



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

REINGENIERÍA DEL PORTAL DEL CURSO “PROYECTO DE SOFTWARE”

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN COMPUTACIÓN

TOMÁS EDUARDO MARCHESSI NAVARRO

PROFESORA GUÍA:  
MARÍA CECILIA BASTARRICA PIÑEYRO

MIEMBROS DE LA COMISIÓN:  
JOHAN FABRY  
JUAN ÁLVAREZ RUBIO

SANTIAGO DE CHILE  
2024

# Resumen

El curso Proyecto de Software provee un servicio a organizaciones ajenas al DCC, donde un equipo conformado por alumnos del DCC realizan un proyecto que estas organizaciones propongan durante un semestre. En este curso se lleva a la práctica competencias, habilidades y conocimientos profesionales en un contexto laboral real de empleador-empleado. Lo anterior, está en línea con el perfil de egreso de la carrera en el ámbito de concebir, desarrollar y mantener soluciones de Software, que den respuesta a los problemas prácticos de forma sistemática y de calidad.

Al momento de escribir esta memoria, este curso ya contaba con un sitio web, desarrollado hace aproximadamente 8 años en el curso de Ingeniería de Software II por estudiantes de la misma facultad, el cual tiene como meta facilitar el proceso de postulación al servicio que ofrece este curso. Este posee una interfaz que permite a las organizaciones informarse con respecto al curso, ver los resultados de proyectos realizados en semestres anteriores y también postular a este servicio. La página cuenta también con una segunda interfaz que permite al profesor a cargo del curso gestionar las postulaciones y otros aspectos de la interfaz de las organizaciones.

Pese a que ambas interfaces cumplían con su propósito, tanto estas como el software del sitio web también presentaban fallas o carencias que afectaban la usabilidad, mantenibilidad y gestión de identidad y accesos del sitio web. Es por esto que, para esta memoria, se propuso realizar una reingeniería al portal, trabajando en el back-end, front-end y base de datos de la aplicación, enfocada en mejorar ambas interfaces y el software del sitio.

Se concluye que la reingeniería realizada en ambas interfaces permitió, mejorar la mantenibilidad del software y la usabilidad de ambas interfaces, así como también la gestión de identidad y acceso de la interfaz del profesor a cargo. Todo esto sin modificar el cómo se utilizan las diferentes vistas de la página y con un software actualizado que cuenta con un soporte robusto y de calidad.

*Dedicado a todos los que me han acompañado en estos casi 25 años.*

# Agradecimientos

A mi familia (especialmente mi papá y mamá) por darme todas las oportunidades que me trajeron aquí.

A mis grupos de amigos, los Bronces Unidos por siempre apoyarme en mis proyectos de computación y a los Andy Boys y HFF por siempre ayudarme con mis actividades tanto dentro como fuera de la universidad.

Al Nacho, Martín, Cata I y Cata P que aunque a veces se hable poco, siempre dan apoyo incondicional.

Al Diego y Porfirio L. por acompañarme durante mi intercambio.

A la Carla y la Fer por ayudarme a entregar mis informes sin tantos errores ortográficos y gramaticales.

A mi profesora guía, por mostrarme este proyecto y darme el voto de confianza pese a que estaba realizando la memoria de forma extraña.

Y finalmente a la tante Carmen Gloria, la primera profe que me puso fichas y no me anotó.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Portalpsw . . . . .	1
1.2. Objetivos . . . . .	5
1.2.1. Objetivo General . . . . .	5
1.2.2. Objetivos Específicos . . . . .	5
<b>2. Marco Teórico</b>	<b>6</b>
2.1. Reingeniería de Software . . . . .	6
2.1.1. Mantenibilidad de software . . . . .	6
2.1.2. Usabilidad de software . . . . .	7
2.1.3. Seguridad . . . . .	7
2.2. Tecnologías . . . . .	8
2.2.1. Django . . . . .	8
2.2.2. Docker . . . . .	8
2.2.3. PostgreSQL . . . . .	8
<b>3. Software del sitio</b>	<b>10</b>
3.1. Obtención y estado inicial . . . . .	10
3.2. Solución . . . . .	10
3.2.1. Solucionar de manera momentánea la situación que no permitía levantar el sitio . . . . .	11
3.2.2. Eliminar el código sin utilidad . . . . .	11

3.2.3.	Actualizar todas las dependencias restantes . . . . .	12
3.2.4.	Solucionar los errores generados por la actualización . . . . .	12
3.3.	Estado final . . . . .	14
<b>4.</b>	<b>Interfaz de los clientes</b>	<b>16</b>
4.1.	Contexto . . . . .	16
4.2.	Problemas de las vistas . . . . .	18
4.2.1.	Vista principal . . . . .	18
4.2.2.	Vista para postular . . . . .	18
4.2.3.	Vista de proyectos realizados . . . . .	18
4.3.	Solución . . . . .	19
4.3.1.	Vista principal . . . . .	19
4.3.2.	Vista para postular . . . . .	20
4.3.3.	Vista de proyectos realizados . . . . .	20
4.4.	Estado final . . . . .	21
4.4.1.	Vista principal . . . . .	21
4.4.2.	Vista para postular . . . . .	21
4.4.3.	Vista de proyectos realizados . . . . .	22
<b>5.</b>	<b>Interfaz del profesor</b>	<b>23</b>
5.1.	Contexto . . . . .	23
5.2.	Problemas de las vistas . . . . .	27
5.2.1.	Vista de postulantes . . . . .	27
5.2.2.	Vista proyectos . . . . .	28
5.2.3.	Vistas de Documentos y portal de revisión . . . . .	28
5.2.4.	Vista de plantilla . . . . .	28
5.3.	Gestión de identidad y acceso de la interfaz . . . . .	28
5.3.1.	Recuperación de contraseña . . . . .	28

5.3.2.	Cierre de sesión . . . . .	29
5.4.	Solución . . . . .	29
5.4.1.	Eliminación de elementos . . . . .	29
5.4.2.	Agregación de funciones . . . . .	31
5.4.3.	Cambios de nombres . . . . .	31
5.4.4.	Agregación de variables en proyectos realizados . . . . .	31
5.4.5.	Mejora de la gestión de identidades y accesos . . . . .	32
5.5.	Estado final . . . . .	34
5.5.1.	Vista postulaciones . . . . .	34
5.5.2.	Gestión de identidad y acceso . . . . .	35
<b>6.</b>	<b>Validación</b>	<b>38</b>
6.1.	Validación portalspw . . . . .	38
6.2.	Validación recuperación de contraseña . . . . .	39
6.3.	Validación cierre automático de sesión . . . . .	39
<b>7.</b>	<b>Conclusión</b>	<b>41</b>
7.1.	Cumplimiento de objetivo . . . . .	41
7.1.1.	Mejorar la mantenibilidad del sitio . . . . .	41
7.1.2.	Mejorar la usabilidad del sitio . . . . .	41
7.1.3.	Abordar las deficiencias en la gestión de identidad y acceso de la interfaz	42
7.2.	Trabajo a futuro . . . . .	42
7.2.1.	Portal de pagos . . . . .	42
7.2.2.	Orden del código . . . . .	43
7.2.3.	Formulario online . . . . .	43
7.2.4.	Recuperación de nombre de usuario . . . . .	43
	<b>Bibliografía</b>	<b>44</b>

# Índice de Ilustraciones

1.1. Criterios a cumplir para poder participar en el curso . . . . .	2
1.2. Interfaz para los potenciales clientes . . . . .	3
1.3. Interfaz para el profesor a cargo del curso . . . . .	4
2.1. Subcaracterísticas que conforman la característica usabilidad . . . . .	7
3.1. Versiones de las dependencias al inicio de la memoria . . . . .	11
3.2. Diferencia entre el admin de Django normal (arriba) y con django-jet (abajo)	13
3.3. Vista de los proyectos realizados debido al error con la base de datos . . . . .	14
3.4. Estado de las dependencias al finalizar la memoria . . . . .	15
4.1. Vista principal de la interfaz de los clientes . . . . .	16
4.2. Vista para postular de la interfaz de los clientes . . . . .	17
4.3. Vista de los proyectos realizados de la interfaz de los clientes . . . . .	17
4.4. Proyectos repetidos en slides diferentes . . . . .	19
4.5. Nueva vista principal . . . . .	21
4.6. Nueva vista para postular . . . . .	21
4.7. Nueva vista de proyectos realizados . . . . .	22
5.1. Vistas accesibles desde la página inicial de la interfaz del profesor . . . . .	24
5.2. Vistas de todas las postulaciones realizadas . . . . .	24
5.3. Vistas de los proyectos realizados en cada convocatoria . . . . .	25
5.4. Vistas de las plantillas . . . . .	25

5.5. Vistas de las plantillas . . . . .	26
5.6. Vistas de Subidas de documentos . . . . .	26
5.7. Vista para notificar a una empresa que postulo . . . . .	27
5.8. Vista específica de una postulación . . . . .	27
5.9. Vista específica de un proyecto aceptado . . . . .	28
5.10. Inicio de sesión de la interfaz del profesor . . . . .	29
5.11. Línea de código comentada que debe ser eliminada . . . . .	30
5.12. Línea de código comentada que debe ser revisada . . . . .	30
5.13. Nueva vista para contactar a los clientes . . . . .	31
5.14. Correo que se envía al usuario para restablecer la contraseña . . . . .	32
5.15. Templates para la recuperación de contraseña . . . . .	33
5.16. Todas las vistas que se pueden seleccionar luego de los cambios . . . . .	34
5.17. Nueva vista de postulaciones realizadas . . . . .	34
5.18. Nueva vista de postulaciones individuales . . . . .	35
5.19. Nueva vista de proyectos realizados . . . . .	35
5.20. Nuevo inicio de sesión . . . . .	36
5.21. Primera vista para recuperar la contraseña . . . . .	36
5.22. Segunda vista para recuperar la contraseña . . . . .	36
5.23. Tercera vista para recuperar la contraseña . . . . .	37
5.24. Segunda vista para recuperar la contraseña . . . . .	37
5.25. Cierre en caso de inactividad por más de 5 minutos . . . . .	37
6.1. Proyectos que postularon en la convocatoria Primavera 2023 . . . . .	39
7.1. Vista del portal de pago . . . . .	43

# Capítulo 1

## Introducción

En esta sección del informe se entrega un breve contexto de la razón por la que se decidió realizar una reingeniería en este sitio web, así como también una descripción reducida de los objetivos propuestos al inicio de la memoria para realizarla de forma exitosa.

### 1.1. Portalpsw

El curso “Proyecto de Software” es uno de los últimos cursos del Departamento de Ciencias de la Computación tanto en el plan curricular 2007 como en el plan curricular innovado 2019 [1], todo estudiante del DCC que desee titularse deberá cursar este ramo que, por malla, corresponde al décimo semestre de la carrera. Este curso comenzó en 1997 y, al día de escribirse este informe, lleva más de 250 proyectos realizados con éxito por diferentes equipos conformados por alumnos de la carrera.

“Proyecto de Software” consiste en formar múltiples equipos de cinco a ocho alumnos que trabajen para a una organización externa al departamento. Tanto los equipos como la organización para la que se trabajará, son designados por el profesor a cargo del curso. Este equipo deberá cumplir los requisitos que la organización solicite para su proyecto y serán guiados por su tutor (un alumno más avanzado o ya titulado) y el profesor. Además de trabajar en el proyecto, al equipo se le solicitará realizar tres presentaciones, en las cuales estarán presentes: su cliente, su tutor, el profesor y otros equipos con sus respectivos tutores y cliente.

Para participar en este curso, las organizaciones deben postular durante el período de convocatoria del semestre anterior. Estas organizaciones deberían cumplir con una serie de criterios para poder participar, los cuales se pueden ver en la Figura 1.1.

Los criterios que deben cumplir las organizaciones clientes son los siguientes:

- Presentar un problema de negocio claro para obtener resultados en el semestre.
- Comprometerse a efectuar un aporte económico, en dos cuotas, una al comienzo y otra al final del proyecto, contra entrega satisfactoria.
- Estar en condiciones de firmar el convenio de colaboración, que puede revisar en [este enlace](#).
- Disponer de una contraparte oficial y otra suplente. Este rol está definido por:
  - Pertenecer a la organización cliente,
  - Poseer conocimiento acabado del problema de negocio a resolver,
  - Tener la confianza de la organización para tomar decisiones autónomamente,
  - Dedicación mínima de 8 horas a la semana para trabajar con el equipo de forma remota y al menos 4 horas de trabajo presencial por cada iteración con los alumnos.
- Disponer de puestos de trabajo en un mismo lugar con equipamiento adecuado, cerca de la contraparte y de los usuarios finales para todas las sesiones presenciales.
- Asistir a las tres presentaciones de avance de los proyectos.
- Comprometerse a mencionar la participación del DCC y del curso Proyecto de Software, en notas de prensa que puedan aparecer relativas al proyecto desarrollado.

Figura 1.1: Criterios a cumplir para poder participar en el curso

Actualmente, este curso cuenta con la página online: <https://portalpsw.dcc.uchile.cl>, la cual fue creada y modificada a través del tiempo por diferentes alumnos del DCC en el ramo “Ingeniería de Software 2”, siendo el semestre de primavera de 2021 la última vez que se trabajó en su código.

El sitio web anteriormente mencionado posee dos tipos de interfaces, creadas para dos tipos diferentes de usuarios. Una de estas (Figura 1.2) es la página pública correspondiente al link presentado en el párrafo anterior, la cual está enfocada en los usuarios que podrían ser futuros clientes de este curso. En ella, representantes de diferentes organizaciones pueden postular sus proyectos para tomar parte en un proceso de selección. De ser elegidos, podrán participar en el curso del próximo semestre. Al mismo tiempo, la página cumple otras funciones como:

- Dar el contexto del curso a los usuarios que visiten el sitio
- La posibilidad de ver los proyectos que se han realizado con éxito en semestres anteriores
- Una explicación detallada de todos los pasos que hay que seguir para poder postular
- La posibilidad de descargar el formulario que se debe rellenar y enviar para postular
- El contacto de la coordinadora del curso



Figura 1.2: Interfaz para los potenciales clientes

La segunda interfaz (Figura 1.3) posee un único usuario que utiliza la página de forma constante. El usuario es el profesor a cargo del curso. Además, secretaria del DCC, el jefe de desarrollo del DCC o el autor de esta memoria poseen cuentas para situaciones específicas, tales como la modificación del sitio web o la revisión de los pagos al postular. Para acceder a esta, se debe acceder al link <https://portalpsw.dcc.uchile.cl/admin> y se debe ingresar un usuario y contraseña. En esta interfaz se puede revisar todas las organizaciones que han postulado al curso, así como también la información que estas proporcionaron al momento de hacerlo. Además de esto, la interfaz permite:

- Crear o editar una convocatoria
- Subir o eliminar los documentos que deben descargarse y rellenar para postular
- Aceptar o rechazar a los postulantes de la convocatoria, así como también informarles sobre la decisión
- Relacionar cada proyecto aceptado con los integrantes del equipo que trabajara en él y una descripción creada por el profesor
- Decidir cuál de los proyectos realizados se muestran en la interfaz pública

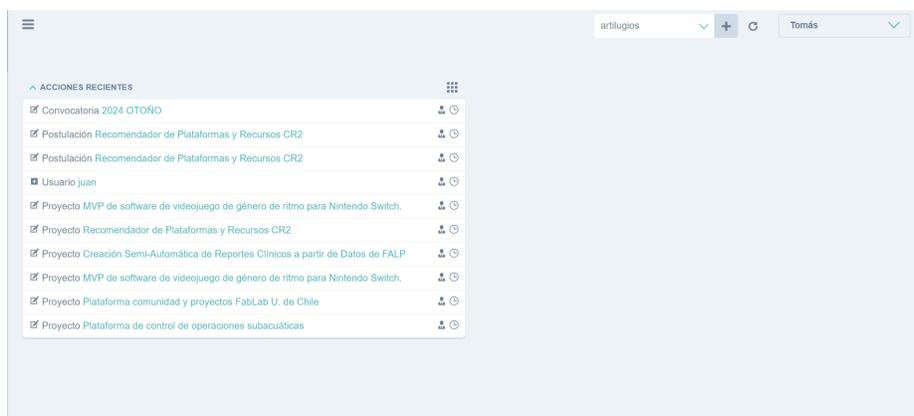


Figura 1.3: Interfaz para el profesor a cargo del curso

Ambas interfaces permiten que el proceso de postulación de la página funcione de manera correcta, como se presenta en la siguiente lista:

- Primero, se inicia una convocatoria en la interfaz de los clientes para postular
- Segundo, los clientes postulan
- Tercero, estos son considerados por el profesor a cargo del ramo
- Cuarto, se les notifica si fueron o no seleccionados
- Quinto, se presentan los proyectos aprobados al finalizar el semestre en la interfaz de los clientes

Sin embargo, al momento de iniciar esta memoria, esta página igualmente contaba con distintas fallas. La interfaz de los clientes contaba con problemas de usabilidad, ya que entregaba información errónea o desactualizada a sus usuarios y, además, en ocasiones se generaban problemas con respecto al documento con el que se postulaba al curso.

Por otro lado, la interfaz del profesor a cargo del curso contaba igualmente con problemas de usabilidad, alguna de las funcionalidades no eran intuitivas de usar debido al nombre que poseían, además, poseía múltiples elementos o funcionalidades que no eran utilizados y solo contribuían al desorden en esta. Por otro lado, había funcionalidades que se debían utilizar para más de un propósito, facilitando el uso desordenado de la interfaz.

Paralelamente, la interfaz del profesor contaba con problemas en su gestión de identidad y acceso. El inicio de sesión presentaba problemas con el método de recuperación de contraseña y, una vez iniciada la sesión, la interfaz contaba con un inconveniente en el caso de inactividad prolongada del usuario.

Además, el software del portal contaba con problemas que afectan su mantenibilidad y, por lo tanto, la de toda la página web. Esta situación se debía a que, junto con los elementos no deseados en la interfaz del profesor que igualmente estaban presentes en el código,

existían también componentes en el código que no tenían efecto alguno en las interfaces, que correspondían a vestigios de trabajos pasados que nunca fueron debidamente eliminados.

Finalmente, y relacionándose directamente a la complejidad que existía al momento de trabajar en el código de proyecto, el sitio web se encontraba funcionando con un software que no había sido actualizado, en teoría, desde su inicio. Esto implicó que el sitio utilizara una versión desactualizada del framework Django, el cual no solo contenía librerías deprecadas, sino que no era la versión estándar que utiliza la oficina de software del DCC, que es quien mantiene la aplicación. En consecuencia, se decidió realizar una reingeniería a este sitio web. En específico, para esta memoria, se decidieron cuatro puntos que engloban los objetivos a completar para realizar la reingeniería del portal que tributa a obtener una versión mejorada del sitio web. A continuación se establecen los objetivos de esta memoria.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Realizar una reingeniería en el sitio web del curso Proyecto de Software, manteniendo su funcionalidad intuitiva para los usuarios públicos, sin embargo, mejorando su mantenibilidad, seguridad, estética y usabilidad.

### 1.2.2. Objetivos Específicos

- Mejorar la mantenibilidad del sitio
  1. Eliminar funciones, importaciones y variables que ya no se utilizan
  2. Actualizar el framework y las librerías del proyecto
- Mejorar usabilidad del sitio
  1. Hacer que se incremente de forma automática la cantidad de proyectos realizados mostrados en la página principal
  2. Homogeneizar los proyectos en la vista de Proyectos Realizados
  3. Restringir libertades con respecto a los documentos para postular
  4. Reestructurar la interfaz del profesor a cargo del curso
- Abordar las deficiencias en la gestión de identidad y acceso de la interfaz
  1. Implementar un cierre automático de sesión para la interfaz privada
  2. Agregar una funcionalidad para que los usuarios de la interfaz privada puedan recuperar su contraseña

Realizar estas acciones permitirá tener un sitio web acorde con la calidad del Departamento de Ciencias de la Computación y, además, facilitará el poder trabajar en este proyecto en el futuro en caso de querer realizar cambios o agregar nuevas funcionalidades.

# Capítulo 2

## Marco Teórico

En este capítulo se exponen los principales conceptos teóricos y tecnológicos involucrados en el desarrollo de esta memoria. También se describen tecnologías que componían el portal, soluciones existentes y la influencia de ambas en el desarrollo de la memoria.

### 2.1. Reingeniería de Software

La reingeniería de software es una práctica que, debido al entorno tecnológico que cambia constantemente, es cada vez más necesaria. La reingeniería es el proceso de analizar y modificar un sistema de software existente para convertirlo en un sistema que cumpla con los requisitos actuales y mejore su estructura interna. Se enfoca en mejorar la usabilidad del software, su mantenibilidad, seguridad, sin necesidad de reescribirlo completamente[10].

Esta práctica puede llevar a uno o más beneficios para el sistema al cual se aplica: puede mejorar su usabilidad y funcionalidad, puede modificarse el lenguaje de programación que se está utilizando, puede facilitar el mantenimiento del software y, por lo tanto, ampliar su vida útil, puede aumentar o mejorar sus medidas de seguridad [7]. En definitiva, mejora la calidad del sistema en general, tanto para los usuarios como para los desarrolladores. En esta memoria nos centraremos en los tres beneficios que se desarrollarán a continuación.

#### 2.1.1. Mantenibilidad de software

El artículo “Developing maintainable software”, publicado en el SSI (Software Sustainability Institute)[11], describe un software mantenible como un software que permite de forma rápida y sencilla: Corregir bugs; añadir nuevas características sin generar bugs, mejorar el rendimiento del software, realizar cambios que prevengan errores en el futuro. Para conseguir esto, se analizarán y modificarán los archivos y dependencias del código.

Existen cuatro tipos de mantenimientos de software[13]:

- Mantenimiento correctivo de software
- Mantenimiento preventivo de software
- Mantenimiento perfectivo de software
- Mantenimiento adoptivo de software

En esta memoria nos centraremos en realizar un mantenimiento preventivo, el cual se centra en prevenir que algún error ocurra y también uno adoptivo, que se centra en reemplazar las tecnologías utilizadas. Estos cambios se explicarán en detalle en el capítulo tres de esta memoria.

### 2.1.2. Usabilidad de software

La usabilidad se refiere a la facilidad con que las personas pueden utilizar un producto o servicio para alcanzar sus objetivos específicos de manera eficiente, eficaz y satisfactoria en un contexto de uso determinado. En otras palabras, mide qué tan fácil, intuitivo y agradable es usar un producto o servicios [9]. Esta característica se subdivide a su vez en seis subcaracterísticas señaladas en la Figura 2.1.

- **Reconocibilidad de la adecuación.** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Aprendizabilidad.** Capacidad del producto que permite al usuario aprender su aplicación.
- **Operabilidad.** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Figura 2.1: Subcaracterísticas que conforman la característica usabilidad

En esta memoria, se trabajó en mejorar tres de estas características, en el caso de la interfaz del profesor, se mejoró la operabilidad, en cuanto a la interfaz de los clientes, fue la protección contra errores y la estética de la interfaz en lo que se enfocó.

### 2.1.3. Seguridad

La seguridad web abarca un conjunto de medidas y tecnologías que protegen a usuarios, dispositivos y redes de ataques cibernéticos a través de Internet. Su objetivo principal es garantizar la integridad, confidencialidad y disponibilidad de la información y los sistemas web. Existen varias formas de seguridad web tales como: criptografía, protección contra malware o gestión de identidad y acceso. En este informe será este último aspecto de seguridad que tendrá relevancia.[15]

La IBM(International Business Machines Corporation) [4] describe la gestión de identidades y acceso(IAM) como la disciplina de seguridad que hace posible que las entidades

correctas (en el caso de esta memoria, personas) usen los recursos correctos (aplicaciones o datos) cuando los necesiten, sin interferencias, usando los dispositivos que quieran utilizar. Es en este tipo de seguridad en la cual nos enfocaremos en esta memoria. Cuando se hable de seguridad de interfaz en esta memoria, será a este concepto al que se refiera.

## 2.2. Tecnologías

Debido a que esta memoria consistió en trabajar sobre una aplicación que ya existía, fue necesario hacer una revisión de cada una de las tecnologías, para así determinar si estas se debían mantener o reemplazar, con el propósito de mejorar el producto final.

### 2.2.1. Django

Django es un framework de aplicaciones gratuito y de código abierto escrito en el lenguaje Python. Este fue desarrollado por Adrian Holovaty, Simon Willison en 2003. [3]

Pese a que, según la encuesta “2023 Developer Survey” realizada por Stack Overflow Django ya no se encuentra entre los frameworks más utilizados (siendo superado principalmente por frameworks que utilizan el lenguaje javascript tales como node.js, react o jquery) [12], el DCC tiene a Django como el framework estándar para la creación de sus aplicaciones. Por lo tanto, en vez de reemplazarlo, se le actualizó de la versión 2.2.25 a la actual 4.2.5.

### 2.2.2. Docker

Docker lleva en funcionamiento desde el 2013 gracias a su autor Solomon Hykes. Es una plataforma de código abierto que permite a los desarrolladores crear, implementar, ejecutar, actualizar y gestionar contenedores, que son componentes estandarizados y ejecutables que combinan el código fuente de la aplicación con las bibliotecas y dependencias del sistema operativo necesarias para ejecutar ese código en cualquier entorno. [5]

A diferencia de Django, Docker se encuentra en la primera posición en la encuesta de herramientas extras que ocupan los desarrolladores web. Además de esto, Docker, al igual que Django, es la plataforma estándar utilizada por el DCC para sus proyectos y, al mantener este tipo de estándares, se facilita la mantención del software a largo plazo.

### 2.2.3. PostgreSQL

Esta base de código abierto, liderado en sus inicios por Michael Stonebraker, lleva en funcionamiento desde 1982. PostgreSQL posee una sólida reputación debido a fiabilidad, flexibilidad y soporte de estándares técnicos abiertos, PostgreSQL soporta tipos de datos

relacionales y no relacionales, lo cual la convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles actualmente. [6]

Al igual que Docker, PostgreSQL se encuentra en el primer lugar con respecto a las bases de datos más utilizadas según la Stack Overflow. Es por esto que, junto a los dos puntos finales mencionados en la sección de Docker, se decidió mantener el uso de PostgreSQL.

# Capítulo 3

## Software del sitio

En este Capítulo de la memoria se comentará el estado en el que se recibió el software, las diferentes soluciones que se aplicaron y el resultado final obtenido.

### 3.1. Obtención y estado inicial

Se obtuvo acceso al repositorio de potalpsw (correspondiente al sitio web), por medio del github del DCC. Una vez descargado el código se identificaron los problemas que este poseía, los cuales se detallarán a continuación.

En primer lugar, el código contaba con muchos archivos, importaciones, funciones, variables y comentarios que deberían haber sido eliminados al finalizar modificaciones de semestres anteriores, ya que no poseían utilidad alguna para el funcionamiento y comprensión del código. Esto generaba desorden en el código, lo cual dificultaba futuros cambios en este.

El segundo problema correspondía a la versión del framework y las otras dependencias del código. Estas versiones ocupadas diferían demasiado de las versiones actuales (Figura 3.1), siendo el caso más evidente y problemático el framework Django, el cual se encontraba en su versión 2.2.25 y la actual es la versión 4.2.5. La diferencia en estas versiones, por sí sola, implicaba dificultades para trabajar en el código, debido a que había que utilizar métodos, funciones y librerías aptas para una versión con varios años de antigüedad. Al mismo tiempo, la versión ocupada por el portal ya se encontraba en proceso de deprecación, a tal punto que el sitio ya no podría volver a levantarse en caso de que este dejara de funcionar.

### 3.2. Solución

Con la situación mencionada en la sección anterior, se decidió implementar una solución de cuatro pasos:

```

1 # https://www.djangoproject.com/download/
2 # Django 2.2 LTS será deprecado a inicios del 2022
3 Django==2.2.25
4 psycopg2==2.9.2
5 requests==2.26.0
6 django-mail-templated==2.6.5
7 django-jet2==1.0.12
8 Pillow==8.4.0
9 django-ckeditor==5.7.1
10 python-jose==3.3.0
11 django_cron
12 django-extensions==3.1.5
13 python-magic==0.4.24
14 django-model-utils==3.2.0
15 django-debug-toolbar==3.2.4
16 django-dbbackup==3.3.0

```

Figura 3.1: Versiones de las dependencias al inicio de la memoria

- Solucionar de manera momentánea la situación que no permitía levantar el sitio
- Eliminar el código sin utilidad
- Actualizar todas las dependencias restantes
- Solucionar los errores generados por la actualización

La razón por la que se optó por eliminar parte del código antes de actualizarlo, se debió a que se generarían errores debido a que algunas funciones, importaciones y librerías ya no serían compatibles con estas nuevas dependencias y con la intención de optimizar el tiempo, se quería evitar solucionar errores producidos por elementos del código que deberían ser eliminados terminada la actualización.

### 3.2.1. Solucionar de manera momentánea la situación que no permitía levantar el sitio

Se cambió la versión Python con la que se estaba trabajando en el archivo dockerfile (utilizado para levantar el sitio web de forma local) de la versión 3.8.12-slim-bullseye a 3.11-slim, esto permitió seguir utilizando la versión antigua de Django y la mayoría de las dependencias, con la excepción de la librería Pillow (librería permite facilitar el manejo de imágenes), la cual se actualizó de la versión 8.4.0 a la versión 9.5.0.

### 3.2.2. Eliminar el código sin utilidad

Esta parte de la solución está dividida en dos partes, en primer lugar se eliminaron elementos pertenecientes a la interfaz del profesor y el código correspondiente a estos, este proceso será comentado en el Capítulo 4 de este informe.

En la segunda parte, se procedió a eliminar las partes del código que no se utilizaban, siendo la mayoría las importaciones al inicio de los archivos. También se eliminó una carpeta

que contenía archivos HTML para funciones que nunca fueron agregadas al sitio. Finalmente, se buscaron funciones que no tuvieran utilizad para el portal, pero se desistió, ya que encontrar una función en algún archivo implicaba un uso de tiempo muy grande en comparación al beneficio de encontrar una.

### **3.2.3. Actualizar todas las dependencias restantes**

A fin de disminuir la cantidad de errores que se generarían una vez actualizadas las dependencias, se reemplazó el repositorio portalpsw por el repositorio “ext-portalpsw”, el cual se creó utilizando el repositorio django-boilerplate. Este último es, actualmente, el template base para Sistemas de Información del Área de Desarrollo de Software, el cual cuenta con una versión de Django actualizada y el proyecto django\_utils, que permite utilizar las credenciales de MiUchile.

Con esto se pretendía contar con una nueva base, que poseyera las características base de las páginas del DCC (y en consecuencia también del portal de Proyecto de Software) con la diferencia clave de que esta tendría sus dependencias actualizadas. A continuación se agregaron las versiones actuales de las dependencias que el nuevo repositorio no poseía y se integró el código del repositorio portalpsw.

Debido a este proceso, se evitó tener que solucionar los problemas que se generarían en los archivos que compartían ambos repositorios, teniendo que centrarse únicamente en tres dependencias que ya no eran compatibles con la nueva versión de Django y cambios en los nombres de algunas variables e importaciones.

### **3.2.4. Solucionar los errores generados por la actualización**

#### **Django-jet2**

Esta dependencia reemplaza los templates del administrador de Django estándar por unos nuevos (Figura 3.2), con vistas más agradables y organizadas[2]. Pese a que al día de hoy django-jet continúa estando funcional, la dependencia con la que se obtenía acceso estaba configurada para funcionar con la versión dos de Django, por lo que al actualizarlo a la versión cuatro, fue necesario reemplazarla.

Inicialmente, se consideró reemplazar Django Jet por una librería que fuera más moderna y fácil de integrar con Django, sin embargo, no se encontró un reemplazo que justificara modificar el código para que funcionara con una nueva librería y que su usuario debiera adaptarse a una nueva funcionalidad. Por lo tanto, se reemplazó django-jet2 por django-4-jet, la cual permitía utilizar la versión de Django Jet apta para la nueva versión de Django.

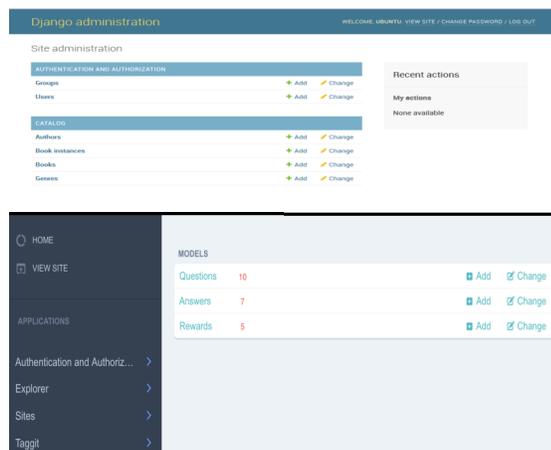


Figura 3.2: Diferencia entre el admin de Django normal (arriba) y con django-jet (abajo)

## Python-magic

Python-magic es una librería actualmente deprecada que se utilizaba para poder confirmar que los documentos enviados por los postulantes eran realmente pdf y no estaban corruptos. Esta dependencia funcionaba utilizando la librería magic para Linux, transformándola a una versión apta para Windows.

Se buscó librerías que fueran compatibles con la nueva versión de Django y que igualmente adaptaran la librería magic para que funcionara en Windows, sin embargo, estas no fueron encontradas. Y las nuevas librerías utilizadas para reemplazar a python-magic no eran compatibles con la funcionalidad que se le daba a esta librería en el sitio web.

Fue por esto que se cambió la búsqueda a una dependencia que permitiera revisar los documentos. Como resultado, se empezó a utilizar django-easy-pdf, el cual, mediante la función PdfReader determinaba si el archivo era o no un PDF y si estaba o no corrupto. Esta dependencia cumplió el propósito inicial, una vez reemplazadas todas las dependencias que ya no servían, el código se pudo ejecutar sin problema alguno. Sin embargo, al intentar validar la postulación, surgió un error, ya que PdfReader era incapaz de trabajar con el archivo que se le entregaba. Es por esto que finalmente se terminó utilizando un módulo estándar de python llamado "mimetypes" el cual no presentó ningún problema en la validación.

## Psycopg2

Finalmente, Psycopg es el adaptador más popular actualmente de PostgresPSQL para python, al igual que con django-jet2, esta se obtenía mediante una dependencia que dejó de funcionar al momento de dejar de utilizar la versión dos de Django, por lo se buscó una nueva forma de utilizar este adaptador y se empezó a utilizar la dependencia psycopg.

Una vez reemplazadas estas dependencias, el sistema se pudo levantar sin problema al-

guno, por lo que, se empezó a buscar errores al ocupar la página debido a los cambios realizados (por ejemplo, el que se mencionó con respecto a pdfReader). Surgieron distintos problemas a la hora de tratar de utilizar distintas vistas, tales como la página principal o la página de proyectos realizados. Estos errores eran causados, ya que algunas funciones utilizaban variables o paquetes que solo funcionaban con las versiones antiguas y era necesario modificarlas, por ejemplo:

- Se reemplazó el paquete `ugettext`, que ya no era factible, por uno que cumplía la misma función llamada `gettext`
- Se reemplazó todas las variables `NullbooleanField` de los archivos `models` por `booleanField`

Por otro lado, igualmente surgieron problemas en la base de datos debido a la actualización, ya que el sistema no era capaz de conectarse con la base que se le estaba entregando. Debido a que este problema no se logró solucionar de forma rápida, se tomó la decisión de volver al antiguo repositorio (Figura 3.3). Esto no implicaba que se perdería todo el trabajo realizado, ya que se modificaron todas las carpetas y archivos para tener la misma información que el repositorio en el que se trabajó, la única excepción al momento de editar los archivos fue lo relacionado con la base de datos.

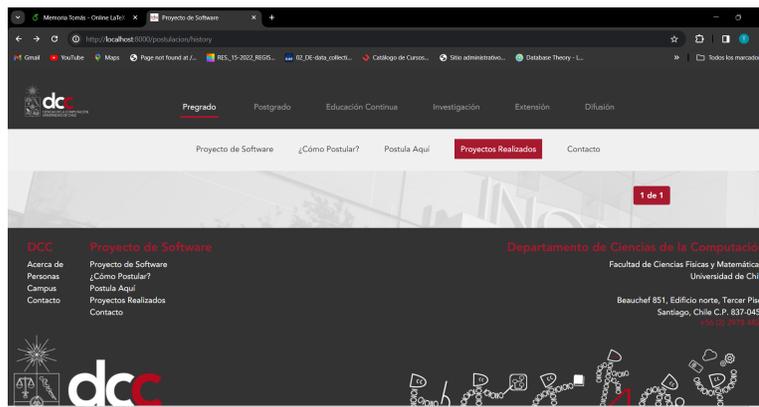


Figura 3.3: Vista de los proyectos realizados debido al error con la base de datos

Siguiendo esta idea, el sitio web volvió a funcionar igual que lo hacía antes de la actualización, con la diferencia de que ahora sería mucho más simple agregar nuevas funcionalidades, ya que se podrían buscar soluciones para las versiones actuales. Es por esto que, pese a que el método que se eligió no funcionó completamente, al no haber implicado un gasto extra de tiempo, se considera que utilizar este método fue un acierto.

### 3.3. Estado final

Una vez terminado este proceso, ambas interfaces del portal funcionaban de igual manera que hacían en su versión original, la diferencia se encontraba que al momento de trabajar en

estas, el código estaba más ordenado y se podía utilizar información y foros actuales para agregar nuevos elementos. Al mismo tiempo, la actualización de las dependencias (Figura 3.4) hizo pasar de un sitio que prácticamente carecía de soporte, a uno que posee un soporte de, a lo menos, 5 años.

```
Django==4.2.5
django-auto-logout==0.5.1
django-ckeditor==6.7.0
django-cron==0.6.0
django-dbbakup==4.0.2
django-debug-toolbar==4.2.0
django-easy-pdf==0.1.1
django-extensions==3.2.3
django-4-jet==1.0.9
django-json-widget==1.1.1
django-mail-templated==2.6.5
django-model-utils==4.3.1
Pillow==10.0.1
psycopg==3.1.12
python-daemon==3.0.1
python-docx==0.8.11
python-jose==3.3.0
requests==2.31.0
weasyprint==60.1
```

Figura 3.4: Estado de las dependencias al finalizar la memoria

# Capítulo 4

## Interfaz de los clientes

En este capítulo de la memoria se dará un contexto sobre esta interfaz y se explicará: los problemas de esta al inicio de la memoria, como estos problemas fueron solucionados, y el estado final de la interfaz, cabe mencionar que el cambio de colores es debido a la nueva versión de Django-jet mencionado en la sección 3.2.4 de este informe.

### 4.1. Contexto

Esta interfaz corresponde al front-end del sitio, cuenta con cinco vistas (todas creadas mediante el uso de HTML, CSS, JavaScript y JQuery) y en esta memoria se trabajó en tres de ellas: la vista principal, que se encarga de entregar información sobre el curso, como se puede apreciar en la Figura 4.1. También informa a los clientes cuando está abierta una convocatoria, tal como se presentó en la figura 1.2 en la introducción.



Figura 4.1: Vista principal de la interfaz de los clientes

Otra de las vistas en las que se trabajó corresponde al formulario que hay que completar para poder postular al curso (Figura 4.2), en este se debe rellenar con información de la

organización que postula, de la persona que la representará a lo largo del semestre y un documento que posee múltiples datos del proyecto propuesto.

Pasos para postular:

- 1 Revisar criterios de participación
- 2 Descargar y llenar perfil de proyecto
- 3 Completar registro en línea

Organización

\* Rol Único Tributario (RUT): Ej: 11111111-1

\* Razón Social

Rubro o Giro

Unidad

Nombre de la unidad dentro de la organización

Seleccionar archivo... **Logotipo (jpg, png, mpj)**

Responsable

\* Nombre y Apellido

\* Email

Proyecto

\* Título **Completar este campo**

Ingrese una pequeña descripción del proyecto.

Seleccionar archivo... **Perfil de proyecto (pdf)**

No soy un robot

**Finalizar Postulación**

Figura 4.2: Vista para postular de la interfaz de los clientes

La última vista en la que se trabajó (Figura 4.3) presenta los proyectos que se han realizado con éxito en este curso, aquí se debe presentar el nombre del proyecto, el logo de la empresa, una pequeña descripción del proyecto, los integrantes del equipo que lo desarrolló y el semestre en el que se realizó.



Figura 4.3: Vista de los proyectos realizados de la interfaz de los clientes

Las últimas dos vistas que no se modificaron corresponden a la que contiene toda la información y documentos que los clientes necesitan para poder postular al curso y la vista que entrega toda la información necesaria en caso de querer contactarse con el profesor a cargo del curso.

## 4.2. Problemas de las vistas

### 4.2.1. Vista principal

La razón por la que se decidió trabajar en esta vista fue debido a los errores que tenía la información entregada:

- El año de inicio del curso es 1997, sin embargo, en el sitio indica que fue en 1998
- Se indica que los grupos están conformados por 5 a 7 alumnos, pero actualmente los equipos se forman por grupos de 5 a 8
- La cantidad mínima de proyectos realizados que se presentaba (120) era muy baja en comparación a la cantidad actual (aproximadamente 250)

### 4.2.2. Vista para postular

A diferencia del caso anterior, no era la parte visual de la vista la que había que modificar. En una convocatoria abierta, las organizaciones eran capaces de postular con un documento diferente al que se solicita (y permite descargar) en esta misma vista, esto implicaba que el profesor a cargo del curso debería solicitar a la organización que enviara el documento correcto. Por otro lado, también se daban caso en los que se postulaba con el documento correcto, sin embargo, debido a que este superaba la cantidad máxima gigabytes, la página dejaba de funcionar.

### 4.2.3. Vista de proyectos realizados

Esta vista presentaba dos tipos de problemas, visualmente, los proyectos ocupaban distinta cantidad de espacio y la información correspondiente a cada proyecto solo era constante en presentar el nombre del proyecto y el logo de la empresa, los equipos no siempre incluían a los tutores y cada proyecto presentaba una de las siguientes combinaciones con el resto de la información:

- Semestre en el que se realizó, equipo que lo realizó, descripción del proyecto
- Semestre en el que se realizó, equipo que lo realizó
- Descripción del proyecto, equipo que lo realizó
- Semestre en el que se realizó, descripción del proyecto
- Descripción del proyecto

Por otro lado, no todos los proyectos que se deseaban mostrar aparecían en esta vista, ya que algunos proyectos se repetían, esto no solo era un problema debido a que era un claro error

en el sitio, sino que también había proyectos que no se mostraban debido a que la repetición utilizaba su lugar(Figura 4.4). Además, tomando en cuenta que son las organizaciones las que señalan si desean que se publique o no el proyecto, el que su proyecto no fuera mostrado se prestaba para malentendidos con el curso.



Figura 4.4: Proyectos repetidos en slides diferentes

## 4.3. Solución

### 4.3.1. Vista principal

Para solucionar los tres problemas que presentaba esta vista, se trabajó el template correspondiente a esta vista “home.html”. Se corrigió el texto “Desde 1998” por “Desde 1997” y se reemplazó el texto “Grupos de 5 a 7 ...” por “Grupos de 5 a 8 ...”.

Para solucionar el problema con respecto a la cantidad de proyectos, se modificó el *div* correspondiente al texto “Más de 120 proyectos realizados”. Este texto fue reemplazado por “Un total de `{{object_list|length}}` proyectos realizados”, donde `{{object_list|length}}` se muestra de color rojo y se reemplaza en la vista principal por la cantidad de proyectos aprobados que se encuentran en el sistema. Este valor se irá actualizando de manera automática semestralmente a medida que nuevos proyectos se vayan realizando.

El texto `object_list|length` obtiene el largo de la lista “`queryset`”, la cual contiene los proyectos realizados con éxito. Esta variable se creó en la clase `HomeView` (clase utilizada para la vista principal), y esta obtiene todos los proyectos que han sido aprobados en la interfaz del profesor. Igualmente, se cambió la clase base de `HomeView`, de “`TemplateView`” a “`generic.ListView`”, para que así la actualización se realizara de forma automática cada vez

que se aprobara un nuevo proyecto en la interfaz del profesor y no se tuviera que volver a levantar el sitio cada vez que terminara la evaluación de los proyectos.

### 4.3.2. Vista para postular

Para evitar que los postulantes no utilizaran un documento, se planteó implementar un sistema (mediante alguna librería) que pudiera confirmar que el documento enviado fuera el solicitado, sin embargo, no se logró dar con ningún sistema ya existente que permitiera esto y se desestimó crear uno en esta memoria, ya que no fue posible llegar a una solución factible que se pudiera realizar sin utilizar el tiempo correspondiente a otros objetivos.

Sin embargo, se modificó el template “applicationForm.html” agregando un *div* con el texto “Debe descargar, rellenar y enviar ESTE documento”, donde la palabra ESTE se encontraba en rojo y subrayado, en caso de presionar dicha palabra, se descargaría el formulario que se debe enviar.

### 4.3.3. Vista de proyectos realizados

Inicialmente, se solucionó las discrepancias en tamaño e información que presentaba el sitio. Para homogeneizar el tamaño, se modificó el CSS del template “project.list.html” para imponer que el alto y ancho del *div* establecido se cumpliera, sin importar la diferencia en la información entregada. También se agregó un ancho y alto obligatorio a los logos de las empresas, ya que no todos tenían las mismas medidas y luego se escalaban para que no se deformaran por el espacio (como en los casos de la Figura 4.3).

Luego, para homogeneizar la información presentada, se reestructuró la vista, para que presentara en todos los proyectos: el logo de la organización; el nombre del proyecto; una descripción del proyecto; el semestre y año en el que se realizó y el equipo que participó, incluyendo alumnos y tutor.

Para esta medida, se modificó en el template “project.list.html”, agregando la fila para colocar el semestre y una distinción entre los alumnos y el tutor. Paralelamente, se agregaron variables a la clase `ApprovedProject`, de la cual se obtenía toda la información de cada proyecto, estos cambios serán discutidos en el capítulo cinco de este documento, ya que corresponden a cambios realizados en la interfaz del profesor.

Finalmente, se modificó la clase `Projects` del archivo `views.py`, la cual se encargaba de entregar al template “project-list.html” todos los proyectos que han sido marcados como visibles en la interfaz del profesor. Aquí se agregó a la línea de código encargada de obtener los proyectos visibles, el método `distinct`, el cual se encarga de que no fuera posible que aparecieran proyectos repetidos.

## 4.4. Estado final

### 4.4.1. Vista principal

Luego de los cambios realizados, la vista principal ya no posee errores en la información correspondiente al curso y, además, ahora permite tener una idea más cercana a la cantidad de proyectos realizados (Figura 4.5), es importante destacar que la cantidad de proyectos mostrados en la figura no corresponde a la cantidad total del sitio, sino a la base de datos con la que se trabajó en esta memoria.



Figura 4.5: Nueva vista principal

### 4.4.2. Vista para postular

Con los cambios realizados en el HTML de esta vista se espera minimizar el envío de documentos erróneos (Figura 4.6).

Pasos para postular:

- 1 Revisar criterios de participación
- 2 Descargar y llenar perfil de proyecto
- 3 Completar registro en línea

**Organización**

- \* Rol Único Tributario (RUT): E: 11111111-1
- \* Razón Social
- Rubro o Giro

**Unidad**

Nombre de la unidad dentro de la organización

Seleccionar archivo... **Logo (.jpg, .png, .svg)**

**Responsable**

- \* Nombre y Apellido
- \* Email

**Proyecto**

- \* Título
- Ingrese una pequeña descripción del proyecto.

Debe descargar, rellenar y enviar **ESTI\_documento**.

Seleccionar archivo... **Perfil de empresa (.pdf)**

**Finalizar Postulación**

Figura 4.6: Nueva vista para postular

### 4.4.3. Vista de proyectos realizados

Se consiguió que desaparecieran los proyectos repetidos e igualmente se logró homogeneizar de manera visual e informativa todos los proyectos presentados (Figura 4.7).

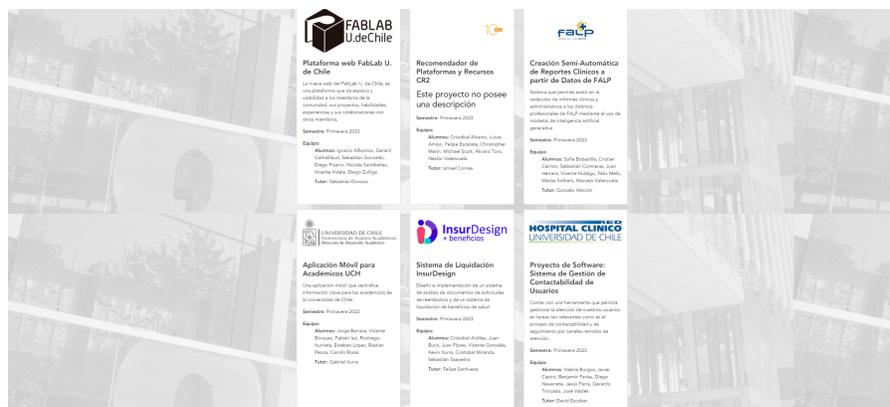


Figura 4.7: Nueva vista de proyectos realizados

# Capítulo 5

## Interfaz del profesor

En este capítulo de la memoria se dará un contexto sobre esta interfaz y se explicará: los problemas de esta al inicio de la memoria, como estos problemas fueron solucionados, y el estado final de la interfaz.

### 5.1. Contexto

Esta interfaz corresponde al administrador de Django y, por lo tanto, al back-end del sitio web. Es accesible por medio de un inicio de sesión que requiere un nombre de usuario y contraseña, cuenta con una vista principal que posee un historial de lo que ha ocurrido en el sitio y desde la cual se puede acceder a las otras 15 vistas que posee (Figura 5.1). En esta memoria se trabajó únicamente en 8 de estas vistas:

- Vista relacionada con las postulaciones
- Vista relacionada con proyectos realizados
- Vista relacionada con las plantillas para postular
- Vista del portal de revisión
- Las cuatro vistas relacionadas con los documentos

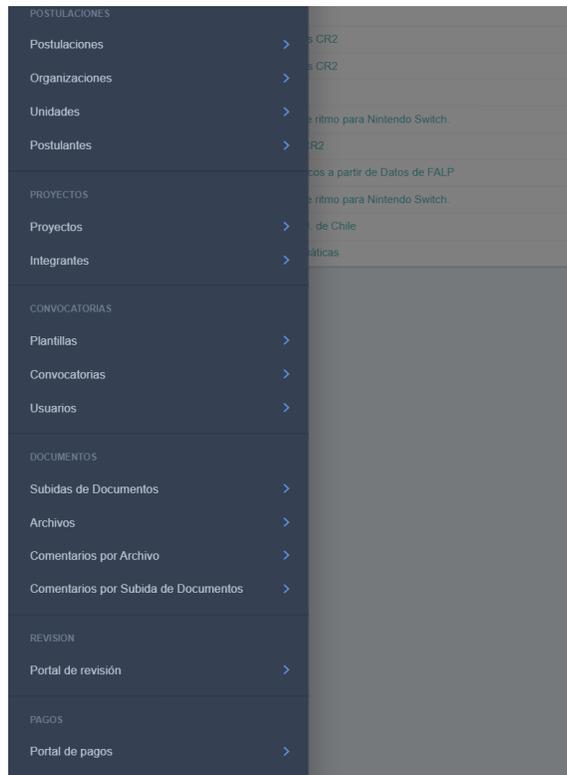


Figura 5.1: Vistas accesibles desde la página inicial de la interfaz del profesor

## Postulaciones

Esta vista presenta una tabla que contiene información básica de todas las postulaciones realizadas (Figura 5.2), igualmente permite descargar el documento de cada postulación entregado y notificar a los postulantes sobre si su proyecto fue aceptado o rechazado. Además de esto, en caso de seleccionar una fila de la tabla, se podrá acceder a una nueva vista en la cual se puede modificar toda la información de dicha fila.

TÍTULO	CONVOCATORIA	ORGANIZACIÓN	UNIDAD	POSTULANTE	ESTADO	NOTIFICADO	PUNTAJE	PERFIL DE PROYECTO	NOTIFICAR	ACCIONES
ERP Restaurantes	2024 OTORO	NevoLab SpA	Innovación	Felipe Viloso	Creada	✗	Ingresar 0	O Descargar Perfil	📧 Felipe Viloso@nevoab.cl	🗑️ ✕
Plataforma de supervisión...	2024 OTORO	error_falta_nombre	Gerencia de Desarrollo Digital	Mauricio Sothers	Creada	✗	Ingresar 0	O Descargar Perfil	📧 mauricio@studio.cl	🗑️ ✕
Recomendador de Plataform...	2023 PRIMAVERA	Universidad de Chile	Desarrollo de software	Francisca Muñoz	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 francisca@uchile.cl	🗑️ ✕
Creación Semi-Automática ...	2023 PRIMAVERA	Fundación Arturo López Pérez	Informática Médica y Data Science	Sergio Penafiel	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 sergio.penafiel@fap.org	🗑️ ✕
MVP de software de videoj...	2023 PRIMAVERA	ADNE GAMES S.P.A.	Ingeniería	Sebastián Gana	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 sebastian@adnegames.com	🗑️ ✕
Plataforma comunidad y pr...	2023 PRIMAVERA	Universidad de Chile	FabLab	Dennis Peric	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 dennis@fablab.uchile.cl	🗑️ ✕
Plataforma de control de ...	2023 PRIMAVERA	Vertex spa	Servicios submatrimos	Luis Martínez	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 luis.martinez@vertexchile.com	🗑️ ✕
Sistema para el acompaña...	2023 PRIMAVERA	Universidad Metropolitana de Ciencias de la Educación	Departamento de Informática	Francisco Segovia	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 francisco.segovia@umce.cl	🗑️ ✕
SmartKam	2023 PRIMAVERA	Lookdata SpA	LookData	Gonzalo Tepeman	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 gtepeman@lookdata.cl	🗑️ ✕
Proyecto de Software: Sis...	2023 PRIMAVERA	Universidad de Chile	Hospital Clínico de la Universidad de Chile Dr. José Joaquín Aguirre	Marcos Baltara	Aceptada	✓	Ingresar 0	O Descargar Perfil	📧 mbaltara@houch.cl	🗑️ ✕

Figura 5.2: Vistas de todas las postulaciones realizadas

## Proyectos Realizados

Aquí se encuentra la información de los proyectos que han sido aceptados para cada convocatoria (Figura 5.3) y es la vista que se encarga de la vista Proyectos Realizados mencionada en la sección 4.1 de este informe. Para editar la información que se presentara en esa vista, se debe hacer clic sobre la fila del proyecto que se desea modificar.

título	CONVOCATORIA	ORGANIZACIÓN	UNIDAD	POSTULANTE	VISIBLE	ACCIONES
<input type="checkbox"/> Sistema de Legitimación IntraDesign	2023 PRIMAVERA	IntraDesign SpA	gerencia	Gabriel Oñares	✓	🔍 ✕
<input type="checkbox"/> Aplicación Móvil para Académicos UCH	2023 PRIMAVERA	Universidad de Chile	Dirección de Desarrollo Académico	Claudio Oñe Azar	✓	🔍 ✕
<input type="checkbox"/> Proyecto de Software: Sistema de Gestión de Contactabilidad de Usuarios	2023 PRIMAVERA	Universidad de Chile	Hospital Clínico de la Universidad de Chile Dr. José Joaquín Aguirre	Marcelo Balboa	✓	🔍 ✕
<input type="checkbox"/> SmaRkam	2023 PRIMAVERA	Loiadata SpA	Loiadata	Gonzalo Tapeman	✕	🔍 ✕
<input type="checkbox"/> Sistema para el acompañamiento a la Trayectoria Formativa	2023 PRIMAVERA	Universidad Metropolitana de Ciencias de la Educación	Departamento de Informática	Francisco Segovia	✓	🔍 ✕
<input type="checkbox"/> Plataforma de control de operaciones subacuáticas	2023 PRIMAVERA	Veritec spa	Servicios submarinos	Luis Martínez	✓	🔍 ✕
<input type="checkbox"/> Plataforma comunitad y proyectos F&A: U. de Chile	2023 PRIMAVERA	Universidad de Chile	F&A:U	Daniela Peric	✓	🔍 ✕
<input type="checkbox"/> MVP de software de videojuego de género de ritmo para Nintendo Switch	2023 PRIMAVERA	AONE GAMES S.P.A.	Ingeniería	Sebastián Gana	✓	🔍 ✕
<input type="checkbox"/> Creación Semi-Automática de Reportes Clínicos a partir de Datos de FALP	2023 PRIMAVERA	Fundación Arturo López Pérez	Informática Médica y Data Science	Sergio Peñafiel	✓	🔍 ✕
<input type="checkbox"/> Recomendador de Plataformas y Recursos CRZ	2023 PRIMAVERA	Universidad de Chile	Desarrollo de software	Francesca Muñoz	✓	🔍 ✕

Figura 5.3: Vistas de los proyectos realizados en cada convocatoria

## Plantillas

Esta vista contiene todas las plantillas que las organizaciones han tenido que rellenar y enviar al momento de postular (Figura 5.4), es importante destacar que no todos estos documentos son necesarios para postular, algunos corresponden a documentos que se utilizaron en convocatorias anteriores.

DESCRIPCIÓN	ACCIONES
<input type="checkbox"/> Nuevo convenio de trabajo híbrido	🔍 ✕
<input type="checkbox"/> Nueva postulación semipresencial	🔍 ✕
<input type="checkbox"/> Plantillas Primavera 2019	🔍 ✕

Figura 5.4: Vistas de las plantillas

## Portal de revisión

Esta vista corresponde a un portal externo a la interfaz, todos los semestres presentan el mismo aspecto que el apreciado en la Figura 5.5, esto no se debe a un error en su funcionamiento, sino que no tiene utilidad para el sitio web.



Figura 5.5: Vistas de las plantillas

## Vistas relacionadas con los documentos

Estas cuatro vistas corresponden a: Subidas de documentos; Archivos; Comentarios por archivo; Comentarios por subida de documentos. Sin embargo, las cuatro vistas tienen la misma apariencia que la presentada en la Figura 5.6.

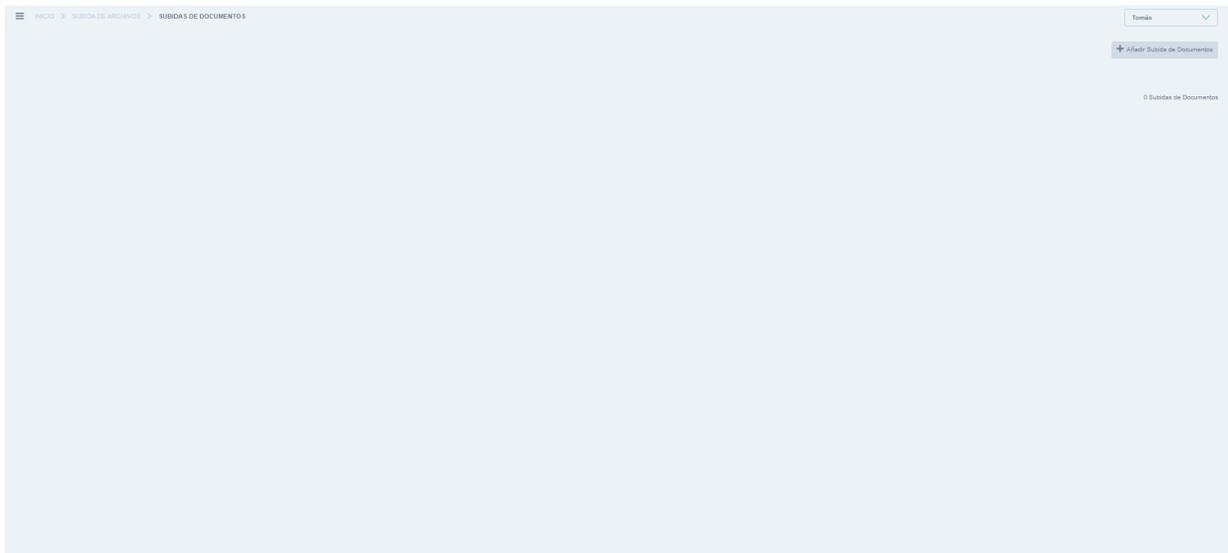


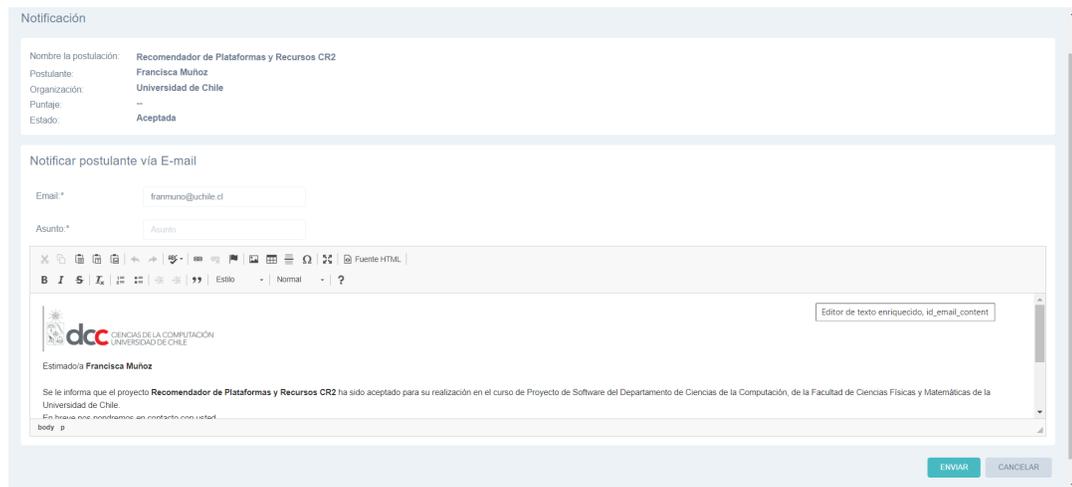
Figura 5.6: Vistas de Subidas de documentos

Las otras siete vistas en las que no se trabajaron fueron: dos vistas relacionadas con las organizaciones que postulaban, una vista relacionada con los representantes de las organizaciones, una vista relacionada con las convocatorias, una vista relacionada con los usuarios de la interfaz, una vista relacionada con los equipos de cada proyecto y finalmente una vista que redirige al usuario al portal de pago.

## 5.2. Problemas de las vistas

### 5.2.1. Vista de postulantes

La tabla que se presentaba en esta vista tenía tres errores, en primer lugar esta tabla contaba con una columna llamada “puntaje”, la cual no había sido utilizada nunca ni se pretendía hacerlo. El segundo problema se debía a que, en caso de querer contactarse con el cliente, se debía utilizar la misma función que permite notificar a las organizaciones sobre el estado de su postulación (Figura 5.7). Esto era un problema, ya que este correo se encontraba escrito de manera automática, por lo que se debía reescribir este correo. Finalmente, se deseaba modificar el nombre del botón “Ir” por uno que fuera más intuitivo.



The screenshot shows a web interface for sending an email notification to a postulant. The form is titled "Notificación" and contains the following fields and content:

- Nombre la postulación:** Recomendador de Plataformas y Recursos CR2
- Postulante:** Francisca Muñoz
- Organización:** Universidad de Chile
- Puntaje:** --
- Estado:** Aceptada

Below this information is a section titled "Notificar postulante vía E-mail" with the following fields:

- Email \*:** frammuno@uchile.cl
- Asunto \*:** Asunto

The email body is composed in a rich text editor with a toolbar. The content of the email body is:

**Estimadora Francisca Muñoz**

Se le informa que el proyecto **Recomendador de Plataformas y Recursos CR2** ha sido aceptado para su realización en el curso de Proyecto de Software del Departamento de Ciencias de la Computación, de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

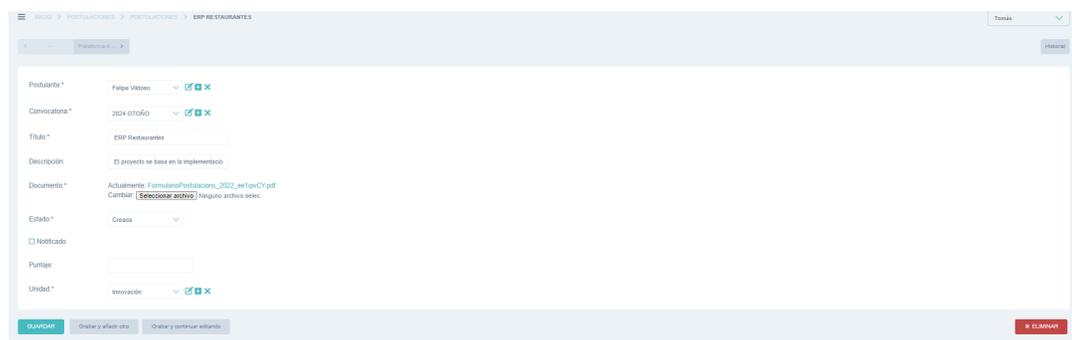
En breve nos contactamos con usted.

body p

At the bottom right of the form are two buttons: "ENVIAR" and "CANCELAR".

Figura 5.7: Vista para notificar a una empresa que postulo

Por otro lado, al acceder a la vista de cada postulación al presionar la fila (Figura 5.8), se presentaban dos elementos que se debían eliminar. El primero era el correspondiente a la columna puntaje, que permitía decidir el valor que iría en la columna de la fila seleccionada. El segundo elemento correspondía a una casilla de verificación que marcaba a la organización como notificada o no en la tabla, esta se debía eliminar, ya que al momento de enviar una notificación a la organización, la columna de notificación de esa postulación marcaba automáticamente que se había realizado la notificación.



The screenshot shows a detailed view of a postulation. The breadcrumb navigation at the top reads: "Inicio > Postulaciones > Postulaciones > ESP Restaurantes". The main content area displays the following information:

- Postulante \*:** Felipe Váñez
- Convocatoria \*:** 2024-07060
- Título \*:** ESP Restaurantes
- Descripción:** El proyecto se basa en la implementación
- Documento \*:** Actualmente: FormularioPostulacion\_2022\_eeIsvCY.pdf. Opción: [Seleccionar archivo](#) | [ninguno archivo selec.](#)
- Estado \*:** Creada
- Notificado
- Puntaje:** [Empty field]
- Unidad \*:** Innovación

At the bottom of the form are three buttons: "Quitar", "Quitar y añadir otro", and "Quitar y continuar editando". A red "ELIMINAR" button is located in the bottom right corner.

Figura 5.8: Vista específica de una postulación

## 5.2.2. Vista proyectos

A diferencia del caso anterior, en este no existían problemas en las filas o columnas de la tabla, para este caso era necesario modificar la vista que aparecía cuando se presionaba una de las filas (Figura 5.9). Se debía eliminar la variable “Ponderación de Relevancia”, ya que, al igual que con “puntaje” en el caso anterior, este no poseía ningún tipo de utilidad.

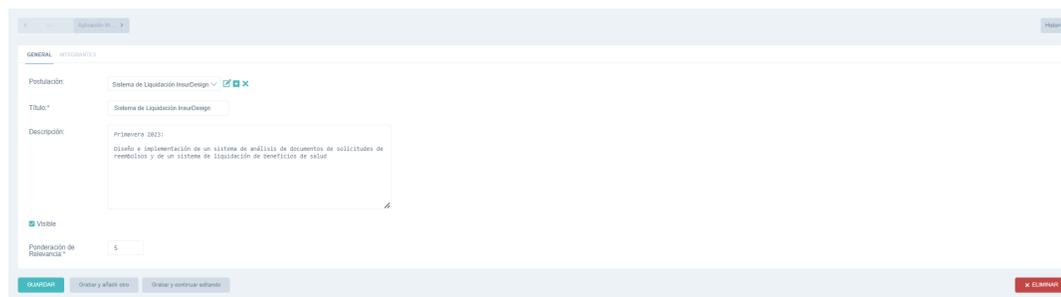


Figura 5.9: Vista específica de un proyecto aceptado

Con respecto a lo que se debían agregar a esta vista, un nuevo elemento que permitiera seleccionar el semestre y año en el cual se realizó el proyecto para que así no estuviera toda la información en el espacio correspondiente a la descripción del proyecto, igualmente se debía agregar una casilla que permitiera ingresar el nombre del tutor del equipo.

## 5.2.3. Vistas de Documentos y portal de revisión

Como se mencionó en la sección anterior, estas cinco vistas no poseen ningún tipo de información y tampoco deberían, se infiere que fueron creadas en algún proceso anterior (o que se creó una funcionalidad que las dejó obsoleta) y que nunca fueron eliminadas.

## 5.2.4. Vista de plantilla

Finalmente, el cambio con respecto a la vista de plantilla correspondía a modificarle el nombre, ya que este no se consideraba adecuado para su propósito.

## 5.3. Gestión de identidad y acceso de la interfaz

### 5.3.1. Recuperación de contraseña

Pese a que la interfaz del profesor contaba con un inicio de sesión, este no contaba con ninguna característica que le permitiera recuperar su contraseña, como se aprecia en la Figura 5.10. En vez de eso, se debía solicitar ayuda al encargado de la oficina de software del DCC para así poder obtener una nueva. Esto es un caso único, ya que en otros sitios de

la universidad, existe una funcionalidad que permite recuperar la contraseña, por ejemplo, Ucourses.



Figura 5.10: Inicio de sesión de la interfaz del profesor

### 5.3.2. Cierre de sesión

Por otro lado, la interfaz no contaba con ninguna característica que le permitiera cerrar de forma automática la sesión de un usuario, sin importar cuanto tiempo hubiera estado inactivo. Esto conllevaba un serio problema para la interfaz, ya que implicaba que la única medida de seguridad existente para evitar que una persona no consentida pudiera ingresar al administrador en caso de tener acceso al computador donde se inició sesión en la interfaz, era que el propio usuario hubiera cerrado esta página de forma manual con anterioridad.

Esta situación era grave, ya que, al obtener acceso a la interfaz del sitio, se obtenía también acceso al portal de pagos, así como también a la capacidad de comunicarse con distintas organizaciones que hayan postulado, haciéndose pasar por el profesor a cargo del curso.

## 5.4. Solución

### 5.4.1. Eliminación de elementos

Como se mencionó en el Capítulo 3 de este informe, junto a la eliminación de los elementos no deseados del código, se realizó también la limpieza de los elementos que se debían quitar mencionados en la sección anterior. Todos los elementos se eliminaron utilizando la misma metodología.

#### Buscar el código correspondiente

Primero, se marcaba las líneas de código correspondientes al elemento que se deseaba eliminar con el comentario: “Marchessi: Eliminar” (Figura 5.11).

```
#Marchessi: Eliminar
score = models.DecimalField(
    max_digits=3,
    decimal_places=2,
    blank=True,
    null=True,
    verbose_name=_("Puntaje"),
    validators=[MaxValueValidator(5), MinValueValidator(0)],
)
```

Figura 5.11: Línea de código comentada que debe ser eliminada

## Revisar elementos relacionados

Luego se revisaban todas las funciones, variables e importaciones que estaban ligados al elemento marcado. Todas estas eran marcadas con un “Marchessi: Revisar”, tal como se aprecia en la Figura 5.12.

```
#Marchessi: revisar
def get_context_data(self, **kwargs):
    ctx = super().get_context_data(**kwargs)
    ctx._setitem_("instance", self.instance)
    score = self.instance.score if self.instance.score else "--"
    ctx["score_front"] = score
    return ctx
```

Figura 5.12: Línea de código comentada que debe ser revisada

## Investigar relaciones de los elementos marcados

Se determinaba si estos solo estaban relacionados con el objeto que se quería eliminar o si cumplían otra función en el sitio. En caso de que fuera la primera opción, se borraba el comentario para así desmarcarlo. En caso contrario, se cambiaba el comentario por un “Marchessi: Eliminar” y se volvía a realizar el paso anterior en este elemento.

## Paso final

Finalmente, se eliminaban todos los elementos marcados con “Marchessi: Eliminar”.

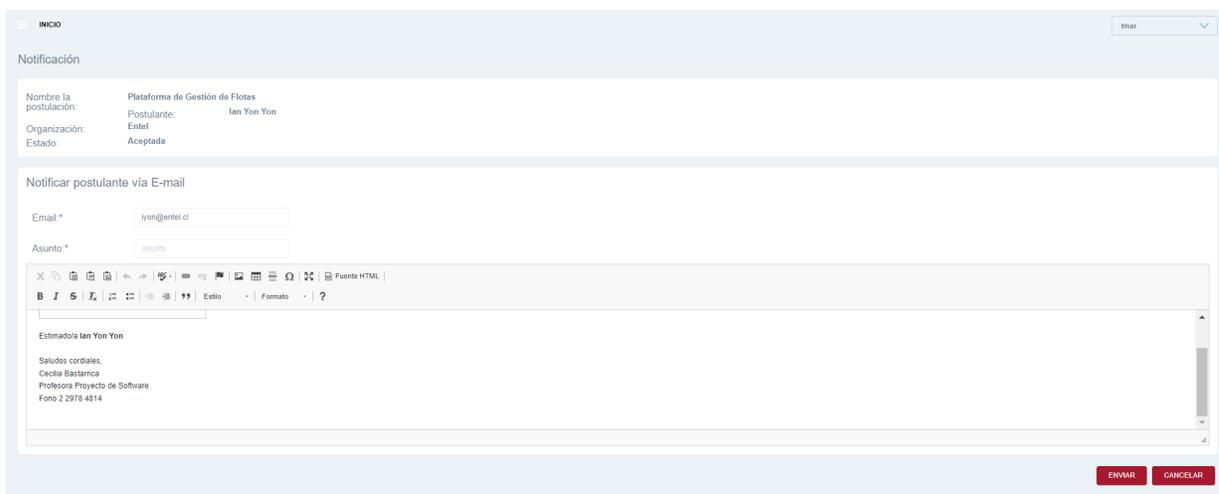
Una vez realizado este proceso en todos los elementos que ya no eran de utilidad en la interfaz, se revisó el sitio para asegurar que las eliminaciones no hubieran tenido un efecto negativo en la interfaz. Al realizar este chequeo, se notó que no se podía acceder a algunas secciones de la interfaz, ya que, surgían errores. Este problema surgió, debido a que algunas tablas de la base de datos trataban de acceder al objeto eliminado “puntaje” y, al no encontrarlo, generaba un error.

Se procedió a eliminar la variable “score” en la base de datos de todos los comandos “CREATE TABLE”, ya que era esta la columna correspondiente al objeto eliminado. Luego de esto se eliminaron las columnas “score” de los proyectos que ya se encontraban en la base

de datos, para así evitar que se creara un error debido a que estas tablas contaban con una columna más de la que deberían.

### 5.4.2. Agregación de funciones

Se procedió a crear una nueva función que permitiera contactar con la empresa sin la necesidad de reutilizar y modificar la funcionalidad de notificar. Para crear esta nueva funcionalidad, se creó una función llamada “send\_mail” en el archivo “admin.py”, en el cual estaban todas las funcionalidades de la interfaz correspondientes a las postulaciones (ver Figura 4.6). Esta función, utilizaba un template llamado “contact\_user.html”. Este archivo cuenta con un saludo predeterminado que se adapta a la organización que se desee contactar y una despedida con los datos del profesor a cargo del curso. El cuerpo del texto estaba vacío para que se pudiera rellenar con el contenido que se deseara (Figura 5.13).



The screenshot shows a web application interface. At the top left, there is a menu icon and the word "INICIO". On the right, there is a user profile dropdown showing "lmar". Below this is a section titled "Notificación" containing a card with the following information: "Nombre la postulación: Plataforma de Gestión de Flotas", "Postulante: Ian Yon Yon", "Organización: Entel", and "Estado: Aceptada". Below the notification card is a section titled "Notificar postulante vía E-mail". It contains an "Email\*" field with the value "lyon@entel.ci" and an "Asunto\*" field with the value "Asunto". Below these fields is a rich text editor with a toolbar and a text area containing the following pre-filled content: "Estimado/a Ian Yon Yon", "Saludos cordiales.", "Cecilia Bastarica", "Profesora Proyecto de Software", and "Fono 2 2978 4814". At the bottom right of the form, there are two buttons: "ENVIAR" and "CANCELAR".

Figura 5.13: Nueva vista para contactar a los clientes

### 5.4.3. Cambios de nombres

Se cambió el texto del botón “Ir” por uno que hiciera más intuitivo su uso (“Ejecutar Acción”) y se modificó el nombre de la vista “Plantilla” por “Convenio”, ambos cambios se realizaron modificando el nombre de las variables que les correspondían.

### 5.4.4. Agregación de variables en proyectos realizados

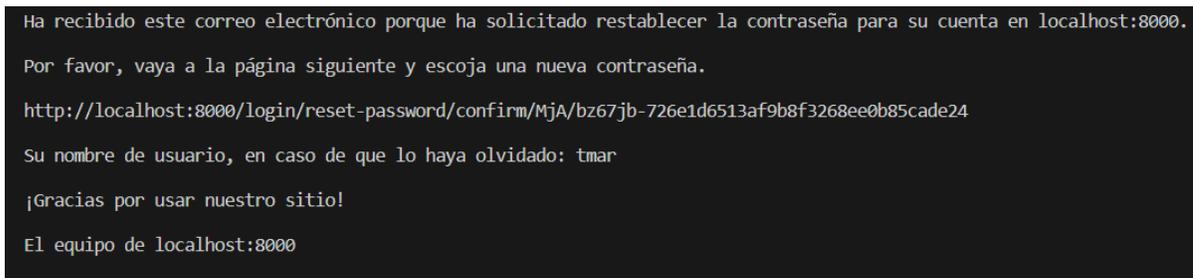
Como se mencionó en el Capítulo 4 de este informe, la agregación de nuevas variables a los proyectos se realizó en conjunto con las variables agregadas a esta vista, ya que es de aquí donde se obtienen los datos se inició por modificar la clase `ApprovedProject`, esta clase se encargaba, en la interfaz del profesor, de las características de los proyectos realizados, a esta clase se le agregaron: el semestre en el que se realizó, el año y los tutores de cada equipo.

Estas tres nuevas variables también debieron ser agregadas en forma de columnas a la tabla de la base de datos correspondiente a los proyectos realizados que aparecen en la página.

### 5.4.5. Mejora de la gestión de identidades y accesos

#### Recuperación de contraseña

Al descubrir que los templates que se utilizarían para añadir esta funcionalidad eran facilitados por Django, se investigó el cómo se utilizaban. Se descubrió que Django no ofrece una opción para recuperar la contraseña, pero sí ofrece una funcionalidad que permite cambiarla mediante un correo electrónico (Figura 5.14) y, además, las vistas y el correo que se utilizan para realizar este proceso se pueden personalizar. Al pertenecer a Django, el proceso de recuperar la contraseña solo funciona si el correo que se utiliza para pedir el cambio corresponde a un usuario inscrito como administrador de Django (o en este caso a la interfaz privada), lo cual aumenta la seguridad del proceso.



```
Ha recibido este correo electrónico porque ha solicitado restablecer la contraseña para su cuenta en localhost:8000.  
Por favor, vaya a la página siguiente y escoja una nueva contraseña.  
http://localhost:8000/login/reset-password/confirm/MjA/bz67jb-726e1d6513af9b8f3268ee0b85cade24  
Su nombre de usuario, en caso de que lo haya olvidado: tmar  
¡Gracias por usar nuestro sitio!  
El equipo de localhost:8000
```

Figura 5.14: Correo que se envía al usuario para restablecer la contraseña

Para utilizar esta funcionalidad, lo primero fue agregar un total de cuatro urls al archivo `urls.py`, cada uno para una vista distinta. Las funciones serían, en el orden de uso de las vistas.

1. Vista para colocar el correo
2. Vista que señalaba que el correo había sido enviado
3. Vista para realizar el cambio de correo
4. Vista que señala que se realizó el cambio

Solo con esto, ya se podía utilizar la recuperación de contraseña, sin embargo, para personalizar los templates y mantener una estética similar a la del sitio del DCC, se agregaron los HTML de la Figura 5.15 a la carpeta `templates` donde se encontraban los demás templates relacionados con el inicio de sesión y se les agregó la línea: `{% block bodyclass%}{% block.super %} login {% endblock %}`, esta línea igual se agregó al template “`login.html`”, ya que de otra forma la transición del inicio de sesión al de recuperación destacaba mucho la diferencia de estética.

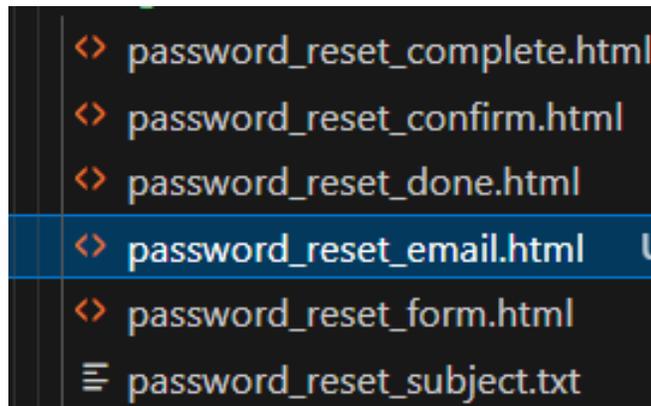


Figura 5.15: Templates para la recuperación de contraseña

### Cierre de sesión automático

Para agregar esta característica, inicialmente se consideró agregar una nueva funcionalidad a la interfaz que consistiera en cerrar la sesión en caso de que el usuario que estuviera ocupando el administrador de Django saliera de este sin cerrar su sesión (por ejemplo, cambiando el URL por cualquiera que no perteneciera al administrador o cerrando la pestaña), sin embargo, se decidió por modificar esta idea por una un poco más flexible, pero de mayor utilidad, esto se debió a dos razones:

- Caso 1: Podía suceder que se dejara abierta la sesión y, sin cambiar de página o apagar el computador, dejarlo abandonado por una cantidad indefinida de tiempo
- Caso 2: Podía suceder que un usuario saliera por accidente, y debería volver a realizar la verificación de usuario, pese a que el tiempo que estuvo fuera del administrador no hubiera sido considerable

Es por esto que se terminó utilizando una de las funcionalidades de Django que permitía solucionar el problema planteado inicialmente y, además, tomaba en cuenta los dos problemas que se mencionaron. Esta nueva característica consiste en establecer un límite de tiempo en el cual la página puede estar sin que se realice ningún tipo de acción y, en caso de realizarse una acción una vez se cumpla el tiempo, se reenviará a la página de inicio de sesión con un mensaje que explique lo que ha sucedido.

Para agregar esta nueva funcionalidad, primero era necesario instalar la dependencia `django-auto-logout` y, a continuación, había que agregar una nueva línea `"django_auto_logout.middleware.auto_logout"` al middleware de Django. Para personalizar el cómo funcionaría este cierre forzoso se agregó una nueva variable que contenía el tiempo en segundos que debía pasar inactivo un usuario para que se cerrara la sesión y el mensaje que mostraría en caso de que este cierre ocurriera.

## 5.5. Estado final

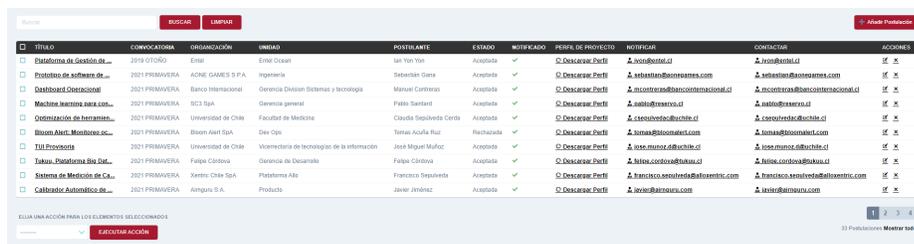
Gracias a la eliminación de vistas que no eran utilizadas en la interfaz, el cambio de nombre realizado, la selección de vistas es ahora más intuitiva de usar y carece de cualquier elemento que no se utilice (Figura 5.16).



Figura 5.16: Todas las vistas que se pueden seleccionar luego de los cambios

### 5.5.1. Vista postulaciones

Debido a la eliminación de la columna puntaje, a la agregación de la columna contactar y al cambio de nombre de la variable ir, ahora se tiene una vista de postulaciones que no posee elementos indeseados y que incluso facilita acciones que antes requerían pasos extra (Figura 5.17).



ID	Título	Convocatoria	Organización	Estado	Postulante	Estado	Notificado	Fecha de Proyecto	Referencia	Contacto	Acciones
<input type="checkbox"/>	Plataforma de Gestión de...	2021-PRIMAVERA	Enel	Cumplido	de Yari Yari	Aceptada	✓	02-Diciembre-2021	➤ yari@enel.cl	➤ yari@enel.cl	✕ ✕
<input type="checkbox"/>	Problemas de software de...	2021-PRIMAVERA	ALCANTARAS S.P.A.	Ingeniería	Sebastián Gana	Aceptada	✓	02-Diciembre-2021	➤ sebastian@alcantaras.com	➤ sebastian@alcantaras.com	✕ ✕
<input type="checkbox"/>	Dashboard Oceanografía	2021-PRIMAVERA	Banco Intercontinental	Gerencia División Sistemas y Tecnología	Manuel Contreras	Aceptada	✓	02-Diciembre-2021	➤ mcontreras@bancointercontinental.cl	➤ mcontreras@bancointercontinental.cl	✕ ✕
<input type="checkbox"/>	Machines seerms para con...	2021-PRIMAVERA	SIC3 SPA	Gerencia general	Pablo Santaró	Aceptada	✓	02-Diciembre-2021	➤ pablo@seerms.cl	➤ pablo@seerms.cl	✕ ✕
<input type="checkbox"/>	Optimización de herramien...	2021-PRIMAVERA	Universidad de Chile	Facultad de Medicina	Claudia Sepúlveda Cerda	Aceptada	✓	02-Diciembre-2021	➤ csepulveda@uchile.cl	➤ csepulveda@uchile.cl	✕ ✕
<input type="checkbox"/>	Bloom Airtel. Mantenimie...	2021-PRIMAVERA	Bloom Airtel SPA	Dev Ops	Tomas Acuña Ruiz	Rechazada	✓	02-Diciembre-2021	➤ tomas@bloomairtel.com	➤ tomas@bloomairtel.com	✕ ✕
<input type="checkbox"/>	TU1 Proveedor	2021-PRIMAVERA	Universidad de Chile	Vicerrectoría de Tecnología de la Información	José Miguel Muñoz	Aceptada	✓	02-Diciembre-2021	➤ jose.muñoz@uchile.cl	➤ jose.muñoz@uchile.cl	✕ ✕
<input type="checkbox"/>	Taxuu. Plataforma Bto Bto...	2021-PRIMAVERA	Falpa Cárrova	Gerencia de Desarrollo	Falpa Cárrova	Aceptada	✓	02-Diciembre-2021	➤ falpa.carrova@taxuu.cl	➤ falpa.carrova@taxuu.cl	✕ ✕
<input type="checkbox"/>	Sistema de Medición de Ca...	2021-PRIMAVERA	Seerms Chile SpA	Plataforma Ato	Francisco Sepúlveda	Aceptada	✓	02-Diciembre-2021	➤ francisco.sepulveda@seermschile.com	➤ francisco.sepulveda@seermschile.com	✕ ✕
<input type="checkbox"/>	Catálogo Automático de...	2021-PRIMAVERA	Ampango S.A.	Producto	Javier Jiménez	Aceptada	✓	02-Diciembre-2021	➤ javier@ampango.com	➤ javier@ampango.com	✕ ✕

Figura 5.17: Nueva vista de postulaciones realizadas

Igualmente, las vistas individuales de las organizaciones ya no poseen elementos que no se utilizan (Figura 5.18).

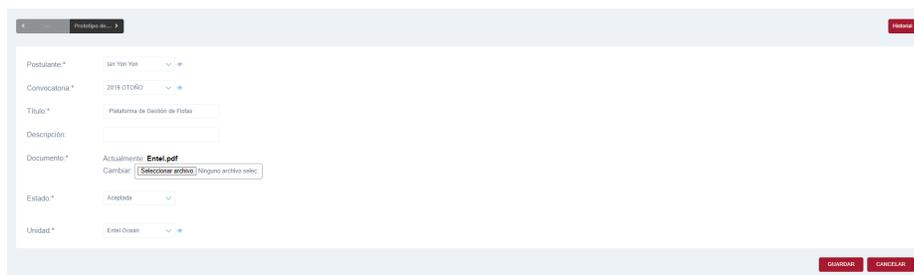


Figura 5.18: Nueva vista de postulaciones individuales

## Vista Proyectos

Gracias a las nuevas variables agregadas, el profesor tendrá más sencillo agregar el mismo tipo de información para cada proyecto, además estas vistas no poseen variables que no se utilizan (Figura 5.19).

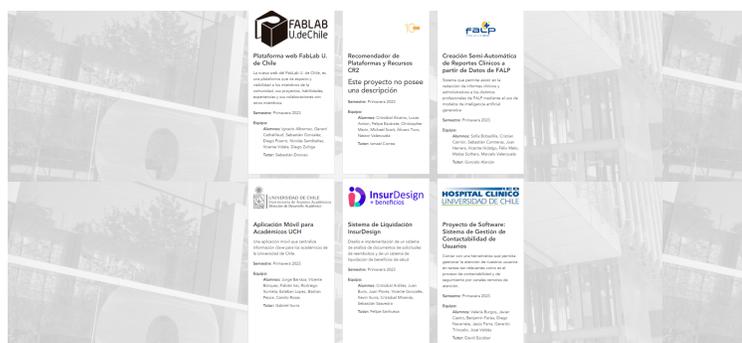


Figura 5.19: Nueva vista de proyectos realizados

### 5.5.2. Gestión de identidad y acceso

#### Recuperación de contraseña

La interfaz del profesor pasó de no poseer un método para recuperar la contraseña a tener uno que funciona en 5 simples pasos.

En primer lugar, se debe presionar el enlace para recuperar la contraseña de la Figura 5.20.



Figura 5.20: Nuevo inicio de sesión

A continuación se debe colocar el correo electrónico del usuario (Figura 5.21), en caso de que el correo este asociado a un usuario ya registrado de la interfaz, se enviará el correo.



Figura 5.21: Primera vista para recuperar la contraseña

Se redirige a una vista que confirma que se envió el correo y permite regresar al inicio de sesión mediante un link (Figura 5.22).



Figura 5.22: Segunda vista para recuperar la contraseña

El correo que llegue llevará un link incluido que al presionarlo redirigirá a la Figura 5.23, en esta se debe colocar la nueva contraseña deseada.



Figura 5.23: Tercera vista para recuperar la contraseña

Finalizado esto, se reenvía a una última vista que informa que el cambio ha sido exitoso (Figura 5.24).



Figura 5.24: Segunda vista para recuperar la contraseña

## Cierre forzoso

El nuevo cierre de sesión automático (Figura 5.25) entrega mayor seguridad de la que se planteó al inicio, al igual que entrega más libertad en caso de que se salga de la interfaz por error.



Figura 5.25: Cierre en caso de inactividad por más de 5 minutos

# Capítulo 6

## Validación

Este capítulo se comentarán las validaciones realizadas a la memoria. Estas validaciones se realizaron considerando que los aspectos no modificados, tales como el proceso para aceptar o rechazar postulantes, no debían ser validadas, ya que al no haber modificado ninguna parte del proceso, la validación realizada por el equipo que creó esta funcionalidad era todavía válida.

### 6.1. Validación portalpsw

La primera validación que se realizó fue replicar el proceso de postulación realizado el semestre de primavera 2023 (ver Figura 6.1), desde la apertura de una nueva convocatoria hasta la publicación de los proyectos aprobados, pero con ciertas diferencias. En primer lugar, no se pidió a las distintas organizaciones que buscaron participar que volvieran a realizar este proceso, sino que sencillamente se reutilizaron los datos de estas, con la única excepción del correo electrónico, el cual fue reemplazado por el correo propio. Esto último se realizó debido a que si se hubieran utilizado los correos con los que originalmente se postuló, las organizaciones recibirían los correos de aceptación (o rechazo dependiendo de la organización), lo cual indudablemente se prestaría para confusiones. Finalmente, el periodo de postulación también fue distinto y de menor duración, ya que en este caso solo se buscaba comprobar el resultado del trabajo realizado.

Igualmente, se validó que la nueva funcionalidad agregada para contactar a los clientes sobre detalles no relacionados con el estado de su postulación, para esto se utilizó la nueva funcionalidad reemplazando el correo de la empresa por el propio con el asunto “Validación” y sin agregar nada a la base que tiene el correo que se puede ver en la Figura 5.13 del Capítulo 5. La validación fue exitosa, ya que se imprimió en la consola el correo con la información tal cual estaba en la interfaz. Es importante mencionar que se imprimió en la consola debido a que para que el correo se envíe el sitio debe estar en producción y la validación se realizó en el sitio levantado localmente. Lo mismo ocurrió con las notificaciones de aceptación o rechazo en la validación del proceso de postulación, el cambio de correo era una medida de seguridad preventiva.

TÍTULO	CONVOCATORIA	ORGANIZACIÓN	UNIDAD	POSTULANTE	ESTADO	NOTIFICADO
<input type="checkbox"/> Proyecto de Software: Sis...	2023 PRIMAVERA	Universidad de Chile	Hospital Clínico de la Universidad de Chile Dr. José Joaquín Aguirre	Marcos Batarce	Aceptada	✓
<input type="checkbox"/> Aplicación Móvil para Aca...	2023 PRIMAVERA	Universidad de Chile	Dirección de Desarrollo Académico	Claudio Olea Azar	Aceptada	✓
<input type="checkbox"/> Automatización de proceso...	2023 PRIMAVERA	Finemp SpA	Founder	Montserrat Orellana	Creada	✗
<input type="checkbox"/> Sistema de Liquidación In...	2023 PRIMAVERA	InsurDesign SpA	gerencia	Gabriel Ditrans	Aceptada	✓
<input type="checkbox"/> Recomendador de Plataform...	2023 PRIMAVERA	Universidad de Chile	Desarrollo de software	Francisca Muñoz	Aceptada	✓
<input type="checkbox"/> Creación Semi-Automática ...	2023 PRIMAVERA	Fundación Arturo López Pérez	Informática Médica y Data Science	Sergio Peñafiel	Aceptada	✓
<input type="checkbox"/> MVP de software de videoj...	2023 PRIMAVERA	AONE GAMES S.P.A.	Ingeniería	Sebastián Gana	Aceptado	✓
<input type="checkbox"/> Plataforma comunidad y pr...	2023 PRIMAVERA	Universidad de Chile	FabLab	Danisa Peric	Aceptada	✓
<input type="checkbox"/> Plataforma de control de ...	2023 PRIMAVERA	Vortex spa	Servicios submarinos	Luis Martínez	Aceptada	✓
<input type="checkbox"/> Sistema para el acompañam...	2023 PRIMAVERA	Universidad Metropolitana de Ciencias de la Educación	Departamento de Informática	Francisco Segovia	Aceptada	✓
<input type="checkbox"/> SmartCam	2023 PRIMAVERA	Lookdata SpA	LookData	Gonzalo Teperman	Aceptada	✓

Figura 6.1: Proyectos que postularon en la convocatoria Primavera 2023

## 6.2. Validación recuperación de contraseña

En primer lugar, se instó a la profesora a cargo del curso a realizar este proceso mediante su correo institucional, ya que es el que se encuentra inscrito en su cuenta de la interfaz privada. Todos los pasos resultaron intuitivos y se logró realizar el cambio de contraseña sin dificultad alguna. A continuación se intentó realizar el mismo proceso con un correo que no pertenecía a ningún usuario del portal, esto con la intención de comprobar si un usuario externo podría de alguna forma obtener una contraseña para ingresar al sitio. El proceso se pudo realizar sin problemas, al igual que en el primer caso, sin embargo, al no pertenecer el correo a un usuario, Django no envió ningún correo para realizar el cambio, lo cual confirmó la seguridad de este método.

## 6.3. Validación cierre automático de sesión

En este caso era necesario realizar 3 validaciones junto a la profesora a cargo del curso:

- Validar que se pudiera reingresar a la cuenta de la interfaz privada sin la necesidad de nombre de usuario y contraseña en caso de una salida accidental
- Validar que, en caso de salir por más del tiempo acordado, no se pudiera volver a ingresar sin el nombre de usuario y contraseña
- Validar que, en caso de no salir del sitio, pero tampoco realizar ninguna acción en la interfaz, este se cerraría una vez transcurrido el tiempo acordado

Para el primer caso, se revisaron dos situaciones. La primera situación fue cerrar la ventana en la cual se estaba utilizando la interfaz privada para luego abrir una nueva ventana y reingresar a la interfaz. La segunda situación fue ingresar a la interfaz para, a continuación, cambiar de sitio web utilizando un nuevo url y luego volver a la interfaz para comprobar que todavía se podía ingresar. En ambos casos el resultado fue exitoso, logrando reingresar a la cuenta sin la necesidad de volver a utilizar el nombre de usuario o contraseña.

La segunda validación se realizó de la misma manera que la primera, con la excepción de que en este caso se esperó más del tiempo acordado para intentar reingresar a la interfaz. Además, se realizó un tercer caso nuevo en el cual se reinicia el computador que se estaba utilizando para luego buscar abrir la interfaz una vez transcurrido el tiempo necesario. Al igual que en la primera validación, se obtuvieron los escenarios esperados, no se pudo reingresar a la cuenta sin utilizar el nombre de usuario y contraseña.

Finalmente, la validación del último caso se realizó de tres maneras. Primero se ingresó a la interfaz privada y luego se abrió una segunda ventana en la cual se estuvo realizando distintas acciones hasta que se cumplió el tiempo acordado, a continuación se volvió a la ventana donde se encontraba la interfaz y se intentó realizar alguna acción. La segunda prueba fue quedarse en la ventana de la interfaz, pero sin realizar ningún tipo de acción por el tiempo estipulado para, una vez cumplido el tiempo, intentar utilizar la interfaz. En ambos casos la sesión se cerró automáticamente mostrando un mensaje que señalaba que se había excedido el tiempo. La última validación consistió en ingresar a la interfaz en dos pestañas diferentes con el mismo usuario, pero solo utilizar una de estas durante el tiempo estipulado, una vez transcurrido el tiempo se utilizó la interfaz de la otra pestaña. En este caso sí se pudo utilizar, lo que implica que el tiempo para que se cierre la sesión está ligado a la sesión del usuario y no a la pestaña que se está utilizando.

# Capítulo 7

## Conclusión

En este capítulo se evaluarán los resultados del trabajo realizado en esta memoria y también se discutirán los cambios que no se integraron en esta.

### 7.1. Cumplimiento de objetivo

#### 7.1.1. Mejorar la mantenibilidad del sitio

Debido a las actualizaciones realizadas en el software, se logró aumentar la mantenibilidad del sitio, pasando de una situación donde el framework se encontraba en proceso de deprecación, a una en la cual se cuenta con la última versión, no solo de Django, sino también de todas las dependencias.

Además, se logró ordenar en cierta medida el código, gracias a la eliminación comentada en el Capítulo 3 y Capítulo 5 de este informe, sin embargo, debido a lo extenso de la tarea, específicamente la revisión de todos los archivos del repositorio, no se logró realizar de forma completa este punto.

#### 7.1.2. Mejorar la usabilidad del sitio

Los cambios realizados en ambas vistas referentes a la información sobre el curso en la interfaz de los clientes, permite que los datos que se desean presentar sean entregados sin errores y de forma clara. Por otro lado, se realizaron cambios que permiten reducir los casos problemáticos con respecto al documento de postulación que se mencionaron en este informe.

Por otro lado, los elementos eliminados de la interfaz y base de datos con respecto a los elementos y funcionalidades no utilizadas, así como también las variables y funcionalidades agregadas a la interfaz del profesor, facilitarán al profesor el uso del sitio tanto en el proceso

de postulación como al final del curso.

### **7.1.3. Abordar las deficiencias en la gestión de identidad y acceso de la interfaz**

En el caso de la recuperación de la contraseña, gracias al uso de las plantillas que Django ofrece, se terminó implementando una funcionalidad que aumentó la usabilidad del sitio, mucho más de lo que habría hecho la solución planteada al inicio. Se obtuvo una característica que logra la recuperación, pero también se asegura de que el correo esté ligado al administrador.

Por otro lado, se consiguió agregar un cierre forzoso a la aplicación, pese a que esto se logró utilizando una solución diferente a la planteada inicialmente, la solución implementada es más útil, ya que es capaz de cerrar la interfaz en caso de inactividad(algo que no se planteó inicialmente) y también da flexibilidad en caso de que se salga por error de la cuenta, dando un tiempo límite para volver a conectarse sin volver a ingresar los datos.

En conclusión, se consiguió completar la mayoría de los puntos que fueron establecidos al inicio de esta memoria, lo cual llevó a obtener una página que cuenta con dependencias en sus últimas versiones y un soporte robusto. Igualmente, se logró que este sitio promoció el curso de mejor manera y sin presentar errores que puedan ser razón de desconfianza en los posibles postulantes. De igual manera, se entregó a la profesora a cargo del curso una interfaz con mayor usabilidad. Por lo tanto, utilizando el concepto establecido en la sección 2.1 del Marco Teórico, se puede afirmar que la reingeniería del portal de Proyecto de Software fue exitosa.

## **7.2. Trabajo a futuro**

Antes de iniciar esta memoria, fueron comentados los distintos objetivos que se podían realizar durante el semestre, la mayoría de este informe se ha centrado en los cuatro puntos que fueron decididos y fue mientras se desarrollaron en estos puntos que se descubrieron nuevos objetivos en los que se podía trabajar. Por lo tanto, esta sección se utilizará para comentar las distintas acciones que se podrían tomar a futuro sobre el sitio para así seguir incrementando la calidad y utilidad de esta.

### **7.2.1. Portal de pagos**

Como se mencionó en la introducción de este documento, en 2021 un equipo conformado por alumnos del DCC trabajó en el mismo código, en esa ocasión fue implementado un portal de pagos (Figura 7.1), el cual, a día de hoy, cuenta con la arquitectura necesaria para empezar a utilizarse, sin embargo, todavía cuenta con errores que deben solucionarse. Por ejemplo,

algunos usuarios (como las secretarías del departamento) tienen acceso a vistas que no deberían. Es por esto que se debe realizar una depuración profunda al código correspondiente a esta funcionalidad antes de que se implemente de manera pública.



Proyectos	Convocatoria	Organización	Última Modificación	Estado
Creación Semi-Automática de Reportes Clínicos a partir de Datos de FALP	2023 PRIMAVERA	Fundación Arturo López Pérez	2023/08/10 16:44	Solicitar Documentos
MVP de software de videojuego de género de ritmo para Nintendo Switch.	2023 PRIMAVERA	ACNE GAMES S.P.A.	2023/08/10 16:44	Solicitar Documentos
Plataforma comunidad y proyectos FatLab U. de Chile	2023 PRIMAVERA	Universidad de Chile	2023/08/10 16:45	Solicitar Documentos
Plataforma de control de operaciones subacuáticas	2023 PRIMAVERA	Vortex spa	2023/08/10 16:45	Solicitar Documentos
Sistema de Liquidación InsurDesign	2023 PRIMAVERA	InsurDesign SpA	2023/08/10 16:49	Solicitar Documentos
Sistema para el acompañamiento a la Trayectoria Formativa	2023 PRIMAVERA	Universidad Metropolitana de Ciencias de la Educación	2023/08/10 16:45	Solicitar Documentos
SmartKam	2023 PRIMAVERA	Lookdata SpA	2023/08/10 16:46	Solicitar Documentos

Figura 7.1: Vista del portal de pago

### 7.2.2. Orden del código

Como se mencionó anteriormente en este documento, la limpieza en el código no se llevó al cabo completamente debido al tiempo que esta acción conllevaría. Se considera que, en caso de combinar esta acción con una reordenación de las carpetas y una documentación de las distintas funciones, se conseguiría un repositorio en el que sería mucho más sencillo trabajar en el futuro.

### 7.2.3. Formulario online

Una vez determinado que una revisión completa de todos los documentos enviados al momento de postular no era factible, se consideró que una solución al problema sería implementar un formulario online, de esta forma sería imposible que se enviaran documentos PDF erróneos.

### 7.2.4. Recuperación de nombre de usuario

Una vez establecida la recuperación de contraseña, fue inevitable notar que de todas formas quedaba una opción de no poder acceder al administrador de Django. Para este problema existen varias soluciones tales como:

- Agregar una función para recuperar el nombre de usuario con el correo
- Eliminar el uso de nombre de usuario y utilizar solo el correo institucional
- Conectar el administrador de Django con la cuenta de usuario de uCampus del profesor a cargo

# Bibliografía

- [1] <https://malla.cadcc.cl/>.
- [2] Django JET. <http://jet.geex-arts.com/>.
- [3] Great Learning. <https://www.mygreatlearning.com/django/tutorials/history-of-django>, 2019.
- [4] IBM. <https://www.ibm.com/es-es/topics/identity-access-management>.
- [5] IBM. <https://www.ibm.com/mx-es/topics/docker>.
- [6] IBM. <https://www.ibm.com/mx-es/topics/postgresql>.
- [7] Imagar. El proceso de reingeniería de software. 2023.
- [8] ISO. <https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad>.
- [9] J Nielsen. Usability engineering. *MA: Academic Press.*, 1993.
- [10] Pressman R.S. Ingeniería del software: un enfoque práctico. 2005.
- [11] S. Crouch. Developing maintainable software, 2013.
- [12] Stack Overflow. <https://survey.stackoverflow.co/2023/>, 2023.
- [13] Thales. Los cuatro tipos de mantenimiento de software.
- [14] Visual Studio Code. <https://code.visualstudio.com/docs>.
- [15] Lawrie Brown William Stalling. Seguridad web: Principios y aplicaciones. *Pearson Educacion*, 2017.