



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

SISTEMA DE APOYO PARA LA MESA DE GÉNERO DCC

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN COMPUTACIÓN

ANDREA JI YE PARK KWAK

PROFESORA GUÍA:
JOCELYN SIMMONDS WAGEMANN

MIEMBROS DE LA COMISIÓN:
ALEJANDRO HEVIA ANGULO
JUAN ARRIAGADA CANCINO

SANTIAGO DE CHILE
2023

Resumen

Genera Igualdad es un proyecto de certificación en estándares de igualdad de género que busca instalar estándares de género para instalar una cultura de igualdad de género en distintas áreas y quehaceres a nivel universitario.

Frente a esta situación, en el año 2021 surge la Mesa de Género del DCC, conformada por integrantes de distintos estamentos dentro del departamento que tenían como primera misión, entender la situación de brecha de género presente en el Departamento de Ciencias de la Computación.

Actualmente, el departamento cuenta con datos recopilados desde distintas fuentes en forma de hoja de cálculo almacenadas en un Drive de acceso restringido. A partir de estos datos, mediante un proceso manual, generaban gráficos que luego eran utilizados como material de apoyo durante las reuniones de la comisión, para el entendimiento del estado en el cual se encontraba el DCC. Entendiendo que los datos deben ser actualizados de forma constante y, que la generación de este material gráfico resultaba ser un proceso engorroso y un sistema difícil de mantener, el problema abordado en esta memoria consistió en crear una herramienta propia del departamento que sirviera como apoyo a la Mesa de Género en cuanto a la generación de gráficos con el objetivo de entender la situación en cuanto a la brecha de género.

La solución propuesta consistió en implementar una plataforma web, capaz de permitir a usuarios cargar archivos y a partir de ellos, calcular métricas e indicadores que pudiesen ser visualizados mediante gráficos.

Comenzando con el análisis de datos almacenados en el Drive, se armó un listado de métricas e indicadores por estamento. Se desarrolló una herramienta con las funcionalidades de carga de datos, cálculo de métricas e indicadores y generación de gráficos que permitieran a los integrantes de la Mesa de Género, visualizar datos del departamento relacionados a género.

Al final del proceso y luego de validaciones con usuarios, se obtuvo una herramienta funcional que cumplía con el objetivo principal de la memoria; una plataforma que apoyara a la Mesa de Género en entender la situación de brecha de género dentro del departamento.

Para ti tía, que siempre estuviste y estarás.

Agradecimientos

A mis papás que siempre dieron todo por mi.

A Lucy, que gracias a su apoyo y compañía, pude llenarme de ánimos en los momentos más difíciles.

A mis infaltables, Camila y Andrés, por todos los momentos, risas. Por su genuina amistad y apoyo incondicional.

A la profesora Jocelyn, por su apoyo, comprensión y buena disposición. Sin ella, nada de esto hubiese sido posible.

A cada uno que fue parte de mi vida universitaria y que gracias a ellos, hizo de esta etapa mucho más armoniosa.

A mi grupo internacional, mención especial a Karen, Karina, Priscila, por sus ánimos, buenos momentos, risas interminables, por su apoyo y presencia, por las grandes alegrías que nunca olvidaré.

A las amistades que logré formar durante este último tiempo, por las madrugadas de risas, por los momentos en los que supieron escucharme, por las rabias que pasamos juntos durante el trabajo.

Por último, para mi tía, por haberme convertido en la persona que soy ahora, por su sabiduría infinita y por su recuerdo que nunca desaparecerá. A ella, desde donde esté descansando: ¡Lo logré!

Tabla de Contenido

1. Introducción	1
1.1. Contexto y oportunidad	1
1.2. Objetivos	2
1.3. Descripción de la solución	2
1.4. Metodología	3
1.5. Validación y resultados	3
1.6. Estructura del documento	4
2. Estado del arte	5
2.1. Trabajos previos relacionados y fuentes alternativas de datos	5
2.1.1. Sistema de alimentación de información docente para el DCC	6
2.1.2. Plataforma de monitoreo de la docencia de pregrado del DCC	7
2.1.3. Implementación de una herramienta de apoyo a la toma de decisiones en instituciones de educación superior en base a la información curricular de Ucampus	8
2.1.4. Ucampus	9
2.2. Herramientas y/o plataformas para visualización de datos	9
2.3. Situación actual	11
2.3.1. Propuesta inicial de métricas e indicadores	12
3. Análisis y Diseño	15
3.1. Análisis	15
3.1.1. Tipos de usuario y requisitos	15

3.2.	Diseño	16
3.2.1.	Arquitectura General	16
3.2.2.	Carga de Datos	18
3.2.3.	Cálculo de Métricas e Indicadores	19
3.2.4.	Visualización de Datos	19
3.2.5.	Diagrama de clases	21
3.2.6.	Interfaces e interacciones en la plataforma	22
4.	Implementación	23
4.1.	Primera Etapa	23
4.1.1.	Configuraciones iniciales y levantamiento del proyecto	23
4.1.2.	Implementación de vista principal	24
4.2.	Segunda Etapa	26
4.2.1.	Creación de modelos	26
4.2.2.	Implementación de carga de datos	27
4.3.	Tercera Etapa	33
4.3.1.	Cálculo de métricas utilizando pandas	33
4.3.2.	Generación de gráficos	35
4.4.	Resumen	44
5.	Validación	46
5.1.	Pruebas funcionales	46
5.2.	Encuesta de Usabilidad y Satisfacción de Usuarie	47
6.	Conclusión	50
	Bibliografía	53

Índice de Tablas

5.1. Tabla comparativa con el total de miembros calculado en Excel versus herramienta	46
5.2. Tabla comparativa con la proporción de género femenino calculado en Excel y herramienta	47
5.3. Resultados de la encuesta por parte de los usuarios	48

Índice de Ilustraciones

1.1. Flujo de la solución propuesta	3
2.1. Módulo de Indicadores de Autoevaluación Ucampus [1]	6
2.2. Plataforma Docencia DCC, dashboard por género [2]	7
2.3. Dashboard de ingresos a la Universidad de Chile por género [3]	8
2.4. Dashboard generado en Tableau	10
2.5. Dashboard generado en Power BI.	10
3.1. Diagrama del funcionamiento de la plataforma	17
3.2. Diagrama del funcionamiento entre los tres módulos	17
3.3. Ejemplos de gráficos disponibles en Plotly	20
3.4. Diagrama UML de clases	21
3.5. Vista propuesta de la plataforma a partir de otros sitios del DCC	22
4.1. Organización de archivos relacionado a diseño dentro del proyecto	24
4.2. Primera iteración de la vista principal de la plataforma	25
4.3. Vista de módulo de carga de datos implementado con formularios	28
4.4. Vista de administración del modelo Academique con botón de importar disponible	30
4.5. Vista de verificación de archivo al modelo Academique previa a la importación	31
4.6. Vista de importación exitosa de archivo al modelo Academique	31
4.7. Vista de error al importar un archivo que no cumple con el modelo	32
4.8. Parte del archivo .csv utilizado como dato de prueba para el cálculo de métricas	33

4.9. Primera iteración de cálculo de métricas	35
4.10. Gráfico generado utilizando Plotly y métricas calculadas relacionado a alumnos por ramo	37
4.11. Gráfico de línea con sus respectivos filtros para la vista Comunidad	39
4.12. Gráfico torta que muestra cantidad de integrantes por género y por año	40
4.13. Gráfico de indicadores de proporción de género femenino	42
4.14. Gráfico de barra agrupado por género según tipo de docente	43
4.15. Panel de gráficos para sección Comunidad	44

Capítulo 1

Introducción

1.1. Contexto y oportunidad

La brecha de género [4] entendida como la distancia que hace falta para alcanzar la igualdad entre géneros respecto a un determinado fenómeno, se ha convertido en un problema a abordar en temas relacionados a derechos humanos, oportunidades laborales, educación, entre otros. Enfocado en la educación superior, no es raro notar que existe una mayor representación de cierto género en carreras o áreas de estudio particulares; ahondando un poco más en este punto, se registra que las estudiantes femeninas (universitarias) superan en número a los estudiantes varones a nivel mundial en carreras como medicina, humanidades y ciencias sociales, sin embargo, su representación en áreas como ciencia, tecnología, ingeniería y matemáticas es mucho menor en comparación a los estudiantes de género masculino [5].

Frente a esta situación, universidades y facultades han implementado diferentes medidas para acortar la brecha de género [6], sobretodo en áreas de estudio destinadas a ciencias, tecnologías, ingeniería y matemáticas. Parte fundamental del proceso es la recolección de datos con el fin de entender la situación actual y así decidir de manera adecuada los pasos a seguir para disminuir la desigualdad de género entre estudiantes.

En el año 2021, el departamento del DCC dio inicio al funcionamiento de la Mesa de Género del DCC. Con el fin de entender la situación dentro del departamento relacionado a la brecha de género, recopilamos datos desde distintas fuentes, incluyendo encuestas hacia la comunidad. Estos datos fueron almacenados en hojas de cálculo en un Drive con acceso restringido, para luego ser utilizados en la creación manual de distintos gráficos que ayudaron a entender el estado al cual se enfrentaba el DCC.

La generación de gráficos y la actualización de datos es un proceso engorroso dado que se realiza de forma manual. Esto consume tiempo y lo hace un proceso ineficiente e inmantenible, puesto que los datos se van actualizando de forma periódica.

A partir de lo anterior, se busca desarrollar un sistema propio del departamento, que permita ingresar datos en algún(os) formato(s) definido(s) (por ejemplo `.csv`), procesarlos y generar métricas e indicadores acompañados de tablas y/o gráficos según sea el caso. El

objetivo de este sistema es ser una herramienta de apoyo para la Mesa de Género del DCC, automatizando la generación de tablas, gráficos e información y facilitando la actualización de datos de forma periódica. Además, el carácter de “sistema propio” garantiza la protección de datos sensibles de cada persona perteneciente al DCC.

1.2. Objetivos

Este proyecto tiene como meta diseñar e implementar un sistema que sea capaz de generar y graficar métricas e indicadores desde la perspectiva de género, a partir de la carga de datos para ser utilizado como herramienta de apoyo a la Mesa de Género del DCC. Para ello, se tomarán los distintos datos recolectados y se analizarán para poder definir métricas e indicadores que aporten a entender la situación del departamento en cuanto a perspectiva de género y así, tomar decisiones para acortar la brecha, tomando como referencia los indicadores desarrollados en el trabajo de memoria mencionado anteriormente.

Con el fin de cumplir el propósito, se plantearon los siguientes objetivos específicos:

1. Analizar datos recopilados de años anteriores y en base a ellos, determinar un listado de métricas e indicadores posibles desde la perspectiva de género.
2. Diseñar vistas del sistema para la presentación de métricas, indicadores y gráficos.
3. Implementar funcionalidades de carga de datos, cálculo de métricas e indicadores y visualización de datos (gráficos) para el sistema.
4. Validar usabilidad y experiencia de usuario del sistema desarrollado mediante test de usabilidad a usuarios finales.

1.3. Descripción de la solución

Para abarcar los objetivos planteados en la sección 1.2, se desarrolló una plataforma web empleando Django¹ destinada a los integrantes de la Mesa de Género del DCC, que permite la carga de datos, cálculo de métricas e indicadores y visualización de datos enfocadas en género. Cada una de estas funcionalidades fueron programadas de forma independiente utilizando Python como lenguaje de programación. Para la subida de archivos, se recurrió a una librería específica detallada en la sección 3.2.2, la cual fue integrada en la vista de administración generada en las configuraciones iniciales de Django. Con esto, se aseguró que la responsabilidad recaerá en roles específicos y se evita la subida errónea o inconsistente de datos. En cuanto a el cálculo de métricas e indicadores y la visualización de estas, fueron trabajadas en una misma aplicación de Django.

En la Figura 1.1 se muestra de forma generalizada el modo en que operan las distintas funcionalidades entre sí. En resumen, los datos subidos a la plataforma son tomados por el

¹<https://docs.djangoproject.com/en/4.2/intro/overview/>

módulo de cálculo para generar métricas e indicadores y estos mismos son plasmados a través de gráficos presentándolos en una vista hacia les usuaries.

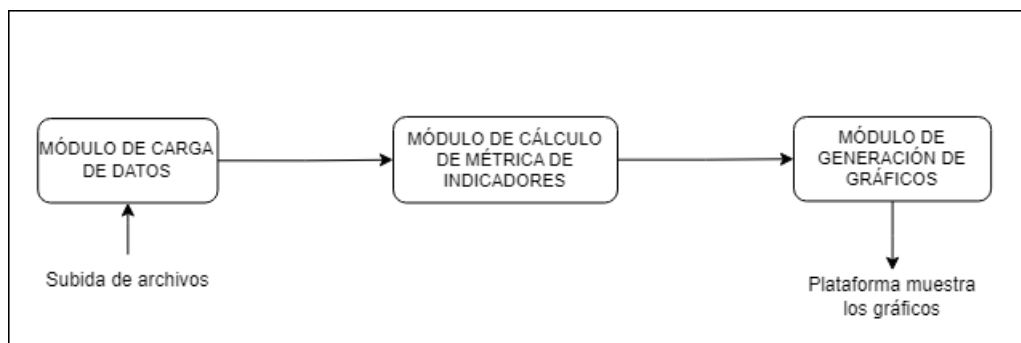


Figura 1.1: Flujo de la solución propuesta

1.4. Metodología

Sobre la metodología de trabajo utilizada, se centró en el alcance por objetivos divididos en ciclos de dos semanas. Los avances eran presentados y revisados en reuniones con la profesora guía, lo que se tradujo en un desarrollo incremental.

Dado que cada una de las principales funcionalidades (carga de datos, cálculo de métricas e indicadores y visualización de gráficos) dependían entre sí, el proceso de implementación se dividió en tres etapas donde se trabajó en cada una de ellas, considerando además la conexión y las pruebas para el correcto funcionamiento de la plataforma al término de cada objetivo alcanzado.

1.5. Validación y resultados

La validación de la plataforma se centró en evaluar utilidad de la información y usabilidad. Para ello, se creó un cuestionario con preguntas relacionadas a estos dos enfoques, que fueron respondidas luego de observar videos sobre el funcionamiento de la plataforma. Esta encuesta fue enviada a integrantes que conformaban y conforman la Mesa de Género del DCC, que serían les usuaries finales de la herramienta.

Además de las validaciones con usuaries, se realizaron pruebas funcionales que aseguraran el correcto funcionamiento de la plataforma. Dichas pruebas fueron efectuadas durante el avance del proyecto y en cada una de las etapas de implementación de funcionalidades e integración entre estas.

Tomando en cuenta los objetivos y los resultados de la encuesta, este proyecto entregó una herramienta funcional que cumple con el objetivo principal de la memoria. Los comentarios recopilados de la encuesta, se centran principalmente en mejoras hacia la experiencia de usuarie.

1.6. Estructura del documento

Luego de la Introducción, se sigue con el capítulo Estado de arte, donde se exponen trabajos previos relacionados a esta memoria, herramientas alternativas de visualización de datos y la situación actual, que abarca contexto sobre qué es la Mesa de Género del DCC, cuál es su objetivo y qué datos se tienen a partir de su trabajo. El tercer capítulo Análisis y Diseño, toma la problemática y a partir de ella explica el diseño de la solución a implementar.

El cuarto capítulo Implementación, presenta el desarrollo detallado de la solución, problemas encontrados y las soluciones implementadas para estos casos. En el capítulo de Validación se analizan las respuestas de los usuarios a la encuesta, junto con los resultados obtenidos de las pruebas funcionales realizadas a la herramienta.

Finalmente, en el capítulo de Conclusión, se profundiza sobre el recuento de los objetivos logrados, aprendizajes obtenidos y trabajo futuro.

Capítulo 2

Estado del arte

En este capítulo se examinará y comprenderá el nivel actual de conocimiento, avances y enfoques existentes en relación con la integración de datos, cálculo de métricas e indicadores y visualización de información en un contexto de género.

Se explicarán diversas memorias y soluciones desarrolladas en años anteriores relacionadas con el tema de esta memoria, identificando funcionalidades, alcances y las limitaciones respecto al enfoque de género.

Posteriormente, se describen herramientas alternativas para la visualización de datos y la razón por la cual no fueron escogidas para formar parte de la solución a desarrollar.

Finalmente, se contextualiza sobre la situación previa al desarrollo de esta memoria; cómo nace la Mesa de Género, su objetivo, datos relevantes recopilados relacionados a la brecha de género y un listado propuesto de métricas e indicadores.

2.1. Trabajos previos relacionados y fuentes alternativas de datos

Existen memorias que se relacionan con el tema propuesto en este informe, sin embargo, cada una de estas abarca una funcionalidad por separado cuando lo que se busca es una solución que una tanto la carga de datos, el cálculo de métricas e indicadores y la visualización de estos. En las secciones 2.1.1, 2.1.2, 2.1.3, se presentan y explican, a grandes rasgos, algunos trabajos realizados en años anteriores, junto con la plataforma que actualmente utiliza la facultad, que para el contexto de este informe, puede ser una alternativa como fuente de datos (sección 2.1.4).

2.1.1. Sistema de alimentación de información docente para el DCC

El trabajo realizado por el alumno de ingeniería civil en computación de la Universidad de Chile, Nicolás Caracci [1], consistía en la creación de base de datos de docencia del DCC que se alimenta de forma automática desde Ucampus, pero es independiente de esta. Esta solución permite que actividades de monitoreo, planificación y toma de decisiones sobre docencia del departamento no sea limitada por la gestión que podría conllevar levantar una solicitud a Ucampus. Se hizo un trabajo de cálculo de indicadores y estadísticas de docencia para ilustrar la utilidad de la información almacenada en esta base de datos, pero estos cálculos no quedaron registrados dentro de ella. Algunos de los indicadores calculados: demanda de docencia en cursos de especialidad, evolución de la demanda de cupos en cursos de computación, evolución de la cantidad de alumnos que toman cursos de computación no tenían relación con el género.

Cabe destacar que en esta memoria se menciona la existencia de un módulo dentro de Ucampus llamado *Indicadores de Autoevaluación* que muestra distintos indicadores asociados a la docencia, por ejemplo, el número de alumnos que ingresa a la carrera separado por género, sin embargo, actualmente (2022), este módulo no se encuentra disponible de forma pública, el acceso está limitado a roles específicos. En la siguiente imagen (Figura 2.1), se puede ver la interfaz de dicho módulo, específicamente se muestran métricas relacionadas a la cantidad de estudiantes por género que ingresaban a Ingeniería Civil en Computación entre los años 2008 y 2020:

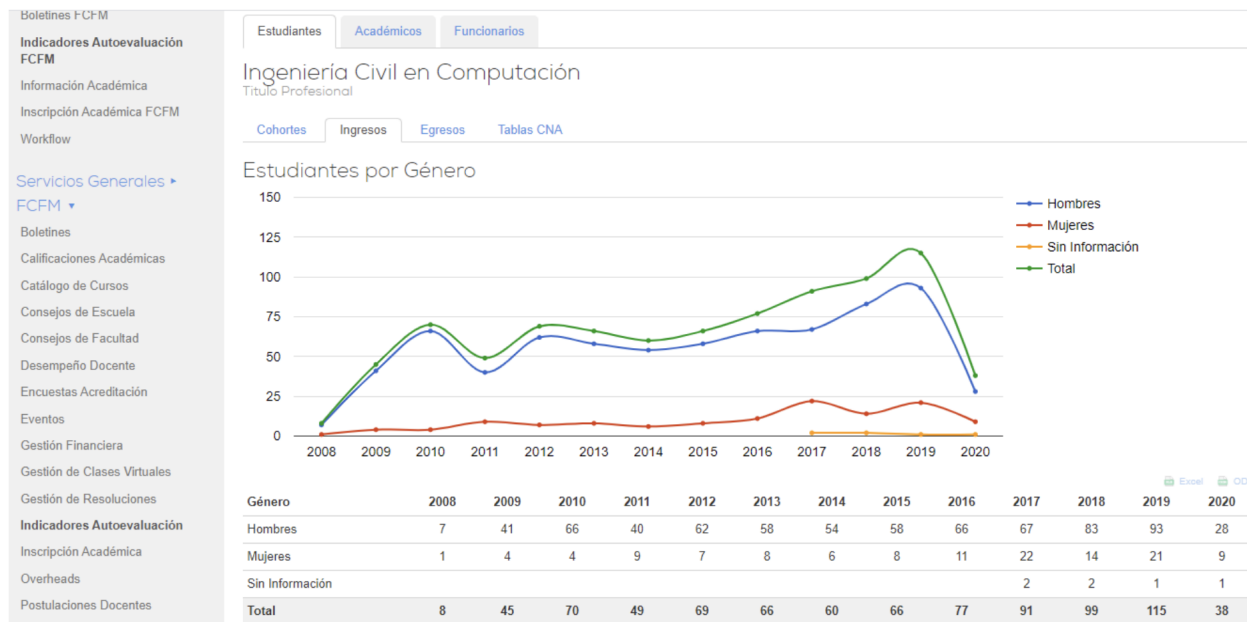


Figura 2.1: Módulo de Indicadores de Autoevaluación Ucampus [1]

2.1.2. Plataforma de monitoreo de la docencia de pregrado del DCC

Esta memoria en particular realizada por Ignacio Vallejos [2], en ese momento estudiante de ingeniería civil en computación de la Universidad de Chile, se concentró en el diseño e implementación de un sistema que calculaba diversos indicadores a partir del trabajo realizado en la memoria *Sistema de alimentación de información docente para el DCC* (antes descrita en la sección 2.1.1) que pueden ser visualizados a través de dashboards, con el fin de facilitar el acceso a jefes docentes y funcionarios del departamento. Dentro de esta plataforma, permitía además filtrar datos con criterios predefinidos, como por ejemplo la cantidad de alumnos entre los años 2015 y 2020, la tasa de aprobación y promedio de notas de aprobación por curso.

En la Figura 2.2 se muestra la plataforma implementada donde se aprecian indicadores relacionados al género, como por ejemplo, cantidad anual/semestral de ingresos/egresos a la carrera. En estricto rigor, al ser una herramienta de apoyo a la docencia, estuvo solo enfocada a estudiantes y no en otros estamentos como funcionarios o docentes. El acceso a la plataforma estaba configurado solo para aquellos que tienen un rol de Jefe docente, por lo que integrantes de la mesa de género que no cumplan con este rol (como estudiantes, por ejemplo) no pueden acceder a dicha información presentada en los dashboards.

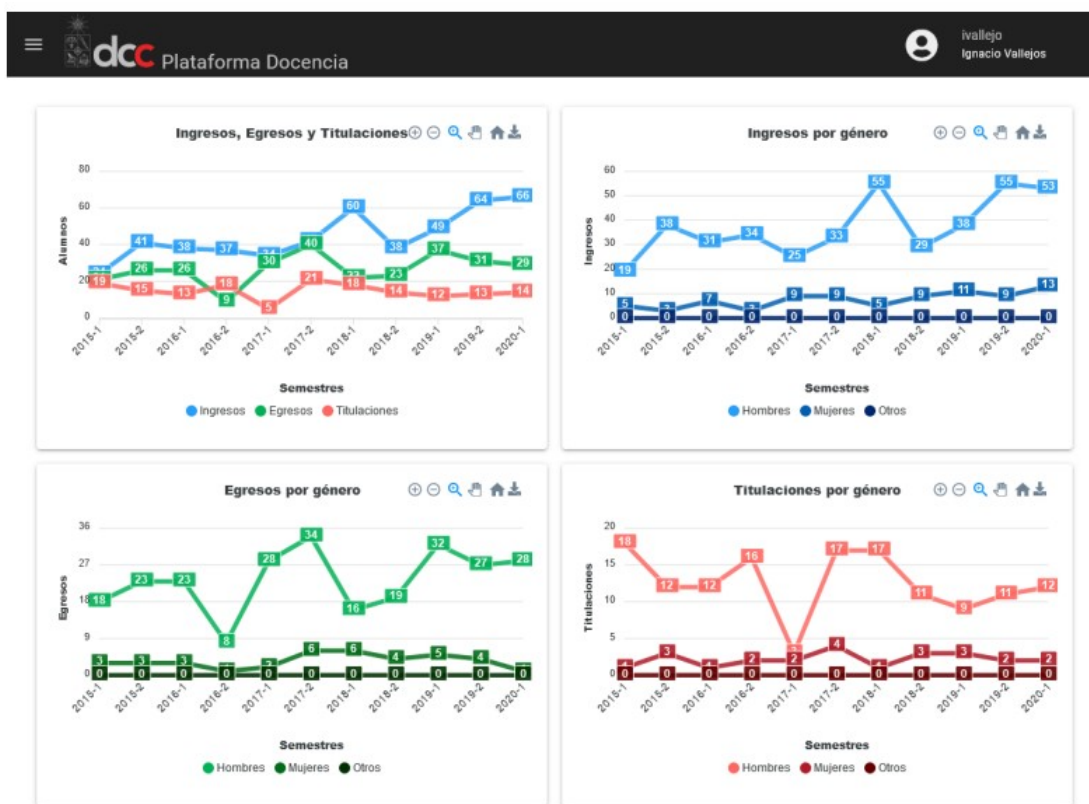


Figura 2.2: Plataforma Docencia DCC, dashboard por género [2]

2.1.3. Implementación de una herramienta de apoyo a la toma de decisiones en instituciones de educación superior en base a la información curricular de Ucampus

La memoria a cargo de Roberto Tapia [3], estudiante de ingeniería civil en computación en la Universidad de Chile, constaba en el diseño e implementación de una herramienta que entrega una visión global del estado de la institución para la facilitación en la toma de decisiones estratégicas.

Las métricas desarrolladas durante esta memoria estaban relacionadas a las requeridas durante el proceso de acreditación de una institución [7], tales como resultados de los proyectos de investigación (publicación en revistas, libros, ensayos, papers), cantidad de estudiantes, organización de personal administrativo y académico (por nombrar algunos). Si bien el género (sexo) fue utilizado para agrupar a los estudiantes en la métrica que determina los nuevos ingresos a la unidad académica por año, no era el principal foco de esta memoria, además de que solo fue calculado para los estudiantes, dejando fuera a docentes y no se mencionan otros estamentos como funcionaries.

En la Figura 2.3 se muestra el dashboard con las métricas de ingresos por género durante los años 2015 y 2021 implementado en esta memoria.

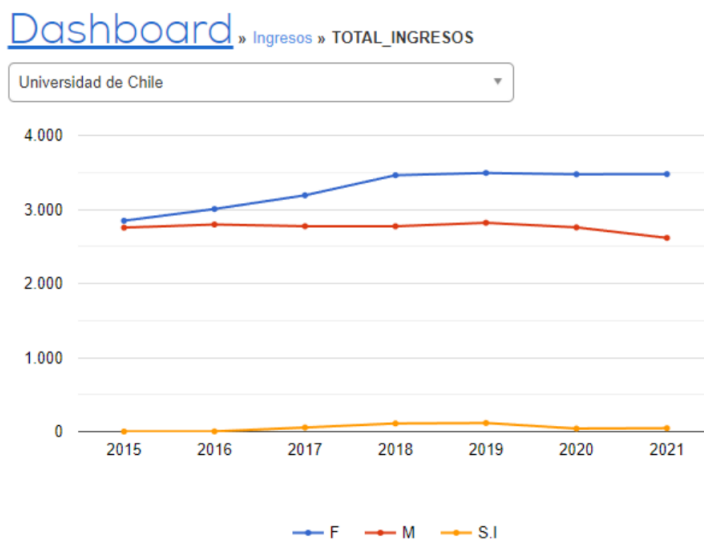


Figura 2.3: Dashboard de ingresos a la Universidad de Chile por género [3]

Este trabajo estaba dirigido a usuarios administrativos en apoyo a gestión. Las métricas e indicadores presentados engloban a estudiantes, académiques e incluso información institucional (años de acreditación, publicaciones por académiques, entre otros) de la Universidad de Chile, de los cuales, solo una (total de ingresos), de las métricas e indicadores implementadas se relaciona a género.

2.1.4. Ucampus

Ucampus es la plataforma utilizada por la Universidad de Chile, responsable en la gestión curricular, la cual abarca distintos módulos para cumplir con diversos requerimientos tales como inscripción académica, catálogo de cursos, certificados, votaciones, reserva de espacios, entre otros, pero no existe ningún módulo de apoyo para una mesa de género (a nivel de universidad) que permita extraer datos con el fin de analizarlos. Si bien, Ucampus disponibiliza una API¹ para el uso de datos, se requiere levantar una solicitud para acceder a ellas; a su vez, la información entregada en los endpoints tiene una visión general a nivel facultad y no a nivel departamento enfocado en el género, como se requiere.

2.2. Herramientas y/o plataformas para visualización de datos

Entre las opciones más populares se encuentran Tableau² y Power BI³. Ambas plataformas son utilizadas para el análisis y visualización de datos, principalmente en el área de Business Intelligence. Permite el manejo de grandes volúmenes de datos, además de la generación de métricas e indicadores (principalmente KPI's⁴) que junto con una interfaz sencilla, un usuario es capaz de armar gráficos a partir de estos cálculos.

Tanto Tableau como Power BI requieren el pago de una licencia para acceder a todas las funcionalidades que ofrecen y aunque cuentan con una versión gratuita, está disponible solo por un tiempo limitado desde que el usuario se registra.

Si bien, ambas herramientas ofrecen la generación de dashboards a partir de diversas fuentes de datos que permiten el mantenimiento de los datos en sus ubicaciones generales, presentan capacidades predefinidas en cuanto a flexibilidad y personalización en cuanto a funcionalidad y diseño; un ejemplo son los filtros precargados que ofrece Tableau. Tanto Power BI como Tableau permiten que le usuario integre y embeba informes y paneles en aplicaciones a través de funcionalidades como: *Publish to web*, *Power BI Embedded*, en Power BI y *Compartir o Tableau Embedded Analytics* en Tableau; hacer uso de estas características requieren un pago adicional de suscripción, pero es importante hacer mención que ambas herramientas tienen limitaciones de seguridad y privacidad, puesto que los informes y paneles son accesibles públicamente en Internet. Es relevante considerar además, que ante cualquier problema técnico, se debe hacer uso del soporte técnico de cada herramienta, lo cual quita en cierta medida, la autonomía que se pueda tener sobre la aplicación si se hace uso de alguna de estas opciones.

En las Figuras 2.4 y 2.5 se pueden apreciar dashboards o reportes generados en Tableau y Power BI a modo de ejemplo. En ellas se observan los distintos tipos de gráficos disponibles, además de la visualización de métricas e incluso nubes de palabras utilizadas como

¹<https://ucampus.uchile.cl/api/0/>

²<https://www.tableau.com/es-es>

³<https://powerbi.microsoft.com/es-es/>

⁴https://es.wikipedia.org/wiki/Indicador_clave_de_rendimiento

representación visual de la frecuencia y valor de las palabras (Figura 2.5).

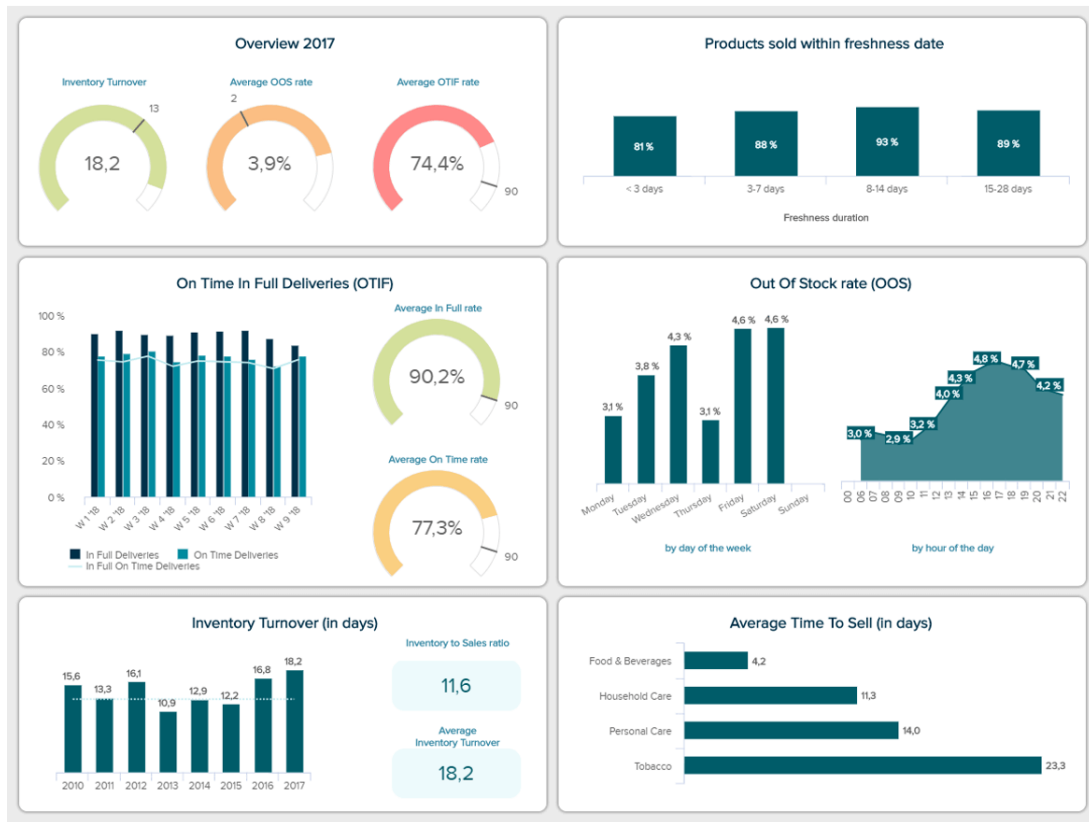


Figura 2.4: Dashboard generado en Tableau



Figura 2.5: Dashboard generado en Power BI.

2.3. Situación actual

La conformación de la Mesa de Género DCC nace en julio del 2021, a partir de la iniciativa *Genera Igualdad*⁵, registrando un grupo de veinte integrantes pertenecientes a los distintos estamentos (estudiantes, CaDCC, funcionarios, académicos, entre otros). La iniciativa consta de un proceso impulsado por la Universidad de Chile junto al Programa de Naciones Unidas para el Desarrollo, en el cual a través de la certificación y reconocimiento de estándares de igualdad de género y no discriminación, busca avanzar hacia una cultura inclusiva y con igualdad de derechos y oportunidades entre géneros. Con el fin de formalizar las iniciativas adoptadas dentro del departamento, la primera tarea de la Mesa de Género fue realizar un diagnóstico de la situación dentro del departamento basado en la perspectiva de género y para ello, recopilamos datos desde distintas fuentes como U-Cursos⁶, Ucampus⁷ y encuestas realizadas a la comunidad. Estos datos fueron almacenados en hojas de cálculo y en la actualidad se encuentran dentro de un Drive con acceso restringido. El registro fue utilizado para crear gráficos manualmente, que contribuyeron a entender el panorama en el que se encontraba el departamento en ese momento.

En relación al Drive de la Mesa de Género, este se encuentra organizado por carpetas que contienen minutas de reuniones junto con sus grabaciones, presentaciones con resultados de encuestas relacionadas a género. Dentro de los archivos, las carpetas de Pregrado y Postgrado son las que contienen los datos registrados en hojas de cálculo, del cual se obtiene el siguiente listado:

PREGADO:

- **Equipos Docentes:** tabla que contiene los tipos de docentes del departamento (Ayudante, Profesor Auxiliar, Profesor Coordinador, entre otros), curso, semestre, sexo y cantidad, entre los años 2013-2021.
- **Estudiantes DCC:** tabla que contiene la cantidad de estudiantes según sexo (femenino, masculino), curso y semestre entre los años 2013-2021.
- **Estadísticas CC1000-CC1002:**
 1. Tabla de auxiliares del curso CC1000-CC1002, indicando año, sexo (femenino, masculino) y cantidad total de auxiliares, entre los años 2017-2021.
 2. Tabla de ayudantes del curso CC1000-CC1002, indicando año, sexo (femenino, masculino) y cantidad total de ayudantes, entre los años 2017-2021.
 3. Tabla de estudiantes del curso CC1000-CC1002, indicando año, sexo (femenino, masculino) y cantidad total de estudiantes, entre los años 2017-2021.
- **Género alumnos CC5002:** Tabla de estudiantes del curso CC5002, indicando cantidad de alumnos según sexo (femenino, masculino), año y semestre, entre los años 2011-2021.

⁵<https://direcciondegenero.uchile.cl/generaigualdad/>

⁶<https://www.u-cursos.cl>

⁷<https://ucampus.uchile.cl>

- **Egresados DCC:** Tabla de estudiantes egresados según tipo (Ingenieros en Computación, Magíster, Doctorado), indicando cantidad según sexo (femenino, masculino) y año, entre el periodo 2011-2021.

POSTGRADO:

- **Tabla de alumnos inscritos en Doctorado:** contiene rut, nombres, apellido paterno, apellido materno, correo, estado (Abandono de Estudios, Eliminado, Graduado, No Activo, Postergación de Estudios, Regular, Retiro Voluntario), año.
- **Tabla de alumnos de Postgrado:** contiene cantidad total de ingresos, cantidad de mujeres, cantidad de hombres.
- **Tabla de alumnos graduados:** contiene cantidad total de graduados, cantidad de mujeres, cantidad de hombres.
- **Tabla de alumnos graduados:** contiene cantidad de ingresos, cantidad de graduados, cantidad de mujeres, cantidad de hombres.
- **Tabla de alumnos que desertan:** contiene cantidad de ingresos, cantidad de alumnos que continúan, cantidad de alumnos que desertan, cantidad de mujeres que desertan, cantidad de hombres que desertan.
- **Tabla de alumnos de Postgrado:** contiene total de ingreso, cantidad de mujeres, cantidad de hombres cada diez años.

Actualmente, no se cuenta con ningún sistema automatizado centrado en el DCC, para generar gráficos o representar métricas e indicadores a partir de la carga de data, todo se hace a través de las funcionalidades de las hojas de cálculo.

2.3.1. Propuesta inicial de métricas e indicadores

Durante el segundo semestre del año 2022, se trabajó en conjunto con algunos integrantes de la Mesa de Género para construir un listado de métricas e indicadores, tomando como base los datos (hojas de cálculo) almacenados en el Drive que dispone el organismo de forma privada. Los integrantes que accedieron a participar en dichas validaciones fueron:

- María Cecilia Bastarrica (profesora asociada al departamento).
- Jocelyn Simmonds (profesora asistente del departamento, profesora guía).
- Nahuel Gómez (alumno e integrante de la comisión de Género del CaDCC 2022).

A partir de los comentarios recibidos durante reuniones efectuadas con los participantes anteriormente mencionados, se propuso el siguiente listado:

Listado de métricas por año:

1. Total de miembros en el DCC.
2. Total de miembros por estamento.
3. Total por género (de todo el departamento).
4. Total por género por estamento.
5. Total de alumnos por curso.
6. Total de alumnos por género según curso.
7. Número de alumnos según tipo de ingreso y género.

Listado de indicadores:

1. Comunidad:
 - Proporción según género.
2. Estudiantes:
 - Proporción según género: entendiéndose como la proporción entre *la cantidad de estudiantes de un tipo de género y la cantidad total de estudiantes*.
 - Proporción según género por curso (año): entendiéndose como la proporción entre *la cantidad de estudiantes de un tipo de género de un curso y la cantidad total de estudiantes de un tipo de género*.
 - Proporción según género por tipo de ingreso (año).
3. Académiques:
 - Proporción según género.
 - Proporción según género por jerarquía.
4. Equipo docente:
 - Proporción según género.
 - Proporción según género por tipo (auxiliar, ayudante).
 - Proporción según género por tipo de ramo⁸.
5. Postgrado:
 - Proporción según género.
 - Proporción según género por tipo (magíster, doctorado).
 - Proporción según género por etapa⁹.

⁸Esto se refiere a qué área pertenece el ramo, ya sea de investigación, teoría o práctico.

⁹Esto aplica solo para Doctorado. Dentro del proceso de Doctorado hay distintas etapas entre las cuales están: Examen de calificación, Proyecto de tesis y Examen de grado.

- Proporción según género por tiempo desde que inició postgrado.

6. Funcionaries:

- Proporción según género.
- Proporción según género por cargo.

Un punto importante a destacar de este listado, es que en un principio no se tenía contemplado el incluir la proporción por etapa para el estamento de Postgrado: esta fue una sugerencia levantada por la profesora Bastarrica. Otro de los indicadores agregados a partir de sus comentarios, fue la proporción según género por jerarquía en Académiques. De la misma manera, gracias a los comentarios de la comisión de Género del CaDCC, se añadió la proporción según género por tipo de ramo en el estamento de Equipo Docente.

Capítulo 3

Análisis y Diseño

De acuerdo a lo planteado en Introducción y Estado del arte, la solución propuesta contempla la implementación de una herramienta que permita a los integrantes de la Mesa de Género, generar métricas e indicadores enfocados en género y que a partir de ellas se puedan generar gráficos para su análisis.

En este capítulo se precisan tipos de usuarios y requisitos que debe cumplir la solución, junto con alternativas consideradas y decisiones que llevaron a la solución desarrollada a lo largo de esta memoria.

3.1. Análisis

En esta sección se describe hacia qué usuarios se enfoca la solución, junto con los requisitos que debiese cumplir la herramienta con el fin de dar solución a la oportunidad presentada en la sección 1.1.

3.1.1. Tipos de usuarios y requisitos

Los usuarios finales de la plataforma serían los integrantes de la Mesa de Género del Departamento de Ciencias de la Computación (DCC), incluyendo estudiantes, personal administrativo, académicos y cualquier participante que se haya unido a la Mesa de Género. Los usuarios buscan medir, analizar y entender las métricas relacionadas con la brecha de género dentro del DCC. El objetivo principal se enfocaría en utilizar estos datos para informar, planificar e implementar medidas que promuevan la igualdad de género. Esto incluiría, pero no se limitaría a, identificar áreas problemáticas, monitorizar el progreso de las intervenciones existentes y proporcionar datos objetivos para apoyar el debate y la toma de decisiones en las reuniones de la Mesa de Género.

Entre los usuarios que interactuarán con la plataforma, se identifican dos tipos de roles: administrador y usuario general. Sus acciones y casos de uso se detallan siguiendo la estructura de Historias de Usuario (HU), a continuación el listado de requisitos funcionales:

1. Como usuario administrador quiero cargar datos de cada estamento en formato `.csv` para generar métricas e indicadores.
2. Como usuario administrador y general quiero visualizar métricas e indicadores relacionados a género, de los distintos estamentos dentro del DCC para entender la situación interna del departamento en cuanto a brecha de género.
3. Como usuario administrador y general quiero filtrar y segmentar datos de diferentes maneras para realizar un análisis más detallado según las necesidades y objetivos de la Mesa.

Es importante mencionar que además de los requisitos funcionales detallados anteriormente, existen requisitos no funcionales. La plataforma debe ser mantenible, lo que implica la posibilidad de agregar nuevas características y/o modificar las ya existentes. Adicionalmente, la plataforma debe ser extensible, es decir, diseñada de tal manera que pueda adaptarse y crecer para acomodar nuevas funcionalidades y capacidades en el futuro, ya sea agregando nuevas métricas e indicadores y/o nuevos estamentos que surjan dentro del departamento.

3.2. Diseño

En esta parte se aborda la arquitectura de la plataforma con el objetivo de cumplir con los requerimientos establecidos anteriormente en la sección 3.1.1. En las siguientes secciones, se describe cómo se organiza el sistema en diferentes componentes y aplicaciones, junto con las distintas herramientas a utilizar para cumplir con las funcionalidades de carga de datos, cálculo de métricas e indicadores y visualización de estos a través de gráficos. Adicionalmente, se explica el modelo de datos utilizado e interfaces generadas a partir de sitios webs relacionados al departamento de Computación.

3.2.1. Arquitectura General

La plataforma sería diseñada utilizando Django, un marco de trabajo de alto nivel para el desarrollo de aplicaciones web en Python que sigue el patrón Modelo-Vista-Controlador (MVC). Este patrón promueve la separación de responsabilidades, facilitando así el mantenimiento y extensibilidad de la plataforma.

En la Figura 3.1 se muestra el funcionamiento general de la plataforma basado en la estructura que maneja Django.

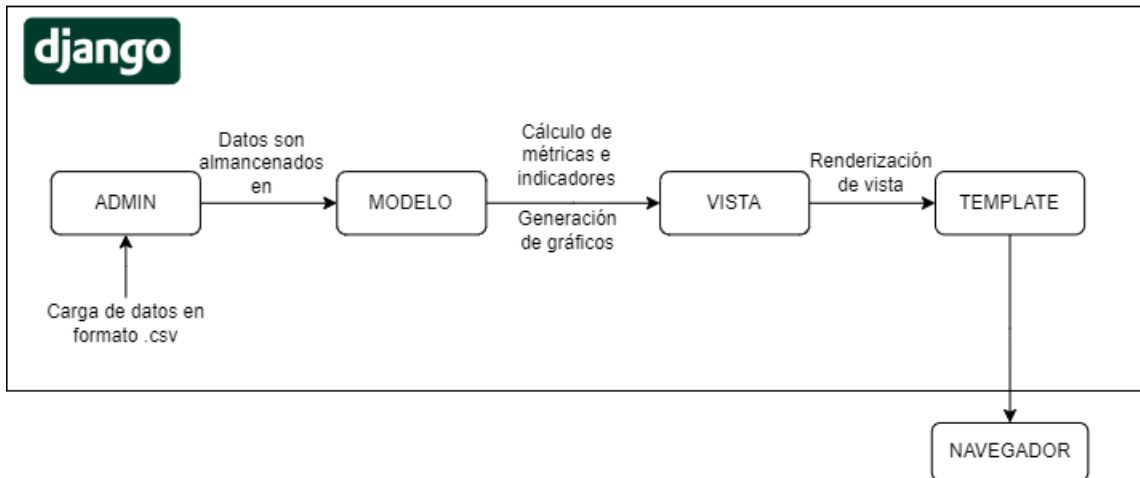


Figura 3.1: Diagrama del funcionamiento de la plataforma

En cuanto a la organización del proyecto, Django permite la creación de aplicaciones modulares y autónomas que pueden ser reutilizadas en diferentes proyectos. Bajo esta definición, existirían tres aplicaciones:

- **genderboard**: módulo encargado de las configuraciones generales y necesarias del proyecto, por ejemplo, las configuraciones de los archivos estáticos que incluyen **templates**, **CSS**, entre otros.
- **core**: módulo donde se define la estructura de los modelos de datos. Además, incluye la funcionalidad de carga de datos a través de la vista de administradore proporcionada por Django.
- **metrics**: módulo encargado del cálculo de métricas e indicadores, además de la generación y renderización de los gráficos interactivos que se muestran al usuario.

La integración entre estas tres aplicaciones se pueden entender mejor con el diagrama presentado en la Figura 3.2. En ella, se muestra que desde el módulo de carga de datos que recibe los archivos desde el usuario, alimenta al módulo de cálculo con los datos subidos. Este módulo con las métricas e indicadores generados, pasa estos resultados al módulo encargado de la generación de gráficos, el cual renderiza las vistas hacia la interfaz.

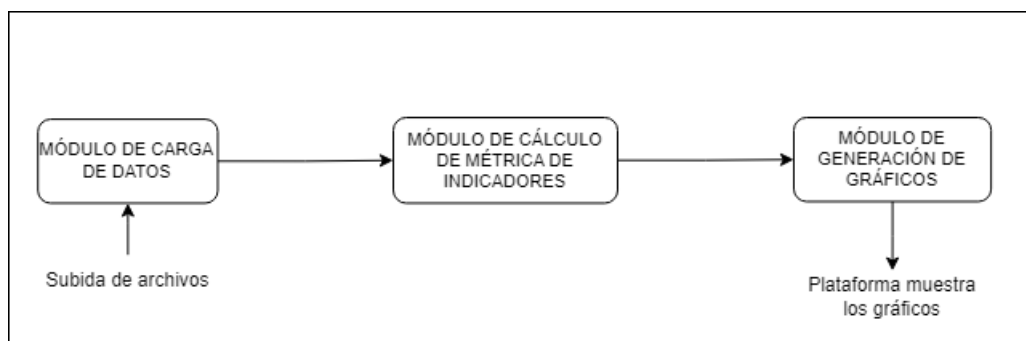


Figura 3.2: Diagrama del funcionamiento entre los tres módulos

Al dividir la lógica del proyecto en aplicaciones distintas, se sigue el principio de responsabilidad única. Esta separación facilitaría la legibilidad del código y su mantenimiento, dado que cualquier modificación se realizaría en las partes individuales del sistema, promoviendo un alto grado de desacoplamiento. Esto también potencia la reutilización del código, especialmente en el uso de modelos en distintas partes del proyecto, evitando así la posible duplicación de código.

Además, la estructura de las aplicaciones ofrece beneficios significativos en términos de escalabilidad. A medida que el proyecto crece y se añadan nuevas funcionalidades o se necesite manejar una mayor cantidad de datos, cada aplicación puede extenderse de manera independiente. Este enfoque no solo minimiza el impacto de los cambios en las demás partes del sistema, sino que también permite un desarrollo y pruebas más focalizados, lo que agiliza la detección y solución de errores.

En conclusión, esta organización fomentaría un código más limpio, facilitaría el mantenimiento, promovería la escalabilidad y proporcionaría una mayor flexibilidad en el desarrollo y la adaptación a nuevas necesidades.

3.2.2. Carga de Datos

La idea inicial para la carga de datos en la plataforma era desarrollar un módulo dentro de Django encargado de esta tarea. Para ello, se pensó en hacer uso de Forms junto con una vista que pidiera al le usuare cargar un archivo en formato `.csv`. Luego, esta vista debía ser integrada al panel de administración proporcionado por el framework. Pero durante el proceso de implementación, esta alternativa resultó ser complicada de integrar al panel de administración.

A partir de ello, se encontró otra alternativa, hacer uso de una librería externa y dejar que ella se encargase de la tarea de la carga de datos.

La librería `django-import-export`¹, permite importar y exportar datos en diversos formatos, incluyendo `.csv`, JSON y Excel, proporcionando una gran flexibilidad en la forma en que se pueden cargar los datos en la plataforma. El administrador podría subir archivos en formato `.csv`, que luego se almacenarían en la base de datos del sistema (en esta primera etapa de forma local) para su análisis y visualización. Considerando que no requería grandes intervenciones en cuanto a código y que además, la integración con el panel de administrador era mucho más sencilla si se comparaba con la alternativa anterior, se decidió tomar esta opción y desarrollarla para cumplir con la funcionalidad de subida de datos.

¹<https://django-import-export.readthedocs.io/en/latest/>

3.2.3. Cálculo de Métricas e Indicadores

Entre las alternativas que lograran resolver el cálculo de métricas e indicadores, se encontraba la opción de utilizar NumPy², una librería utilizada para realizar operaciones numéricas en arreglos multidimensionales. Si bien, esta opción y pandas³ son muy parecidas, la deficiencia de la primera es que no cuenta con una estructura que permita el manejo flexible de datos tabulares, mientras que pandas ofrece `dataframes`. Adicional a esto, NumPy no proporciona la funcionalidad de agrupar por etiquetas, por ejemplo, agrupar datos por año, género, tipo; característica necesaria dado que la idea principal es poder filtrar los datos por las columnas antes mencionadas.

Es por eso que, se decidió hacer uso de pandas.

Pandas proporciona estructura de datos flexibles y eficientes que permiten realizar operaciones en grandes conjuntos de datos con facilidad. Las métricas e indicadores se calcularían a partir de los datos cargados en la plataforma, para luego visualizarlos a través de gráficos interactivos. El proceso de cálculo se realizaría dentro de la sección `views` proporcionado por Django.

3.2.4. Visualización de Datos

Para la visualización de métricas e indicadores, se hará uso de Plotly⁴, una librería de gráficos en Python que produce gráficos interactivos de alta calidad. Los gráficos de Plotly se generarán dinámicamente a partir de las métricas e indicadores calculados y se mostrarán en la interfaz de usuario de la plataforma, permitiendo la descarga de gráficos en formato de imagen `.png`, además de la selección de datos por medio de filtros. Los filtros personalizados implementados en Plotly, permiten la visualización de datos de manera inmediata, traduciéndose en una mejor experiencia hacia el usuario.

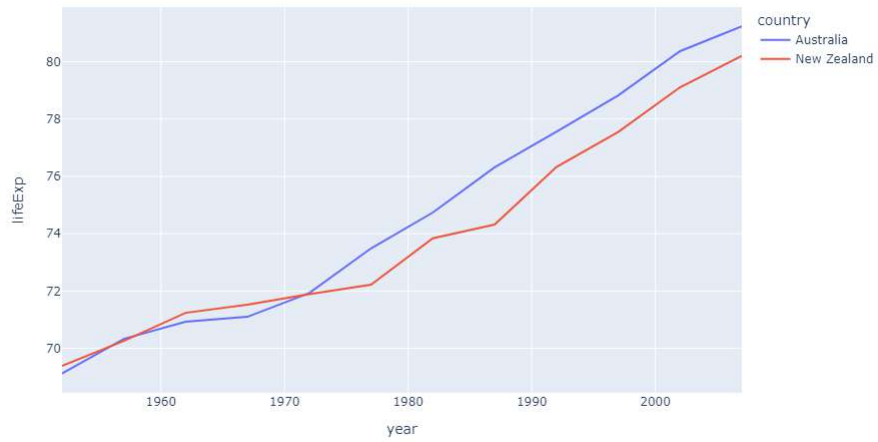
Plotly ofrece una amplia gama de funcionalidades, desde la personalización de gráficos en cuanto a colores y fuentes de ejes y leyendas, hasta herramientas interactivas que permiten a los usuarios acercarse y alejar zonas dentro de los gráficos, hacer clic en determinados elementos para obtener más detalles y activar o desactivar conjuntos de datos directamente desde el gráfico (filtros predeterminados de Plotly).

En la Figura 3.3 se pueden apreciar los distintos tipos de gráficos pensados para utilizar: gráfico de línea para representar tendencias temporales relacionados a la cantidad de integrantes por género entre un periodo de tiempo. Gráfico de torta para representar las distribuciones según género.

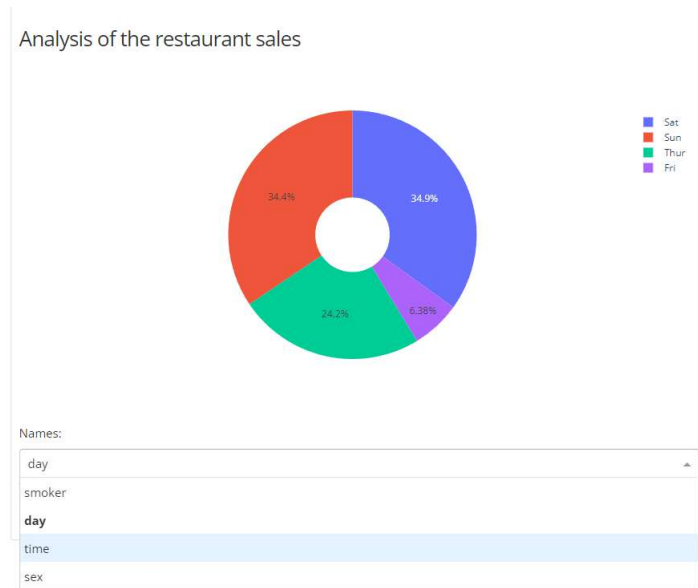
²<https://numpy.org/>

³<https://pandas.pydata.org/docs/>

⁴<https://plotly.com/python/>



(a) Gráfico de línea



(b) Gráfico de torta junto con filtros personalizados

Figura 3.3: Ejemplos de gráficos disponibles en Plotly

3.2.5. Diagrama de clases

Con el objetivo de presentar la información procesada de forma clara y ordenada y tomando en cuenta la estructura que maneja Django en cuanto a estructura de datos almacenados, la mayoría de las entidades/estamentos dentro del departamento serán representados por una clase, a excepción de Estudiantes y Comunidad DCC; para el caso de Estudiantes, la tabla o clase asociada que se utilizará para el cálculo de sus métricas e indicadores llevará por nombre **Ramos**, mientras que para el grupo que reúne a toda la comunidad DCC en general, será nombrada como **Universo**, todo esto para asegurar coherencia entre los nombres de las tablas y los datos que se almacenan en cada una.

Bajo la premisa de facilitar el análisis de la brecha de género y permitir la identificación de patrones y tendencias a lo largo del tiempo, las clases se construirán de acuerdo al diagrama mostrado en la Figura 3.4. Cada clase es independiente, por lo que no existen relaciones entre ellas.

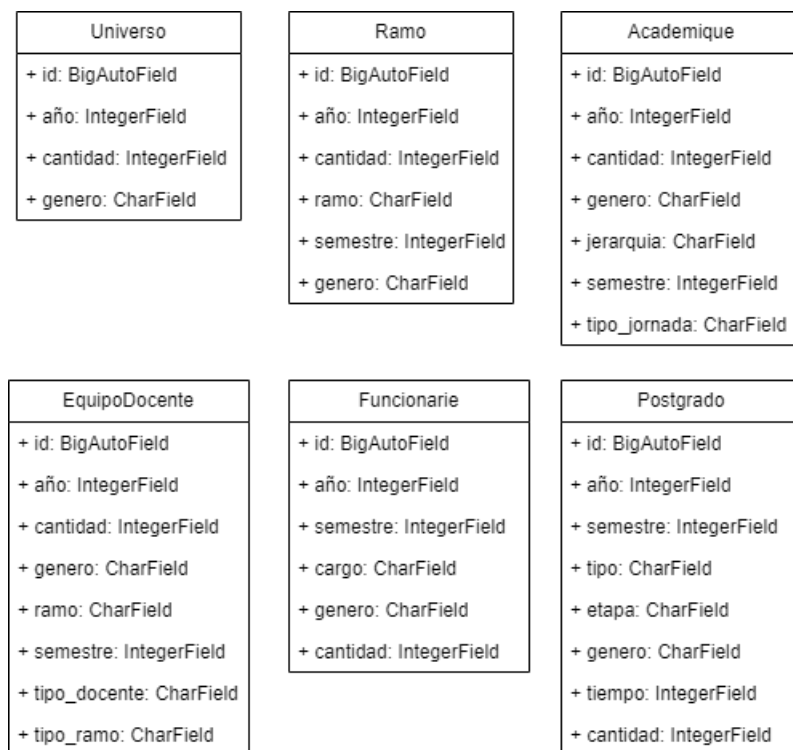


Figura 3.4: Diagrama UML de clases

De forma aclaratoria, a continuación se detallan brevemente algunos campos presentes la clase **Postgrado**:

- **tipo**: determina si pertenece al grupo de magíster o doctorado.
- **etapa**: este campo aplica solo para Doctorado. Dentro del proceso de Doctorado existen distintas etapas, las cuales son: Examen de calificación, Proyecto de tesis y Examen de grado.
- **tiempo**: cantidad de años desde que inició postgrado.

3.2.6. Interfaces e interacciones en la plataforma

Para facilitar la navegación y el análisis, la interfaz de usuario se dividirá en secciones correspondientes a los diferentes estamentos del departamento. Estos incluyen Comunidad, Estudiantes, Académiques, Equipo Docente, Funcionaries y Postgrado. Cada sección presenta métricas e indicadores a través de gráficos (línea, torta, entre otros) con filtros, permitiendo a los usuaries explorar y analizar datos relevantes.

En cuanto a la paleta de colores y diseño de interfaz, se tomará como base, las plataformas existentes relacionadas al DCC, siendo estas el sitio web oficial del departamento⁵ y el portal de Proyecto de Software⁶. A partir de lo anterior, en la Figura 3.5 se aprecia un bosquejo inicial de lo que sería la plataforma. En esta, es importante destacar el diseño de la barra lateral que reemplazaría el menú superior, pensando en la posibilidad futura de que nazcan nuevos estamentos.



Figura 3.5: Vista propuesta de la plataforma a partir de otros sitios del DCC

Cada uno de los gráficos generados por la plataforma tendrán sus correspondientes filtros, que permitirán al usuarie fijar un año o género (según corresponda). Esta funcionalidad cumple con el listado presentado en la sección 3.1.1, específicamente las Historias de Usuario 2 y 3.

⁵<https://www.dcc.uchile.cl/>

⁶<https://portalpsw.dcc.uchile.cl/>

Capítulo 4

Implementación

En el presente capítulo se detalla la implementación de la plataforma en base al análisis y diseño descrito en el Capítulo 3. De esta manera, esta parte se divide en tres secciones: la primera contempla configuraciones iniciales e implementación de la funcionalidad de carga de datos, siguiendo con la segunda parte que registra el proceso de desarrollo para el módulo de cálculo de métricas e indicadores, finalizando con la tercera y última etapa, enfocada en la generación de gráficos a partir de las métricas e indicadores calculados.

4.1. Primera Etapa

Como se menciona anteriormente, en esta etapa inicial de implementación, se procede a las configuraciones iniciales del proyecto, haciendo énfasis a la estructura que llevaría la herramienta junto con la implementación de las primeras vistas de la plataforma.

4.1.1. Configuraciones iniciales y levantamiento del proyecto

Siendo Django el framework escogido para el desarrollo de la herramienta web, dado que es un marco de trabajo que proporciona una estructura organizada y clara que facilita la creación de aplicaciones web de manera rápida y eficiente, además de proporcionar librerías y herramientas integradas que automatiza tareas, como por ejemplo, la utilizada en este desarrollo relacionada a la carga de archivos por la vista de administración integrada por Django. Dentro de los beneficios de haber escogido este framework, se encuentra además el patrón Modelo-Vista-Controlador (MVC)¹, que emplea como lógica de arquitectura que separa la aplicación, la presentación de datos y la manipulación de solicitudes de usuario, otorgando un código modular y de fácil mantenimiento.

Para las configuraciones iniciales, se procedió a la instalación de Django y creación del proyecto en el entorno de desarrollo a través de la línea de comandos proporcionado por

¹<https://es.wikipedia.org/wiki/Modelo-vista-controlador>

el mismo framework, el cual generó la estructura inicial del proyecto, incluyendo archivos y directorios necesarios para dar comienzo a la implementación. Dentro de la estructura inicial se incluyó una aplicación que por defecto se crea, llamándola **genderboard**, la cual contiene todas las configuraciones correspondientes a las generadas por la línea de comandos del framework.

Con el proyecto ya configurado, se procedió a la implementación del diseño de la plataforma, explicado en la siguiente sección.

4.1.2. Implementación de vista principal

En cuanto al desarrollo de la vista, se tomó el *mockup* de la Figura 3.2.6 como referencia. Considerando la posibilidad de futuras modificaciones que pudiese llegar a sufrir este diseño, se decidió separar los componentes gráficos de barra superior y barra lateral en archivos distintos, esto para facilitar el entendimiento y realización de cambios sobre las plantillas, además de obtener mayor flexibilidad y modularidad en el diseño, junto con disminuir la duplicación de código en vistas que requieran estos mismos elementos.

La organización del proyecto, de acuerdo a esta primera iteración quedó tal como se señala en la Figura 4.1. Durante esta etapa, solo se tenía creada la aplicación **genderboard**, responsable de las configuraciones iniciales del proyecto, junto con el diseño de la primera vista, presentada en la Figura 4.2.

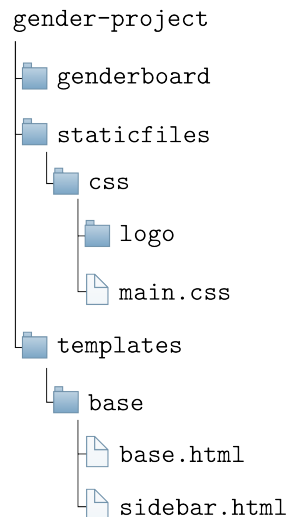


Figura 4.1: Organización de archivos relacionado a diseño dentro del proyecto

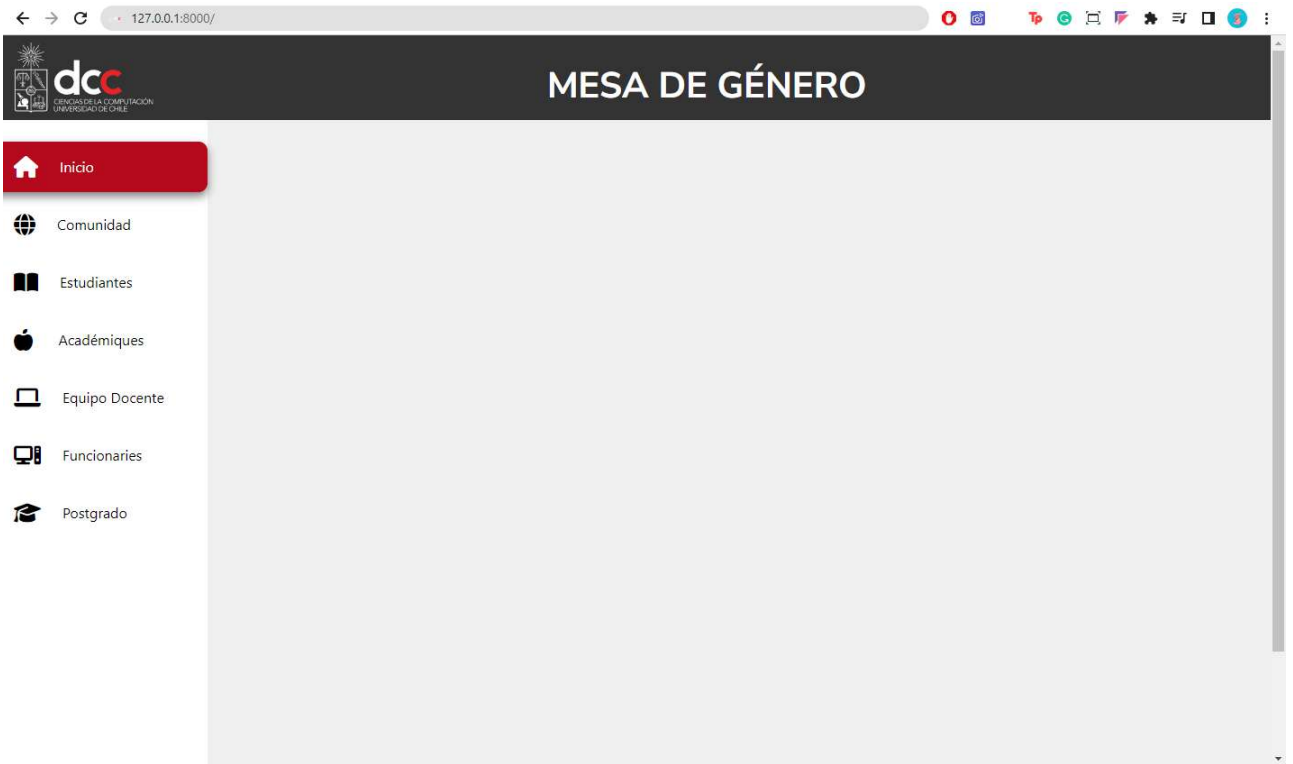


Figura 4.2: Primera iteración de la vista principal de la plataforma

Es importante mencionar que para el desarrollo de esta vista y de la plataforma en general, se tomó en cuenta solamente la resolución 1440×990 px, utilizada para pantallas de laptop. Cada una de las secciones/botones queda destacada en el color correspondiente dependiendo de lo que diga la barra de navegación. Por ejemplo, si la url contiene `/comunidad`, la barra lateral tendrá destacado en rojo la sección de Comunidad correspondiente.

Con la vista principal implementada, se continuó el proceso de carga de datos en la plataforma, la cual se explica en la siguiente etapa.

4.2. Segunda Etapa

Esta sección se enfoca en el desarrollo del módulo de carga de datos de la plataforma, proceso que involucró la creación de modelos dentro del proyecto, la integración de la librería `django-import-export` para la carga de datos y posteriormente, el proceso de carga masiva de datos de prueba.

4.2.1. Creación de modelos

Antes de iniciar con el proceso de implementación de la funcionalidad de carga de datos, fue necesario crear los modelos que representarían la estructura y lógica de los datos en la plataforma. Para ello y siguiendo la idea de separar responsabilidades/funcionalidades de la herramienta en distintos módulos, se creó una nueva aplicación en el proyecto. Como se explicó en la sección 3.2.1, en la aplicación `core` se realizaron todos los cambios correspondientes a la creación y configuración de modelos antes de dar paso a la implementación de la funcionalidad de carga de datos. Con la creación de otra aplicación encargada de una nueva funcionalidad, le otorga al proyecto el carácter modular que permite agregar, quitar y/o modificar estos módulos dependiendo de las necesidades futuras; gracias a esto el proyecto será mucho más flexible y facilitará la escalabilidad al momento de agregar nuevas funcionalidades, a medida de que vaya creciendo, por ejemplo, para el caso en que se modifiquen los estamentos, ya se agregando nuevos y/o modificando sus atributos correspondientes.

La creación de modelos se hizo en base a lo establecido en la sección 3.2.5, donde cada estamento se tradujo en un nuevo modelo dentro de la aplicación `core`, para ello, se modificó el archivo `models.py` donde se registró la estructura de datos de cada una de las entidades. En el Código 4.1, se muestra un ejemplo de la construcción de la clase para el modelo de datos relacionado a `Academique`.

```
1 class Academique(models.Model):
2     ano = models.IntegerField()
3     semestre = models.IntegerField()
4     tipo_jornada = models.CharField(max_length=2)
5     jerarquia = models.CharField(max_length=255)
6     genero = models.CharField(max_length=1)
7     cantidad = models.IntegerField()
8
9     def __str__(self):
10        return self.genero
```

Código 4.1: Clase correspondiente al modelo de datos `Academique` dentro de la aplicación `core`

Con los modelos creados para cada una de las entidades, se procedió a desarrollar la funcionalidad de carga de datos presentada en la siguiente sección.

4.2.2. Implementación de carga de datos

Esta etapa correspondió al desarrollo de la funcionalidad de carga de datos a la plataforma. Como primera opción, se generó una vista en el archivo `views.py` de la aplicación `core` con un método `get` que renderizaba un archivo `.html`. Para esta alternativa, se hizo una primera prueba para el modelo `Ramo`, donde su esquema se puede ver en el Código 4.2. El archivo `views.py` modificado se puede observar en el Código 4.3. Además de registrar el modelo, se agregó un formulario en el archivo `forms.py` de la aplicación correspondiente. Los formularios en Django se utilizan para crear y procesar formularios web, permitiendo ingresar datos y enviarlos al servidor. Para este caso, se registraron dos formularios `RamoForm` y `uploadForm`. En el Código 4.4 se ve la implementación mencionada.

```
1 class Ramo(models.Model):
2     ramo = models.CharField(max_length=255)
3     ano = models.IntegerField()
4     semestre = models.IntegerField()
5     genero = models.CharField(max_length=1)
6     cantidad = models.IntegerField()
7
8     def __str__(self):
9         return self.ramo
```

Código 4.2: Archivo `models.py` con la estructura de modelo `Ramo`

```
1 from django.http import HttpResponse
2 from django.shortcuts import render
3 from django.views.generic.base import View
4 from .forms import UploadForm
5
6
7 class UploadView(View):
8
9     def get(self, request, *args, **kwargs):
10        return render(request, 'upload.html', {"form": UploadForm()})
```

Código 4.3: Archivo `views.py` de la aplicación para la implementación de funcionalidad de carga de datos

```
1 from django.forms import FileField, Form, ModelForm
2 from .models import Ramo
3
4 class RamoForm(ModelForm):
5     class Meta:
6         model = Ramo
7         fields = ["ramo", "ano", "semestre", "genero", "cantidad"]
8
9 class UploadForm(Form):
10    ramo_file = FileField()
```

Código 4.4: Archivo `forms.py` de la aplicación para la implementación de funcionalidad de carga de datos

En el Código 4.4 se definen los dos formularios antes mencionados. `RamoForm` es del tipo `ModelForm`, un modelo que se utiliza para crear un formulario automáticamente basado en los campos y validaciones definidas en un modelo de base de datos. Para este caso, `RamoForm` se utilizará para crear o actualizar instancias del modelo `Ramo`. Los campos del formulario se definen en `fields`, presentado en la línea 7 del código y estos campos corresponden con los atributos del modelo `Ramo`.

UploadForm es un formulario simple que contiene un campo `ramo_field` del tipo `FileField`. Este formulario se utiliza para permitir a los usuarios cargar archivos, en este caso, un archivo de ramos.

Con las estructuras necesarias, se creó un archivo `.html` para la renderización de esta vista en la plataforma. En el Código 4.5 se muestra su implementación y en la Figura 4.3 se muestra la vista implementada.

```
1 <html lang="en">
2   <body>
3     <form
4       action="/upload/"
5       enctype="multipart/form-data"
6       method="post"
7     >
8       {% csrf_token %}
9       {{ form }}
10      <br/>
11      <input type="submit" value="Submit">
12    </form>
13  </body>
14 </html>
```

Código 4.5: html para la vista de carga de datos

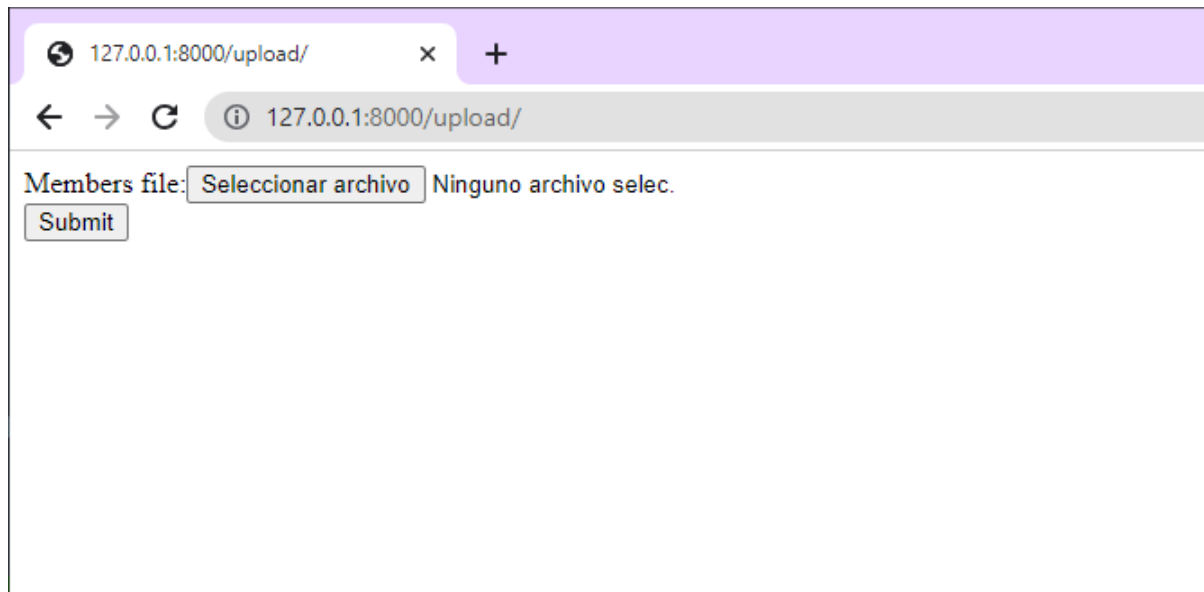


Figura 4.3: Vista de módulo de carga de datos implementado con formularios

Luego de analizar la solución anterior, se tomó la decisión de buscar otra alternativa, dado que la implementación de esta opción resultó ser engorrosa, tomando en cuenta que todavía faltaba por incluirla dentro de la vista de administración otorgada por Django, además de ajustar el diseño para que tuviera concordancia con la plataforma.

Finalmente, se optó por implementar una librería que cumpliera con esta funcionalidad.

Para ello, se hizo uso de la librería `django-import-export`, instalándola al entorno virtual mediante comandos de Miniconda. Para su configuración, se siguió la guía de instalación

disponible en la documentación oficial de la librería². Teniendo las configuraciones iniciales de la librería, se procedió a integrar la funcionalidad de carga de datos con la vista de administradore que tiene disponible Django.

Dentro de la aplicación `core`, encargada de la carga de datos, fue necesario editar el archivo `admin.py` para crear clases correspondientes a cada una de las entidades especificadas en 3.2.5. Por cada modelo se definen dos clases, una que indica la plantilla de cómo se deben procesar los datos para el modelo (atributos) y otra que habilita las opciones para importar los datos directamente desde la interfaz de administración. A continuación, se muestra una parte en el Código 4.6 para mayor entendimiento.

```
1 from django.contrib import admin
2 from import_export import resources
3 from import_export.admin import ImportExportActionModelAdmin
4 from .models import *
5
6 class AcademiqueResource(resources.ModelResource):
7     class Meta:
8         model = Academique
9 class AcademiqueAdmin(ImportExportActionModelAdmin):
10     resource_classes = [AcademiqueResource]
11
12 admin.site.register(Academique, AcademiqueAdmin)
```

Código 4.6: Implementación de librería `django-import-export`

En las líneas 6 a 8 del Código 4.6 anterior, se crea la clase correspondiente al modelo `Academique` que define la estructura y se centra en cómo se deben procesar los datos del modelo. `AcademiqueResource` toma el modelo traído desde `Academique` para validar que el archivo a cargar cumple con este.

En la clase `AcademiqueAdmin` de la línea 9 del Código 4.6 se disponibiliza la funcionalidad de importar la data desde la vista de administradore. En la línea 12 del mismo código, se registran los modelos y sus clases personalizadas en el sitio de administración de Django, que permite acceder a la importación/exportación en la interfaz para cada modelo.

Con esta funcionalidad implementada, la vista de administración cuenta con los botones para importar datos en cada uno de los modelos registrados en la plataforma, tal y como se puede apreciar en la Figura 4.4.

²<https://django-import-export.readthedocs.io/en/latest/installation.html>

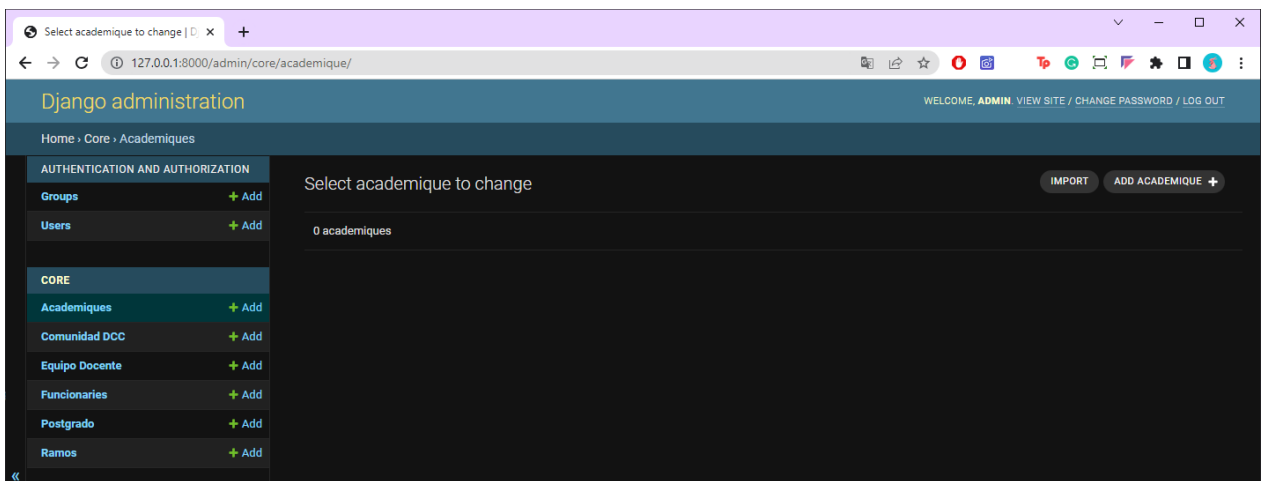


Figura 4.4: Vista de administración del modelo `Academique` con botón de importar disponible

En la Figura 4.5 se observa la pantalla de confirmación previa a la carga de datos al sistema, misma en la que se realizan las validaciones correspondientes donde se verifica que el archivo `.csv` a subir cumpla con los atributos registrados en la sección 4.2.1. Finalmente, una vez que le usuario confirma que los datos son correctos, la misma librería otorga una vista de caso exitoso a la plataforma, tal y como se observa en la Figura 4.6.

Esta librería en particular, tiene un manejo integrado de errores que aseguran que los datos a cargar cumplan con el modelo de datos determinado. En la Figura 4.7, se muestra el caso en que un usuario intenta subir un archivo que no cumple con el modelo establecido.

Es importante destacar que en el caso de error abarcado en la Figura 4.7, se utilizó un archivo que no cumplía con la estructura base del modelo `Academique` que abarca los siguientes parámetros: `año`, `semestre`, `tipo_jornada`, `jerarquía`, `género`, `cantidad`, es decir, al archivo le faltaba uno o más parámetros en su cabecera (en este caso, `semestre`). Con esto, se asegura que la carga de archivos hacia la plataforma corresponda a su debido modelo. Para el alcance de este proyecto, se configuró la librería solo para que permitiera la carga de datos nuevos a los modelos; la herramienta contempla funcionalidades para la actualización de datos dentro del archivo entregado que, de ser necesario para futuras implementaciones bastaría con cambiar parámetros dentro de la configuración de la misma librería³.

³https://django-import-export.readthedocs.io/en/latest/advanced_usage.html?highlight=update#create-or-update-model-instances

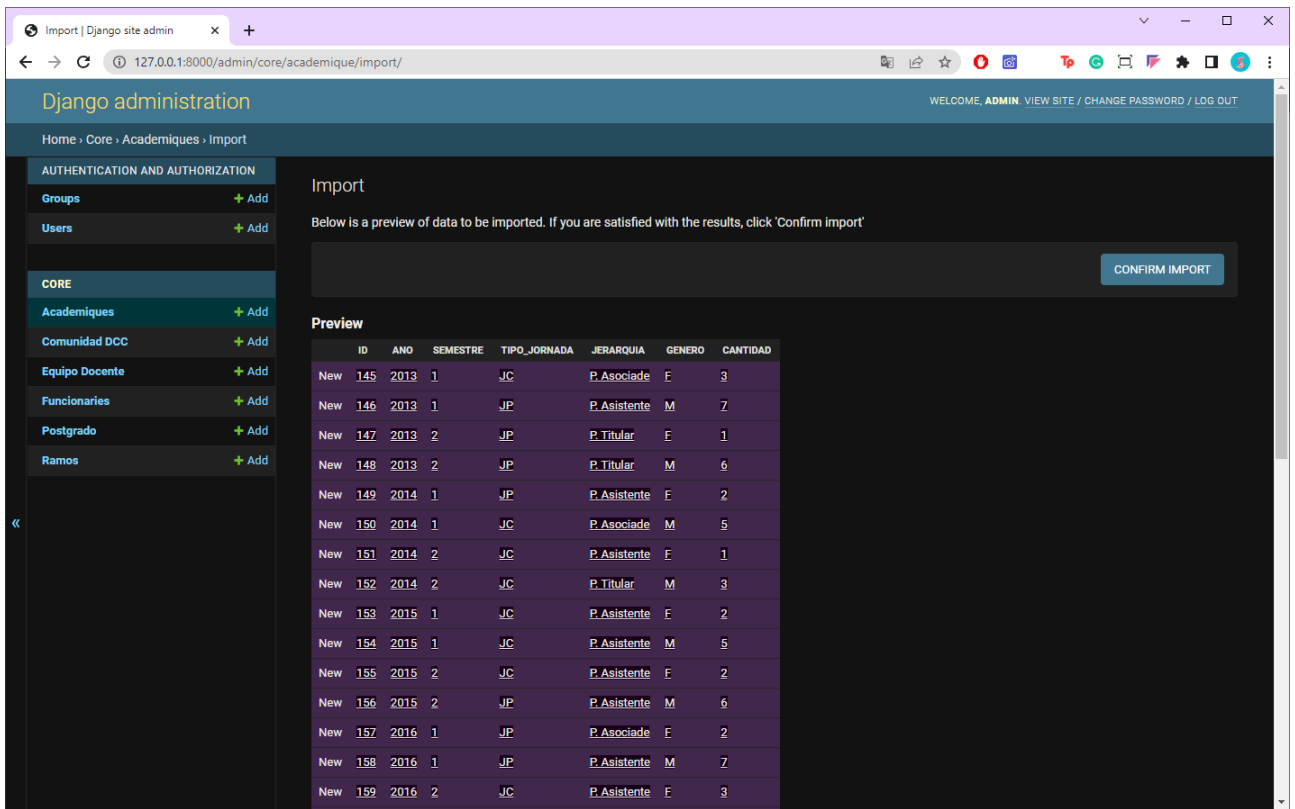


Figura 4.5: Vista de verificación de archivo al modelo Academique previa a la importación

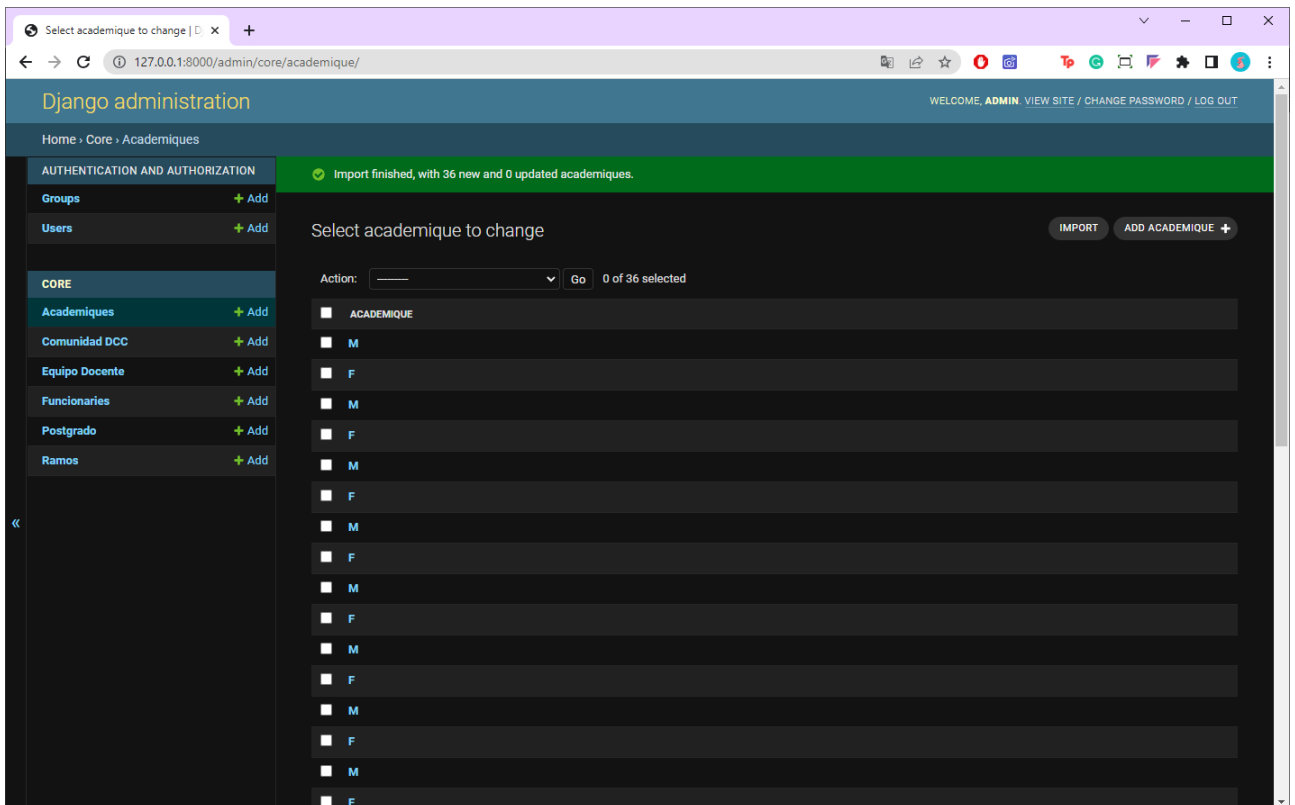


Figura 4.6: Vista de importación exitosa de archivo al modelo Academique

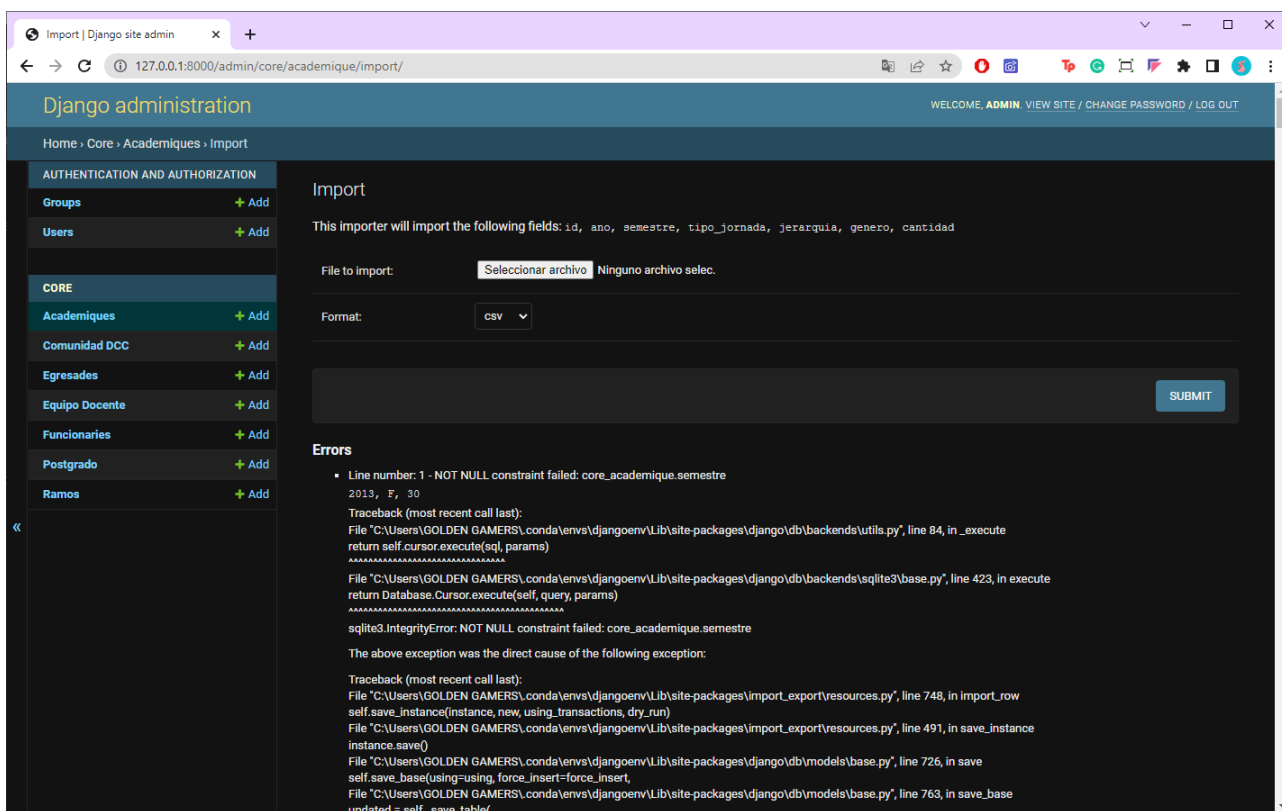


Figura 4.7: Vista de error al importar un archivo que no cumple con el modelo

Como paso culmine en esta etapa, se ejecutó la carga masiva de datos de prueba para alimentar la plataforma y así trabajar con la funcionalidad de cálculo de métricas e indicadores. Para cada uno de los modelos, se subieron archivos de formato `.csv` creados a partir de los datos que se almacenan en el Drive de la Mesa de Género, siguiendo la estructura establecida detallada en la sección 3.2.5 (Figura 3.4) y los pasos ejecutados con el modelo de `Academique` mostrado en las Figuras 4.5 y 4.6.

4.3. Tercera Etapa

La siguiente sección detalla el desarrollo del cálculo de métricas e indicadores. Primero se comienza calculando métricas a partir de un archivo `.csv` cargado de forma local. Posterior a ello, se procedió a generar gráficos a partir de las métricas calculadas, renderizando la vista para que se mostrara en la plataforma. Finalmente, se finalizó con el cálculo de indicadores, generando su gráfico correspondiente.

El proceso de cálculo se hizo para cada uno de los estamentos abarcados, considerados los modelos descritos en la sección 3.4.

4.3.1. Cálculo de métricas utilizando pandas

La siguiente funcionalidad implementada fue la de cálculo de métricas e indicadores, tomando como datos de entrada los entregados por el módulo de carga de datos desarrollado en la sección 4.2.2. Lo primero que se hizo fue crear la aplicación correspondiente a esta funcionalidad, bajo el nombre de `metrics` según el diagrama expuesto en el Capítulo 3 de Análisis y Diseño, específicamente en la Figura 3.2. Dentro de esta aplicación, las principales modificaciones fueron realizadas en el archivo `views.py`, en el cual se maneja toda la lógica relacionada a la plataforma; desde la consulta a la base de datos, ejecutar cálculos y la generación de respuesta hacia le usuarié.

Si bien, el módulo de carga de datos abarcado en la sección 4.2.2 ya se encontraba implementado, para cumplir con la independencia entre funcionalidades y módulos, se comenzó haciendo uso de archivos locales en formato `.csv` antes de hacer la integración directa con el módulo de carga de datos.

Comenzando con el cálculo de métricas y tomando en cuenta el listado expuesto en la sección 2.3.1, se inició creando una `view` de prueba dentro del archivo `views.py` de la aplicación `metrics`. Haciendo uso de `pandas` se procedió a crear `dataframes` desde los archivos locales que contenían datos de prueba. En el Código 4.7 se muestra un ejemplo del uso de la librería para el cálculo de total de integrantes por año y en la Figura 4.8 se muestra parte del archivo `.csv` utilizado como dato de prueba.

Nº	ramo	ano	semestre	genero	cantidad
1	CC1000	2014	1	F	233
2	CC1000	2014	1	M	608
3	CC1000	2014	2	F	11
4	CC1000	2014	2	M	34
5	CC1000	2015	1	F	221
6	CC1000	2015	1	M	629
7	CC1000	2015	2	F	9
8	CC1000	2015	2	M	15
9	CC1000	2016	1	F	223
10	CC1000	2016	1	M	604

Figura 4.8: Parte del archivo `.csv` utilizado como dato de prueba para el cálculo de métricas

```

1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4 import pandas as pd
5 import numpy as np
6
7 def total_members(request):
8     df = pd.read_csv('metrics/data/estudiantes_DCC.csv', encoding='latin-1', usecols = ['ramo
9         ', 'ano', 'semestre', 'genero', 'cantidad'])
10
11     #dataframe ordenado por año
12     cantidad_por_año = df.groupby('ano')['cantidad'].sum().reset_index()
13
14     # Convertir los datos a una lista de diccionarios
15     data = cantidad_por_año.to_dict('records')
16
17     # Pasar los datos a la plantilla
18     context = {'data':data}
19
20     # Renderizar la plantilla
21     return render(request, 'metrics/total_members.html', context)

```

Código 4.7: Primera implementación de cálculo de prueba de métricas

En la línea 8 del Código 4.7, se carga un archivo `.csv` en un `dataframe` de `pandas`, seleccionando las columnas que corresponderían a los atributos registrados en los modelos de la sección 4.2.1, para este caso y tomando el archivo mostrado en la Figura 4.8, serían `ramo`, `año`, `semestre`, `genero`, `cantidad`. Luego, se procede a agrupar este `dataframe` por año, tal y como se muestra en la línea 12; este método proporcionado por la librería, suma todos aquellos datos por año. Desde la línea 14 a la 21, se procede a convertir los datos en otro formato para entregarlos a la vista y mostrarlos en la plataforma.

Adicional a esto y para poder visualizar los cálculos resultantes de forma temporal en la plataforma, se creó un archivo `.html` bajo el nombre de `total_members.html` dentro de la carpeta `templates/metrics` de la aplicación `metrics` y su implementación se puede ver en el Código 4.8. Lo relevante de aquí, es que tal y como se muestra en las líneas 14-21 del Código 4.7, pasa como parámetro un `context` que lleva los datos agrupados por años. En el Código 4.8 simplemente se invocan estos datos en las líneas 14-19 respectivamente. El resultado de esta primera prueba se muestra en la Figura 4.9.

```

1 {% extends "../base/base.html" %}
2 {% block title %} METRICAS {% endblock %}
3
4 {% block template %}
5     <h1>Total de miembros por año</h1>
6     <table>
7         <thead>
8             <tr>
9                 <th>Año</th>
10                <th>Cantidad</th>
11            </tr>
12        </thead>
13        <tbody>
14            {% for row in data %}
15                <tr>
16                    <td>{{ row.ano }}</td>
17                    <td>{{ row.cantidad }}</td>
18                </tr>
19            {% endfor %}
20        </tbody>
21    </table>
22 {% endblock %}

```

Código 4.8: html para la vista de prueba de cálculo de métrica

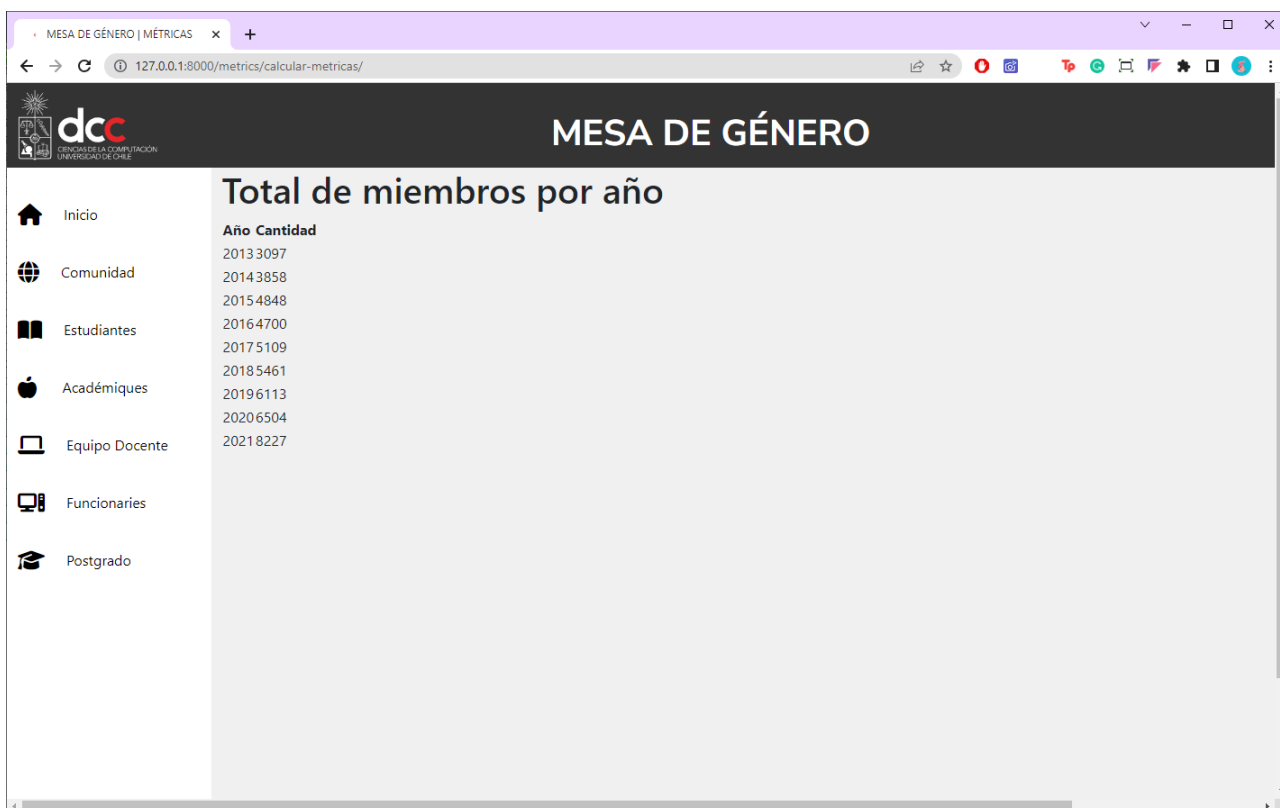


Figura 4.9: Primera iteración de cálculo de métricas

Con esta implementación, se demostró que las métricas podían ser calculadas haciendo uso de **pandas** y visualizadas en la plataforma. La idea en este punto era seguir con el cálculo de las demás métricas y luego implementar el módulo de generación de gráficos pero, para la verificación de los cálculos obtenidos, se tomó la determinación de ir trabajando paralelamente el módulo de cálculo junto con el de visualización; por lo tanto, se procedió a implementar la funcionalidad de generación de gráficos a partir de métricas e indicadores calculados.

4.3.2. Generación de gráficos

Para la generación de gráficos, tal y como se indicó en la sección 3.2.4, se hizo uso de **Plotly**; esta librería de visualización de datos interactiva, además de contar con diversidad de gráficos, puede ser integrada con **Dash**, tecnología desarrollada por **Plotly** que permite crear aplicaciones web interactivas para la visualización de datos. Según estas funcionalidades, se pensó que serviría para la construcción del dashboard para cada una de las entidades.

Se intentó entonces integrar **Dash** con **Django** y para ello, se hizo uso de una librería externa, **django-plotly-dash**⁴. Esta librería permitiría integrar aplicaciones **Dash** basadas en **Plotly** dentro de un proyecto desarrollado en **Django**. La razón por la cual no se optó por esta solución, fue porque la comunicación entre el módulo de carga de datos y la aplicación de **Dash** no funcionaba correctamente entre ellas. A partir de ello e indagando en la documen-

⁴<https://django-plotly-dash.readthedocs.io/en/latest/>

tación, se encontró que esta había sido actualizada tres años atrás⁵, por lo que seguramente las dificultades de comunicación entre los módulos pudo deberse a las versiones con la que se contruyó la librería `django-plotly-dash`, quedando descartada esta opción.

Siendo el uso de `Dash` una opción inviable para las necesidades del proyecto, se procedió a utilizar solamente `Plotly` sin ninguna integración externa. De esta manera, se buscó evitar futuros problemas relacionados con incompatibilidades entre versiones de otras librerías, además de `Plotly`, que podrían afectar la estabilidad y funcionalidad del sistema.

Retomando la implementación de la funcionalidad y de acuerdo al Modelo-Vista-Controlador que maneja `Django`, el uso de `Plotly` para la generación de gráficos debía ser mediante una vista en el archivo `views.py`. Dado que no se utilizaría una aplicación separada para abarcar este requerimiento, se decidió incluir las líneas de código referente a los gráficos en cada una de las vistas que calculaban las métricas e indicadores de cada uno de los estamentos. Paralelamente, se hizo la integración del módulo de carga de datos para dejar de cargar archivos de forma local. En el Código 4.9 se observa el desarrollo antes descrito.

```
1 from django.shortcuts import render
2 from core.models import Ramo
3 import plotly.express as px
4 from plotly.offline import plot
5
6 import pandas as pd
7
8
9 # funcion que calcula métricas y genera graficos para el modelo Ramo
10 def total_members(request):
11     data = Ramo.objects.all().values() # obtencion de registros desde tabla
12     df = pd.DataFrame(list(data)) # se conviertes los datos en un dataframe
13
14     # dataframe agrupado por ramo, año, semestre
15     cantidad_por_año = df.groupby(['ramo', 'año', 'semestre'])['cantidad'].sum().reset_index
16         ()
17
18     # utilizacion de plotly para crear grafico de lineas interactivo
19     fig = px.line(
20         x = [x for x in cantidad_por_año.año],
21         y = [y for y in cantidad_por_año.cantidad],
22         color=cantidad_por_año['ramo'],
23         width=800
24     )
25
26     fig.update_layout(
27         title="Total de alumnes por curso",
28         xaxis_title="Año",
29         yaxis_title="Cantidad",
30     )
31
32     chart = plot(fig, output_type="div")
33
34     # Pasar los datos a la plantilla
35     context = {'chart': chart}
36
37     # Renderizar la plantilla
38     return render(request, 'metrics/total_members.html', context)
```

Código 4.9: Implementación de gráfico con `Plotly`

⁵<https://github.com/GibbsConsulting/django-plotly-dash/blob/master/docs/introduction.rst>

En resumidas cuentas, la función `total_members` extrae datos de la tabla `Ramo` de la base de datos alimentada desde el módulo de carga de datos que recibe un archivo `.csv` y lo almacena de forma local en la plataforma y a partir de estos datos, crea un gráfico de líneas interactivo utilizando `Plotly`. Este gráfico es pasado como parámetro a la plantilla donde se renderiza en la plataforma web. En el Código 4.10 se muestra la plantilla `.html` mencionada y en la Figura 4.10 se ve el resultado de la implementación.

```
1 <!-- calcular_metricas.html -->
2 {% extends "../base/base.html" %}
3 {% block title %} METRICAS {% endblock %}
4
5 {% block content %}
6
7     {% autoescape off %}
8         {{ chart }}
9     {% endautoescape %}
10
11 {% endblock content %}
```

Código 4.10: Plantilla `.html` utilizada para la visualización del gráfico generado con `Plotly`

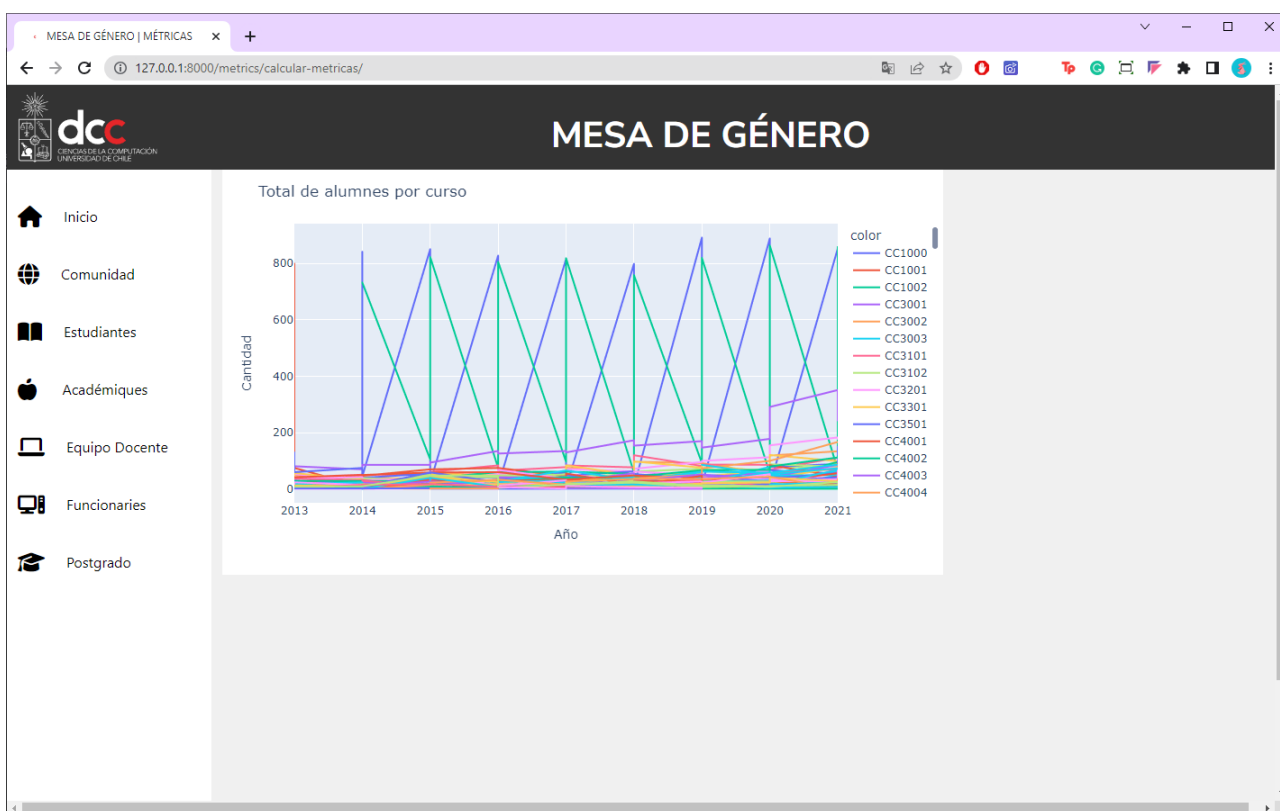


Figura 4.10: Gráfico generado utilizando `Plotly` y métricas calculadas relacionado a alumnos por ramo

A partir del gráfico mostrado en la Figura 4.10, se vio la necesidad de agregar filtros que permitan organizar de mejor manera la información mostrada, dado que a simple vista, tantas líneas generan confusión y no cumple con el fin de entregar información de forma clara. En este punto de la implementación y luego de verificar que las funcionalidades de carga de datos, cálculo de métricas e indicadores y la generación de gráficos funcionan correctamente entre sí, se procedió a desarrollar las vistas definitivas para cada uno de los estamentos, que incluyeran las métricas e indicadores correspondientes junto con gráficos y sus respectivos filtros.

El Código 4.11 muestra parte de la vista implementada para la sección Comunidad de la plataforma (que toma en cuenta a todos los integrantes pertenecientes al departamento), específicamente la generación de botones y filtros. Entendiendo el funcionamiento de estos botones, cada uno representa un gráfico de línea generado. Por ejemplo, en las líneas 3-10, se agrega el botón correspondiente al gráfico General que presenta los datos de la cantidad de integrantes en el departamento según género. En las líneas 7-8, el atributo `visible` dentro de `args`, se utiliza para proporcionar las actualizaciones que se deben realizar en el gráfico cuando se activa un botón en específico. Un caso claro de esta propiedad se puede observar en la línea 15, donde se establece como no visible (`False`) todos los elementos correspondientes a los géneros únicos y, al final se agrega un `True`, que significa que el último elemento que corresponde a la línea que representa el `Total`, se establece como visible, tal y como se muestra en la Figura 4.11(b).

Es importante mencionar que entre las líneas 18-24, se generan los filtros asociados a tipos de género que se registran en los datos; así, se asegura que las opciones disponibles vengan desde el archivo cargado y no se limita a un listado definido dentro de esta vista. El gráfico junto con sus filtros se pueden ver en la Figura 4.11.

```

1  # Botones de seleccion
2  buttons = []
3  buttons.append(
4      dict(
5          label="General",
6          method="update",
7          args=[{"visible": [True] * len(df["genero"].unique()) + [False],
8                "layout": {"title": {"text": f"Comunidad DCC General"}}}]
9      )
10 )
11 buttons.append(
12     dict(
13         label="Total",
14         method="update",
15         args=[{"visible": [False] * len(df["genero"].unique()) + [True], "title": f"
16               Comunidad DCC - Total"}]
17     )
18     for i, genero in enumerate(df["genero"].unique()):
19         button = dict(
20             label = str(genero),
21             method="update",
22             args=[{"visible": [False] * i + [True] + [False] * (len(df["genero"].unique()) -
23                   i - 1) + [False],
24                   "title": f"{genero}"}]
25         )
26         buttons.append(button)
27 fig.update_layout(
28     updatemenus=[
29         dict(
30             buttons=buttons,

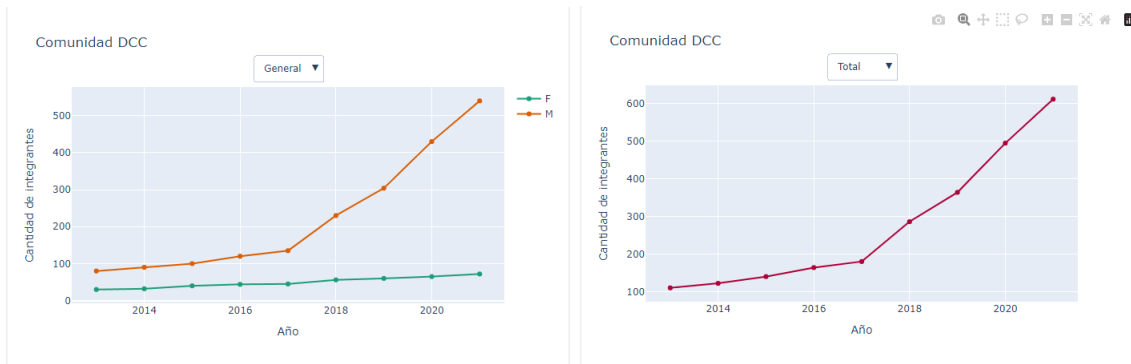
```

```

31     direction="down",
32     showactive=True,
33     x=0.5,
34     y=1.15,
35     xanchor="center",
36     yanchor="top"
37 )
38 ]
39 )

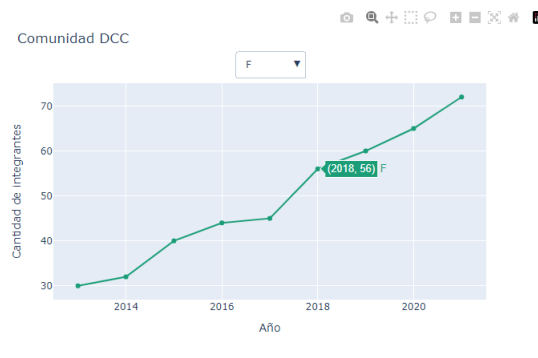
```

Código 4.11: Parte del código para la vista de Comunidad con filtros integrados en su gráfico



(a) Filtro General

(b) Filtro Total



(c) Filtro género femenino (F)

Figura 4.11: Gráfico de línea con sus respectivos filtros para la vista Comunidad

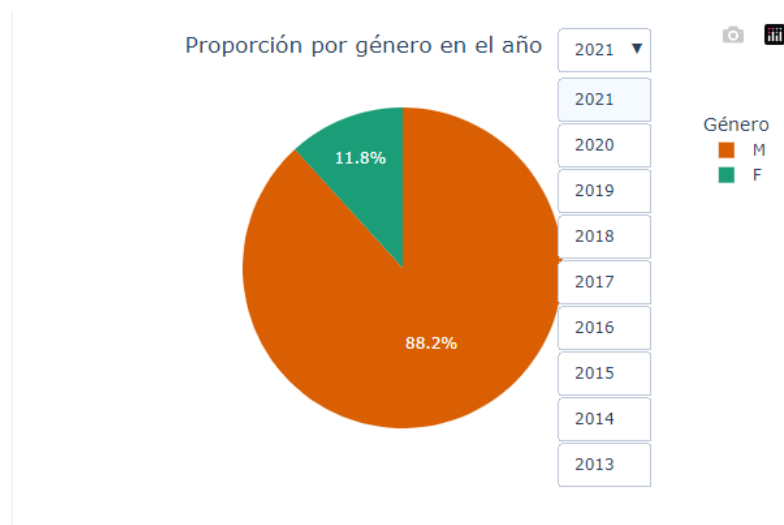
Ya teniendo todos los elementos necesarios para cada una de las interfaces en la plataforma, se procedió a calcular las métricas e indicadores, tomando como referencia el listado presentado en la sección 2.3.1, asociándolos a un tipo de gráfico para visualizar la información de forma más clara. Por ejemplo, en la Figura 4.11 se muestra un gráfico de línea para representar la cantidad de integrantes a lo largo de los años.

A continuación, se presentan los gráficos más utilizados para métricas e indicadores en cada de las secciones de la plataforma.

El gráfico utilizado para representar la cantidad de integrantes por género en cada año, fue un gráfico de torta con los filtros correspondientes por año, como el de la Figura 4.12 y la parte correspondiente a su implementación dentro de la vista en el archivo `views.py` dentro de la aplicación `metrics` se muestra en el Código 4.12.



(a) Gráfico de torta



(b) Despliegue de filtros por año

Figura 4.12: Gráfico torta que muestra cantidad de integrantes por género y por año

```

1 # Crear figura del gráfico de torta
2 anos = sorted(df["ano"].unique())
3 # Obtener el último año
4 ultimo_ano = max(anos)
5 # Se crea dataframe para los datos que se mostraran en el grafico de torta. En este caso
6 # son los integrantes agrupados por genero segun año
7 df_pie = df[df['ano'] == anos[(anos.index(ultimo_ano))]].groupby(['genero'])['cantidad'
8 ].sum().reset_index()
9 # se crea el grafico de torta con el dataframe
10 fig_pie = px.pie(df_pie, values='cantidad', names='genero', color='genero',
11                 color_discrete_sequence=colors)
12 # Agregar botones/filtros para cada uno de los años, asociando los datos con cada año
13 # correspondiente
14 buttons = []
15 # se toma el listado de años al reves para que se muestre el mas actual
16 for ano in reversed(anos):
17     df_ano = df[df['ano'] == ano]
18     labels = df_ano['genero'].tolist()
19     values = df_ano['cantidad'].tolist()

```

```

19 # cada boton generado por año se agrega al diccionario de botones
20 # para el filtro de grafico de torta
21 buttons.append(
22     dict(
23         label=str(ano),
24         method="update",
25         args=[{
26             "values": [values],
27             "labels": [labels]
28         }]
29     )
30 )

```

Código 4.12: Parte del código para el gráfico de torta

Para visualizar los indicadores calculados, se utilizó un tipo de gráfico disponible dentro de la librería Plotly para este tipo de datos. En el Código 4.13 se muestra la línea donde se realiza el cálculo de la proporción entre género femenino versus total de integrantes. En la línea 6, se realiza una combinación entre `df_female` y `df_total`, haciendo uso de la columna que contiene los años como pivote. En la línea 7 se crea una nueva columna llamada `proporcion` en `df_prop`, que contiene el cálculo de la proporción de género femenino para cada año. Esta proporción se calcula con la Fórmula 4.1.

$$\frac{\text{Suma total de género femenino por año}}{\text{Totales generales de cantidad para cada año}} \times 100 \quad (4.1)$$

```

1 # Dataframe con la cantidad total de integrantes por año
2 df_total = df.groupby(['ano'])['cantidad'].sum().reset_index()
3 # Dataframe con la cantidad de género femenino por año
4 df_female = df[df['genero'] == 'F'].groupby('ano')['cantidad'].sum().reset_index()
5 # Calcular la proporción de género femenino para cada año
6 df_prop = df_female.merge(df_total, on='ano')
7 df_prop['proporcion'] = (df_prop['cantidad_x'] / df_prop['cantidad_y'])*100

```

Código 4.13: Línea que corresponde al cálculo de proporción entre género femenino versus total de integrantes

El indicador generado se muestra en la Figura 4.13. Para el caso en que se seleccione el año de comienzo de los registros de datos (menor año), no se hace la comparativa y solo se muestra la proporción entre el género femenino y el total de integrantes de ese año, tal y como se ve en la Figura 4.13(b).

La implementación de este gráfico se muestra en el Código 4.14. Entre las líneas 3–6 se establece la condición del caso antes mencionado para cuando se trata del primer año registrado en los datos.

Proporción de género femenino por año

2021 ▼

Comparativa entre 2021 y 2020

11.76%
▼ -10.41%

Proporción de género femenino por año

2013 ▼

27.3%
-

(a) Indicador de proporción para género femenino para el año 2021

(b) Indicador de proporción para género femenino para el año 2013

Figura 4.13: Gráfico de indicadores de proporción de género femenino

```
1 for i, prop in enumerate(df_prop['proporcion']):
2     # se calcula el delta como la diff entre la proporción del año, con la del año anterior
3     if i > 0:
4         delta = df_prop['proporcion'].iloc[i-1]
5     else:
6         delta = prop
7     # crea un indicador para la proporción actual
8     indicador = go.Indicator(
9         mode = 'number+delta',
10        value=prop,
11        delta={'relative': True, 'reference': delta, "valueformat": ".2%"},
12        number={'suffix': '%'},
13        title={'text': f'Comparativa entre {df_prop["ano"].iloc[i]} y {df_prop["ano"].iloc[i-1]}',
14              'if i > 0 else f''},
15        visible=(i == len(df_prop['proporcion'])-1) # Hace visible solo el primer indicador
16    )
```

Código 4.14: Líneas de código que corresponden a la generación de gráfico de indicador

Si bien, en la mayoría de interfaces implementadas (secciones de la barra lateral) se muestran estos tres gráficos, existen variaciones en otras secciones, como por ejemplo la de Equipo Docente, que contiene un gráfico que muestra la cantidad de integrantes según tipo de docente, clasificado por género, tal y como se muestra en la Figura 4.14. En este gráfico se presentan los datos para el año 2021, donde cada barra representa un tipo de docente, junto con la cantidad de integrantes según género, diferenciado por colores.

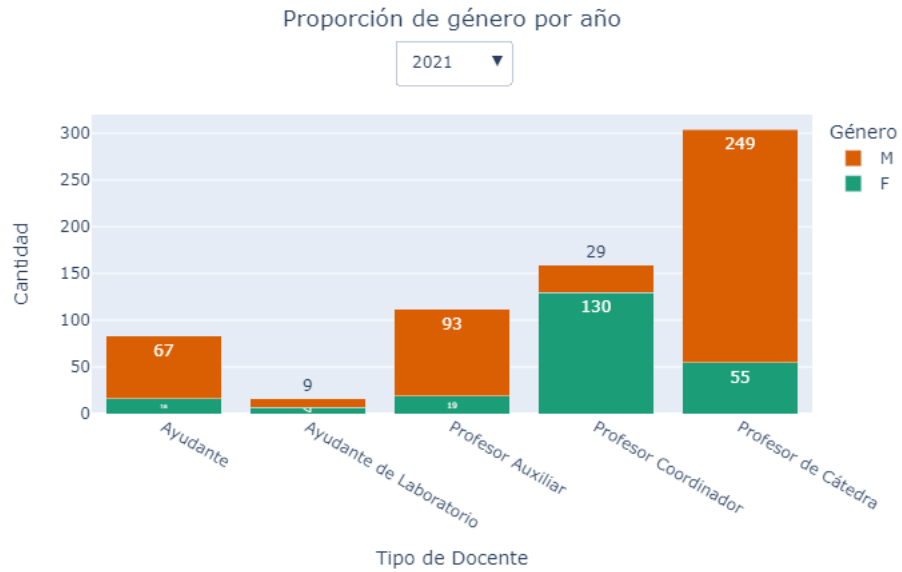


Figura 4.14: Gráfico de barra agrupado por género según tipo de docente

Finalmente, se procedió a implementar cada una de las vistas correspondientes a los estamentos Comunidad, Estudiantes, Académiques, Equipo Docente, Funcionaries y Postgrado. Un panel completo dentro de la plataforma se vería como en la Figura 4.15. Esta vista muestra el diseño final utilizado para el proyecto, la cual no cuenta con una sección Inicio, como aparecía en la Figura 3.5, dado que la herramienta en sí es para mostrar métricas e indicadores hacia usuarios que conforman la Mesa de Género del DCC (usuarios finales) y no una plataforma destinada (durante el desarrollo de este proyecto) para otro grupo que no tenga conocimiento o no forme parte de esta organización.

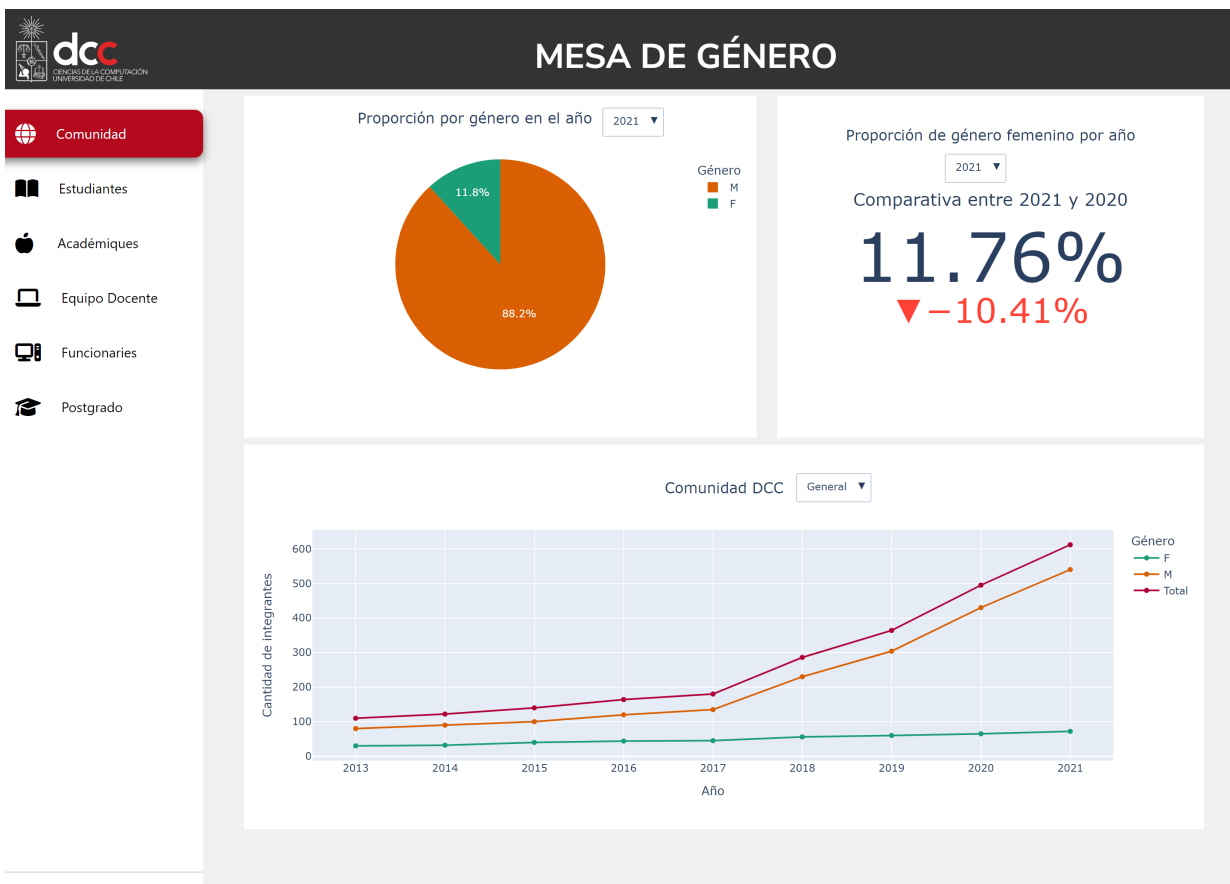


Figura 4.15: Panel de gráficos para sección Comunidad

4.4. Resumen

En este capítulo se explicó el proceso de implementación de la plataforma, comenzando con sus configuraciones respectivas, haciendo uso de **Django** como framework para la estructura de la plataforma web. Seguido de ello, como parte de la primera etapa, se realizó el diseño propuesto en la sección 3.2 en sus componentes principales, como lo es la barra superior y lateral. Con las configuraciones realizadas y la primera vista creada, se dio paso a la segunda etapa que contempló la creación de modelos siguiendo el esquema presentado en la Figura 3.4, junto con el desarrollo de la funcionalidad de carga de datos, proceso en el cual, se tomó la decisión de utilizar la librería **django-import-export**, luego de los problemas identificados a partir de la creación de una vista dedicada a la subida de archivos, que incluía la utilización de **Forms**, no siendo la solución que se buscaba en cuanto a extensibilidad. Finalmente, en la tercera etapa, se muestra todo lo relacionado al cálculo de métricas e indicadores, haciendo uso de la librería **pandas**, incluyendo la generación de gráficos a partir de **Plotly** y la integración de estas dos funcionalidades junto con la de carga de datos expuesta en la segunda etapa (sección 4.2.2).

Se adjunta además, el listado de métricas e indicadores que fueron implementados de forma efectiva en esta etapa final del proyecto.

- Métricas:
 - Total de miembros en el DCC.
 - Total de miembros por estamento.
 - Total por género.
 - Total por género por estamento.
 - Total de alumnos por código de curso.
 - Total de alumnos por género según código de curso.
- Indicadores:
 - Proporción según género (para cada estamento).

Este listado a diferencia con el propuesto en la sección 2.3.1, es reducido pero, es considerable como una base sólida dado que, es capaz de mostrar la situación del departamento en cuanto a género por etapa, ya sea por periodo de tiempo (entre años) o en un año en particular. Además, el carácter extensible del proyecto no limita a que a este listado se puedan seguir añadiendo métricas e indicadores en un futuro.

Capítulo 5

Validación

5.1. Pruebas funcionales

Para validar las funcionalidades de carga de datos, cálculo de métricas e indicadores y la visualización de datos a través de la generación de gráficos, estos fueron probados a medida de que se iba desarrollando el sistema, dado que las funcionalidades dependían entre ellas para el correcto levantamiento de la herramienta. Las pruebas funcionales se realizaron al momento de la carga masiva de datos al sistema, verificando que, a partir de los archivos subidos, se mostraban los gráficos correspondientes con los cálculos correctos en las vistas de la plataforma.

Parte de la verificación de los cálculos de métricas, se realizó comparando lo entregado por la herramienta, junto con los cálculos realizados en una hoja de cálculo. La comparación entre los resultados se puede apreciar en la Tabla 5.1, para el caso de la sección Comunidad, donde los resultados entregados por la hoja de cálculo coinciden con los entregados por la plataforma. Como la implementación de métricas fue la misma para cada uno de los estamentos, bastó con hacer una validación.

Año	Total Excel	Total Herramienta
2013	110	110
2014	122	122
2015	140	140
2016	164	164
2017	180	180
2018	286	286
2019	364	364
2020	495	495
2021	612	612

Tabla 5.1: Tabla comparativa con el total de miembros calculado en Excel versus herramienta

En cuanto al cálculo de indicadores, se realizó el mismo proceso anterior, utilizando la sección de Comunidad como caso de prueba, siguiendo la Fórmula 4.1. En la Tabla 5.2 se muestran los resultados obtenidos por la hoja de cálculo que son coincidentes a los entregados por la plataforma.

Año	Proporción Excel %	Proporción herramienta %
2013	27,3	27,7
2014	26,2	26,2
2015	28,6	28,6
2016	26,8	26,8
2017	25	25
2018	19,6	19,6
2019	16,5	16,48
2020	13,1	13,13
2021	11,8	11,76

Tabla 5.2: Tabla comparativa con la proporción de género femenino calculado en Excel y herramienta

5.2. Encuesta de Usabilidad y Satisfacción de Usuario

Es importante mencionar que si bien, se consideró la opción de utilizar la encuesta SUS (System Usability Scale)¹, este sistema y sus preguntas estaban enfocadas a la usabilidad que si bien, es importante validar este aspecto, se buscaba ahondar más en la utilidad de la información entregada en cada uno de los gráficos y vistas.

A partir de la aseveración anterior, se armó una encuesta que fuera capaz de abarcar preguntas relacionadas a usabilidad y utilidad de la información. En la siguiente lista se presentan las preguntas planteadas hacia los usuarios:

1. En una escala del 1 al 5, ¿considera de utilidad los datos presentados en la vista?
2. En una escala del 1 al 5, ¿qué tan claro y comprensible encuentra la visualización de los datos en la vista?
3. En una escala del 1 al 5, ¿qué tan fácil es para usted interpretar las tendencias y patrones en los datos presentados en la vista?
4. En una escala del 1 al 5, ¿qué tan satisfecho está con la cantidad de información que se muestra en la vista?
5. En una escala del 1 al 5, ¿qué tan valioso considera que es la vista en términos de proporcionar información relevante?
6. ¿Utilizaría este sistema en caso de necesitar métricas e indicadores enfocadas en género? Califique su respuesta en una escala de 1 al 5.

¹https://en.wikipedia.org/wiki/System_usability_scale

7. En una escala del 1 al 5, ¿qué tan rápido puede encontrar la información que necesita en la vista?
8. En una escala del 1 al 5, ¿qué tan atractivo y estéticamente agradable encuentra el diseño visual de la plataforma?

Donde las preguntas 1, 3, 4, 5 evalúan utilidad de la información, mientras que las restantes están relacionadas a la usabilidad. Se utilizó una escala ordinal de 5 puntos; entre mayor valor se otorgue en la respuesta, mayor es la satisfacción y/o utilidad percibida por parte del usuario.

Estas preguntas fueron planteadas en un formulario dividido en tres secciones, con tres videos distintos que mostraban distintas vistas de la plataforma. La primera parte consistía en la vista de Comunidad DCC, siendo la vista principal al ingresar a la plataforma. La segunda parte mostraba la sección de Equipo Docente que contenía gráficos un poco más complejos si se comparaba con los presentados anteriormente en la primera parte y, en la tercera parte se presentaba la vista de Funcionarios que contenía gráficos más generales.

Mediante el uso de Google Forms, se integraron las preguntas y sección adicional de comentarios, junto con los videos que mostraban el funcionamiento de la plataforma en las secciones de Comunidad, Equipo Docente y Funcionarios. El formulario fue repartido a integrantes de la Mesa de Género del DCC; participantes en las validaciones explicadas en la sección 2.3.1, funcionarios, alumnos, auxiliares, académicos, entre otros. De la convocatoria, se obtuvieron seis respuestas, las cuales se detallan en la Tabla 5.3. Cabe destacar que si bien la encuesta, iba dirigida hacia los integrantes de la Mesa de Género, fue de carácter anónimo, dado que se buscaba una visión desde la Mesa de Género y no una individual relacionada con el estamento al cual pertenecían, por ejemplo.

	P1	P2	P3	P4	P5	P6	P7	P8	Puntaje
U1	1	1	1	1	1	1	1	1	8
U2	5	4	5	5	5	5	5	5	39
U3	5	5	3	5	5	5	4	3	35
U4	5	5	5	5	4	5	5	5	39
U5	5	5	4	5	5	5	5	5	39
U6	5	5	3	4	5	4	5	5	36

Tabla 5.3: Resultados de la encuesta por parte de los usuarios

A partir de los resultados expuestos en la Tabla 5.3, cinco de los seis usuarios evaluaron la herramienta con puntos arriba de tres, interpretando que la plataforma es usable y su información es de utilidad. Es importante destacar que entre los cinco usuarios, todos otorgaron el puntaje máximo a la pregunta relacionada con la utilidad de los datos presentados en la vista. La pregunta que presentó más variación entre puntajes fue la número tres, que tenía que ver con la facilidad de interpretar tendencias y patrones en los datos presentados en la vista.

Entre los comentarios entregados por estos cinco usuarios, se tiene:

- *“A pesar que el manejo de información y datos es bastante, la simplificación de la misma es óptima y entendible para el lector, a lo igual que su comprensión”.*
- *“No queda muy claro qué significa el porcentaje negativo en la esquina superior derecha”,* haciendo alusión al gráfico de indicadores (ver Figura 4.15).
- *“Considero que esta vista está muy bien lograda, lo que no me gusta es que se tiene que scrollear para llegar al segundo gráfico, por lo que se puede perder, podrían estar dentro de elementos desplegables”.*
- *“Buena elección de colores”.*

En general, los comentarios recibidos proporcionaron una visión equilibrada de los aspectos positivos, como por ejemplo, a la comprensibilidad de la información, así como sugerencias constructivas en la navegación dentro de la vista.

Ahora, para entender al usuario que proporcionó el puntaje mínimo a cada una de las secciones, se hizo revisión de sus comentarios donde esclarece: *“El video pasa muy rápido y el gráfico es extremadamente pequeño no se ve ni se entiende nada, creo que para algo tan rápido e ilegible se debe acompañar de un locutor que explique cada gráfico como modo de apoyo esa es mi sugerencia”*, lo cual dio a entender que sus respuestas fueron a partir del video anclado a la encuesta, punto importante a mencionar dado que sus respuestas no fueron a partir de la información otorgada por la plataforma, si no por dificultades técnicas relacionadas a la visualización del video y forma en que se presentó este. La sugerencia debería ser considerada para futuras invitaciones a encuestas según se requiera.

En resumidas cuentas, se obtuvieron variados comentarios respecto a la experiencia de usuario en la plataforma, interpretados como oportunidades de mejora. La herramienta en general fue evaluada positivamente, considerando la entrega de información clara y la intención de le usuario en hacer uso de ella.

Capítulo 6

Conclusión

Como se hace mención en la sección 1.2, el objetivo de esta memoria fue diseñar e implementar un sistema que sea capaz de generar métricas e indicadores desde la perspectiva de género, a partir de la carga de datos como herramienta de apoyo a la Mesa de Género del DCC. En base a los objetivos específicos planteados, se hará un análisis junto con las tareas desarrolladas con el fin de determinar el cumplimiento de cada uno de ellos.

El primer objetivo específico enuncia: *Analizar datos recopilados de años anteriores y en base a ellos, determinar un listado de métricas e indicadores posibles desde la perspectiva de género*, abarcado durante el curso previo Introducción al Trabajo de Título, proceso que se detalla en el capítulo de Estado del arte. A partir de este análisis se propuso además, el modelo de datos a implementar para cada uno de los estamentos abarcados en esta memoria: Comunidad, Estudiantes, Académiques, Equipo Docente, Funcionaries y Postgrado.

Para el segundo objetivo específico: *Diseñar vistas del sistema para la presentación de métricas, indicadores y gráficos*, el diseño fue propuesto durante el segundo semestre del 2022, tomando elementos principales de otras plataformas relacionadas al departamento, como paleta de colores y distribución de contenido, como la barra superior y la lateral. La implementación final de la plataforma abarcó la mayoría de los elementos presentados en la propuesta (ver Figura 3.5) y si bien, la distribución de los gráficos no fue completamente fiel al *mockup*, cumplió con el objetivo de presentar métricas, indicadores y gráficos, explicada en el capítulo 4 de Implementación.

El tercer objetivo específico que requería: *Implementar funcionalidades de carga de datos, cálculo de métricas e indicadores y visualización de datos (gráficos) para el sistema*, fue cumplido a través de distintas etapas de desarrollo explicadas en el capítulo 4. Luego de las configuraciones iniciales y desarrollo de la primera vista, las etapas posteriores consistieron en el trabajo desarrollado para cada una de las funcionalidades de forma incremental; comenzando con la carga de datos, luego con el cálculo de métricas y finalizando con la generación de gráficos, que integrados, conforman la herramienta completa.

Por último, el cuarto objetivo relacionado a: *Validar usabilidad y experiencia de usuario del sistema desarrollado mediante test de usabilidad a usuarios finales*, fue cumplido aunque de forma parcial, dado que no se realizaron test de usabilidad donde un usuario fuera capaz de

interactuar con al plataforma de forma directa. Las validaciones se hicieron mediante videos y presentaciones en reuniones, dado que la plataforma web no fue desplegada a ningún servidor que permitiese a usuaries acceder mediante alguna url.

Pero dado que la herramienta ya se encuentra desarrollada, solo faltaría un proceso de despliegue. Por lo tanto y, considerando que se abordaron la mayoría de los objetivos específicos de forma completa, se concluye que se cumplió con el objetivo principal de *Diseñar e implementar un sistema que sea capaz de generar y graficar métricas e indicadores desde la perspectiva de género, a partir de la carga de datos para ser utilizado como herramienta de apoyo a la Mesa de Género del DCC.*

Aprendizajes obtenidos

En cuanto a tecnologías utilizadas, se aprendió de que es mejor delegar tareas específicas a librerías ya creadas, pero siempre teniendo en cuenta la documentación y por sobre todo, las actualizaciones sobre estas; por ejemplo, para el desarrollo de la funcionalidad de carga de archivos, se encontró una librería actualizada y que al ser integrada a la herramienta, optimizó tiempo y código.

Si bien, el uso de Plotly facilitó tanto la generación de gráficos a partir de el cálculo de métricas e indicadores, como la agrupación de `dataframes` por columnas para el cálculo de métricas, durante la marcha y como opinión personal, se encontró que la creación de filtros fue complicado, especialmente cuando se deseaba, por ejemplo, implementar dos filtros en un mismo gráfico.

Ante esto último, se reflexiona que quizá hubiese sido bueno hacer pruebas de concepto que abarcaran filtros personalizados con datos más complejos, como en el caso de querer filtrar por año y semestre.

De todo el proceso que significó el desarrollo de la memoria, se destaca la importancia de la organización. Gracias a que el trabajo fue dividido en tareas específicas y con revisiones periódicas junto a la profesora guía, el alcance del proyecto fue logrado, pese a que se presentaron situaciones que requirieron más tiempo de lo estipulado, no se vio afectada la fecha de finalización de la memoria. El trabajo adelantado además se tradujo en tiempo utilizado para avanzar en otras tareas, permitiendo cierta holgura y menos estrés al presentarse eventualidades que involucraran alguna pausa durante el desarrollo de la memoria.

Por último, comenzar un proyecto de cero es un proceso que contempla muchas etapas; fue importante establecer alcances logrables y acotados de acuerdo al tiempo para asegurar un resultado funcional que cumpliera con los objetivos. Destacar además sobre la importancia de la comunicación efectiva, dado que esta herramienta fue construida en conjunto junto con la Mesa de Género, a quien se agradece su cooperación y buena disposición. Lo mismo para la profesora guía, que gracias a sus comentarios y retroalimentación en etapas tempranas permitieron la finalización de un trabajo que la memorista, consideró suficiente y bien logrado.

Trabajo Futuro

Algo que se podría abarcar en un futuro, sería reestructurar el proyecto; crear una nueva aplicación para cada estamento y trabajar como módulo separado en cada una, las vistas que incluyen los cálculos y la generación de gráficos para luego unirlos en la aplicación principal de métricas e indicadores `metrics`. Incluir además un módulo para Egresados que no fue abarcado durante este desarrollo.

Otra opción interesante sería unificar o estandarizar la aplicación según los estándares definidos por el departamento del DCC, tanto en la obtención de datos, como en la visualización de información entregada, además de extender esta herramienta a otros departamentos dentro de la facultad.

Las sugerencias planteadas en los comentarios presentes en el capítulo 5, abarcando aquellas que tengan que ver con la mejora hacia la experiencia de usuario.

Bibliografía

- [1] Nicolás Andrés Caracci Veloso. Sistema de alimentación de información docente para el DCC, 2020.
- [2] Ignacio Arturo Vallejos Quinsacara. Plataforma de monitoreo de la docencia de pregrado del DCC, 2021.
- [3] Roberto Tapia Avedaño. Implementación de una herramienta de apoyo a la toma de decisiones en instituciones de educación superior en base a la información curricular de U-Campus, 2022.
- [4] Servicio de Información de Educación Superior. Brechas de género en educación superior 2020. <https://bibliotecadigital.mineduc.cl/handle/20.500.12365/16821>, 2021.
- [5] Times Higher Education and UNESCO-IESALC. Gender equality: How global universities are performing. https://www.iesalc.unesco.org/wp-content/uploads/2022/03/SDG5_Gender_Report-2.pdf, 2022.
- [6] UNESCO-IESALC. Global universities address gender equality but gaps remain to be closed. <https://www.iesalc.unesco.org/en/2022/03/08/global-universities-address-gender-equality-but-gaps-remain-to-be-closed/>.
- [7] Comisión Nacional de Acreditación. Guía para la autoevaluación interna de acreditación institucional universidades.
- [8] Jacob Nielsen. Why you only need to test with 5 users. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>, Marzo 2000.
- [9] José Miguel Pacheco Aguilera. Sistema de apoyo a la gestión docente del DCC, 2022.