermedia el identificador de la persona, RUT, un hash que se calcula desde la información de la persona, una fecha con la última vez que se actualizó esta información y un contador de la última vez que se actualizó la información de perfil de la persona. Se hace notar que acá se hace el filtro de las personas que son del interés del departamento al revisar los códigos de las carreras asociadas en archivos de configuración.

- **Actualización de roles por persona**

Para obtener los roles que tienen las personas en el departamento, para cada persona que deba ser actualizada, ya sea porque ha transcurrido más del tiempo establecido en el archivo de configuración o porque nunca ha sido revisado antes, se revisan las carreras a través de la API de *U-Campus*, *"/carreras_alumnos?rut={rut}"*, se calcula el hash de esta información y si han habido cambios con respecto al hash anterior, que parte en vacío (" "), procede a actualizar la información de la persona, si no, continúa con la que sigue tras actualizar la fecha de la última revisión de los roles de la persona.

- **Actualización de perfiles por persona**

Para obtener la información de las personas del sistema, se revisa si la fecha de actualización supera la dada por el límite de configuración. En caso de que se deba actualizar, se consulta la API de *U-Campus*, *"/personas?rut={rut}"*, desde donde se recupera el

RUT, dígito verificador, alias, apellido paterno, apellido materno, dirección, email, fecha de nacimiento, foto, género, primer nombre, segundo nombre, teléfono.

4.1.3. Gestión de Roles

U-Comunidad distingue dos tipos de roles, *simple* y *agregado*, tal como se ilustra en la Figura 3.5). Los roles *simples* son los roles organizacionales a los que se puede asociar directamente a los miembros de la comunidad, y corresponden a los roles de la tabla homologada (ver Figuras A.1, A.2 y A.3). Los roles *agregados* son roles más generales, que agrupan a los miembros de la comunidad que cumplen otros roles. Un ejemplo de rol agregado es el rol *estudiante del DCC*, que agrupa a los miembros de la comunidad que son estudiantes de cada programa del pregrado (licenciatura e ingeniería) y posgrado (magíster ti, magíster en ciencias y doctorado), cada uno de estos programa posee un rol simple que los representa (*estudiante de licenciatura*, *estudiante de ingeniería*, *estudiante de magíster ti*, *estudiante de magíster en ciencias* y *estudiante de doctorado*).

U-Comunidad registra ambos tipos de roles en la base de datos. Como se explicó en la Sección 4.1.1 y se ilustra en la Figura 4.2, la tabla *roles* contiene la lista de roles, ya sean *simples* o *agregados*, y la tabla *involved_roles* contiene la relación entre roles, indicando para cada rol en qué roles está agregado. Sin embargo, la gestión de estos roles se realiza en forma diferente.

Por un lado, la gestión de los roles simples está expuesta por el API de *U-Comunidad*. Como se explica en la Sección 3.3 y se detalla en la Tabla 3.2, el API permite a otros componentes obtener (*GET*), crear (*POST*), actualizar (*PUT*) y eliminar (*DELETE*) estos roles simples.

Por otro lado, la gestión de los roles agregados no está expuesta por el API de *U-Comunidad*. En cambio, éstos se gestionan mediante archivos de configuración del sistema que *U-Comunidad* procesa al momento de iniciarse y actualiza la información correspondiente en la base de datos. Esta decisión de diseño se basa en que *U-Comunidad* mantiene en forma materializada la lista de roles de cada miembro de la comunidad. En otros términos, al momento de consulta, *U-Comunidad* tiene la lista de roles de cada miembro ya computada. Es al momento de carga, cuando se asigna un miembro a un rol, que *U-Comunidad* registra la asignación al rol simple, y además computa y registra la asignación a los roles agregados. De esta forma, se optimiza la consulta, que es la función más frecuente que se espera del servicio. Como consecuencia, un cambio en la definición de roles agregados implica recomputar la asignación de roles. En la implementación actual esto se realiza cuando el sistema inicia. Sin embargo, se propone como trabajo futuro proveer este servicio a través del API.

Por último, *U-Comunidad* tiene una categoría especial de roles llamados roles de *sistema*. Estos corresponden a roles *simples* y *agregados* que no pueden ser manipulados (creados, actualizados ni eliminados) a través del API, solamente pueden ser consultados. Los roles de sistema corresponden a aquellos roles cuyos miembros son cargados y actualizados por un *Agente* y, por lo tanto, representa información extraída de fuentes externas. En la implementación actual, los roles correspondientes a estados de los estudiantes, como por ejemplo

estudiante congelado del DCC (Agregado), egresado de pregrado (Agregado) o desertor de magíster ti (Simple), son considerados roles de sistema ya que el *Agente U-Campus* gestiona dicha información. Los roles que no son roles de sistema se manipulan directamente a través del API, que a su vez se manipula y/o gestiona a través de una interfaz de usuario (mediante el prototipo del Servicio de Gestión del Personal descrito en la Sección 4.2.1). Tal como se define en la arquitectura de *U-Comunidad* en la Sección 3.2 y se ilustra en las Figuras 3.3 y 3.4, se prevé la construcción de múltiples agentes a cargo de la automatización de la carga y actualización de los miembros de la comunidad abarcando todos los roles organizacionales. Por lo tanto, a medida que se implementen más agentes, más roles simples pasarán a ser roles de sistema. Mientras estos agentes no estén disponibles, la asignación de miembros a roles puede realizarse a través del API y mediante el prototipo del Servicio de Gestión de Personal.

4.2. Validación

El sistema *U-Comunidad* está pensado para ofrecer información y servicios a otros micro-servicios y no está pensado para atender a usuarios finales. Dado que los servicios descritos en los escenarios no existían en el momento en que se realizó esta memoria, la validación necesitó la construcción de un portal para mostrar las funcionalidades y los casos de uso. Esta aplicación web sirve de prototipo del servicio de *Gestión del Personal*.

4.2.1. Servicio de Gestión del Personal

Dado que no existía tal servicio al momento de realizar el trabajo de esta memoria, no solo se construyeron los componentes que conforman *U-Comunidad*, sino que también se desarrolló una prueba de concepto del servicio de *Gestión del Personal*, que permite al *Área de Administración del Departamento* gestionar los roles de las Categorías B y C indicadas en la Sección 2.4.3. Este prototipo es una aplicación, que, como se puede ver en la Figura 4.1, es *fuentes* y consumidora a la vez de *U-Comunidad*, hecho que debe ser corregido, como trabajo futuro, para mantener la arquitectura lógica mostrada en la Sección 3.3. El objetivo de esta prueba de concepto es poder mostrar la funcionalidad de *U-Comunidad* por medio de este portal.

La aplicación web está construida en AngularJS y emula ser su propio *agente* al usar directamente la API de *U-Comunidad* para evitar construir, además, un API para el *Servicio de Gestión del Personal*.

En esta prueba de concepto se validaron los escenarios y requerimientos más importantes, no obstante, se utilizaron otras peticiones a la API que facilitaron el uso de la *web de gestión de personal del DCC*. A pesar de que no son esenciales para la utilidad de *U-Comunidad*, son ejemplos de la necesidad de extender la API para facilitar funcionalidades en los servicios. Por ejemplo, en vez de pedir la lista de identificadores para mostrarlos en una página, se necesitó un perfil mínimo (RUT, nombre y correo) para evitar saturar con la petición del

perfil para cada identificador de la lista a mostrar.

Para la validación del prototipo y *U-Comunidad*, se tienen en cuenta los tres primeros escenarios de usos descritos en la Sección 2.2: *autorización*, *autenticación* y *contenidos dirigidos* junto con los requisitos descritos en la Sección 2.3: Obtención del perfil y los roles para un identificador y, a partir de un rol, obtener una lista de identificadores que cumplen tal rol.



Figura 4.3: Vista del login del prototipo

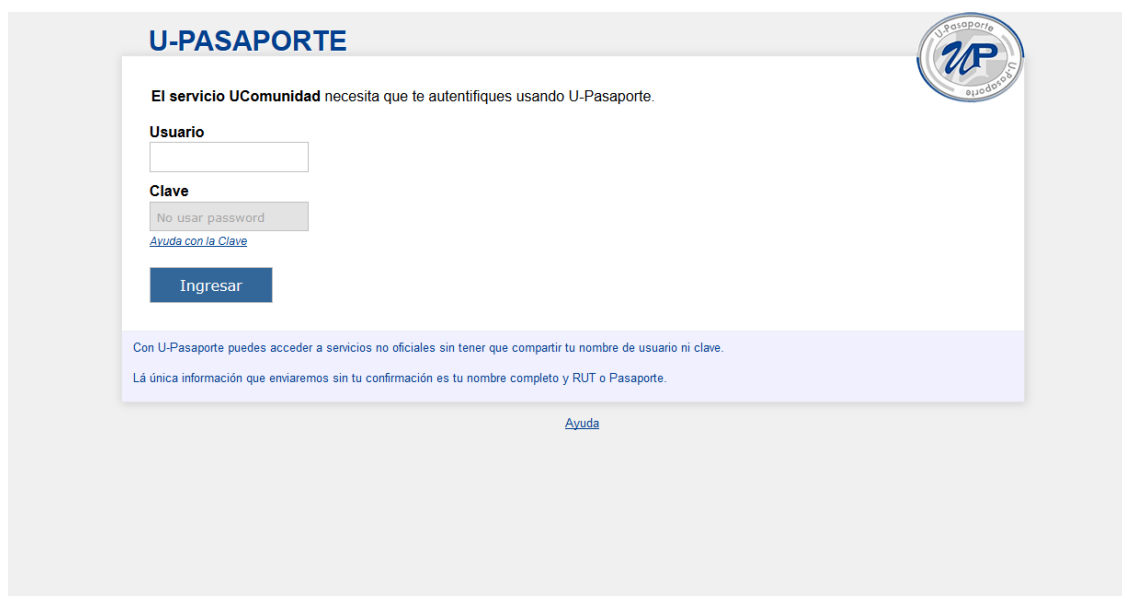


Figura 4.4: Vista de la redirección a login de U-Pasaporte

Para mostrar lo anterior, se utiliza las vistas del prototipo para guiar el proceso de validación. En esta implementación, la *autenticación* se hace a través de *U-Pasaporte*, como muestran las Figuras 4.3 y 4.4. Tras ingresar las credenciales a *U-Pasaporte*, con la librería

JSON Web Token (JWT), *Spring* une la *autenticación de U-Pasaporte* con los roles. Para ello, busca en la base de datos todos los roles asociados al usuario, luego genera un *token* que contiene el RUT de la persona y los roles asociados. Como estamos modificando la información del mismo sistema *U-Comunidad*, en su implementación describimos los permisos que tienen los usuarios para modificar. En esta prueba, se permite al rol *estudiante* solo ver, mientras que al rol *Jefe de Administración*, se le permite editar la información.

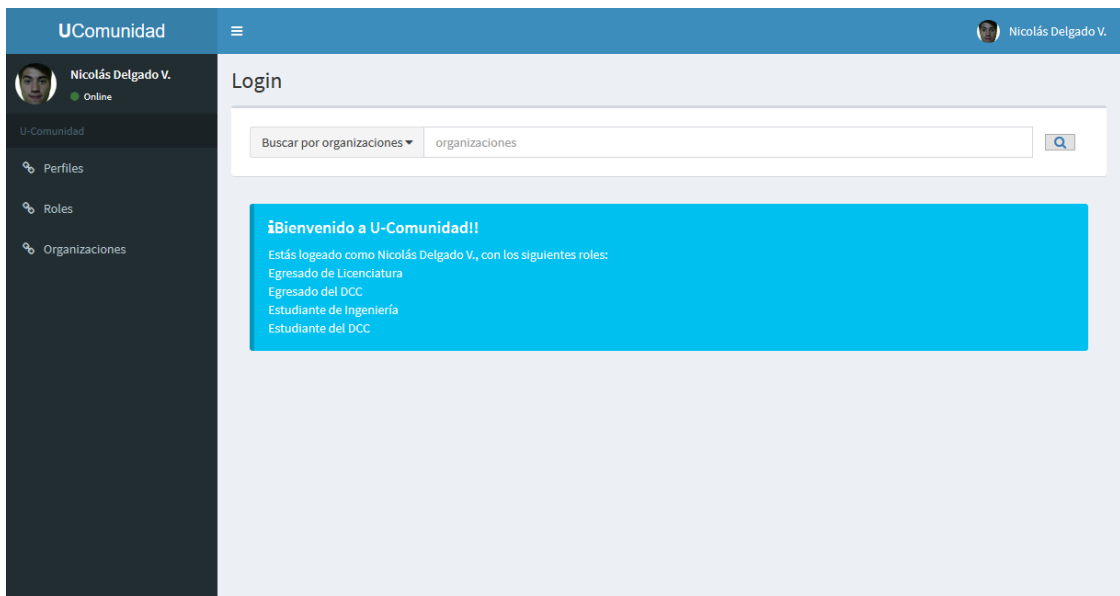


Figura 4.5: Vista de bienvenida a un estudiante

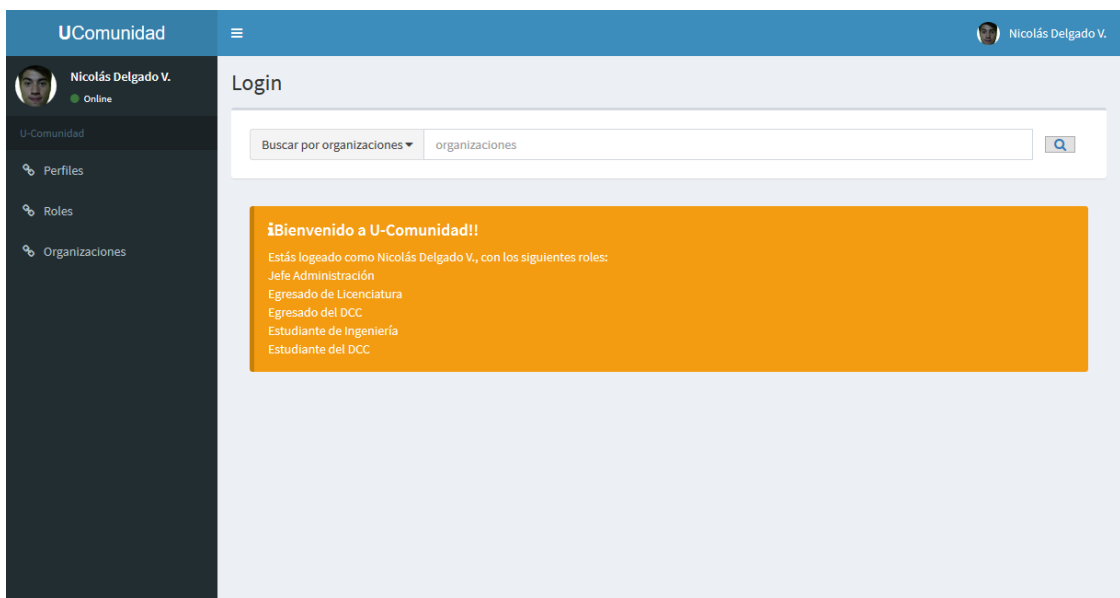


Figura 4.6: Vista de bienvenida a un funcionario

El *token* anteriormente descrito es recibido por el sitio web, el cuál lee la información de perfil y los roles que posee la persona para mostrar los contenidos que le correspondan. Para mostrar este comportamiento de contenidos dirigidos, se agregó el rol *Jefe de Administración* a la cuenta de un estudiante. Las diferencias se pueden ver en los mensajes de bienvenida de

la Figura 4.5 y 4.6. También se puede ver que el prototipo muestra la información del perfil como la foto, los nombres y los apellidos.

Para mostrar las diferencias de autorización, se puede ver la diferencia en las Figuras 4.7 y 4.8, donde dependiendo del rol del usuario la página muestra la posibilidad de edición del contenido por ser *Jefe de Administración*, mientras que en el rol de estudiante las opciones están escondidas. Aún evadiendo esta vista y realizando las peticiones directamente con el servicio, el *token* indica a *U-Comunidad* los roles que posee el usuario y, por lo tanto, no permitiría a un estudiante modificar la información.

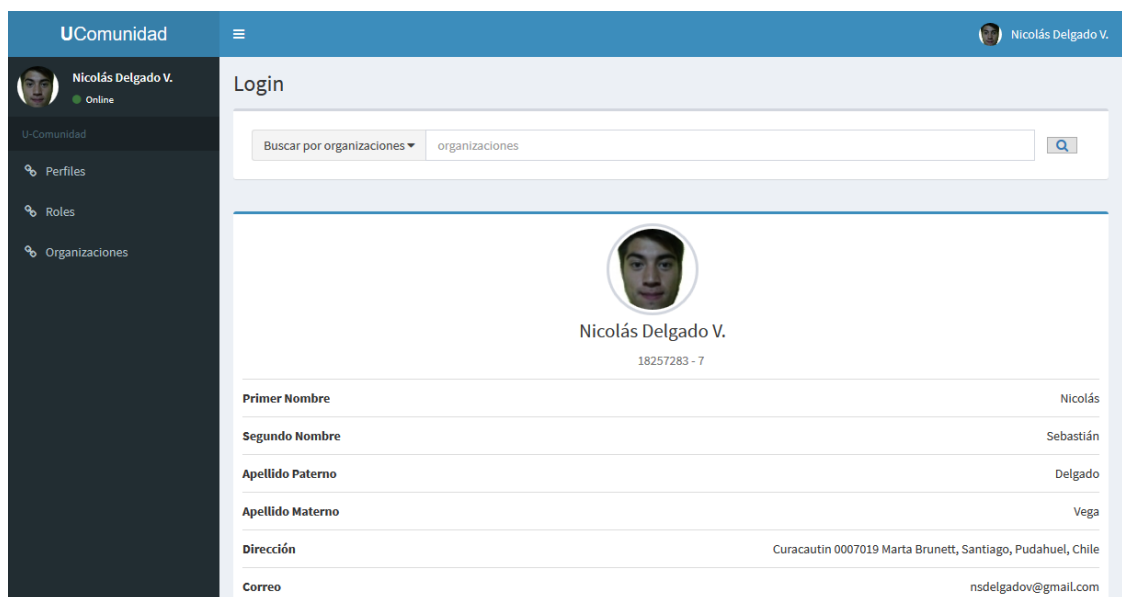


Figura 4.7: Vista del perfil a un estudiante

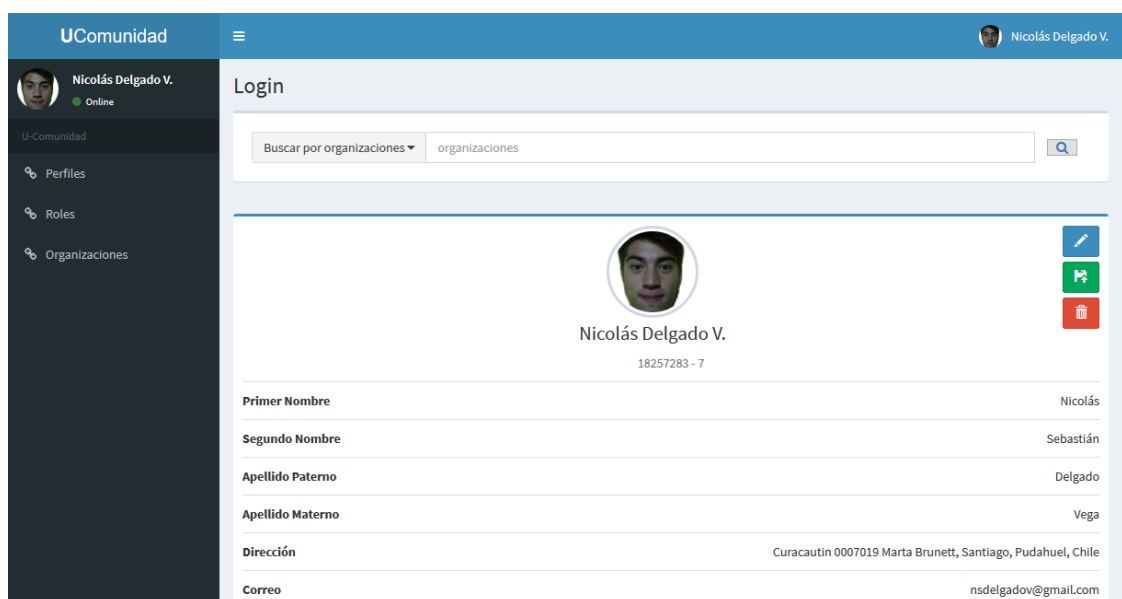


Figura 4.8: Vista del perfil a un funcionario con poder de edición

Para mostrar el cumplimiento de los requerimientos, en particular la información asociada

a un RUT, se puede ver la Figura 4.9 donde se muestra la información asociada al perfil y los roles que posee el estudiante.

The screenshot shows the user profile page for Nicolás Delgado V. in the UComunidad system. The page includes a search bar, a profile picture, and a list of roles.

Perfil de Nicolás Delgado V.
18257283 - 7

Primer Nombre	Nicolás
Segundo Nombre	Sebastián
Apellido Paterno	Delgado
Apellido Materno	Vega
Dirección	dirección
Correo	correo@email.cl
Teléfono	9 8765 4321
Fecha de Nacimiento	1993

Id	Rol
licenciatura_egresado	Egresado de Licenciatura
DCC_egresado	Egresado del DCC
ingenieria_estudiante	Estudiante de Ingeniería
DCC_estudiante	Estudiante del DCC

Copyright © 2016 Company. All rights reserved.
Icons made by Chanut is Industries from www.flaticon.com is licensed by CC 3.0 BY

Figura 4.9: Vista del perfil completo de un estudiante

Para mostrar que con un *rol* se obtiene la lista de quienes lo cumplen, el prototipo permite ver la lista de RUTs que poseen el *rol*, como se ve en la Figura 4.10.

El prototipo construido permitió validar que los escenarios de uso pueden lograrse con el servicio *U-Comunidad* desarrollado, y que éste cumple con los requisitos funcionales definidos. A su vez, la implementación del prototipo y del servicio permitió mostrar no solo que la arquitectura definida cumple con la visión de *U-Comunidad*, sino también que a partir de esta arquitectura puede implementarse el sistema. Se concluye así que la arquitectura definida captura las decisiones de diseño críticas que permiten la construcción del sistema completo, y que guía su evolución mediante la inclusión de nuevos agentes.

The screenshot displays a web application interface for managing roles. The header shows 'UComunidad' and the user 'Nicolás Delgado V.' is logged in. The left sidebar contains navigation links for 'Perfiles', 'Roles', and 'Organizaciones'. The main content area is titled 'Roles' and features a search bar with the text 'doctorado'. Below the search bar, the details for the role 'Rol: Estudiante del Doctorado' are shown, including its ID ('doctorado_estudiante'), description ('Rol de sistema.'), and organization ('Doctorado'). A table lists the RUTs associated with this role, with the first 10 entries visible. The pagination control at the bottom indicates 'Showing 1 to 10 of 38 entries' and includes buttons for 'Previous', '1', '2', '3', '4', and 'Next'.

Rut
10022893
14146463
15429061
16069895
16857477
16921298
16944712
16978403
17024264
17024961

Showing 1 to 10 of 38 entries

Previous 1 2 3 4 Next

Copyright © 2016 Company. All rights reserved.
Icons made by Chanut is Industries from www.flaticon.com is licensed by CC 3.0 BY

Figura 4.10: Vista del detalle de un rol, se puede ver la lista de ruts asociados.

Capítulo 5

Conclusiones y trabajo a futuro

El objetivo general, “*Proveer al DCC de una fuente centralizada de información acerca de los miembros de su comunidad y los roles organizacionales que ellos cumplen, mediante un servicio de información que capture automática y periódicamente la información de otras aplicaciones consideradas como fuentes válidas.*” (Sección 1.2) se cumplió en su totalidad. El repositorio que centraliza la información de los miembros de la comunidad ha sido implementado, posee una de las *fuentes* que mayor información entrega, *U-Campus*, y se tiene un prototipo funcional del servicio de *Gestión del Personal* que permite ser una *fente* temporal de todos los *roles* faltantes, incluyendo los de *PG Scopio*.

Con el levantamiento de información de los *roles* del departamento se pudo comprobar la necesidad de centralizar esta información en *U-Comunidad*, además de poder visualizar la carencia del sistema de *Gestión del Personal* y la necesidad del sistema de *Monitoreo y Consistencia de la Información*.

Con *U-Comunidad* y la prueba de concepto del servicio de *Gestión del Personal* se mejorarán las interacciones de la plataforma de microservicios con los escenarios mostrados en la Sección 3.5: mejorar la seguridad y generar contenidos dirigidos en todas las aplicaciones, reduciendo tiempo de mantención y actualización de ambos trabajos que requerían hacerse de forma manual. Además, permiten generar notificaciones dirigidas y alertas de inconsistencia de portales en los que el departamento no tiene control.

Dentro de los aprendizajes obtenidos, se cuenta la aplicación de nuevas tecnologías. En el lado del servidor, el framework Spring con sus facilidades para crear la API REST, procesos automáticos como el agente de *U-Campus* y autenticación con JWT y *tokens*, como se hace con *U-Pasaporte*. Por el lado del cliente, se reforzaron los conocimientos de AngularJS. En cuanto al negocio, se entendió lo complejo que es mantener un sistema con la cantidad de aplicaciones que posee el DCC y el hecho de trabajar con un componente de la arquitectura de microservicios hizo entender la importancia que tiene definir el diseño antes de comenzar la implementación. Además de interiorizar los beneficios de esta arquitectura, tales como saber de antemano el impacto que generará el microservicio solo con tener la interfaz clara y la posibilidad de probar la funcionalidad a pesar de no tener el resto de los componentes implementados.

Como trabajo futuro queda implementar el servicio de *Gestión del Personal* adecuándolo a la arquitectura de microservicios y crear su agente correspondiente, desacoplando así la actual implementación del prototipo. También queda pendiente el trabajo de los servicios que aún no existen y que dan mayor utilidad a *U-Comunidad* como lo son el servicio de *Notificaciones* y el de *Monitoreo de Contenido Externo*, además de poner en producción el mismo sistema de *U-Comunidad*. Por otro lado, la necesidad de saber la historia de los *roles* no fue abordada en esta memoria.

Para potenciar aún más *U-Comunidad*, se podría construir un sistema o servicio que permita la administración de los archivos de configuración que contienen la definición de la jerarquía de las *Organizaciones* y de los *Roles Agregados*.

Bibliografía

- [1] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal. *Pattern-Oriented Software Architecture - Volume 1: A System of Patterns*, Wiley Publishing, 1996.
- [2] Cesare Pautasso, Olaf Zimmermann and Frank Leymann. *RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision*, 17th International World Wide Web Conference (WWW2008)
- [3] Rodríguez Aranda, F. (2016). *Reingeniería de U-Dashboard: Sistema de apoyo a la gestión del DCC*.
- [4] Schmitt Rivera, D. (2013). *Desarrollo de un sistema de apoyo a la gestión de fondos de investigación universitarios*.

Apéndice A

Roles

N^0	Rol
1	Académico Jornada Completa
2	Académico Jornada Parcial
3	Administrativo
4	Asistente de Coordinación PEC
5	Asistente de Gestión Estratégica
6	Asistente de vinculación externa
7	Auxiliar
8	Claustro de Doctorado
9	Claustro de Magíster en ciencias
10	Claustro de Magíster TI
11	Comité Académico del Magíster en ciencias
12	Comité Académico del Magíster TI
13	Comité Académico del Programa de doctorado
14	Comité de Postgrado
15	Comité docente de pregrado
16	Consejo departamental
17	Coordinación Página Web
18	Coordinador Académico Área Gestión Informática
19	Coordinador Académico Área Ciencia e Ingeniería de Datos
20	Coordinador Académico Área Informática Educativa

Tabla A.1: Lista de todos los roles homologados I.

N ^o	Rol
21	Coordinador Académico Área Seguridad Computacional
22	Coordinador Académico del programa de educación continua
23	Coordinador de Apliaciones
24	Coordinador de biblioteca/Coordinador de la Biblioteca del DCC
25	Coordinador de Comunicaciones
26	Coordinador de Consultorías
27	Coordinador de Gestión Estratégica
28	Coordinador de investigación
29	Coordinador de Posgrado
30	Coordinador de prácticas profesionales
31	Coordinador de primer año
32	Coordinador de Proyectos Externos
33	Coordinador de Revista
34	Coordinador de titulación
35	Coordinador de vinculación externa
36	Coordinador del PEC
37	Coordinador Docente
38	Coordinador Magíster en Ciencias de la Computación
39	Coordinador Magíster en Tecnologías de la Información
40	Coordinador Programa de Doctorado en Computación
41	Coordinador Proyecto de Software
42	Coordinadora Académica Área Ingeniería y Calidad de Software
43	Director
44	Director de Docencia CaDCC
45	Director de Finanzas CaDCC
46	Director de Secretaría y Comunicaciones CaDCC
47	Egresados
48	Egresados DCC
49	Ejecutiva Programas Corporativos
50	Ejecutivo en ventas
51	Encargado del Museo
52	Estudiantes del PEC
53	Estudiantes doctorado
54	Estudiantes magíster en ciencias
55	Estudiantes magíster TI

Tabla A.2: Lista de todos los roles homologados II.

N ^o	Rol
56	Estudiantes pregrado
57	Integrantes Junta evaluadora/Junta evaluadora departamental
58	Jefe Administrativo
59	Jefe de Estudio
60	Jefe de Estudio de posgrado
61	Jefe de Programa Educación Continua
62	Jefe de Sistemas
63	Periodista
64	Postdoctor
65	Presidente CaDCC
66	Profesional
67	Profesor
68	Profesores Colaboradores y Visitantes Programa Doctorado
69	Profesores Colaboradores y Visitantes Programa Magíster
70	Secretaria de Administración
71	Secretaria de Dirección
72	Secretaria de Investigación
73	Secretaria Docente
74	Secretario de Postgrado
75	Subdirector
76	Subjefe de Sistemas
77	Técnico de Sistemas
78	Traductor Inglés
79	Presidente CaDCC
80	Vicepresidente CaDCC
81	Vicepresidente CaDCC
82	Director de Secretaría y Comunicaciones CaDCC
83	Director de Finanzas CaDCC
84	Director de Docencia CaDCC

Tabla A.3: Lista de todos los roles homologados III.

N ^o	Rol
1	Académico Jornada Completa
2	Académico Jornada Parcial
3	Administrativo
4	Auxiliar
5	Coordinador Comunicaciones
6	Coordinador Docente
7	Coordinador PEC
8	Coordinador Prácticas
9	Coordinador Revista DCC
10	Coordinador: Doctorado
11	Coordinador: Investigación
12	Coordinadora Magíster en Ciencias de la Computación
13	Coordinadora Magíster en Tecnologías de la Información
14	Director
15	Jefa Administrativa
16	Jefa Docente
17	Jefe Docente
18	Jefe Programa Educación Continua
19	Jefe Unidad de Sistemas
20	Periodista
21	Profesional
22	Secretaria
23	Subdirector
24	Técnico
25	Traductor Inglés

Tabla A.4: Roles obtenidos desde entrevista a Jefa Administrativa.

N ^o	Rol
1	Académico jornada completa
2	Académico jornada parcial
3	Administrativo
4	Asistente Dirección
5	Asistente Capacitación
6	Asistente contable
7	Asistente de Investigación
8	Asistente Docente
9	Auxiliar
10	Ayudante
11	Electricista
12	Jefa Administrativa
13	Jefe de Estudio
14	Jefe de Sistemas
15	Jefe Programa
16	Periodista

Tabla A.5: Roles obtenidos desde el directorio telefónico de la página del DCC, pedido a *Sistemas*.

N ^o	Rol
1	AJC DCC
2	AJP DCC
3	Asistentes de Coordinación PEC
4	Claustro Doctorado
5	Claustro Magíster TI
6	Comité Académico del Magíster TI
7	Comité Académico del Programa de doctorado
8	Comité de Postgrado Comité asesor posgrado Coordinación posgrado
9	Comité docente de pregrado
10	Consejo departamental
11	Contacto Proyecto de Software
12	Coordinación Académica Educación Continua
13	Coordinación Página Web Coordinador (Sitio Web DCC)
14	Coordinación Prácticas de vacaciones
15	Coordinación primer año Coordinador de Primer Año
16	Coordinación Programa corporativo Coordinador Académico Programas Corporativos Coordinación Programas diploma de postítulos
17	Coordinación Proyectos externos Coordinador Proyectos Externos
18	Coordinación sistemas Coordinador (Sistemas) Coordinador de Sistemas
19	Coordinación titulación Coordinadora de Titulación
20	Coordinador Académico Área Gestión Informática
21	Coordinador (Consultorías)
22	Coordinador (Revista Bits)
23	Coordinador Académico Área Ciencia e Ingeniería de Datos
24	Coordinador Académico Área Informática Educativa
25	Coordinador Académico Área Seguridad Computacional
26	Coordinador Académico Magíster Profesional en Tecnologías de la Información Coordinación Magíster en TI Coordinador Programa de Magister en TI Coordinador Académico Magíster TI
27	Coordinador de biblioteca/Coordinador de la Biblioteca del DCC
28	Coordinador de Comunicaciones

Tabla A.6: Roles obtenidos desde la página del DCC I.

N ^o	Rol
29	Coordinador de investigación
30	Coordinador Programa de Doctorado en Ciencias mención Computación Coordinador Programa de Doctorado en Computación
31	Coordinador Programa de Magíster en Ciencias mención Computación Coordinadora Programa de Magíster en Ciencias mención Computación
32	Coordinadora Académica Área Ingeniería y Calidad de Software
33	Coordinadora IEPI
34	Director de Docencia CaDCC
35	Director de Finanzas CaDCC
36	Director de Postgrado
37	Director de Secretaría y Comunicaciones CaDCC
38	Director del departamento
39	Egresados
40	Ejecutiva en ventas (PEC)
41	Ejecutiva Programas Corporativos (PEC)
42	Eléctrico Técnico de Sistemas
43	Encargado del Museo
44	Estudiantes doctorado
45	Estudiantes magíster en ciencias
46	Estudiantes magíster TI
47	Estudiantes pregrado
48	Integrantes Junta evaluadora Junta evaluadora departamental
49	Jefa de Estudio de posgrado
50	Jefa de Estudios Jefe de estudios
51	Jefe Administrativo
52	Jefe de área de RRHH
53	Jefe de Sistemas
54	Jefe Docente Coordinador Docente Coordinación Docencia pregrado
55	Jefe Programa de Educación Continua Jefe de Educación Continua
56	Periodista
57	Personal de Servicio
58	Presidenta Junta evaluadora
59	Presidente CaDCC
60	Profesional Asistente Administrativo
61	Profesores Colaboradores y Visitantes Programa Doctorado
62	Profesores Colaboradores y Visitantes Programa Magíster

Tabla A.7: Roles obtenidos desde la página del DCC II.

N ^o	Rol
63	Profesores PEC
64	Secretaria de Administración
65	Secretaria de Dirección
66	Secretaria de Investigación
67	Secretaría de Postgrado
68	Secretaria de Recursos Humanos
69	Secretaría del Programa de Doctorado
70	Secretaria Docente
71	Subdirector Departamento
72	Subjefe de Sistemas
73	Vicepresidente CaDCC

Tabla A.8: Roles obtenidos desde la página del DCC III.

N ^o	Rol
1	Administrativo
2	Auxiliar
3	Jefe Administrativo
4	Profesional
5	Profesor Adjunto/Instructor Adjunto
6	Profesor Asistente/Profesor Asociado/Profesor Titular
7	Técnico

Tabla A.9: Roles obtenidos de las páginas blancas de la universidad.

N ^o	Rol
1	Comité de postgrado
2	Egresados DCC
3	Magister ti
4	PEC
5	Profesores externos
6	Profesores jornada parcial
7	Vinculación externa

Tabla A.10: Roles obtenidos de las listas de correos actuales del DCC.

N^0	Rol
1	Asistente Gestión Estratégica
2	Asistente Vinculación Externa
3	Claustro Magíster en ciencias
4	Comité Académico del Magíster en ciencias
5	Coordinador de APLICACIONES
6	Coordinador Gestión Estratégica
7	Coordinador Vinculación Externa
8	Estudiantes PEC
9	PostDoctorados

Tabla A.11: Roles adicionales obtenidos del Área de Gestión Estratégica.