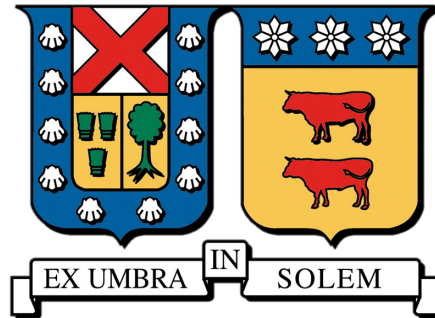


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



**“DESARROLLO DE ALGORITMOS DE
DISTRIBUCIÓN PARA AUTOMATIZAR LA
FORMACIÓN DE EQUIPOS EN EL PROGRAMA DE
MEMORIAS MULTIDISCIPLINARIAS EN LA USM”**

SANTIAGO ALFONSO LÓPEZ BURNS

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO
CIVIL TELEMÁTICO**

PROFESOR GUÍA: MARCOS ZÚÑIGA BARRAZA PhD
PROFESOR CORREFERENTE: MOHAMED ABDELHAMID PhD

OCTUBRE 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: DESARROLLO DE ALGORITMOS DE DISTRIBUCIÓN PARA AUTOMATIZAR LA FORMACIÓN DE EQUIPOS EN EL PROGRAMA DE MEMORIAS MULTIDISCIPLINARIAS EN LA USM

Nombre del candidato(a): Santiago Alfonso López Burns

Carrera / Grado: Ingeniería Civil Telemática

Campus: Casa Central Valparaíso ; Departamento:

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Marcos Zúñiga Barraza, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 08/10/2025

; Firma:

Estudiante o Candidato(a):

Fecha: 08/10/2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Este escrito está dedicado a las personas que me acompañaron de una u otra forma en mi paso por la educación superior. Doy gracias a mis tutores, compañeros y compañeras de trabajo y estudio, y por sobre todo a mis compañeros de memoria Javier Ahumada y Cristian Vega, amigos con los que decidimos embarcar en este desafío. Finalmente, agradezco a mis padres, por siempre tener más fe en mí de la que yo tengo en mí mismo, por ser un apoyo incondicional en todo este proceso y por seguir manteniendo el hábito de la sobremesa luego de todos estos años.

Resumen

Este informe presenta el diseño e implementación de una plataforma destinada a gestionar las postulaciones de estudiantes y automatizar la formación de equipos multidisciplinarios para el programa de memorias multidisciplinarias (PMM) en la Universidad Técnica Federico Santa María (USM). Usando principios de optimización y técnicas de asignación de recursos, la solución busca mejorar el rendimiento del proceso, garantizando equipos diversos y que consideren las preferencias individuales de los estudiantes. Se analiza el estado del arte, identificando herramientas y tecnologías existentes, y se propone un enfoque personalizado que combina la eficiencia de los algoritmos con la accesibilidad y relevancia para el contexto académico. Los resultados esperados incluyen una mayor satisfacción de los usuarios, reducción del tiempo requerido para la asignación y una mejora en la efectividad de los equipos formados. Este trabajo contribuye al desarrollo de soluciones tecnológicas adaptadas a las necesidades específicas del sector educativo.

Abstract

This report presents the design and implementation of a platform intended to manage student applications and automate the formation of multidisciplinary teams for the multidisciplinary reports program (PMM) at the Federico Santa María Technical University (USM). Using optimization principles and resource allocation techniques, the solution seeks to improve process performance, ensuring diverse teams that consider individual student preferences. The state of the art is analyzed, identifying existing tools and technologies, and a personalized approach is proposed that combines the efficiency of algorithms with accessibility and relevance for the academic context. Expected results include greater user satisfaction, reduction in time required for assignment, and improvement in the effectiveness of the teams formed. This work contributes to the development of technological solutions adapted to the specific needs of the educational sector.

Índice de figuras

1.1. Encuesta a estudiantes, postulaciones y formación de equipos PMM . . .	7
1.2. Encuesta a estudiantes, postulaciones y formación de equipos PMM . . .	8
1.3. Encuesta a estudiantes, postulaciones y formación de equipos PMM . . .	8
1.4. Encuesta a estudiantes, postulaciones y formación de equipos PMM . . .	9
1.5. Distribución de Postulaciones para los Desafíos del 2021	10
1.6. Distribución de Postulaciones para los Desafíos del 2022	10
1.7. Distribución de Postulaciones para los Desafíos del 2023	11
1.8. Distribución de Postulaciones para los Desafíos del 2024	11
1.9. Encuesta a estudiantes, postulaciones y formación de equipos PMM . . .	12
1.10. Carreras participantes para el año 2021.	13
1.11. Carreras participantes para el año 2022.	13
1.12. Carreras participantes para el año 2023.	14
1.13. Carreras participantes para el año 2024.	14
1.14. Campus participantes para el año 2021.	15
3.1. Esquema general de la solución	30
3.2. Diagrama de contexto	33
3.3. Diagrama de Arquitectura	34
4.1. Resultados: Algoritmo Original	49
4.2. Resultados: Algoritmo Propuesta	50

Índice general

1. Introducción	5
1.1. Problema a resolver	5
1.2. Acercamiento a la solución	17
1.3. Objetivos	17
1.3.1. Objetivo General	17
1.3.2. Objetivos Específicos	18
1.4. Estructura	19
2. Estado del Arte y de la Técnica	21
2.1. Tecnologías existentes	21
2.2. Marco teórico	22
2.2.1. Algoritmos para la problemática de “Timetabling”	22
2.3. Discusión	27
3. Desarrollo de la solución	29
3.1. Estructura general de la solución	29
3.2. Algoritmo Original	35
3.2.1. Prototipo inicial	35
3.2.2. Finalización del algoritmo	37
3.3. Algoritmo Propuesta	39
3.3.1. Prototipo inicial	39
3.3.2. Finalización del algoritmo	44

4. Resultados	47
4.1. Métricas de evaluación	47
4.2. Información entregada por algoritmos y muestra de evaluación según métricas	49
5. Conclusiones	51

Capítulo 1

Introducción

1.1. Problema a resolver

En la Universidad Técnica Federico Santa María (UTFSM), existe el Programa de Memorias Multidisciplinarias (PMM), que se enfoca en el desarrollo de proyectos para entregar productos y/o soluciones con la creación de grupos de trabajo, estos con la particularidad de que los alumnos de dicho grupo sean de distintas carreras, para así llevar a cabo propuestas de solución más robustas a los problemas que se presenten. Esta metodología de trabajo está implementada en los campus Casa Central y San Joaquín, y se está usando como método para que los alumnos de Ingeniería Civil Telemática, en primer lugar, postulen a su título profesional. A su vez, se ha propuesto esta metodología de trabajo para otras carreras, siendo como ejemplos Construcción Civil, Ingeniería en Diseño de Productos, Ingeniería Industrial, entre otros, con el objetivo de generar los equipos de carácter multidisciplinario que busca el programa. Sin embargo, Telemática es la única carrera que, en su malla curricular, tiene a PMM como su opción de trabajo por defecto, lo cual implica que, a menos que el estudiante decida realizar una investigación personal, este tendrá que ser parte de esta metodología de trabajo, mientras que para el resto de carreras que participan del proceso, es netamente resultado de la motivación de alumnos no telemáticos a ser parte del proceso de memorias.

Para la formación de los equipos de trabajo, se tiene acceso a la página web “desafios-tecnologicos.usm.cl”, en la cual los estudiantes tienen oportunidad de leer y postular a un máximo de 3 desafíos de todos los que se proponen, y cada postulación tiene asociada una prioridad, es decir, indica el nivel de interés y/o ganas del estudiante a trabajar en el desafío en particular. Los desafíos, por su parte, son levantados tanto por la misma UTFSM como por entidades/organizaciones varias, cada uno con sus descripciones, carreras afín, requerimientos y proyecciones, así como grado de innovación, y que es la información que los alumnos tienen a disposición para tomar sus decisiones de postulación.

Lamentablemente, este proceso tiene una serie de complicaciones, siendo la primera de estas, el hecho de que la formación de equipos no está automatizada, y para un problema de estas características, el desarrollo manual resulta muy tedioso y atareado. También, como se está trabajando con preferencias de estudiantes, es posible encontrarse con distribuciones no homogéneas de postulantes por desafío, dejando así proyectos sin postulantes y a su vez desafíos saturados. Esto se puede deber a que la gran mayoría de alumnos que participan son de una carrera en particular, por lo que proyectos muy enfocados en su área, así como la familiaridad de los postulantes con los conceptos y tecnologías requeridas para los desafíos, pueden resultar más tentadores al momento de escoger.

Por otro lado, existen aspectos técnicos a tener en cuenta, como lo son el campus/sede a la que el alumno pertenece, para no tener problemas muy grandes en cuanto a la organización del equipo, y por sobre todo, el hecho de que se quieren formar equipos de carácter multidisciplinario, limitando así la distribución de alumnos en los distintos desafíos, dificultando aún más el proceso.

Entonces, se tendría un problema complejo, en donde se identifica una problemática que vale la pena resolver, sobre todo teniendo en cuenta que el número de participantes en el proceso aumenta con cada iteración del programa de memorias.

Un claro ejemplo de esto ocurrió este año 2024, ya que fue la primera versión del programa donde los estudiantes de Telemática en el campus San Joaquín se unían al proceso, y en donde se presencié el aumento de dificultad en la distribución de desafíos, así como la frustración y descontento tanto de los usuarios que estuvieron a cargo de manejar la creación de equipos como de los alumnos en general.

Para validar los supuestos, se realizó una encuesta que busca recopilar información sobre cómo los estudiantes del proceso PMM 2024 experimentaron la jornada de formación de equipos, para así tener un mejor entendimiento del contexto actual en la que se desenvuelve el problema, y con la cual se obtuvo la siguiente información relevante, en forma de gráficos de torta:

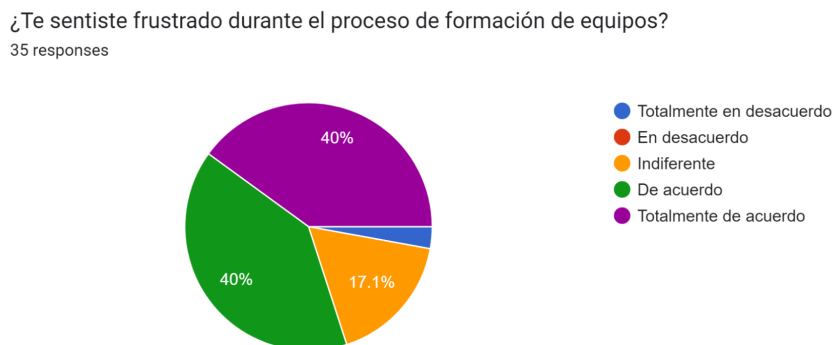


Figura 1.1: Encuesta a estudiantes, postulaciones y formación de equipos PMM

Solo si participaste únicamente por zoom, ¿te sentiste participe/escucharon tus solicitudes?
12 responses

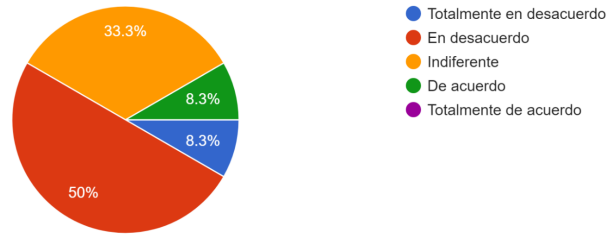


Figura 1.2: Encuesta a estudiantes, postulaciones y formación de equipos PMM

¿Sientes que el proceso de formación de los equipos fue eficiente?
35 responses

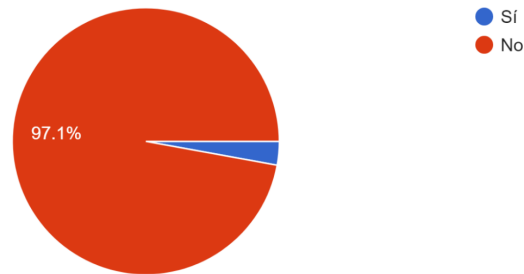


Figura 1.3: Encuesta a estudiantes, postulaciones y formación de equipos PMM

Como se puede apreciar en los resultados de la encuesta, en la Figura 1.1, aproximadamente el 67 % del total de estudiantes que respondieron la encuesta y se encuentran cursando el programa en 2024, se presenta un gran descontento/frustración en el proceso de formación de equipos, así como la falta de escucha por parte de las personas que participaron de forma remota el día de la sesión, denotada en la Figura 1.2. También, en la Figura 1.3, existe una clara mayoría, que percibió una baja eficiencia en el proceso de formación de equipos en general.

De los desafíos existentes, ¿Cuántos te gustaron/interesaron?
35 respuestas

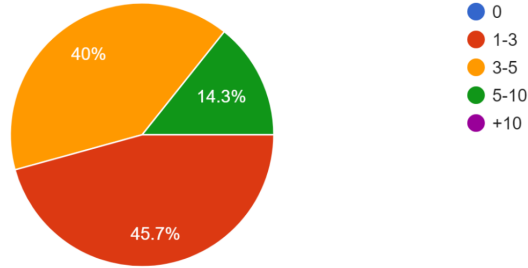


Figura 1.4: Encuesta a estudiantes, postulaciones y formación de equipos PMM

Otro punto importante que hay que destacar, es que a pesar de la gran cantidad de desafíos, donde los estudiantes tuvieron la oportunidad de escoger entre 38 disponibles, en la Figura 1.4 muestra que el 46 % de las personas encuestadas encontró de su interés no más de 3 desafíos, mientras que la otra mayoría no supera las 5 opciones de interés, a pesar de que el año 2024 es el año que ha tenido más desafíos disponibles, evidenciando que muchos de los temas de desafíos probablemente no fueron de interés para ninguno de los participantes.

Esto también se puede ver reflejado en la distribución de las postulaciones en las distintas versiones del programa, en las Figuras 1.5, 1.6, 1.7 y 3.3 se aprecia que la distribución de las postulaciones no es homogénea, donde se saturan las postulaciones en muy pocos desafíos de los totales.

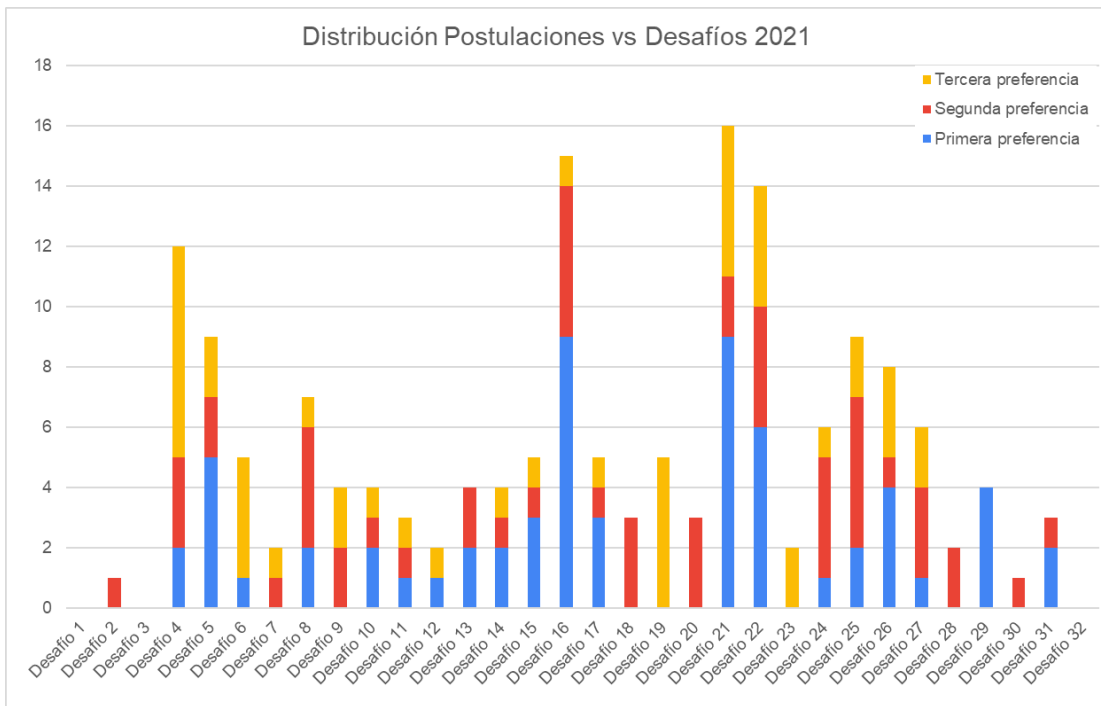


Figura 1.5: Distribución de Postulaciones para los Desafíos del 2021

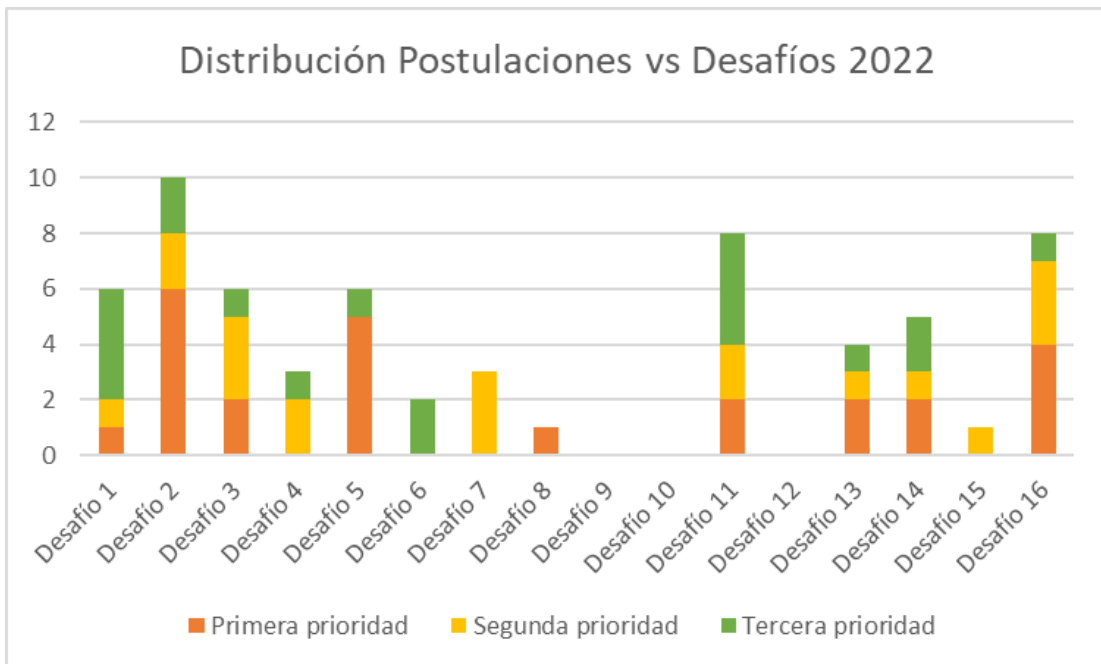


Figura 1.6: Distribución de Postulaciones para los Desafíos del 2022

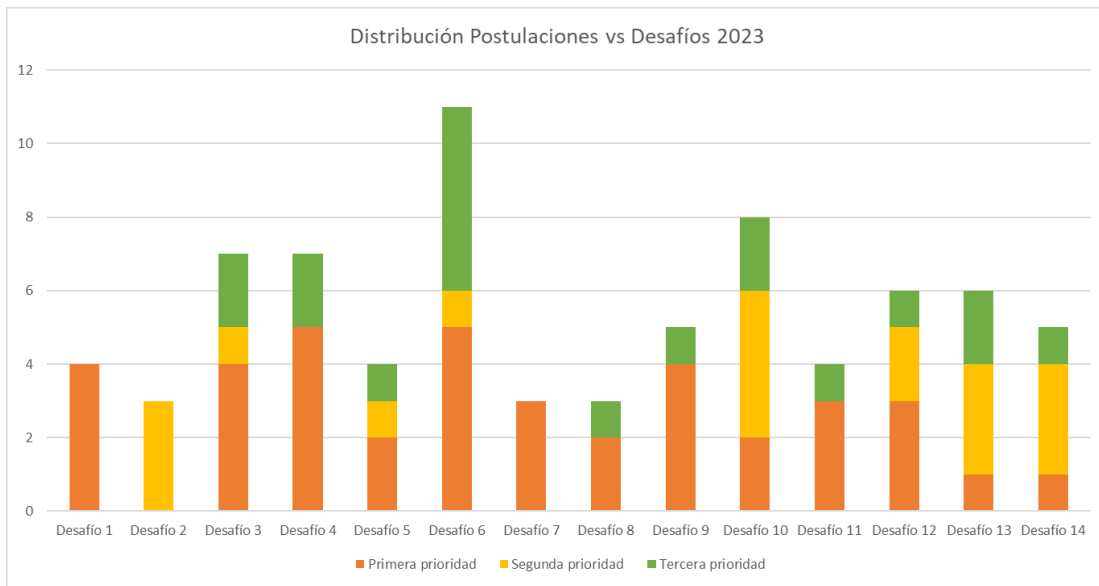


Figura 1.7: Distribución de Postulaciones para los Desafíos del 2023

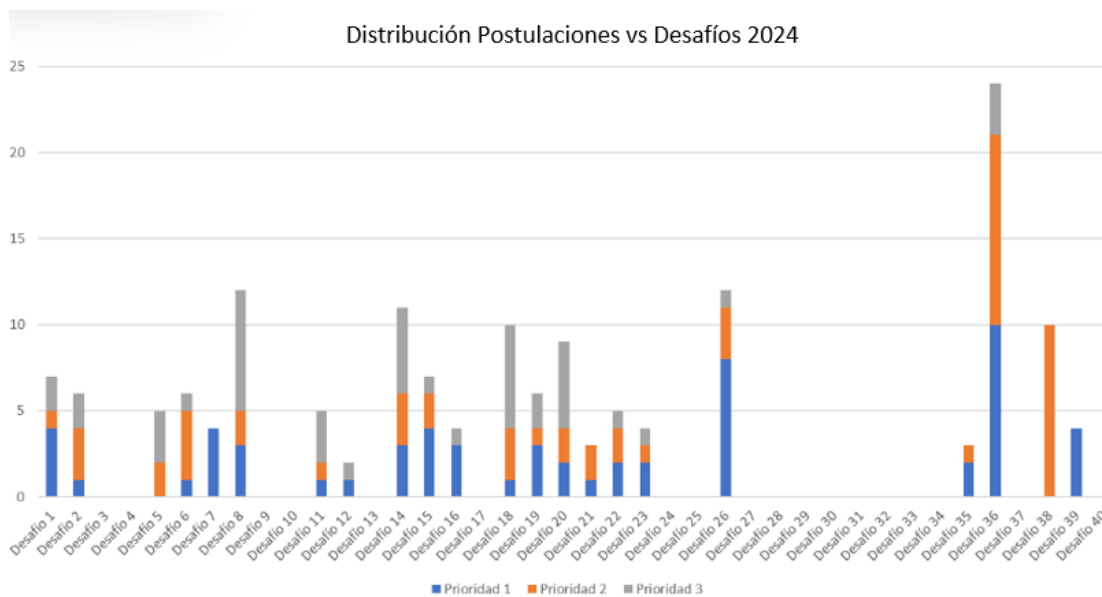


Figura 1.8: Distribución de Postulaciones para los Desafíos del 2024

Finalmente, hay que destacar que el proceso de distribución de desafíos actual, a pesar de todo lo mencionado anteriormente, en la Figura 1.9 se demuestra que logró que un 77 % de los encuestados quedaran seleccionados en una de sus 3 postulaciones previas al proceso de creación de equipos.

¿Lograste quedar en alguna de tus postulaciones?

35 respuestas

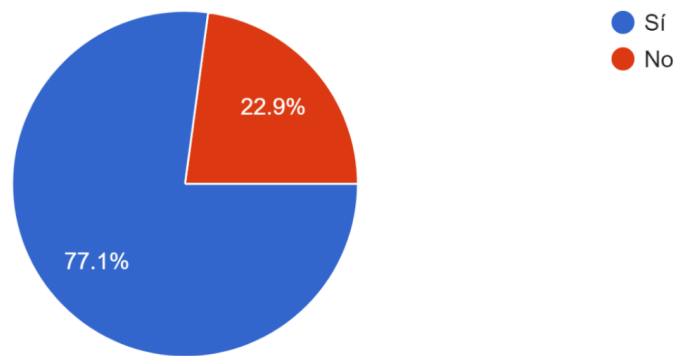


Figura 1.9: Encuesta a estudiantes, postulaciones y formación de equipos PMM

En el contexto de realizar equipos multidisciplinarios, es de interés analizar la cantidad de estudiantes que participan y sus características. La posibilidad de formar equipos con capacidades distintas es una oportunidad única que entrega el Programa de Memorias Multidisciplinarias. En sus distintas versiones han participado múltiples carreras, en las Figuras 1.10, 1.11, 1.12 y 1.13 destaca la activa participación de la Carrera de Ingeniería Civil telemática, siendo la carrera que más estudiantes aporta al programa, aunque también han existido versiones remarcables por su heterogeneidad, en especial para el año 2021.



Figura 1.10: Carreras participantes para el año 2021.

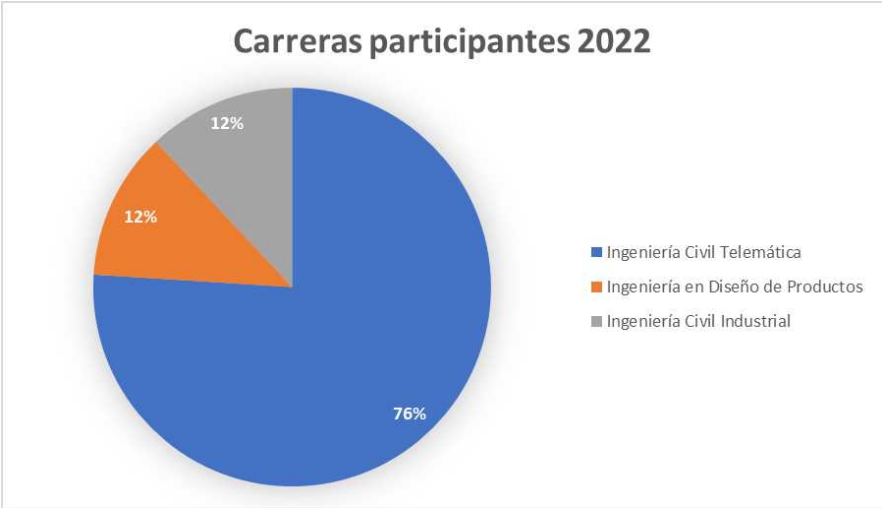


Figura 1.11: Carreras participantes para el año 2022.

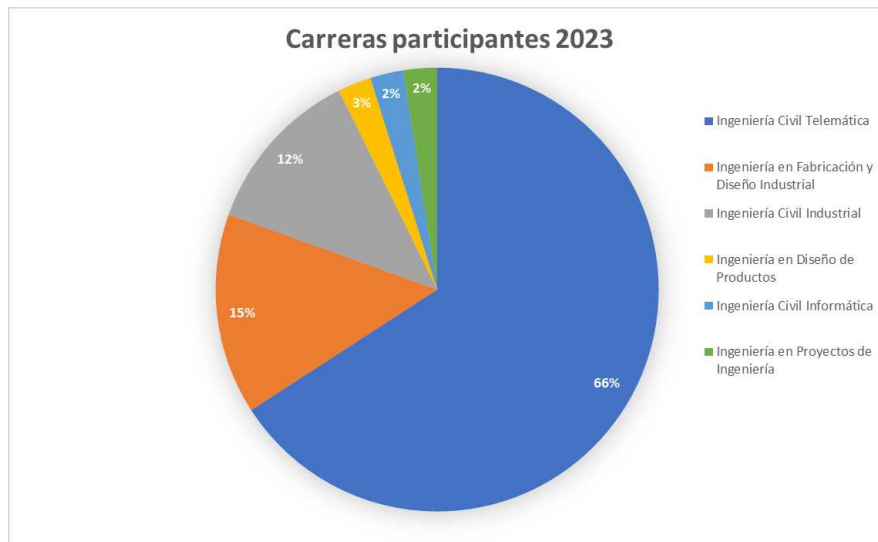


Figura 1.12: Carreras participantes para el año 2023.

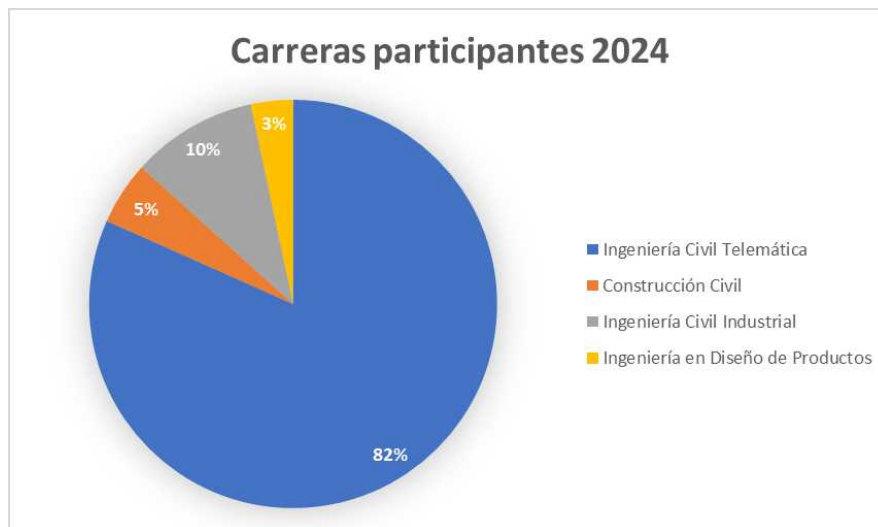


Figura 1.13: Carreras participantes para el año 2024.

El Programa de Memorias Multidisciplinarias se realiza en múltiples campus, existiendo la posibilidad de equipos multidisciplinarios que no solo junte estudiantes de distintas carreras, sino también de distintos campus. Su distribución puede apreciarse en la Figura 1.14.

Distribución Campus 2021

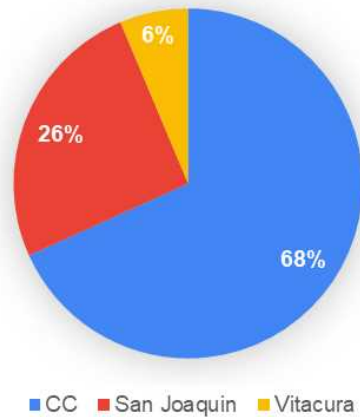


Figura 1.14: Campus participantes para el año 2021.

Toda esta información indica que existen varios enfoques a tener en cuenta al momento de generar los equipos para el programa de memorias, y es un aspecto que vale la pena analizar.

A raíz de que las postulaciones no se distribuyen de forma homogénea, esto probablemente producido por el porcentaje de participación de Ingeniería Civil Telemática en relación a otras carreras en el programa, se presenta un aumento de dificultad en relación a la formación de equipos que, además de promover la formación de los mismos con estudiantes de varias carreras, cumplan con las preferencias de las postulaciones y preferencias de los estudiantes.

Por otro lado, el manejo de las postulaciones de los estudiantes de forma manual, sin un medio estandarizado con el cual gestionar la formación de los equipos multidisciplinarios también presenta una dificultad a tener en cuenta, y que a su vez puede aportar a la generación de un proceso frustrante y tedioso, como se muestra en las gráficas anteriormente presentadas.

Con esto, se puede formular una hipótesis que ponga a prueba las observaciones generadas por la encuesta: **La implementación de una plataforma que gestione las postulaciones de estudiantes y automatice la formación de equipos multidisciplinarios**

mejorará el rendimiento del proceso, mediante la reducción del tiempo empleado en la formación de equipos y entregando un balance entre las preferencias de los estudiantes junto con las necesidades propias del programa de memorias, apoyando así en el crecimiento constante del programa.

La hipótesis presentada permite la formulación de preguntas que conduzcan el trabajo a desarrollar y que, a su vez, estructuran el desarrollo del escrito. Preguntas a considerar son:

- ¿Qué requisitos técnicos debe manejar la plataforma para poder gestionar correctamente las postulaciones y generar automáticamente los equipos multidisciplinares?
- ¿Qué técnicas son adecuadas para abordar un problema de este estilo?
- ¿Qué métricas pueden ser usadas para evaluar el rendimiento de las distribuciones automatizadas?
- ¿Qué criterios son necesarios de considerar para garantizar una distribución de estudiantes efectiva y de calidad?
- ¿En qué medida se ve afectada la satisfacción de los estudiantes al usar métodos que automaticen la generación de equipos?

Estas nuevas problemáticas permiten generar una estructura clara sobre el trabajo a realizar, y son respondidas en distintos puntos a lo largo de esta tesis, abarcando así puntos clave con los cuales asegurar la construcción de una solución efectiva para el problema presentado.

1.2. Acercamiento a la solución

La resolución del problema se centró en desarrollar un sistema que automatice la formación de equipos mediante algoritmos y aprendizaje por reforzamiento, optimizando la distribución de estudiantes según criterios del programa. Además, mejora el servicio web con una interfaz más intuitiva, incorporando nuevas funcionalidades y garantizando un manejo eficiente de solicitudes, asegurando un proceso ágil, transparente y de calidad para los usuarios.

1.3. Objetivos

1.3.1. Objetivo General

Dada la dificultad introducida en la sección 1.1, se tiene como objetivo general: **Mejorar la experiencia del proceso de memorias multidisciplinarias (PMM) a través de la formación automatizada de los equipos de desafíos, asegurando un proceso más rápido y transparente que el actual**, atacando puntos claves que conforman el programa PMM, tales como la distribución de estudiantes, proceso a automatizar mediante el uso de algoritmos y herramientas de aprendizaje por reforzamiento, así como el apartado del servicio web, cambiando varios puntos del mismo, implementando una interfaz más intuitiva y amigable, que entregue mejoras en cuanto a la calidad de uso, visualización, implementación de nuevas funcionalidades y manejo correcto de peticiones a interacciones de la plataforma.

1.3.2. Objetivos Específicos

Para este informe en particular, se van a explicar los procedimientos que se llevaron a cabo para el planteamiento, versión inicial y final implementación de los algoritmos de distribución que formaron parte del prototipo funcional, finalizando la etapa de PMM. Estos algoritmos, así como las implementaciones de modelos de aprendizaje por reforzamiento, conforman las opciones de solución que se terminaron desplegando en la plataforma web, para ser ocupadas por los usuarios administradores y de esta forma solicitar la formación de equipos para el programa.

- **Generar un algoritmo inicial, que logre distribuir a los estudiantes en los equipos de trabajo teniendo en cuenta las postulaciones y el tamaño máximo de equipos.**

Obtener la estructura principal del algoritmo de distribución, donde se realiza una selección secuencial (no aleatoria) del orden de los estudiantes a asignar, que hace uso de las preferencias contenidas en las postulaciones de los estudiantes y teniendo en cuenta la restricción del número máximo de estudiantes por equipo multidisciplinario.

- **Generar un algoritmo multicriterio para la distribución de estudiantes, que logre tener en cuenta las preferencias de postulación, las restricciones asociadas y el factor de aleatoriedad.**

Desarrollar un algoritmo que realice distribuciones tomando en consideración las preferencias de estudiantes, que cumple con las restricciones de tamaño de equipos así como el factor de multidisciplinariedad de los mismos, escogiendo estudiantes de manera aleatoria y que entregue una distribución final enfocada en generar la mayor cantidad de equipos posible.

- **Validar el correcto funcionamiento del algoritmo con datos reales.** Simular distribuciones para casos reales de postulación con tal de analizar cómo se comporta el algoritmo en base a la calidad de las distribuciones que entregue.

- **Comprobación del correcto funcionamiento de los algoritmos en la plataforma web.**

Una vez se tienen los algoritmos correctamente implementados y conectados al sistema general (en este caso, el prototipo funcional), se comprueba la correcta recepción de los datos de la plataforma para ser ocupados en el proceso de distribución y entregar los equipos resultantes.

Esta memoria se enfoca en documentar el aporte al desarrollo de los primeros 3 objetivos específicos mencionados, ya que, al haber completado la propuesta de solución en conjunto, existen análisis y otros aportes que escapan del enfoque puntual del informe.

1.4. Estructura

El resto del contenido del informe de estructura de la siguiente manera:

- **Capítulo 2: Estado del Arte y la Técnica**

Capítulo donde se abordarán investigaciones, técnicas, estrategias y conceptos relevantes en el mundo de los problemas de distribución.

- **Capítulo 3: Desarrollo de la solución: Algoritmos**

Lugar en donde se entrará en detalle al proceso de desarrollo de los algoritmos de distribución, entregando el esquema general de cada uno, su manejo de restricciones y los criterios de calidad, así como las restricciones que eventualmente indican si la distribución realizada es o no satisfactoria.

- **Capítulo 4: Resultados**

En esta etapa se evalúan los resultados obtenidos de las distribuciones, se definen los métodos con los cuales se busca calificar el resultado obtenido, atacando varias vertientes del problema a resolver, entregando así una vista general de la solución que se entregue.

■ **Capítulo 5: Conclusiones**

Se finaliza con una discusión sobre lo implementado y el impacto que tuvo en el problema propuesto así como la dirección del trabajo futuro con respecto a los algoritmos y la solución en general.

Capítulo 2

Estado del Arte y de la Técnica

Esta sección está dedicada a entender cómo se han abordado problemas similares en la actualidad, la manera en que se generan nuevas soluciones al respecto, analizando la competencia tecnológica existente y determinando las herramientas seleccionadas para su uso en este contexto.

2.1. Tecnologías existentes

El problema de asignación de recursos ha dado pie a la generación de diversas tecnologías para su resolución. En la actualidad, soluciones empresariales son destacables, como:

- Google OR-Tools [1]: Un conjunto de bibliotecas de optimización usadas para tareas como la asignación de tareas o el ruteo de vehículos.
- IBM ILOG CPLEX Optimizer [2]: Herramienta usada para resolución de problemas de programación lineal y entera, orientada a proyectos a gran escala con restricciones complejas.
- CATME [3]: Sistema de asignación automática implementado en varias instituciones educativas, utiliza algoritmos de optimización para todo tipo de tareas,

teniendo como ejemplo la formación de equipos de trabajo en ambientes escolares.

- Heurísticas personalizadas: Opción elegida por varias empresas, donde desarrollan soluciones internas haciendo uso de heurísticas presentes en la literatura, como lo son por ejemplo los algoritmos genéticos, y presentan soluciones personalizadas a problemas puntuales, que no necesariamente presenten una solución generalizada en la industria.
- OptaPlanner [4]: Sistema basado en inteligencia artificial, emplea heurísticas avanzadas para abordar problemas de asignación en tiempo real, siendo adaptativo y responsivo para restricciones dinámicas.

Las tecnologías presentadas han sido ampliamente ocupadas en la industria para el contexto de la distribución de recursos y formación de equipos, asignación de horarios y estrategias de planificación. No obstante, la implementación de cada uno de estos enfoques tiene distintos desafíos en cuestión de costos, complejidad técnica, compatibilidad de integración con las tecnologías de las empresas, personalización en situaciones puntuales, entre otros, lo que generalmente motiva, al menos en primera instancia, a evaluar el desarrollo de soluciones internas adaptadas al contexto puntual del problema.

2.2. Marco teórico

2.2.1. Algoritmos para la problemática de “Timetabling”

Para este apartado, se realizó una búsqueda sobre las tecnologías ocupadas para problemas de tipo “Timetabling” [5], concepto referente a problemas de asignación de horarios para organización de eventos en determinados periodos de tiempo. La definición dada se puede ver reflejada en el problema a resolver, ya que si bien no se realiza una distribución de horarios en sí, el desafío implica automatizar un proceso de distribución de recursos (en este caso, estudiantes y desafíos de memoria) teniendo en cuenta

parámetros, restricciones y casos especiales (i.e: prioridad en las postulaciones, cantidad máxima de estudiantes por desafío y situaciones donde un estudiante en particular debe sí o sí pertenecer a un desafío específico).

En primer lugar, se buscó sobre algoritmos que trabajasen en este tipo de contextos, y se encontró que, para una gran variedad de ambientes académicos/escolares, se han planteado soluciones que buscan reducir el tiempo empleado en los problemas de tipo “Timetabling”, así como implementación de herramientas que permitan trabajar con la complejidad de este tipo de contextos de una manera mucho más asistida para el usuario encargado de la distribución de recursos.

Entre las tecnologías y/o métodos que se mencionan en los documentos encontrados, se destacan los conceptos de “Hill Climbing”[6], “Algoritmo Genético” [7], “Sistema Max-Min” [8], “Tabu Search” [9] y “Método de MonteCarlo” [10], los cuales se definen a continuación:

- **Hill Climbing:** Estrategia basada en optimizaciones locales, siguiendo una dirección de ascenso/descenso a partir de su posición y variando una única variable en cada iteración.
- **Algoritmo Genético:** Técnica de búsqueda y optimización basada en los principios de la evolución natural, en donde se parte con una cantidad inicial de soluciones posibles (en este contexto llamadas individuos), y en donde, después de cierta cantidad de iteraciones, se seleccionan ciertos individuos con los cuales crear una nueva cantidad inicial de soluciones, con tal de seguir iterando, mejorando las soluciones (calidad del individuo) con cada iteración en base a una función objetivo.
- **Sistema Max-Min:** Usado para tomar decisiones en condiciones de incertidumbre o conflicto, es un sistema que implementa el concepto de “minimizar el máximo” o “maximizar el mínimo” con tal de obtener el mejor resultado posible en situaciones no favorables (en el peor caso).

- **Tabu Search:** Algoritmo de optimización creado para resolver problemas de búsqueda combinatoria, haciendo uso de búsqueda local y un sistema de memoria, con el objetivo de no caer en ciclos repetitivos o máximos/mínimos locales.
- **Método de MonteCarlo:** Método ocupado para problemas complejos en cuanto a su dimensionalidad o naturaleza, mediante la generación de muestras aleatorias con las cuales realizar una gran cantidad de iteraciones, y comparar los resultados obtenidos de cada iteración.

Se tienen también la descripción de los componentes esenciales que participan en un problema de optimización, y que son acordes al contexto del problema a resolver, puesto que, en esencia, el problema de Timetabling es un problema de distribución de recursos. Con esto, se define:

- **Parámetros:** Valores constantes o datos fijos del problema, que no cambian durante el proceso de optimización y definen el entorno de condiciones y posibilidad de las soluciones (el espacio de acción para llegar a la solución).

Para el contexto dado, se tienen los siguientes parámetros (en este caso, conjuntos):

- E = Conjunto de desafíos
- S = Conjunto de estudiantes

Estos conjuntos son los que permitirán al algoritmo iterar, con el objetivo de encontrar soluciones que sean adecuadas al problema a resolver.

- **Variables:** Son los elementos que varían y se ajustan al momento de resolver la problemática, y representan las decisiones que se toman en relación al uso de los recursos disponibles en cada paso del algoritmo.

Existen varios tipos de variables, siendo los más comunes:

- Variables continuas: Pueden tomar cualquier valor dentro de un intervalo determinado

- Variables discretas: Les corresponde un número finito de valores, generalmente números enteros, y son ocupadas para decisiones individuales no fraccionables.
- Variables binarias: Caso particular de variables discretas, en donde los valores disponibles a tomar son 0 y 1, y se ocupan comúnmente para decisiones de selección o asignación.

Dicho lo anterior, se tiene que, para resolver el problema en cuestión, se hará uso de una variable binaria, que represente la relación entre un estudiante y si está o no asignado a los desafíos disponibles, expresada de la siguiente forma:

$$X_{i,j} = \begin{cases} 1 & \text{si el estudiante } i \text{ es seleccionado en el desafío } j \\ 0 & \text{en caso contrario} \end{cases}$$

Con esto, se tendrá, al finalizar la distribución de estudiantes, un medio el cual indique a qué desafío fue asignado cada estudiante.

- **Restricciones:** Son las limitaciones y/o condiciones que se deben cumplir para la validación de las soluciones que se presenten para el problema en cuestión. Estas representan limitaciones en la capacidad y disponibilidad de los recursos, así como acotar el comportamiento y el enfoque de las soluciones que se generen. Para el contexto del problema, se pueden tener restricciones relacionadas a la conformación de los equipos, que describen cómo estos deberían ser finalmente creados. Se puede restringir tanto en la cantidad de estudiantes por equipo como la naturaleza de los mismos, y en una etapa inicial, se plantean restricciones tales como:

- Capacidad de los equipos (cantidad de estudiantes por equipo acotada):

$$X \leq \sum_{i=0}^E X_{i,j} \leq Y \quad \forall j$$

siendo X e Y los límites que se estimen convenientes para que el problema se considere resuelto.

- Condicionar a que, en la colución dada, cada estudiante debe sí o sí estar asignado en un desafío:

$$\sum_{i=0}^S X_{i,j} = 1 \quad \forall i$$

Estas restricciones, además, tienen la particularidad de poder ser duras o blandas, y permiten abordar el problema desde distintas aristas. En este contexto, las restricciones recién descritas se consideran duras, ya que deben cumplirse categóricamente en cada solución que se presente, teniendo como restricciones blandas, por ejemplo, la particularidad de los equipos formados a ser multidisciplinarios, ya que, si bien es una cualidad que se busca obtener, existen casos en donde se formen equipos que no presenten esta característica, pero eso no necesariamente implicaría la completa invalidez de la solución presentada.

- **Métricas:** Se definen como criterios que permiten evaluar, desde otros puntos de vista, la calidad de las soluciones a presentar. Estas métricas se relacionan con los objetivos del problema, y ayudan a medir qué tan buena es, según un criterio dado, la distribución que se presente.

Para el contexto del problema, se presentan métricas que, en mayor o menor obviada, implican una evaluación de la solución en base a la satisfacción de los estudiantes con los resultados de sus postulaciones. Bajo este criterio, se tiene la siguiente métrica principal:

- Puntaje de satisfacción: Permite identificar qué tan satisfechos quedarían los estudiantes en una distribución dada. Se describe como:

$$\frac{1}{s} \sum_{i=0}^S \frac{1}{P_i(E_i)}$$

una sumatoria en donde cada estudiante aporta puntaje de satisfacción acorde a la prioridad del desafío donde fue asignado.

Métricas de este tipo permiten identificar mejores y peores soluciones dependiendo del contexto de cada métrica, y además son un buen punto de análisis cuando se presentan soluciones muy cercanas en ciertos puntos, ya que, dependiendo además de las necesidades del problema, habrá situaciones en que ciertas métricas generen un mayor aporte que otras, siendo una ayuda para ser tomadas como criterios en la decisión de, por ejemplo, una distribución ganadora.

2.3. Discusión

La propuesta de solución realiza un aporte al campo de asignación de recursos, entregando una plataforma diseñada específicamente para la gestión de postulaciones y formación de equipos multidisciplinarios, además de estar orientada a usuarios con necesidades propias del ámbito académico, en comparación con las tecnologías existentes como Google OR-Tools u OptaPlanner, enfoques más dirigidos a la industria.

Por otro lado, la automatización del proceso de formación de equipos mejora la eficiencia y reduce significativamente el tiempo y los errores asociados con los métodos manuales.

Finalmente, la plataforma, al combinar elementos de optimización con principios y enfoques pedagógicos, no solo permite resolver el problema técnico de la asignación, sino que también fomenta la colaboración y el aprendizaje efectivo, aportando valor agregado y diferenciándola en comparación con las soluciones existentes.

Capítulo 3

Desarrollo de la solución

En esta sección se explicará el procedimiento empleado para el desarrollo de los algoritmos de distribución, y la evolución que estos tuvieron durante el proceso de PMM.

También, se procederá a detallar el funcionamiento de ambos algoritmos y cuales son sus diferencias. Estos algoritmos provienen de distintos contextos, el primero siendo generado en la sesión de creación de equipos para la edición 2024 del programa PMM, al cual, desde ahora en adelante, llamaremos Algoritmo Original, mientras que el otro algoritmo nace con el objetivo de poder comparar resultados y ser presentado como una herramienta más para la generación de equipos multidisciplinarios, al cual nos referiremos como Algoritmo Propuesta.

3.1. Estructura general de la solución

Para el diseño y desarrollo de la solución, se planteó prototipo funcional que atacase todas las problemáticas mencionadas en la introducción, proporcionando una plataforma web interactiva e intuitiva para los usuarios, donde se pudiesen realizar postulaciones y gestionar las mismas de una forma sencilla, así como ofrecer tecnologías de distribución para automatizar la formación de equipos multidisciplinarios.

Estas estrategias de distribución se utilizan por medio de endpoints independientes, y

presentan enfoques basados en algoritmos tradicionales, aprendizaje por reforzamiento y la técnica de búsqueda de Monte Carlo.

La decisión de entregar varias alternativas para un mismo caso de uso se fundamenta en que, primeramente, se puede realizar una comparativa justa entre tecnologías, ya que todos los enfoques propuestos son evaluados según un sistema de métricas, y a su vez permite que los administradores puedan seleccionar la solución más adecuada, según las necesidades y/o intereses específicos que se tengan, y también entrega un factor de escalabilidad al prototipo funcional, teniendo un diseño que facilita la actualización y refinamiento de métodos ya implementados, así como la incorporación de métodos nuevos a la plataforma.

Este trabajo explica sólo una parte del trabajo realizado para el prototipo funcional ya que, debido a que este proyecto se llevó a cabo en equipo, existen otros componentes de la solución entregada que no serán vistos en detalle en este escrito.

En cuanto a la solución general, se presenta el siguiente esquema general del sistema:

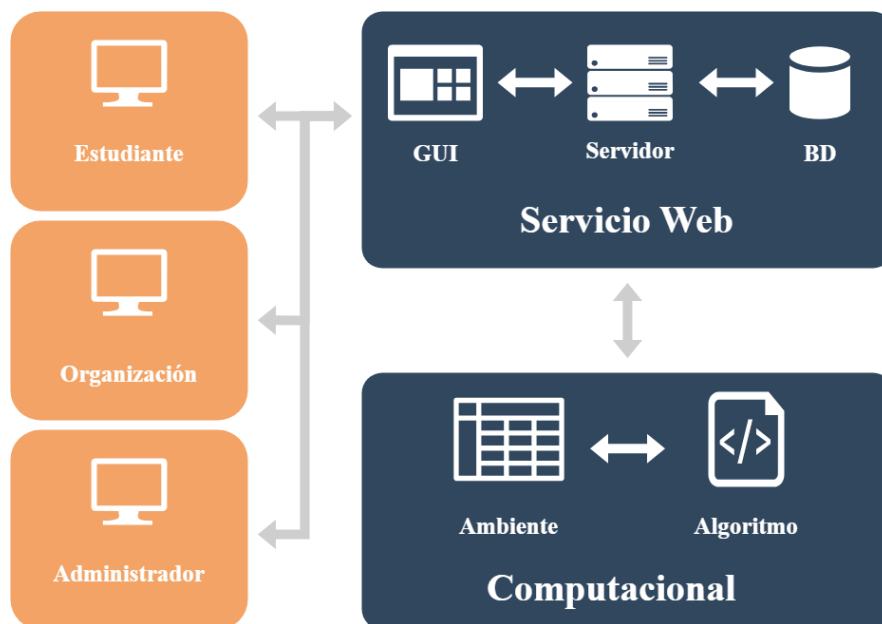


Figura 3.1: Esquema general de la solución

En el esquema se aprecia el flujo de funcionamiento de la solución. Básicamente, se reciben las postulaciones e información de los desafíos por parte de estudiantes e

instituciones/empresas, datos los cuales serán ocupados por el ambiente computacional para realizar las respectivas postulaciones (es aquí donde se ubica el trabajo expuesto en este documento).

El sistema como tal posee una serie requisitos funcionales como no funcionales, como se ve a continuación:

- Requisitos funcionales:

- **RF 1:** El sistema debe automatizar la distribución de desafíos entre los estudiantes basándose en las preferencias y restricciones especificadas.
- **RF 2:** El sistema debe generar equipos multidisciplinarios según los parámetros proporcionadas al algoritmo asegurando una distribución justa y equitativa de los desafíos.
- **RF 3:** El sistema debe permitir a los administradores gestionar y actualizar la lista de desafíos disponibles.
- **RF 4:** El sistema debe permitir a los administradores establecer y modificar las restricciones y criterios de formación de equipos.
- **RF 5:** El sistema debe permitir a los usuarios autenticarse mediante credenciales (correo y contraseña).
- **RF 6:** El algoritmo debe permitir la asignación manual de desafíos por parte de los administradores en casos especiales.
- **RF 7:** El algoritmo debe proporcionar resultados preliminares para evaluar asignaciones previas a su publicación.
- **RF 8:** El sistema debe mantener un registro de logs accesible para monitorear el correcto funcionamiento.

- Requisitos no funcionales:

- **RNF 1:** El algoritmo debe ser robusto y tolerante a fallos, garantizando la integridad de los datos en caso de interrupciones.

- **RNF 2:** El sistema debe ser escalable para soportar un aumento en la cantidad de usuarios y datos sin degradar el rendimiento.
- **RNF 3:** Solo los administradores autenticados y autorizados deben tener acceso a la gestión de desafíos.
- **RNF 4:** El algoritmo debe tener una interfaz de configuración amigable para que los administradores puedan ajustar parámetros sin necesidad de intervención técnica.
- **RNF 5:** El sistema debe implementar controles de acceso adecuados para prevenir modificaciones no autorizadas.
- **RNF 6:** La interfaz de usuario debe permitir sólo a los administradores tener la capacidad de realizar asignaciones manuales a desafíos específicos de manera sencilla y clara.
- **RNF 7:** Los resultados preliminares deben estar accesibles únicamente por los administradores, asegurando que no se filtren a los estudiantes antes de la publicación final.
- **RNF 8:** El sistema debe tener registros detallados que almacenen las operaciones realizadas para asegurar la trazabilidad y monitoreo.
- **RNF 9:** La interfaz del sistema debe ser intuitiva y fácil de usar para todos los tipos de usuarios.
- **RNF 10:** El sistema debe ser compatible en navegadores de dispositivos móviles y escritorios, adaptándose a diferentes tamaños de pantalla.

Para los algoritmos de distribución a implementar, se presta particular atención a los RF 6 y RF 7, así como el RNF 1 y RNF 4. Sin embargo, vemos que el RNF 4, si bien es referente al apartado del algoritmo, no se desarrolla en este documento, ya que su completitud depende del diseño en la plataforma web, dejando de esta forma los otros 3 requisitos como necesidades a cumplir en las distribuciones que se generen.

Viendo un poco más en detalle el flujo de trabajo de la solución, se presenta el diagrama de contexto para el que finalmente será el prototipo funcional:

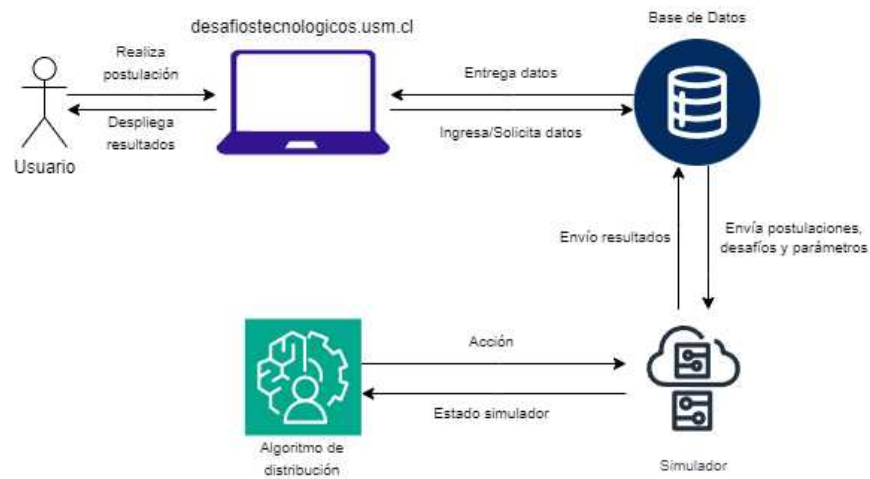


Figura 3.2: Diagrama de contexto

Es aquí donde vemos los procesos que deben ocurrir para que el algoritmo pueda hacer uso de las postulaciones y preferencias de los estudiantes, dando a entender que son una funcionalidad más de una solución más grande.

Por otro lado, se presenta también el diagrama de arquitectura de la solución, donde se aprecia la estructura modular de la misma, mostrando sus módulos y la interacción de los mismos:

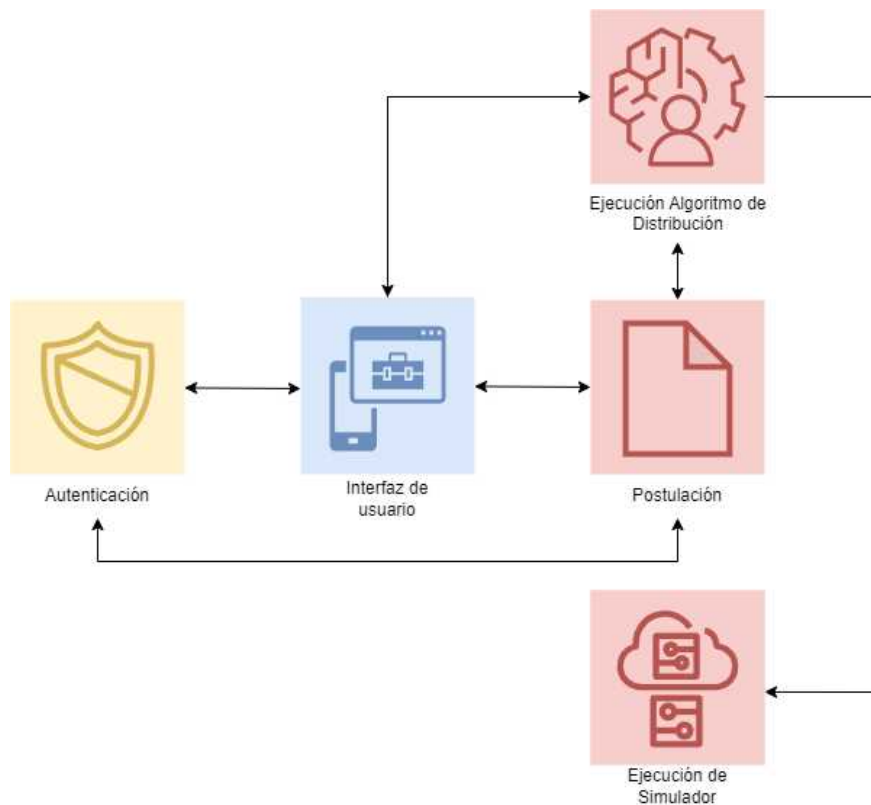


Figura 3.3: Diagrama de Arquitectura

En esta imagen, se presenta el módulo **Ejecución Algoritmo de distribución**. Éste es el módulo donde se aplica y desarrolla lo mencionado, y se puede ver explicado de mejor forma mediante la definición formal del mismo:

- **Propósito:** Módulo encargado de realizar la distribución de los desafíos entre los estudiantes, teniendo en cuenta los parámetros ingresados por el usuario Administrador, así como los datos referentes a las postulaciones de los usuarios Estudiantes.
- **Alcance:** El módulo debe realizar una distribución efectiva de los desafíos disponibles entre los estudiantes, asegurando que se cumplan todas las restricciones y parámetros que se ingresaron.
- **Dependencias:** El módulo depende tanto de la Interfaz de Usuario como de la

Autenticación, ya que de esta forma el usuario Administrador puede decidir en que momento iniciar la distribución, y además se tendrá la información necesaria para iterar de manera correcta.

- Supuestos: El algoritmo funcionará como corresponde todas las veces que se le solicite, y aplicando de manera correcta los cambios (si es que se realizan) en los parámetros.
- Restricciones: El usuario Administrador debe proporcionar todos los valores de parámetros para que el algoritmo realice la distribución.
- Estructura General: En cuanto a las entradas del módulo, se tienen tanto los datos de los usuarios Estudiantes y sus postulaciones, así como los parámetros que el usuario Administrador proporcione. La salida del módulo corresponde a los equipos conformados para cada desafío.

3.2. Algoritmo Original

3.2.1. Prototipo inicial

La iteración principal de esta etapa es levantar el esqueleto y estructura inicial con el cual va a trabajar el algoritmo, y en donde, inicialmente, se hace uso de los datos provenientes del ambiente de simulación, que permiten una ingesta constante de datos generados aleatoriamente, para ver así los comportamientos iniciales del algoritmo.

La estructura general del mismo se puede ver expresada en el siguiente ejemplo:

Algorithm 1 Asignación de estudiantes a desafíos

```
1: Entrada: Lista de desafíos  $E$ , Lista de Preferencias  $P$ 
2: Salida: Asignación de estudiantes a desafíos  $E'$ 
3:
4: for cada desafío  $e$  en  $E$  do
5:    $s_l \leftarrow$  obtener estudiantes que postularon a  $e$ 
6:    $l \leftarrow$  estudiante de la lista  $s_l$ 
7:   if  $e$  tiene espacio disponible then
8:     asignar  $l$  a  $e$ 
9:     if  $e$  ya no tiene espacio then
10:       continuar al siguiente desafío
11:   else
12:     buscar nuevo estudiante  $l$  en  $s_l$ 
13:
14:   Return  $E'$ 
```

donde se expresa que el algoritmo busca iterar en base a los desafíos y formar equipos completos en cada uno, completando así eventualmente la distribución.

Plan de Testing

Se comenta primeramente que el algoritmo itera, para cada lista de estudiantes que postularon a un desafío en particular, revisando de forma secuencial a los miembros de la misma, por lo que no se entrega inicialmente el nivel de justicia esperado para una distribución de este estilo. Además, se indica que la única restricción que se considera en esta versión es la del tamaño de los equipos a formar para cada desafío, por lo que la complejidad real de la situación no se ve representada.

Es por esto que, como plan de Testing propuesto, se tiene primeramente que ase-

gurar el nivel de justicia mínimo requerido, ya que estamos tratando con postulaciones y preferencias de cada estudiante, por lo que una selección aleatoria de los estudiantes permite entregarle el mismo nivel de “suerte” a todos los involucrados.

Por otro lado, se espera llegar a poder simular la complejidad real del problema a resolver, agregando iterativamente nuevas restricciones y condiciones de borde que representen los desafíos que el proceso de distribución realmente conllevan, de tal forma que se logren distribuciones de estudiantes útiles y factibles en ambientes reales.

3.2.2. Finalización del algoritmo

Luego de aplicar la estrategia definida anteriormente en el plan de Testing para el Algoritmo Original, se obtiene, en una explicación generalizada, un algoritmo que permite generar equipos para los desafíos ordenados según la cantidad de postulaciones, donde, para cada desafío, se buscan los estudiantes “más interesados” en el mismo, formando equipos completos y modificando las prioridades de los rezagados para eventualmente formar los equipos y terminar la distribución, tal y como se hace actualmente en el programa de memorias.

El procedimiento de puede ver desarrollado a continuación:

- Paso 1: En esta primera etapa, para dar comienzo a la distribución, se busca el desafío con la mayor cantidad de postulaciones, ya sean de primera, segunda o tercera prioridad. La idea es que, para cada iteración del algoritmo se conforme un equipo de trabajo, y de esta forma ir iterando con los varios desafíos disponibles.
- Paso 2: Luego de haber determinado el desafío con más postulaciones actualmente, se buscan todos los estudiantes que hayan postulado a este usando su primera prioridad, y con los cuales se escogerán de forma aleatoria, integrantes para ese

equipo, siempre y cuando este equipo a conformar cumpla con las restricciones de carreras y tamaño de estudiantes por equipo.

- Paso 3: Luego de esto, inevitablemente se tendrán estudiantes que no habrán quedado seleccionados en el desafío, lo cual implica que una de sus prioridades (la primera prioridad) no estará disponible en el resto de la distribución. Es por esto que, teniendo en cuenta que los estudiantes que participan de la asignación aleatoria para un desafío son sólo los estudiantes que tienen su primera prioridad en dicho desafío, se cambian las preferencias respectivas de los estudiantes que hayan quedado fuera después de la formación de un grupo de trabajo.

Para entender de mejor forma este paso, se tiene un ejemplo:

Algorithm 2 Asignación de estudiantes a desafíos

- 1: **Entrada:** Lista de desafíos E , Lista de Preferencias P
 - 2: **Salida:** Asignación de estudiantes a desafíos E'
 - 3:
 - 4: **for** cada desafío e en E (ordenados por cantidad de postulaciones) **do**
 - 5: $p \leftarrow$ obtener lista de estudiantes que tengan al desafío e como prioridad 1
 - 6: $q \leftarrow$ grupo de estudiantes para el desafío e
 - 7: **if** estudiante en p no fue seleccionado en q **then**
 - 8: cambiar preferencias de estudiante en p no seleccionado para que vuelva a tener un desafío con prioridad 1
 - 9: Continuar al siguiente desafío con mayor cantidad de postulaciones
 - 10: **Return** $E' = 0$
-

- Paso 4: Una vez modificadas las prioridades necesarias se continúa al siguiente desafío con la mayor cantidad de postulaciones.

Entonces, habiendo analizado se procede a la automatización del proceso mediante implementar lo descrito, haciendo uso de dos clases en Python. Las clases son referentes a la lógica de distribución, por una parte, y una clase para el cálculo de métricas

(revisar la sección 4.1), por la otra.

La clase de distribución contiene métodos que son referentes al procedimiento anteriormente descrito, en donde se cuentan la cantidad de postulaciones para determinar el primer desafío a iterar, los límites de carrera para, luego de haber realizado las primeras rondas de generación de equipos, los estudiantes que no hayan quedado seleccionados en ninguna de sus postulaciones son agrupados (siempre cumpliendo con los requisitos de carrera) para conformar los grupos finales.

Finalmente, los estudiantes que falten por ser asignados, son incorporados a grupos con el objetivo de aumentar la cantidad de integrantes de ese grupo, cumpliendo con no superar la cantidad máxima de estudiantes permitidos por grupo.

Posteriormente, se hace uso de los grupos conformados y las prioridades originales de los estudiantes para, por medio de la clase de cálculo de métricas, evaluar el rendimiento de la distribución entregada.

3.3. Algoritmo Propuesta

3.3.1. Prototipo inicial

Se comienza por levantar la estructura inicial del algoritmo, considerando los aspectos primordiales de una distribución con tal de tener un esqueleto de lo que debe hacer la distribución, así como los factores fundamentales en los cuales prestar atención.

En un inicio, en un estado previo al uso de datos reales, el algoritmo se alimentaba del ambiente de simulación creado para la generación aleatoria de postulaciones, con el objetivo de, por un lado, entregar una forma de corroborar el correcto funcionamiento inicial del algoritmo, y proporcionar los datos necesarios para el entrenamiento de los modelos de aprendizaje por reforzamiento, y que juntos componen la totalidad de tecnologías implementadas para realizar distribuciones.

Con esto en mente, se tienen los siguientes pasos que este algoritmo inicial implementó:

- **Obtención del estado del ambiente:** Tiene por objetivo recibir la información (estado) del ambiente de simulación (a describir en el apartado de ambiente de simulación) para trabajar con él. El ambiente, una vez leído, proporciona la lista de desafíos (E) y las preferencias (P) de los estudiantes, y es la información que el algoritmo usará para realizar la distribución.
- **Paso del algoritmo (asignación de estudiante):** En esta etapa se realiza una iteración del algoritmo, en donde se selecciona a un estudiante, se revisan sus prioridades y, de ser posible, se le asigna al desafío con la prioridad más alta que tenga espacio disponible, repitiendo este proceso hasta haber asignado a cada uno de los estudiante de la lista.

Se tiene que el procedimiento a seguir, para cada estudiante en la lista, es el siguiente:

Algorithm 3 Asignación de estudiantes a desafíos

- 1: **Entrada:** Lista de desafíos E , Lista de Preferencias P
 - 2: **Salida:** Asignación de estudiantes a desafíos E'
 - 3:
 - 4: **for** cada estudiante s en S **do**
 - 5: $p \leftarrow$ obtener preferencia más alta de s
 - 6: $e \leftarrow$ desafío correspondiente a la preferencia p
 - 7: **if** e tiene espacio disponible **then**
 - 8: asignar s a e
 - 9: **else**
 - 10: evaluar la siguiente preferencia de s
 - 11: **Return** E'
-

con lo cual se obtendrá, como se explicita en el pseudocódigo, la lista de desafíos con estudiantes asignados (E').

- **Comprobación de cumplimiento de restricciones:** Una vez completada la distribución, se verifica que las restricciones del problema hayan sido cumplidas. Estas restricciones son en relación a la naturaleza que tienen los equipos del programa de memorias multidisciplinarias, así como la estructura esperada que estos mismos conformen: Se tiene un número mínimo y máximo de estudiantes por desafío (en este caso, un equipo puede estar formado por mínimo 2 y máximo 3 participantes), donde se quiere apuntar a generar equipos de carácter multidisciplinario, es decir, equipos donde los participantes pertenezcan a dos o más carreras distintas, y se busca también que un estudiante, además de tener que estar sí o sí seleccionado luego de la distribución, este sólo haya sido asignado a un desafío.

La formulación de las restricciones se ve de la siguiente manera:

- Para cada desafío, el equipo formado al terminar la distribución debe ser mayor a 1 y menor a 3 estudiantes:

$$1 \leq \sum_{i=0}^E X_{i,j} \leq 3 \quad \forall j$$

- Cada estudiante, luego de haber finalizado la distribución, tiene que estar asignado en algún desafío, y a su vez tiene que estar asignado solamente en un desafío:

$$\sum_{i=0}^S X_{i,j} = 1 \quad \forall i$$

- donde:
 - E = Lista de desafíos
 - S = Lista de estudiantes

- Variable binaria de selección de estudiante i en desafío j :

$$X_{i,j} = \begin{cases} 1 & \text{si el estudiante } i \text{ es seleccionado en el desafío } j \\ 0 & \text{en caso contrario} \end{cases}$$

- **Métricas de rendimiento** Si bien la distribución puede cumplir con las restricciones dadas, se necesitan métricas propias del contexto del problema que permitan medir la calidad de la solución que presenta el algoritmo. Para ello, se implementan métricas como, por ejemplo, la satisfacción, la cual indica cuántos estudiantes pudieron ingresar a sus desafíos de alta prioridad; la métrica de equipos PMM, que indica la cantidad de equipos formados que presenten la característica de ser multidisciplinarios; y la métrica de distribución equitativa, que muestra qué tan uniforme es la distribución de los estudiantes en los desafíos, teniendo en cuenta sus prioridades.

La finalidad de estas métricas es la de poder tener datos con los cuales realizar comparaciones, primeramente entre cada distribución que entregue el algoritmo, pudiendo identificar así distribuciones que sean objetivamente mejores que otras y, por otro lado, presentar resultados que permitan comparar el rendimiento del algoritmo con el ambiente de simulación, ya que los datos propios del ambiente (por ejemplo, la recompensa acumulada), terminan siendo poco representativos de la calidad del resultado una vez que el algoritmo termine de iterar.

Las métricas, por su parte, también se pueden describir con el formato de restricciones. Por ejemplo:

- Puntaje de Satisfacción:

$$\frac{1}{s} \sum_{i=0}^S \frac{1}{P_i(E_i)}$$

- donde:

- S = Lista de estudiantes

- $P_i(E_i)$ = Preferencia con la que el estudiante i fue asignado en un desafío (si el estudiante fue asignado en su primera prioridad, $P_i(E_i) = 1$, si es su segunda prioridad, $P_i(E_i) = 2$, y si es su tercera, $P_i(E_i) = 3$).

De esta forma, mientras más alta sea la prioridad con la que el estudiante fue asignado a un desafío, más habrá aportado al puntaje de satisfacción, y este aporte por estudiante va disminuyendo conforme la asignación corresponda a una prioridad más baja.

Con todo lo anterior, además de obtener un modelo funcional inicial del algoritmo donde, como se explica en el procedimiento presentado, se tienen en cuenta sólo las restricciones mencionadas, además de una métrica de rendimiento implementada, se tiene además el levantamiento preliminar del módulo donde se encontrará el algoritmo para ser llamado eventualmente en el prototipo funcional.

Es por esto entonces que se plantea una serie de procedimientos que tienen como objetivo moldear el enfoque de avance de este modelo Dummy presentado, con el cual sentar las bases de desarrollo para la obtención de un algoritmo que pueda representar una distribución más cercana a la real:

Plan de Testing (Modelo Dummy de algoritmo)

Para el planteamiento del Testing del algoritmo, es necesario indicar el estado inicial con el cual se va a trabajar. En una etapa primeriza, se tuvo como objetivo corroborar el correcto funcionamiento del sistema completo, es decir, la conexión de la plataforma Web con el servicio computacional. Luego de esto, en el aspecto computacional específicamente, se introdujo una versión “Dummy” del algoritmo al sistema, con tal de comprobar que la conexión de éste con el ambiente de simulación se realizaba de buena manera.

Es por esto que, en relación a la complejidad del algoritmo en sí, buscando que este eventualmente realice una buena distribución de estudiantes, se tiene un estado

inicial (Dummy) precario, donde la selección del estudiante a asignar es secuencial, con las únicas restricciones siendo la condición de tamaño de los equipos, y que cada estudiante debe sí o sí ser asignado en un desafío, teniendo además la única métrica implementada, siendo ésta el Puntaje de Satisfacción.

Dado lo anterior, se tiene que el plan de Testing consiste en que el algoritmo progresivamente debe, primeramente, integrar un sistema aleatorio de selección de estudiantes a asignar en los desafíos, además de cumplir con más restricciones, ya que las implementadas actualmente no representan la complejidad del problema actual. Luego de esto, se deben implementar más métricas con las cuales poder evaluar desde más aristas la calidad de las distribuciones, y así también entregar soluciones enfocadas en cumplir con restricciones puntuales, según lo estime el usuario Administrador.

Finalmente, realizar pruebas con diversas configuraciones, tanto de restricciones a cumplir y métricas a analizar, como prueba de casos en donde las preferencias de los estudiantes presenten casos de inflexión y complicación para el algoritmo, evaluando así su rendimiento en una amplia gama de casos.

3.3.2. Finalización del algoritmo

Se siguió desarrollando la propuesta de distribución explicada en 3.3.1, donde ya se había descrito el procedimiento a seguir para acercar la solución a resolver un problema real, y que tuviese las suficientes variables en consideración, con tal de representar la complejidad intrínseca del problema.

Ya se tenía la estructura general definida, en donde, a diferencia del método usado por el cliente, se itera (ahora en orden aleatorio y no según el orden de estudiantes definido en el archivo JSON) sobre los estudiantes, revisando sus preferencias y asignándolos (de ser posible) a un desafío que se encuentre en sus preferencias, intentando siempre que sea asignado en la preferencias más alta que se pueda permitir. Este proceso se vuelve a realizar hasta que todos los estudiantes hayan sido asignados a un desafío, verificando también las restricciones anteriormente mencionadas.

También, a modo de actualización, se comenta que, para las restricciones definidas en 3.3.1, se tiene que la segunda se mantiene tal cual, mientras que la primera restricción ahora pide que los equipos conformados estén dentro de valores de 2 a 4 estudiantes:

$$2 \leq \sum_{i=0}^E X_{i,j} \leq 4 \quad \forall j$$

con el objetivo de conformar equipos con una mayor cantidad media de estudiantes.

Con respecto a la implementación del algoritmo, al igual que el método propuesto por el cliente, se indica que ahora la forma de obtener el estado del ambiente de simulación (la obtención de la información referente a las postulaciones) es mediante el consumo de las colecciones del archivo JSON. Además, al igual que en la implementación de la estrategia actual, se hace uso de dos clases en Python, una referente a la lógica de asignación de estudiantes a desafíos, y otra para el cálculo de métricas que permiten evaluar el rendimiento de la distribución realizada.

La clase para la distribución de estudiantes lee el archivo JSON, leyendo la información relevante y formando los equipos conforme va avanzando por los estudiantes y sus preferencias. En casos donde un estudiante no puede ser asignado en ninguna de sus preferencias, estos pasan a ser gestionados por un método de la clase que los asigna, cumpliendo con las restricciones de carrera, a equipos con tal de cumplir con las restricciones de cantidad mínima de estudiantes para efectivamente conformar un equipo. Finalmente, y a diferencia del método anterior, las preferencias de los estudiantes nunca son modificadas, por lo que son directamente proporcionadas a la clase de métricas, para que haga uso de estas y, en base a la distribución final obtenida, evaluar el proceso.

Capítulo 4

Resultados

4.1. Métricas de evaluación

Para los resultados de los algoritmos empleados, se desarrolló, tal y como se mencionó en varios puntos anteriores, un sistema de métricas de evaluación, que buscan atacar los varios enfoques que tiene el programa PMM, y la calidad que las distribuciones generadas por cada algoritmo presentan, para cada una de esas métricas.

Las métricas en cuestión son las siguientes:

- **Satisfacción Promedio:** La única métrica definida anteriormente, que describe el aporte que hace un estudiante según el nivel de preferencia que tenga el desafío en el que el estudiante fue asignado (de esta forma, mientras más alta sea la prioridad del desafío al que termina perteneciendo un estudiante, mayor será el puntaje de aporte a la métrica, y viceversa conforme menor sea la prioridad). Mientras más cercano a 1, mejor será la distribución.
- **Estudiantes en Primera Prioridad:** Nace de interpretar una satisfacción total como el caso donde todos los estudiantes participantes quedan seleccionados en su desafío de primera prioridad. Cuenta la cantidad de estudiantes que fueron seleccionados en su desafío de primera prioridad. Mientras mayor sea el número, mejor será la distribución.

- Estudiantes fuera de Preferencias: Contrario a la métrica anterior, se considera el nivel más bajo de satisfacción cuando ningún estudiante queda en ninguna de sus preferencias, por ende, cuenta la cantidad de estudiantes que cumplen con esta condición. Por esto, mientras mayor sea este número, peor será la distribución.
- Desafíos sin Equipo: Un objetivo a tener en cuenta es generar la mayor cantidad de equipos PMM con la idea de abarcar la mayor cantidad de desafíos. debido a esto, mientras mayor sea este número, peor será la distribución.
- Desviación Estándar tamaño Equipos: La formación de los equipos PMM, si bien tiene holguras en cuanto a la cantidad de integrantes, si buscan ser más o menos uniformes, buscando generar equipos de 3 integrantes, teniendo equipos de 2 o 4 como opciones, en casos de que temas como restricciones y la mayor o menor consideración por las preferencias del estudiante influyan en mayor medida. Con esto, se puede hacer un análisis interesante con la métrica de cantidad de desafíos con equipo, y para la métrica por sí sola, al estar hablando de una desviación estándar, mientras más cercana a 0 sea, implica que la distribución generó equipos con una cantidad de integrantes más uniforme.
- Promedio de carreras por Equipo: Métrica enfocada en el aspecto multidisciplinario de los equipos conformados, ya que se busca que los grupos tengan varias carreras diferentes, y así asegurar la variedad en los equipos. Dicho esto, mientras más grande sea este valor, implica que hay una mayor cantidad de carreras por grupo conformado.

Se puede ver entonces, que las métricas definidas buscan dar enfoques distintos en lo que es el rendimiento y la calidad de la distribución resultante, permitiendo así, de ser necesario, enfocar las distribuciones a maximizar una métrica en específico, o incluso un grupo de ellas, y de esta forma poder medir aspectos propios del problema, como lo son la satisfacción de los estudiantes, el factor de multidisciplinariedad de los equipos formados, la cantidad de estudiantes no satisfechos en base a la distribución,

entre otros.

4.2. Información entregada por algoritmos y muestra de evaluación según métricas

Para comprobar entonces el funcionamiento de los algoritmos, se ocuparon las postulaciones correspondientes a la versión 2024 del programa de memorias, con el objetivo de analizar el comportamiento de las distribuciones en casos reales.

Un ejemplo del resultado que entregan las distribuciones, para cada algoritmo desarrollado, se puede ver a continuación:

- Algoritmo Original:

```
=== Métricas de Asignación ===  
Satisfacción promedio: 0.711  
Estudiantes en primera prioridad: 40  
Estudiantes fuera de preferencias: 12  
Desafíos sin equipo: 20  
Desviación estándar tamaño equipos: 0.497  
Promedio de carreras por equipo: 1.83  
  
Porcentajes:  
Primera prioridad: 62.5%  
Segunda y Tercera prioridad: 18.8%  
Fuera de preferencias: 18.8%
```

Figura 4.1: Resultados: Algoritmo Original

- Algoritmo Propuesta:

```

=== Métricas de Asignación ===
Satisfacción promedio: 0.779
Estudiantes en primera prioridad: 44
Estudiantes fuera de preferencias: 7
Desafíos sin equipo: 18
Desviación estándar tamaño equipos: 0.600
Promedio de carreras por equipo: 1.65

Porcentajes:
Primera prioridad: 68.8%
Segunda y Tercera prioridad: 20.3%
Fuera de preferencias: 10.9%

```

Figura 4.2: Resultados: Algoritmo Propuesta

Se aprecia la información detallada sobre los equipos conformados, agregando también especial atención a temas como la satisfacción obtenida por las distribuciones, el porcentaje de aparición para cada categoría de prioridad (1, 2 o 3), entre otros.

Con esto, se tienen entonces, los resultados de ambos algoritmos para las métricas establecidas en la siguiente tabla:

Métrica	Algoritmo Original	Algoritmo Propuesta
Estudiantes en su 1° preferencia	39	44
Estudiantes fuera de sus postulaciones	10	6
Satisfacción promedio	71 %	78 %
Estudiantes promedio por equipo	3.5	3.2
Carreras promedio de un equipo	1.83	1.7
Total de equipos conformados	18	20

Capítulo 5

Conclusiones

En relación a los resultados obtenidos, se puede apreciar que, dependiendo de qué es lo que se quiera asegurar con las distribuciones generadas, se hará uso de un algoritmo u otro, y es que, si buscamos equipos con una cantidad de estudiantes más homogénea, donde se premie más la generación de equipos de carácter multidisciplinario, y en donde se busque una composición más o menos grande de participantes por equipo, el algoritmo original entrega mejores resultados.

En cambio, podemos querer encontrar una distribución que se centre en aumentar las preferencias principales de los estudiantes, donde una satisfacción promedio de alto valor resulta ser el atractivo principal, y donde se generen una mayor cantidad de equipos, cuidando de minimizar la cantidad de estudiantes que no queden en ninguna de sus preferencias, en cuyo caso el algoritmo propuesta es una sólida opción.

Sea cual sea el caso, es importante aclarar que el objetivo principal es el de lograr automatizar con éxito un proceso tedioso y que innecesariamente era realizado de forma manual, donde, por los valores de las métricas obtenidas, se obtienen resultados prometedores, que sin duda contemplan las postulaciones, condiciones y preferencias de los estudiantes, teniendo también una mirada enfocada en mantener el objetivo prin-

cial del programa de memorias, y que finalmente se traduce en un servicio disponible como herramienta en la plataforma del prototipo funcional, para así brindar su apoyo en versiones futuras de PMM.

Temas interesantes, como lo son el concepto de justicia o preferencia en un ámbito complejo como lo son problemas de distribución, abren espacio a conversaciones que entrelazan estas palabras de forma que parecen bastante más cercanas de lo que se cree generalmente, y que demuestran que pueden coexistir en un espacio como este, donde temas más subjetivos se presentan como restricciones, condiciones o incluso, en este caso, incentivos para ir por un camino en vez del otro, buscando encontrar soluciones que demuestren compromiso tanto con una efectiva formación de equipos como en los intereses y elecciones de los participantes de los mismos, siempre, en la medida de lo posible.

Sin duda, los cambios en las necesidades, prioridades y objetivos del programa pueden llegar en cualquier momento, por lo que, tener un sistema que entregue buenos resultados en las distintas etapas del proceso, proporcionando información relevante para la toma de decisiones (en este caso, la formación de equipos multidisciplinarios), es una gran ventaja, la cual, seguido con la creciente popularidad de PMM, teniendo cada vez más estudiantes participando, eventualmente se puede convertir en una necesidad o requerimiento mínimo para una gestión cómoda y satisfactoria del proceso.

Bibliografía

- [1] Owen Lacey. How audacia implements google or-tools into software development projects. July 2021. <https://audacia.co.uk/technical-blog/how-audacia-implements-google-s-or-tools-into-projects>.
- [2] Phillipe Laborie. Ibm ilog cp optimizer for detailed scheduling illustrated on three problems. May 2009. https://www.researchgate.net/publication/221353689_IBM_ILOG_CP_Optimizer_for_Detailed_Scheduling_Illustrated_on_Three_Problems.
- [3] Misty Loughry, Matthew Ohland, and David Woehr. Assessing teamwork skills for assurance of learning using catme team tools. May 2014. https://journals.sagepub.com/doi/abs/10.1177/0273475313499023?casa_token=Tvd9jj21RhQAAAAA%3A6XR0BCeNCca3Joq3_s6fYo30WN9rk6PtVXYlJ9BHNe4PPiKfoQe3XlbVMr3avp4raVBQ4CFpm2h_UN8&journalCode=jmda.
- [4] Marián Macik. Case management task assignment using optaplanner. October 2016. <https://is.muni.cz/th/ms65e/dp-macik.pdf>.
- [5] Claudia Andrea Chacón Ossa. Modelamiento y resolución de asignación de horarios para bachiller de la malla nueva de ingeniería civil informática. November 2019. <https://repositorio.usm.cl/entities/publication/1139db43-0244-4266-8279-5534c452e711>.

- [6] Kevin Alexander Lagos Lavín. Interrogation scheduling problem: Una estrategia basada en preferencias de profesores y alumnos. August 2021. <https://repositorio.usm.cl/handle/11673/52587>.
- [7] Víctor F Suarez, Álvaro Guerrero, and Omar D Castrillon. Programación de horarios escolares basados en ritmos cognitivos usando un algoritmo genético de clasificación no-dominada, nsga-ii. August 2022. https://www.scielo.cl/scielo.php?pid=S0718-07642013000100012&script=sci_abstract.
- [8] Andrés Saldaña Crovo, Cristian Oliva San Martín, and Lorena Pradenas Rojas. Modelos de programación entera para un problema de programación de horarios para universidades. December 2007. https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052007000300005.
- [9] Desconocido. Estudio del problema de school timetabling utilizando técnicas de ia. February 2009. http://sedici.unlp.edu.ar/bitstream/handle/10915/3993/Presentaci%C3%B3n_.pdf-PDFA.pdf?sequence=3.
- [10] Alberto Daudén Figueras. Desarrollo de un algoritmo para la distribución de exámenes finales en el itba. June 2019. <https://upcommons.upc.edu/handle/2117/182834?show=full>.