



FACULTAD DE INGENIERÍA
UNIVERSIDAD DE BUENOS AIRES

Trabajo Profesional de Ingeniería en Informática

MUSSA

**Generador de Planes de Carrera
personalizados**

Jennifer Andrea Woites

Tutor: Lic. Rosa Wachenchauzer

Co-Tutor: Ing. Diego Essaya

Abril, 2018

Índice

1. Introducción	2
1.1. Motivación	2
2. Alcance	3
3. Tecnologías y Herramientas	3
4. Página Web MUSSA: Acceso Público	5
4.1. Carreras y materias de la facultad	5
4.2. Búsqueda de Docentes	6
4.3. Resultados de Encuestas de Docentes	7
4.4. Búsqueda de Materias	8
4.5. Resultados de Encuestas de un curso	9
5. Página Web MUSSA: Usuarios	12
5.1. Login / Sign In / Cambio de contraseñas	12
5.2. Perfil de usuario	13
5.3. Materias habilitadas para cursar	16
5.4. Formularios	17
5.5. Encuestas	18
5.6. Generación y Visualización del Plan de Carrera	20
6. Página Web MUSSA: Acceso Administrador	23
6.1. Administrar Horarios	23
6.2. Administrar Cursos	24
6.3. Administrar Docentes	25
7. Algoritmos para la generación del Plan de Estudios personalizado	27
7.1. Arquitectura y Flujo General	27
7.2. Configuración de Parámetros	28
7.3. Algoritmo Programación Lineal Entera (PLE)	30
7.3.1. Programación Lineal Entera: Qué es y cómo funciona . .	30
7.3.2. Branch & Cut	31
7.3.3. Algoritmo Desarrollado	34
7.4. Algoritmo Greddy	40
7.5. Pruebas	42
8. ¿Integrando con el SIU Guaraní?	62
9. Mejoras y Nuevas Funcionalidades Futuras	64
10. Conclusiones	66
11. Agradecimientos	68
12. Referencias y Material consultado	69

1. Introducción

El siguiente documento presenta el Trabajo Profesional de Ingeniería en Informática de la estudiante Jennifer Andrea Woites, padrón 93274. Los docentes que estuvieron a cargo son: Lic. Rosa Graciela Wachenchauzer como tutora del trabajo profesional, y el Ing. Diego Essaya como cotutor.

El tema del trabajo profesional es «MUSSA¹ - Generador de Planes de Carrera personalizados.»

El objetivo general del presente trabajo, es la realización de una web responsive que permita a los alumnos la visualización de sus materias, la administración de las mismas, la utilización de encuestas y la generación de un plan de carrera en base a parámetros personalizados, siendo ésto último el ítem más importante del trabajo.

1.1. Motivación

Durante los años en que el alumno lleva adelante la carrera, muchas veces se ve obligado a no seguir con las materias tal como figuran en el plan por diversos motivos: cupos llenos en los cursos que prefiere, decisión para cursar con otros compañeros, necesidad de cursar menos materias por cuatrimestre, incompatibilidad laboral, razones personales, etc. Estos alumnos se ven forzados a tener que rediseñar el plan para poder acomodar las materias en un tiempo razonable de realización, que además sea compatible con las preferencias que él o ella tengan.

A esto, se suman materias que solo se dictan en el primer o segundo cuatrimestre, las correlatividades, entre otras, que dificultan la decisión de qué materia realizar antes que otra.

Luego, se debe tener en cuenta que en muchos de los planes de estudio, las materias electivas no están clasificadas por ramas o intereses y que es parte de las decisiones que debe tomar el alumno el elegir qué materias desea cursar, muchas veces decidiendo más por el nombre que por el contenido en sí porque no necesariamente sabe si le está aportando conocimiento en el área a la que le gustaría dedicarse. En otros casos, por comentarios de algunos compañeros que previamente cursaron dichas materias, es capaz de asesorarse y decidir con un razonamiento más amplio.

Actualmente, la decisión de qué materia se cursa primero y cuál después, se toma "manualmente", es decir, se observa el punto en el que se está parado y se decide en base a lo mejor que se puede hacer en el corto plazo ya que es muy difícil observar el panorama completo cuanto más lejos de la meta se está.

En el marco del programa de re inserción de alumnos que han abandonado la carrera y que le restan pocas materias por recibirse; es útil poder contar con una herramienta que priorice las materias en base a los tiempos que esta persona tiene disponible y tratar de recomendarle los mejores cursos para que pueda recibirse con prontitud, de forma de poder incrementar el número de profesionales recibidos en Argentina.

» ¹Materias Universitarias en un Sistema Simplificado Automático

Se busca entonces, poder facilitar estas tareas a través de una plataforma web que permita obtener la información requerida para clasificar las materias (por ejemplo, a través de las encuestas), que facilite la generación de las notas para trámites como pedidos de créditos o excepciones de correlatividades que han de presentarse en la facultad posteriormente, y que, a través de parámetros configurados por el usuario, pueda armar automáticamente un plan de carrera que se ajuste a sus necesidades, pudiendo ser mutado si las mismas cambian con el paso del tiempo.

2. Alcance

El alcance del trabajo es:

1. Desarrollar una página web responsive que permita:
 - Login / Sign In / Cambio de Contraseña
 - Almacenamiento y edición de datos personales, carreras, materias aprobadas / desaprobadas / con final pendiente / en curso
 - Visualizar la lista de materias habilitadas para cursar
 - Descarga de formularios: Nota al decano y listado de materias
 - Completar y visualizar las encuestas de cursos / docentes
 - Búsquedas de materias con filtros
 - Visualizar y Generar Plan de Estudios
2. Permitir la administración desde la web:
 - Cursos
 - Docentes
 - Carga de horarios desde PDF
3. Modelar y desarrollar el algoritmo para la generación del plan de estudios personalizada

3. Tecnologías y Herramientas

A continuación se detallan las herramientas y tecnologías más importantes que fueron utilizadas durante el desarrollo del trabajo.

Como entorno de desarrollo se había elegido Sublime, pero posteriormente se cambió a IntelliJ IDEA ya que proveía de el debugger incorporado y otras funcionalidades básicas que no estaban disponibles en Sublime.

Para el control de versiones fue utilizado GitHub².

Para la administración del proyecto se utilizó GDocs y Trello.

²El proyecto MUSSA se encuentra en <https://github.com/jennywoites/MUSSA>

Para la generación de documentación, tanto de la propuesta como del presente informe se utilizó Latex.

Se eligió como lenguaje Python junto a Flask[13] para el backend. Flask fue elegido a pesar de no tener experiencia por estar escrito en Python y ser un framework minimalista con una curva de aprendizaje relativamente rápida.

El frontend fue desarrollado con HTML5, JavaScript, CSS y Bootstrap[17] para que sea responsive.

Para la base de datos se eligió MySQL[18]. Además fue utilizado SQLAlchemy [15] como ORM³.

Para el comienzo del proyecto se utilizó Flask-User[16]. Este proyecto ofrece una estructura inicial con las acciones básicas de login/signin y manejo de contraseñas. Fue difícil de configurar al principio ya que no se tenía experiencia con Flask en general, pero una vez configurado solo hizo falta realizar las customizaciones correspondientes. El problema principal que se tuvo al utilizar Flask-User, en lugar de haber realizado los servicios manualmente, es que cuando se quiso desarrollar la API no se la pudo independizar completamente del proyecto ya que para ello, requería solucionar el cross domain o realizar una autenticación en ambos puntos. Como punto a favor es que Flask-User simplificó mucho el desarrollo de esa sección y proveyó de una estructura básica al proyecto en general.

Para poder realizar los planes de forma asíncrona, se utilizó Celery⁴[19] para la cola de tareas asíncrona junto a Redis⁵[20] como estructura para los datos de las tareas.

Para la generación del plan con el algoritmo de programación lineal entera, se utilizó PuLP[11]. PuLP es una librería Open Source escrita en Python. Es usada para describir problemas de optimización como modelos matemáticos. PuLP puede llamar a numerosos solvers de programación lineal, tales como CBC, GLPK, CPLEX, Gurobi, etc, para resolver el modelo y luego utilizar comandos de Python para manipular y mostrar el resultado de la solución. Durante el desarrollo del algoritmo se probaron diferentes solvers, pero finalmente fue elegido CBC ya que fue el que mejor respondía en tiempos para el algoritmo ejecutado, además de ofrecer multi-threading para la resolución del problema.

³ORM: Object Relational Mapper. Un ORM simplifica el acceso a una base de datos por parte del programador convirtiendo toda sentencia sql a operaciones con objetos.

⁴Celery es una cola de tareas asíncrona open source basada en el pasaje de mensajes distribuido

⁵Redis es un almacén de estructura de datos de valores de clave en memoria rápido y de código abierto

4. Página Web MUSSA: Acceso Público

4.1. Carreras y materias de la facultad

Las diferentes carreras de la facultad están compuestas por una serie de materias obligatorias, posibilidad de un idioma obligatorio, posibilidad de rendir un examen de suficiencia de idioma, materias electivas, trabajo profesional, tesis, etc.

Para esta primer implementación se cargaron y modelaron los planes de estudios de Ingeniería en Informática y Licenciatura en Análisis de Sistemas (1986, plan ‘viejo’). Posteriormente en futuras implementaciones, se añadirán las carreras restantes.

Cada materia tiene como datos su código, el nombre, los diferentes cursos disponibles con sus docentes y horarios, la cantidad de créditos y sus correlatividades.

Los horarios de los cursos de las materias mantienen un histórico, y se agrega una nota visual en caso de que sea un curso nuevo (en cuyo caso se repiten los horarios del cuatrimestre actual para el cuatrimestre faltante). Para cada curso de la materia, se indica si se dicta el 1º cuatrimestre, el 2º o ambos. Los cursos que no se dicten ningún cuatrimestre ya no serán mostrados en la oferta horaria, pero quedarán disponibles para que el administrador los pueda volver a habilitar.

Se decidió que cada materia de las diferentes carreras sea tratada como un objeto diferente por más que compartan el código, ya que lo más usual es que las correlativas sean diferentes, que una sea obligatoria de una carrera y de la otra sea electiva y a veces que no todos los cursos estén disponibles para ambas.

Los cursos en cambio, se modelarán como uno solo para el mismo código y se indicará las carreras para las que se encuentra habilitado. En el caso de los cursos que dictan diferentes códigos de materias en el mismo horario, serán tratados como cursos independientes.

Los docentes podrán estar vinculados a uno o más cursos.

Desde el acceso público, se podrán hacer búsquedas de materias, búsquedas de docentes, ver algunos links útiles e información de las comisiones por carrera y los datos de contacto del proyecto en caso de que quieran hacer aportes al proyecto.

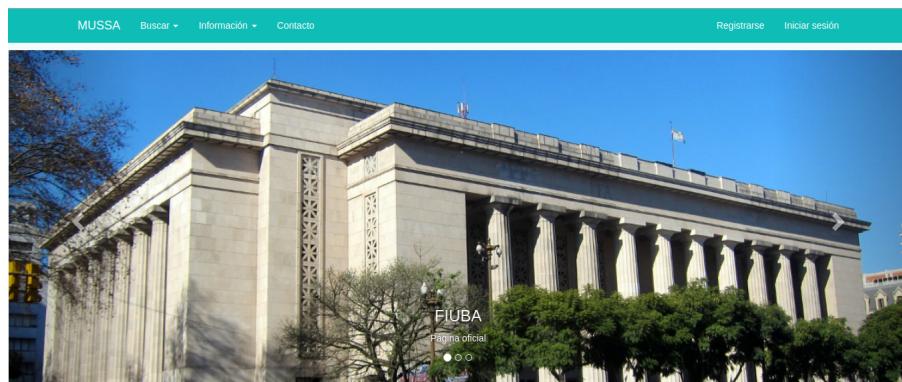


Figura 1: Página Inicio de MUSSA

Links Útiles

Links	Descripción
Facultad de Ingeniería	Página oficial de la Facultad de Ingeniería de la Universidad de Buenos Aires
Organizador de Horarios	Organizador de Horarios para que puedas probar combinaciones de horarios para el cuatrimestre actual en base a los horarios publicados por la facultad.
Campus	Campus virtual FIUBA. Algunas materias utilizan el campus como vía de comunicación o para compartir archivos, subir trabajos prácticos, etc.
FIUBA Consultas	Grupo de Facebook de alumnos de la FIUBA. En él se hacen consultas muy diversas (incluidas las de ingresantes), algunos chistes, información importante, etc.
Wiki Foros-FIUBA	Ejercicios, trabajos prácticos y exámenes de las distintas materias subidos por alumnos.
Foros FIUBA	Foro de alumnos de FIUBA. Mucha información ya ha quedado desactualizada pero es posible encontrar información sobre algunas materias y preguntas de diversa índole.
Solicitud de Título	Requisitos y trámites necesarios para tramitar el título FIUBA.
Seguimiento del Título	Requisitos y trámites necesarios para tramitar el título FIUBA.

Figura 2: Links Útiles

Comisiones por Carrera

Nombre	Carreras	Links
Casa Informática	Ingeniería en Informática	Lista de Correo
	Licenciatura en Análisis de Sistemas	Facebook
ComElec	Ingeniería en Electrónica	Lista de Correo
		Página Web

Figura 3: Comisiones por carrera

4.2. Búsqueda de Docentes

La búsqueda de docentes se realiza buscando por parte del apellido o parte del nombre (no requiere que comience con las letras indicadas sino que éstas se encuentren en el campo indicado).

Como resultado se listarán todos los docentes que cumplan con el criterio de búsqueda. Para ellos, se podrá visualizar el nombre (si lo tienen) y el apellido, además de las materias que dicta. Luego, desde allí se puede acceder a las encuestas asociadas a éste docente.

Nombre	Apellido	Materias	Encuestas
Diego	Essaya	• 7540 - ALGORITMOS Y PROGRAMACION I Curso: 4	

Figura 4: Búsqueda de docentes

4.3. Resultados de Encuestas de Docentes

Para un docente dado, se puede acceder a las encuestas del mismo. Allí, se mostrará el detalle de las materias que dicta con su curso correspondiente y se podrán acceder a las encuestas por cuatrimestre o ver todas reunidas en un mismo lugar.

Ya sea en la visualización por cuatrimestre o en la vista completa, los comentarios realizados al docente se agruparán por curso en que dicta.

Código	Materia	Curso	Carrera
6671	SISTEMAS GRAFICOS	1	10 - Ingeniería en Informática (Plan 1986)

Año	Cuatrimestre	Resultados de la encuesta
2017	2	
2017	1	
2016	2	
2016	1	
2015	1	
2014	2	
2014	1	

Figura 5: Lista de resultados de encuestas para un docente dado

Abbate

2 cuatrimestre / Año: 2017

► 6671 - SISTEMAS GRAFICOS | Curso 1 | 10 - Ingeniería en Informática (Plan 1986)

Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo.

Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem.

Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc.

Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo.

Figura 6: Resultados de encuestas docente para un cuatrimestre específico

4.4. Búsqueda de Materias

La búsqueda de materias se podrá realizar filtrando por los siguientes datos:

- Carrera
- Código de materia
- Nombre (o parte del nombre) de la materia
- Palabras clave

Buscar Materias

Código de materia:
7540

Nombre de la materia:

Carreras:

Todas las carreras
 09 - Licenciatura en análisis de sistemas
 10 - Ingeniería en Informática

Palabras Clave: Agregar

Buscar

Código	Materia	Tipo de materia	Carrera	Acciones
7540	ALGORITMOS Y PROGRAMACION I	OBLIGATORIA	09 - Licenciatura en análisis de sistemas (Plan 1986)	
7540	ALGORITMOS Y PROGRAMACION I	OBLIGATORIA	10 - Ingeniería en Informática (Plan 1986)	

Figura 7: Búsqueda de Materias

Para cada materia se podrán acceder a los datos generales de la misma, en los que se indicará la carrera, la cantidad de créditos, el link a la materia equivalente para otra carrera, las correlativas y el listado de cursos. En el listado

de cursos es posible visualizar el puntaje del mismo (están ordenados por este criterio), y desde allí se puede acceder a los resultados de las encuestas del curso.

6109 - PROBABILIDAD ESTADISTICA B

09 - Licenciatura en análisis de sistemas (Plan 1986)

Otras carreras en las que se dicta esta materia:
10 - Ingeniería en Informática (Plan 1986)

Creditos: 6

Materias correlativas:

Código	Materia
6103	ANALISIS MATEMATICO II A
6108	ALGEBRA II A

Figura 8: Visualización de una materia

Cursos:					
Los cursos marcados con * son cursos nuevos y aún no se cuenta con información completa de los cuatrimestres en los que estará disponible.					
Curso	Docentes	Horarios	Cuatrimestre en que es dictada	Puntaje	Encuestas
20	Altieri-Vera -Puebla	LUNES de 09:00 a 11:00 MIERCOLES de 09:00 a 13:00	Solo el 2º cuatrimestre	3.71	
22	Bello-Ferrari-Albert Raymundo	LUNES de 19:00 a 21:00 MIERCOLES de 19:00 a 23:00	Solo el 2º cuatrimestre	3.67	
29	Garcia-Pastor	MARTES de 09:00 a 13:00 JUEVES de 09:00 a 11:00	Solo el 2º cuatrimestre	2.60	
25	Martinez-Ferrari-Bouza	LUNES de 15:00 a 17:00 MIERCOLES de 15:00 a 19:00	Solo el 2º cuatrimestre	2.00	
* 23	Grynbarg-Ferrari-Bouza	LUNES de 15:00 a 17:00 MIERCOLES de 15:00 a 19:00	Solo el 2º cuatrimestre	0	

Figura 9: Cursos de una materia

4.5. Resultados de Encuestas de un curso

En los resultados de las encuestas de un curso, se pueden visualizar las mismas por cada cuatrimestre para el que haya encuestas finalizadas o como un compilado de todas las respuestas.

Resultados de Encuestas

6103 - ANALISIS MATEMATICO II A

12 | Sirne-Hernandez-Juarez

[Ver todos los resultados](#)

Año	Cuatrimestre	Resultados de la encuesta
2017	1	
2013	2	

Figura 10: Cursos de una materia

Los resultados de las encuestas por curso están divididos en 5 secciones: General, Contenido, Clases, Exámenes y Docentes. Algunas de las respuestas se visualizan como textos, otras como nubes de palabras, otras con gráficos de torta y/o de barras, otras con tablas y otras con mapas de calor. La forma de visualizar los datos dependerá del tipo de pregunta.

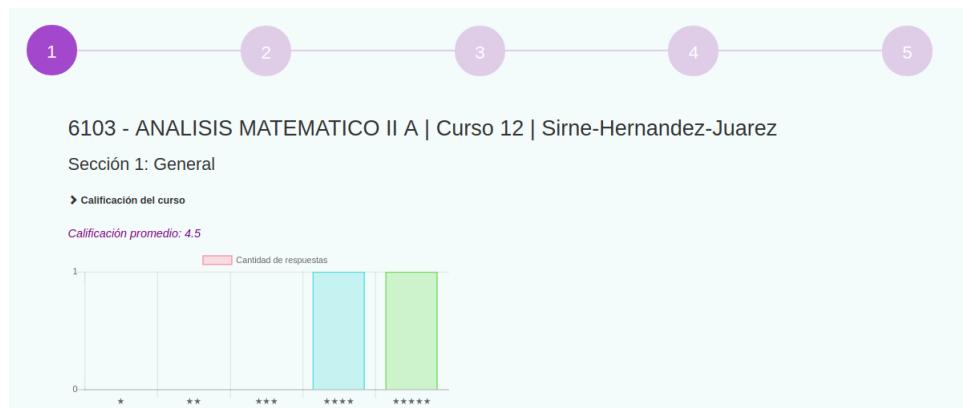


Figura 11: Secciones de los resultados de la encuesta

This figure shows a survey question and its response. The question is "¿Los temas se superponían con otras asignaturas?". The response text is: "Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem." Below this, another text block says: "Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus."

Figura 12: Ejemplo visualización de resultados de textos



Figura 13: Ejemplo visualización de resultados con gráfico de torat y de barras

» Materias que los alumnos creen que deberían ser cursadas antes de cursar esta materia.

Código	Materia	Cantidad de Respuestas
24	INTRODUCCION AL CONOCIMIENTO DE LA SOCIEDAD Y EL ESTADO	2
27	ALGEBRA	2
28	ANALISIS MATEMATICO	2
3	FISICA	2
40	INTRODUCCION AL PENSAMIENTO CIENTIFICO	2
5	QUIMICA	2

Figura 14: Ejemplo visualización de resultados con tabla



Figura 15: Ejemplo visualización de resultados con nubes de palabras

» Horario real de cursada

	13:00	13:30	14:00	14:30	15:00	15:30	16:00	16:30	17:00	17:30	18:00	18:30
LUNES	0	0	0	0	0	0	0	0	0	0	0	0
MARTES	0	1	2	2	2	2	2	2	2	2	1	0
MIERCOLES	0	0	0	0	0	0	0	0	0	0	0	0
JUEVES	0	0	0	2	2	2	2	2	2	1	0	0
VIERNES	0	0	0	0	0	0	0	0	0	0	0	0
SABADO	0	0	0	0	0	0	0	0	0	0	0	0

Figura 16: Ejemplo visualización de resultados con mapa de calor

5. Página Web MUSSA: Usuarios

5.1. Login / Sign In / Cambio de contraseñas

Tal como se indicó en la sección de herramientas y tecnologías para el desarrollo de esta sección fue utilizado Flask-User y luego customizado el mismo.

Para registrarse es necesario ingresar un e-mail, el nombre, apellido y una contraseña. Esto enviará un email a la dirección indicada para que confirme la registración. Una vez confirmada ya podrá ingresar normalmente al sistema utilizando solamente el e-mail y la contraseña. El padrón no es solicitado en la registración pero se podrá guardar desde los datos académicos, de esta forma los alumnos del CBC podrán hacer uso del sistema incluso antes de tener un padrón asignado.

The screenshot shows a registration form titled "Registro". It includes fields for Email, Nombre, Apellido, Contraseña, and Vuelva a introducir la clave. There is also a link to "¿Ya registrado? Identifíquese." and a "Registro" button.

Email	¿Ya registrado? Identifíquese.
Nombre	
Apellido	
Contraseña	
Vuelva a introducir la clave	
<input type="button" value="Registro"/>	

Figura 17: Registración de usuario

The screenshot shows a login form titled "Identificarse". It includes fields for Email and Contraseña, a "Recordarme" checkbox, and a "Identificarse" button. There are links for "Nuevo aquí? Regístrese." and "Olvidó su clave?".

Email	¿Nuevo aquí? Regístrese.
Contraseña	¿Olvidó su clave?
<input type="checkbox"/> Recordarme	
<input type="button" value="Identificarse"/>	

Figura 18: Inicio de sesión

En el caso de que el usuario haya olvidado su contraseña podrá ingresar a la sección correspondiente e ingresando el e-mail solicitar el cambio. Solo se le enviará el link de recupero de contraseña al e-mail que esté registrado. En ese e-mail recibirá el link para realizar el cambio de contraseña. Es posible también

cambiar la contraseña una vez ingresado al sistema en caso de que el usuario no la haya olvidado y simplemente desee modificarla desde el perfil del usuario.

The screenshot shows a light blue web page with a title 'Contraseña olvidada'. Below it is a field labeled 'Su dirección de correo' with a placeholder email input box. At the bottom is a blue button labeled 'Enviar mensaje de reinicio de clave'.

Figura 19: Recuperar contraseña

5.2. Perfil de usuario

El usuario registrado podrá modificar su nombre y/o apellido desde su perfil. Se decidió que no se almacenarán otros datos sensibles como el DNI, domicilio o teléfono.

Desde el registro académico el usuario podrá ingresar un padrón (no obligatorio). Si ingresa el padrón, éste debe ser único, es decir que no debe haber otro usuario que ya posea ese padrón. El alumno podrá registrar la/s carrera/s en la/s que se encuentra inscripto.

The screenshot shows a web page titled 'Registro Académico'. It has a search bar for 'Padrón' with an 'Actualizar' button. Below is a section titled 'Carreras en las que estoy inscripto' with a dropdown menu showing '10 - Ingeniería en Informática (1986)' and an 'Agregar' button. A table lists the student's inscriptions:

Código	Carrera	Plan	Acciones
09	Licenciatura en análisis de sistemas	1986	

Figura 20: Padrón del alumno y carreras en las que está inscripto

Al inscribirse en una carrera, se le habilitarán las materias de la misma para que pueda agregarlas como ‘En curso’, ‘Con Final Pendiente’, ‘Desaprobada’ o ‘Aprobada’. Si una materia se desaprueba, entonces se vuelve a habilitar la materia para ser agregada, pero ésta debe tener la cursada en un cuatrimestre posterior (ya que si fue aprobada no se puede volver a cursar). Al agregar una materia no se hace la verificación de correlativas ya que no se desea restringir la carga, especialmente porque esta versión del proyecto aún no incluye las excepciones de correlatividades.

Agregar Materia

Carrera:

09 - Licenciatura en análisis de sistemas (1986)

Materia:

6109 - PROBABILIDAD ESTADISTICA B

Curso:

20 | Altieri-Vera -Puebla

Estado:

Aprobada

Cuatrimestre de aprobación de cursada: Año:

1 2018

Calificación:

8

Fecha: 28/03/2018

Acta o Resolución:

58458-55522

Forma de Aprobación:

Examen

Guardar **Cancelar**

Figura 21: Agregar Materia Alumno

Cuando se carga una materia es posible elegir el curso en el que está siendo cursada / fue cursada dicha materia. En el caso de las materias del CBC no hay cursos disponibles. En el caso de que el curso en el que fue realizada la materia ya no esté disponible (o que el alumno ya no recuerde el curso por algún motivo), es posible indicar que no se seleccionará curso. Cuando la materia no tiene un curso seleccionado no genera entrada de encuesta, ya que cada encuesta está asociada a un curso en particular y no a la materia en general.

La visualización de las materias se puede realizar con las materias de todas las carreras al mismo tiempo, o filtrándolas por carrera. En caso de que se seleccione una carrera en particular (o solo se esté inscripto a una carrera), se mostrará además el progreso.

Mis Materias

[Agregar Materia](#)

Seleccioná la carrera para la cual querés visualizar las materias:

Todas mis carreras

Código	Materia	Carrera	Curso	Estado	Cuatrimestre / Año aprobación cursada	Nota Final	Fecha	Acta o Resolución	Forma de Aprobación	Acciones
24	INTRODUCCION AL CONOCIMIENTO DE LA SOCIEDAD Y EL ESTADO	10 - Ingeniería en Informática (Plan 1986)	Sin designar	Aprobada	1C / 2018	8	27/03/2018	53325235235	Examen	
6109	PROBABILIDAD ESTADISTICA B	10 - Ingeniería en Informática (Plan 1986)	25 Martinez-Ferrari-Bouza	Aprobada	1C / 2018	5	29/03/2018	523235523	Examen	
7113	INFORMACION EN LAS ORGANIZACIONES	10 - Ingeniería en Informática (Plan 1986)	001 Markdorf-Ducrey-Vever-Paez-Galli	Aprobada	1C / 2018	9	29/03/2018	325325523523	Examen	
6107	MATEMATICA DISCRETA	09 - Licenciatura en análisis de sistemas (Plan 1986)	1 Acero-Bolívar	Final Pendiente	1C / 2018	2	-	-	Examen	

Figura 22: Listado de materias de todas las carreras en las que el alumno está inscripto

El progreso indica el porcentaje de avance total de la carrera, el promedio obtenido y el porcentaje de avance en cada uno de los requerimientos de la carrera (cantidad de materias del CBC, créditos en electivas, créditos en materias obligatorias, créditos en materias de orientación (si corresponde), créditos en trabajo final de la carrera (tesis o trabajo profesional si corresponden)).

Mis Materias

[Agregar Materia](#)

Seleccioná la carrera para la cual querés visualizar las materias:

09 - Licenciatura en análisis de sistemas (1986)

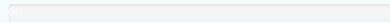
Promedio: 7.00

Avance:

3%

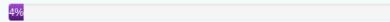
Materias CBC:

0 materias / 6 materias



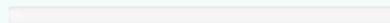
Materias Obligatorias:

6 créditos / 130 créditos



Materias Electivas:

0 créditos / 40 créditos



Código	Materia	Carrera	Estado	Cuatrimestre / Año aprobación cursada	Nota Final	Fecha	Acta o Resolución	Forma de Aprobación	Acciones
7113	INFORMACION EN LAS ORGANIZACIONES	001 Markdorf-Ducrey-Vever-Paez-Galli	Aprobada	1C / 2018	7	28/03/2018	32553523	Examen	
6107	MATEMATICA DISCRETA	1 Acero-Bolívar	Final Pendiente	1C / 2018	2	-	-	Examen	

Figura 23: Progreso y materias filtradas Licenciatura en Análisis de Sistemas

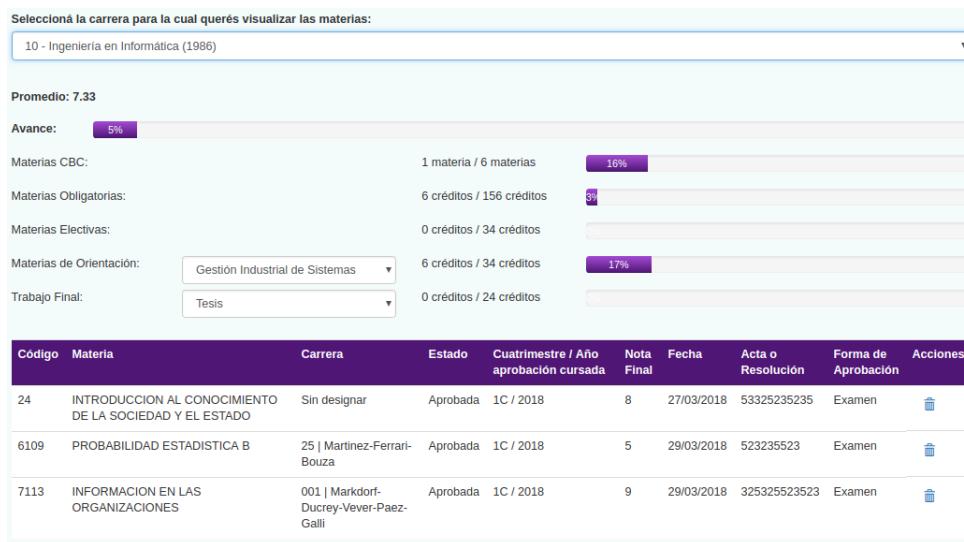


Figura 24: Progreso y materias filtradas Ingeniería en Informática

5.3. Materias habilitadas para cursar

En base a las materias que el alumno tiene ‘Aprobadas’ (no incluye las de final pendiente), se mostrará el listado de materias habilitadas para cursar. Como las correlatividades no son transitivas puede suceder que el alumno haya agregado que aprobó Análisis Matemático II pero aún no haya aprobado las materias del CBC, en ese caso se le mostrarán como habilitadas las materias del CBC ya que no poseen ninguna correlativa y todas aquellas materias que sólo hayan tenido a Análisis Matemático II como única correlativa. Posteriormente cuando en futuros desarrollos se agreguen las excepciones de correlatividades, en este punto se mostrarán también aquellas materias que pueden ser cursadas con pedido de excepción ya que se puede decir que estarían ‘habilitadas’.

Materias habilitadas para cursar

09 - Licenciatura en análisis de sistemas (Plan 1986)

Código	Materia	Tipo de materia	Ver
3	FISICA	CBC	
5	QUIMICA	CBC	
24	INTRODUCCION AL CONOCIMIENTO DE LA SOCIEDAD Y EL ESTADO	CBC	
27	ALGEBRA	CBC	
28	ANALISIS MATEMATICO	CBC	
40	INTRODUCCION AL PENSAMIENTO CIENTIFICO	CBC	

Figura 25: Listado de materias habilitadas para cursar separados por carrera inscripta

5.4. Formularios

El alumno puede generar formularios PDF. Los formularios disponibles para esta versión son la nota al decano (formulario más genérico que es solicitado para la mayoría de los trámites) y el listado de materias.

Generar Formularios

En esta sección podrás generar automáticamente los diferentes formularios disponibles:

Formulario	Generar
Listado de Materias	
Nota dirigida al decano	

Figura 26: Formularios disponibles

Para el caso del listado de materias se pueden seleccionar las carreras (de las que está inscripto) que desea incluir. Además se debe seleccionar si se desean incluir sólo las materias aprobadas y desaprobadas, las de final pendiente, las que están en curso o más de una de ellas.

Listado de Materias

Carreras:

09 - Licenciatura en análisis de sistemas (Plan 1986)
 10 - Ingeniería en Informática (Plan 1986)

Incluir:

Materias Aprobadas y Desaprobadas
 Materias con Final Pendiente
 Materias en Curso

Figura 27: Configuración formulario listado de materias

Para la nota dirigida al decano, se debe establecer el objeto y motivo de la nota. Además se deben indicar los datos personales y de contacto que son solicitados por este modelo. Opcionalmente, se puede ingresar una nota extendiendo los motivos de la solicitud. Tal como lo solicita el template publicado por la facultad, además de estos datos añadirá el listado de materias aprobadas, desaprobadas, con final pendiente y en curso según corresponda.

Nota dirigida al decano

Objeto:

Motivo de la solicitud:

Teléfono / Celular:

Domicilio:

Localidad:

DNI:

Año de ingreso a la Facultad:

Nota extendiendo el motivo de la solicitud (opcional):

Figura 28: Formulario: Nota dirigida al decano

5.5. Encuestas

Cuando se agrega una materia con final pendiente, aprobada o desaprobada con un curso específico, se crea una entrada para completar la encuesta correspondiente. Cada alumno tendrá su listado de encuestas realizadas y podrá contestar una sola vez la combinación [curso + cuatrimestre + año] que corresponde a la materia que ha agregado a su historial.

Mientras que haya encuestas pendientes, se mostrará un ícono en el menú correspondiente. Cuando las encuestas pendientes son finalizadas el ícono desaparecerá.

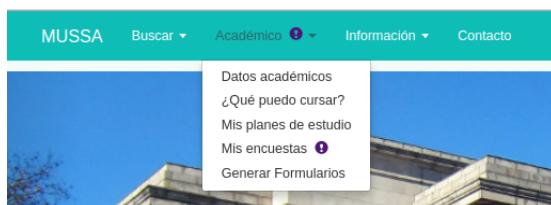


Figura 29: Notificación de encuestas pendientes

En la sección de encuestas se pueden encontrar las encuestas pendientes para ser completadas y visualizar las encuestas que ya fueron finalizadas por el alumno.

The screenshot shows a dashboard titled 'Mis Encuestas' with two main sections: 'Pendientes' (Pending) and 'Finalizadas' (Completed).

Pendientes:

Materia	Curso	Carrera	Aprobación de Cursada	Editar
6107 - MATEMATICA DISCRETA	1 Acero-Bolívar	09 - Licenciatura en análisis de sistemas (Plan 1986)	1C / 2018	

Finalizadas:

Materia	Curso	Carrera	Aprobación de Cursada	Ver
6109 - PROBABILIDAD ESTADISTICA B	25 Martinez-Ferrari-Bouza	10 - Ingeniería en Informática (Plan 1986)	1C / 2018	

Figura 30: Historial de encuestas / Encuestas pendientes

Cada encuesta a completar cuenta con preguntas en 5 secciones: General, Contenido, Clases, Exámenes y Docentes. Cada una de las secciones es guardada de forma parcial de forma que se pueda comenzar con la encuesta en un momento y continuarla más adelante. Una vez que la encuesta se finaliza (y se la guarda como tal) ya no podrá ser editada.

Las preguntas de las encuestas incluyen el puntaje del curso, horario real de cursada, dificultad de los temas, régimen de aprobación, comentarios de los docentes, etc.

Se incluye también una pregunta para indicar la temática de la materia (que será utilizada para clasificar las materias electivas en la generación del plan de carrera) y palabras clave (que son utilizadas en la búsqueda de materias).

The screenshot shows a survey form for course 6109 - PROBABILIDAD ESTADISTICA B | Curso 25 | Martinez-Ferrari-Bouza.

Section 1: General

Rating scale from 1 to 5:

- 1
- 2
- 3
- 4
- 5

Question: ¿Qué te pareció el curso en general?

Rating: ★★★★☆

Question: ¿Te aportó algo el curso por sobre tus conocimientos previos?

Response: Nada Mucho

Question: ¿Se superponen los temas con los de otras materias? Si es así, indicar qué temas y con cuáles asignaturas.

Text area: Esto es un texto de respuesta

Figura 31: Encuesta para completar

La encuesta ha sido modelada como un set de preguntas, donde cada pregunta puede ser de un tipo diferente y conforme a ello será la manera en que será renderizada en la web. Entre los tipos de preguntas se encuentran las de texto, horarios, correlativas, números, estrellas, puntaje, entre otras.

Los resultados de las encuestas que se pueden ver también desde el modo público, solo incluyen las encuestas que ya han sido finalizadas.

5.6. Generación y Visualización del Plan de Carrera

Para permitir el armado del plan, se deberán poder establecer las preferencias del alumno. Las preferencias básicas que pueden elegirse son:

- Carrera (Orientación y Trabajo Final de Carrera si corresponde)
- Máxima cantidad de materias por cuatrimestre
- Horarios en los que el alumno no puede cursar
- Cursos específicos en los que el alumno quiera cursar (puede elegir uno, ninguno o varios cursos). Se elige el curso pero no en qué cuatrimestre éste será cursado.

Además, en caso de tener materias con final pendiente, se deberá indicar cuándo se puede considerar la materia como aprobada, de forma tal que se restrinja el cuatrimestre mínimo de las materias que la tienen como correlativa.

Nuevo Plan de Estudios

Selecioná las preferencias con las cuales querés generar el nuevo plan.

>> Carrera:

09 - Licenciatura en análisis de sistemas (Plan 1986)
 10 - Ingeniería en Informática (Plan 1986)

>> Orientación elegida:

Gestión Industrial de Sistemas
 Sistemas Distribuidos
 Sistemas de Producción

>> Máxima cantidad de materias por cuatrimestre:

5

>> Horarios en los cuales no podés cursar:

LUNES De: 07:00 A: 07:30 Agregar horario

>> Indicar para cada materia con final pendiente a partir de qué momento puede ser considerada aprobada:

Código	Materia	Cuatrimestre en la que la materia estará aprobada
6108	ALGEBRA II A	Considerar la materia como aprobada

>> Si querés, podés elegir algunos cursos específicos en los que deseas cursar: ⓘ

Materia:

Selección una materia

Curso:

Selección un curso

Agregar curso

Materia	Curso	Eliminar
6211 - MECANICA RACIONAL	1 Faigon-Sambuco Salomone	

Figura 32: Preferencias para generación del plan de estudios

Adicionalmente, se podrán establecer las siguientes opciones avanzadas:

- Algoritmo de generación del plan: PLE (Programación Lineal Entera) o Greedy
- Máxima cantidad de cuatrimestres de duración del plan (el algoritmo siempre tratará de minimizarlos)

- Puntaje mínimo requerido para los cursos: Los cursos no puntuados tienen un punaje de 0. La restricción de punaje no es tenida en cuenta en el caso de los cursos que se seleccionan manualmente. Las materias que se elijan no tendrán un punaje menor al seleccionado a menos que sean obligatorias con un único curso disponible. En estos casos siempre se seleccionará en orden la de mayor punaje con mayor cantidad de encuestas completas.
- Máxima cantidad de horas de cursada, por semana
- Máxima cantidad de horas de trabajo extra además de la cursada, por semana
- Cuatrimestre y Año de inicio del plan (se selecciona el cuatrimestre y año actual por defecto). Este dato es importante no solo para la visualización sino también porque hay algunas materias que solo se dictan el 1º o 2º cuatrimestre.
- Preferencias de temáticas de las materias electivas: El porcentaje indicado será el porcentaje mínimo de cada temática respecto del total de créditos en electivas restantes que debe realizarse. Si una materia tiene más de una temática elegida, sus créditos serán contados para ambos porcentajes.

Opciones Avanzadas ▾

>> Algoritmo de confección del plan:

Greedy (más rápido) ⓘ
 PLE (solución óptima) ⓘ

>> Máxima cantidad de cuatrimestres de duración: ⓘ

>> Mínimo puntaje requerido para los cursos a seleccionar: ⓘ

>> Máxima cantidad de horas de cursada por semana: ⓘ

>> Máxima cantidad de horas extra que le querés dedicar por semana al conjunto de materias: ⓘ

>> Cuatrimestre y año de inicio del plan: ⓘ
 Número de cuatrimestre: Año:

>> Porcentaje de cada temática deseada con el que querés que sean elegidas las materias electivas: ⓘ

<input type="text" value="DATA MINING"/>	<input type="text"/>	<input type="button" value="Agregar"/>
Temática	Porcentaje	Eliminar
Cualquier temática	100 %	

Figura 33: Preferencias Avanzadas para generación del plan de estudios

Cuando el plan esté generado, se mostrará una notificación en el menú.



Figura 34: Notificación Plan Generado no visualizado

Es posible visualizar el listado de los planes de estudio que se han solicitado. Cada uno de ellos tendrá el estado ‘En curso’ (aún se está generando o está en la cola de espera), ‘Incompatible’ (no hay una solución óptima que cumpla con los parámetros indicados) o ‘Finalizado’ (plan compatible finalizado de generar).

Planes de Estudios			
Generar nuevo plan de estudios			
Fecha de generación del plan	Carreras	Estado	Acciones
3/4/2018	• 09 - Licenciatura en análisis de sistemas (Plan 1986)	Finalizado	
3/4/2018	• 10 - Ingeniería en Informática (Plan 1986)	En Curso	
3/4/2018	• 10 - Ingeniería en Informática (Plan 1986)	Incompatible	

Figura 35: Historial Planes Generados

Los planes que se encuentren en estado ‘Finalizado’ o ‘Incompatible’ se los podrá eliminar.

Los planes que se encuentren en estado ‘Finalizado’ pueden ser visualizados.

La visualización del plan de estudios consiste en el listado de los cuatrimestres con las materias para cada uno de ellos. En el caso de que se hayan cargado materias que no hayan sido contempladas en el momento de la generación del plan, éstas serán agregadas en el cuatrimestre correspondiente y eliminadas del asignado originalmente (si corresponde).

Las materias con estados diferentes se mostrarán con otros colores para facilitar la diferenciación entre ellas.

Plan de Estudio							
Fecha de creación: 3/4/2018							
Carreras:							
Código	Carrera						Plan
09	Licenciatura en análisis de sistemas						1986
1º Cuatrimestre de 2017							
Código	Materia	Carrera	Curso	Horarios	Puntaje	Estado	Encuestas
28	ANALISIS MATEMATICO	09	Sin curso asignado	-	Pendiente de cursar		
24	INTRODUCCION AL CONOCIMIENTO DE LA SOCIEDAD Y EL ESTADO	09	Sin curso asignado	-	Pendiente de cursar		
5	QUIMICA	09	Sin curso asignado	-	Aprobada		
2º Cuatrimestre de 2017							
Código	Materia	Carrera	Curso	Horarios	Puntaje	Estado	Encuestas
3	FISICA	09	Sin curso asignado	-	Final Pendiente		
40	INTRODUCCION AL PENSAMIENTO CIENTIFICO	09	Sin curso asignado	-	Pendiente de cursar		

Figura 36: Visualización Plan Generado

6. Página Web MUSSA: Acceso Administrador

El usuario con rol de administrador tiene habilitado un menú de Administración para los horarios, los cursos y los docentes.

6.1. Administrar Horarios

Los horarios y los cursos son cargados al sistema a través del PDF que es publicado por la Facultad todos los cuatrimestres. Esta es la única fuente de datos disponible desde la facultad para obtener los datos de los horarios completos, ya que la otra opción implica parsear una página web desde el sistema SIU Guaraní.

Cuando se carga el PDF se verifica que los horarios que se están cargando no se hayan cargado antes y que el cuatrimestre y año sea superior al último que se cargó (para que no se carguen horarios viejos modificando los datos actuales).



El formulario se titula "Cargar horarios de cuatrimestres". Contiene un campo para "Archivo de horarios" con un botón "Seleccionar archivo" y un mensaje "Ningún archivo seleccionado". Hay campos para "Número de cuatrimestre" (que muestra "1") y "Año" (que muestra "2018"). A continuación, hay un botón "Enviar".

Figura 37: Carga de Horarios PDF

Al principio se consultó con la facultad para poder tener un servicio y obtener estos datos (entre otros) y se tuvo una respuesta negativa.

El problema con este mecanismo es que el PDF la facultad lo publica y suele contener muchos errores que son corregidos conforme pasan las semanas en el SIU Guaraní, pero no se vuelve a publicar el archivo, por lo que los horarios erróneos deben modificarse manualmente con la administración de cursos. En versiones futuras, se desea agregar a los alumnos la opción de reportar cursos con datos incorrectos de forma de que se simplifique la tarea del administrador.

Debido a estos errores, lo ideal es utilizar el PDF actual como carga inicial y luego facilitar un administrador con un nuevo rol habilitado para cada departamento, de forma que cada departamento pueda hacer los cambios de los cursos que se han cambiado solo ese cuatrimestre y no de todos. Es ideal que los departamentos luego quieran utilizar el sistema, ya que podrían hacer uso de las encuestas para monitorizar los cursos correspondientes y poder hablar con los docentes ya sea para felicitarlos o para solicitarles que realicen ajustes en sus materias.

Otro problema que se tiene con el PDF es que los docentes que figuran en los cursos no son todos los que el curso realmente tiene, por lo que habría que agregar el resto del equipo docente manualmente si se quiere tener encuestas de ellos. Además, la única referencia al docente es su apellido, por lo que no es posible saber si el docente con apellido X que da la materia M1 es el mismo (o no) que el docente con apellido X que dicta la materia M2. Este link de docentes debe realizarse manualmente desde la administración de docentes.

6.2. Administrar Cursos

En la administración de cursos es posible visualizar y buscar los cursos por código de materia; ver los docentes que lo dictan, las carreras para las cuales está habilitado y en qué cuatrimestre se dicta.

Administrador Cursos							
Cursos guardados:							
Materia	Curso	Docentes	Carreras	Horarios	1º Cuatrimestre	2º Cuatrimestre	Acciones
6103	1	Acero-Porta	09 - Licenciatura en análisis de sistemas	MARTES de 09:00 a 13:00	NO	SI	
			10 - Ingeniería en Informática	JUEVES de 09:00 a 13:00			
6108	5B	Boggi-Oliva	09 - Licenciatura en análisis de sistemas	LUNES de 09:00 a 13:00	NO	SI	
			10 - Ingeniería en Informática	MIERCOLES de 09:00 a 13:00			

Figura 38: Administración de Cursos

Cuando se cargan los horarios PDF de un cuatrimestre, se dan de alta todos aquellos cursos nuevos (un curso es nuevo si el código/número de curso no existe para ese código de materia) y se marca como que se dicta ambos cuatrimestres, pero que es curso nuevo, registrando el cuatrimestre que fue actualizado (el 1º o el 2º). El curso seguirá siendo nuevo hasta que esté registrado en ambos cuatrimestres. Que esté registrado no quiere decir que se dicte en ambos, sino que se cargaron horarios tanto del 1º cuatrimestre como del 2º desde que el curso se dio de alta.

Si el curso ya existía, entonces se actualiza la lista de docentes y los horarios en los que es dictada la materia.

Al editar un curso, es posible modificar las carreras para las cuales está habilitado, agregar o quitar docentes del curso y modificar los horarios del mismo.

Modificar curso: 1 - Materia: 6103

Carreras habilitadas:

- 09 - Licenciatura en análisis de sistemas
- 10 - Ingeniería en Informática

Cuatrimestres en que se dicta la materia:

- Primer cuatrimestre
- Segundo cuatrimestre

Docentes:

Apellido, Nombre	Materias que dicta	Eliminar
Acero	6103 - ANALISIS MATEMATICO II A Curso 1 09 - Licenciatura en análisis de sistemas (Plan 1986)	
Porta	6103 - ANALISIS MATEMATICO II A Curso 1 09 - Licenciatura en análisis de sistemas (Plan 1986)	

Horarios:

Dia	De	A:	Eliminar
LUNES	07:00	07:30	
MARTES	09:00	13:00	
JUEVES	09:00	13:00	

Figura 39: Edición de un curso

6.3. Administrar Docentes

En la administración de docentes, es posible agrupar los docentes que se seleccionen. El agrupamiento sólo se permite si todos los docentes a agrupar tienen mismo apellido y nombre (o no tienen nombre). De esta forma, es posible solucionar el problema del docente con apellido X que dicta la materia M1 y otro docente con apellido X que dicta la materia M2, siempre que el administrador posea este conocimiento.

Administrar docentes

[Agrupar seleccionados](#)

Apellido	Nombre	Materias que dicta	Acciones
Zitto		6103 - ANALISIS MATEMATICO II A Curso: 4B 09 - Licenciatura en análisis de sistemas (Plan 1986)	<input checked="" type="checkbox"/>
Zitto		6103 - ANALISIS MATEMATICO II A Curso: 4A 09 - Licenciatura en análisis de sistemas (Plan 1986)	<input checked="" type="checkbox"/>
Zola		6608 - CIRCUITOS ELECTRONICOS I Curso: 1 10 - Ingeniería en Informática (Plan 1986)	<input type="checkbox"/>

Mostrando 681 a 683 de 683 filas [10](#) registros por página [1](#) ... [65](#) [66](#) [67](#) [68](#) [69](#) >

Figura 40: Administración de Docentes

Al editar un docente, es posible cambiar su nombre, apellido, y cursos que dicta. Se debe tener especial cuidado en que la modificación de estos, como el nombre y el apellido, ya que las encuestas que estén vinculadas a este id docente seguirán estando vinculadas pero mostrarán un nombre diferente del

que mostraban originalmente (mostrarán el nombre actualizado). Estos cambios deben realizarse sólo en caso de errores de escritura o de falta de datos y no para modificar un docente por otro.

Nombre
Mauricio

Apellido
Waldo

Carrera
09 - Licenciatura en análisis de sistemas (Plan 1986)

Materia
6108 - ALGEBRA II A

Curso
Seleccione un curso

Agregar materia

Código	Materia	Curso	Carrera	Acciones
6108	ALGEBRA II A	7A	09 - Licenciatura en análisis de sistemas (Plan 1986)	

Mostrando 1 a 1 de 1 filas

Figura 41: Edición de un Docente

7. Algoritmos para la generación del Plan de Estudios personalizado

Se analizaron varios algoritmos y se decidió utilizar un algoritmo que combina Greedy con un algoritmo de Fuerza Bruta; y por otro lado un algoritmo de Programación Lineal Entera. Éste último fue inspirado en un paper[1] en el cuál utilizaban este tipo de algoritmos para la asignación de horarios en base a la disponibilidad de aulas y docentes.

Se analizaron los papers[1, 2, 3, 4, 5, 6] sobre algoritmos de programación basados en preferencias, pero se decidió descartarlos ya que la similaridad con el algoritmo a desarrollar no era tan próxima. Estos algoritmos se basan en tomar decisiones basados en una preferencia del usuario a partir de otra decisión tomada; por ejemplo: si elegí ir en avión con la compañía C1 a la ciudad X, entonces prefiero pagar (si aún es posible) con la tarjeta T1 sobre la tarjeta T2 y sobre efectivo, ya que T1 aplica mejores descuentos para ir en avión con C1. En el caso del algoritmo del plan de estudios las preferencias no son condicionadas a otras decisiones sino que afectan al todo.

7.1. Arquitectura y Flujo General

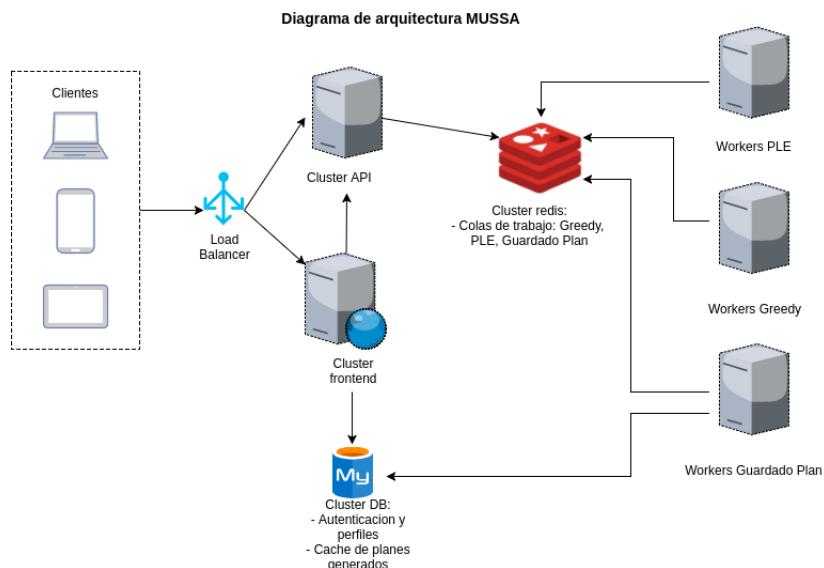


Figura 42: Arquitectura MUSSA

En la arquitectura elegida, se desacoplan las cuatro acciones necesarias para la generación del plan: la solicitud, el procesamiento, la persistencia y la notificación. Esto se logra transformando estas acciones en tareas asíncronas utilizando Celery como manejador de tareas y Redis como cola de trabajo.

Cada tarea, dispondrá de un cluster de servidores (workers) para el procesamiento de manera que se puede controlar la velocidad (throughput) mediante

la configuración de threads en paralelo y cantidad de workers.

Cuando se solicita la generación del plan de estudios desde el browser, se invoca a la API (es posible tener varias instancias levantadas en el servidor). Ésta, realiza la configuración de parámetros, filtrando las materias en base a las aprobadas, horarios incompatibles, etc.

Luego se genera el hash con los parámetros relevantes y se verifica si fue generado anteriormente un plan de estudios con esas mismas características. Si fue generado, se crea el plan copiando los datos de la base de datos de los ids de las materias para el nuevo plan. Sino, se envía la tarea al broker correspondiente donde es encolada. El broker dependerá del algoritmo elegido ya que ambos funcionan independientemente de forma tal que sea posible configurar en el servidor los recursos asignados según el uso y los tiempos que se vayan registrando a medida que se utilice el sistema.

Cuando el worker correspondiente esté disponible, tomará la tarea y generará el plan. Una vez generado, se crea la tarea de guardado que se encola en el broker de guardado del plan, que es el único que accede a la base de datos por pertenecer a la API.

Cuando el worker está disponible, se guarda el hash con los datos de las materias (además del plan solicitado) para poder ser utilizado la siguiente vez. De esta forma, aunque el algoritmo tarde varios minutos, la siguiente vez que se genere ya no será necesario todo este tiempo.

Una vez que el plan queda en estado finalizado, se notifica al usuario.

Por otra parte, se considera la posibilidad de escalar la capacidad del frontend y la API anteponiendo un balanceador de carga HTTP (por ejemplo haproxy⁶ o AWS ELB⁷) y la cantidad necesaria de servidores detrás. Esto permite agregar capacidad dinámicamente para soportar los picos de carga que se producen en períodos de inscripciones sin tener esa capacidad ociosa durante el resto del tiempo.

7.2. Configuración de Parámetros

Cuando se solicita la generación del plan de estudios, y antes de realizar el hash para verificar si el plan fue generado anteriormente o se requiere generar, se realiza un filtrado de las materias.

Se parte del plan completo para la carrera elegida y se actualizan los créditos requeridos en materias electivas, obligatorias de orientación etc. dependiendo de la orientación y trabajo final de carrera elegidos (si corresponde). Se cargan las materias incompatibles que determinarán aquellas materias que no pueden cursarse si se aprobó otra de ellas (por ejemplo, como sucede con las materias de idiomas que sólo es posible cursar una de ellas).

Se eliminan del listado de materias aquellas que el alumno ya aprobó y las que aún se encuentran con final pendiente pero se darán por aprobadas. En el caso de que la materia no se de por aprobada desde el momento de la generación del plan sino en cuatrimestres posteriores, se actualiza con estas últimas el cuatrimestre mínimo que requieren aquellas materias que tienen a la que se dará por aprobada como correlativa.

⁶<https://www.haproxy.org/>

⁷Amazon Web Services Elastic Load Balancer - <aws.amazon.com/elasticloadbalancing>

Se separan las materias del CBC (que siempre serán agregadas en grupos de tres al principio del plan generado) y se configuran como materias obligatorias aquellas de la orientación elegida y como electivas aquellas de las otras orientaciones (si corresponde). Además, se copia el trabajo profesional para dividirlo en dos ‘submaterias’ ya que debe realizarse en dos cuatrimestres consecutivos.

Se eligen los cursos específicos definidos por el usuario. Si la materia era electiva se la transforma en obligatoria con ese único curso disponible y se actualizan los créditos requeridos en materias electivas.

Se verifica que los horarios de los cursos que se van a agregar como disponibles cumplan con la restricción de los horarios en los que el alumno no puede cursar. Si no la cumplen, esos cursos son eliminados.

Se verifica además el puntaje del curso antes de agregarlo. Si el curso es de una materia electiva y no cumple con el puntaje mínimo, es descartado. Si la materia es obligatoria y no hay ningún curso que cumpla con el puntaje mínimo, se agregarán aquellos cursos con mayor puntaje (de forma que sea el más alto posible aunque no cumpla con la restricción).

Se actualizan los créditos en electivas, actualizando el valor a 0 si se hicieron más electivas que las requeridas. Y en el caso de haber completado los créditos en electivas, se eliminan del listado todas las materias electivas ya que no es necesario cursar ninguna más de ese tipo.

Se actualizan los créditos mínimos por temática. Estos valores se calcularán como la cantidad de créditos mínimos necesarios para cada tipo de temática aplicando el porcentaje a la cantidad de créditos mínimos en materias electivas general. Una materia puede tener más de una temática, en cuyo caso sus créditos contarán para todas las temáticas requeridas (que la materia también posea) de igual manera.

Se determina si el primer cuatrimestre será par o impar (debido a que no todas las materias se dictan en el 1º y 2º cuatrimestre), utilizando como datos de entrada el cuatrimestre de inicio del plan determinado por el usuario y la cantidad de cuatrimestres que ocuparán las materias del CBC (considerando que se distribuyen de hasta 3 materias por cuatrimestre independientemente de la configuración del usuario para el resto de la generación del plan).

Se verifica que todas las materias obligatorias tengan cursos disponibles (ya que pueden haber sido eliminadas por incompatibilidad de horarios con el usuario). En ese caso, se le muestra el error al usuario y se le indica el listado de materias obligatorias que requieren que elija el curso manualmente si desea mantener las restricciones actuales (ya que éstas no son aplicadas a los cursos elegidos manualmente) o para que modifique las restricciones.

Se eliminan del listado aquellas materias que hayan quedado sin cursos disponibles y se eliminan las incompatibles que ya no pertenezcan al plan (porque se las haya eliminado o aprobado en pasos anteriores).

Se verifica que los créditos en materias electivas (con las materias que han quedado disponibles) sean suficientes para cumplir con los créditos mínimos (tanto de forma general como por temática elegida). Si no son suficientes, entonces le muestra un mensaje al usuario indicando que no es posible generar el plan con esos parámetros (esto puede deberse a que la restricción de horarios es muy fuerte y por eso se eliminaron muchas materias electivas, porque no cumplen con el puntaje mínimo, etc.).

A continuación, se actualizan los horarios con las franjas mínimas y máximas y los días de la semana no utilizados. Al modificar los horarios para que

el horario de finalización sea en una franja menor, reduce las comparaciones en el caso del algoritmo Greedy, y en el caso del algoritmo de PLE reduce significativamente las ecuaciones.

Finalmente se genera el hash con los parámetros. Si este hash existe, el plan es copiado automáticamente y ya queda disponible para el usuario. En caso de que no exista, se envía la tarea a la cola Greedy o a la cola PLE según corresponda para que sea tratada cuando el worker esté disponible. En ambos casos, se le muestra al usuario el mensaje de que el plan fue enviado a generar y que se le notificará cuando esté finalizado. En el primer caso, esta notificación es inmediata.

7.3. Algoritmo Programación Lineal Entera (PLE)

7.3.1. Programación Lineal Entera: Qué es y cómo funciona

La programación lineal es una técnica matemática utilizada en la investigación de operaciones, que permite la optimización de una función objetivo a través de la aplicación de diversas restricciones a sus variables.

Un modelo de Programación Entera es aquel cuya solución óptima tiene sentido solamente si una parte o todas las variables de decisión toman valores restringidos a números enteros, permitiendo incorporar en el modelamiento matemático algunos aspectos que quedan fuera del alcance de los modelos de Programación Lineal.

En este sentido los algoritmos de resolución de los modelos de Programación Entera difieren a los utilizados en los modelos de Programación Lineal, destacándose entre ellos el Algoritmo de Ramificación y Acotamiento (o Branch & Bound), Branch & Cut, Planos Cortantes, Relajación Lagrangeana, entre otros.

Los modelos de Programación Entera se pueden clasificar en 2 grandes áreas: Programación Entera Mixta (PEM / MIP) y Programación Entera Pura (PEP / ILP). En el caso del algoritmo que se utilizará en este proyecto, es un algoritmo de Programación Entera Pura.

El primer paso para la resolución de un problema de programación lineal consiste en la identificación de los elementos básicos de un modelo matemático.

- Definir el objetivo
- Identificar las variables de decisión
- Identificar los datos del problema
- Identificar la función objetivo
- Identificar las restricciones

En principio, puede parecer que los problemas lineales enteros son más fáciles de resolver que los continuos, dado que el número de soluciones factibles a analizar, cuando el conjunto de oportunidades está acotado, es finito, éste número suele ser lo suficientemente grande como para que resulte imposible su comparación.

La mayoría de los métodos de resolución comienzan su ejecución con la resolución del problema lineal asociado (PLA) consistente en eliminar las condiciones de integridad, obteniéndose en consecuencia un problema de programación lineal que puede ser resuelto mediante el algoritmo simplex.

La resolución del PLA en primer lugar, tiene como ventaja que si la solución a dicho problema verifica las condiciones de integridad de las variables, ésa será la solución al problema entero.

Si la solución al PLA no verifica las condiciones de integridad, entonces se requiere utilizar otro método que permita resolver el problema entero.

El solver utilizado es CBC (COIN⁸ Branch and Cut solver). Este solver es open-source desarrollado en C++ para la resolución de problemas de programación entera mixta.

7.3.2. Branch & Cut

El método de Branch & Cut utiliza dos técnicas:

- Cutting Planes (Planos cortantes): Para resolver un problema de programación lineal entera (PLE), primero se consideran las relajaciones del problema (por lo general una relajación de programación lineal) y repetidamente se ‘cortan’ partes del politopo⁹ (agregando nuevas restricciones) con la esperanza de obtener una solución entera. Para encontrar los planos cortantes se debe resolver la separación del problema (separation problem) que implica encontrar una desigualdad válida del problema de PLE que haya sido violada.
- Branch & Bound (Ramificación y Acotamiento): Un árbol enumerado de todas las posibles configuraciones de variables es parcialmente atravesado, computando los límites locales y globales superiores e inferiores, los cuales son usados para evitar partes del árbol que no pueden producir el valor óptimo.

⁸El acrónimo completo es ‘COIN-OR’ (Computational Infrastructure for Operations Research) Por simplicidad utilizan solamente ‘COIN’.

⁹Objeto geométrico con lados planos. Generalización a cualquier dimensión de un polígono bidimensional o un poliedro tridimensional

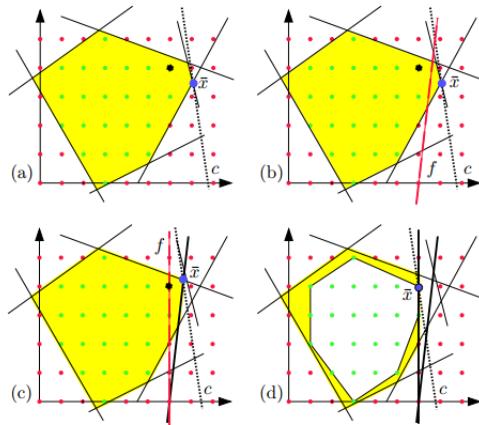


Figura 43: Ejemplo visual de planos cortantes

El casco convexo de todos los vectores de incidencia posibles (los puntos verdes en la imagen anterior) forman el problema del politopo P (área en blanco). Si se consideran las desigualdades del problema de PLE, generalmente describen un politopo más grande (área en amarillo), aunque este politopo no contiene un punto entero no factible (puntos rojos).

Si la solución \bar{x} de la relajación PL es integral, corresponde a un vector incidencia factible que representa una solución óptima.

De otra forma, se busca por una inecuación válida $f_x \leq f_0$ que ‘corta’ la solución \bar{x} . Por ejemplo,

$$f_y \leq f_0 \forall y \in P \quad (1)$$

$$f_{barx} > f_0 \quad (2)$$

El set $\{x \mid f_x = f_0\}$ es llamado plano cortante.

La búsqueda por un plano cortante es llamada la separación del problema. Cualquier plano cortante que es encontrado se agrega al programa lineal y éste es resuelto otra vez.

Hay un tipo especial de planos cortantes en los que se tiene interés, son los nombrados facetas del problema del politopo. Las facetas de un problema de un politopo d -dimensional son facetas de la dimensión $(d-1)$. Éstos son en un sentido los ‘mejores’ planos cortantes dado que están directamente ligados al problema del poliedro. Por ello, es importante identificar las desigualdades que definen facetas.

Se repite el paso de corte hasta que una solución entera que cumple todas las restricciones es encontrada (la cual resuelve el problema original de optimización), o hasta que no es posible encontrar una desigualdad que haya sido violada dentro de las clases de desigualdades válidas.

En este caso de utiliza el branch (ramificación). Esto es, se elige una variable x_i y se resuelven las dos subclases: el caso $x_i = 0$ y el caso $x_i = 1$. Esta aplicación repetida produce un árbol enumerado de posibles casos.

Se llama local al límite superior para el problema de PLE, si es obtenido desde un subproblema del árbol.

Si la solución encontrada para un subproblema es factible para el problema original y tiene el mejor puntaje que cualquier otra solución obtenida hasta el momento, entonces se guarda su valor para ser el nuevo límite global inferior para la función objetivo original.

Como consecuencia, solo se continúa con los subproblemas cuyo límite superior local es mayor o igual al límite global inferior.

El algoritmo Branch & Cut es un algoritmo de división y conquista que se approxima a resolver el problema dividiéndolo en problemas más pequeños.

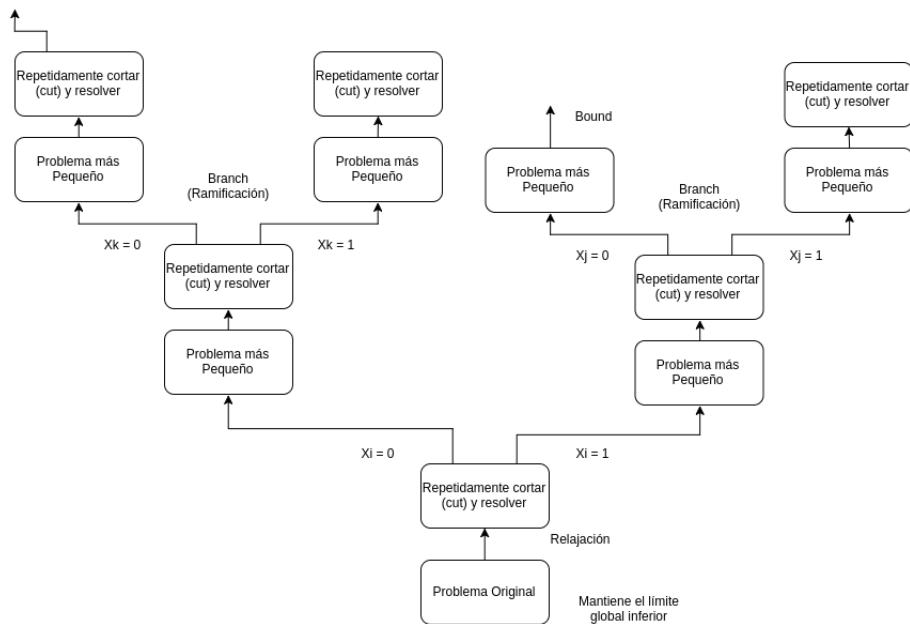


Figura 44: Esquema general del funcionamiento del algoritmo de Branch & Bound

Ventajas y Desventajas del Algoritmo

- Por utilizar las restricciones presentes en el PL desde límites inferiores anteriormente generados, se ahorra una gran cantidad de tiempo y de asignaciones de memoria ya que las restricciones no necesitan ser guardadas ni el tableau PL regenerada.
 - Cualquier plano cortante válido presente en el PL de límites inferiores previamente generados, puede ser usado para inicializar el set de restricciones cuando se genera el límite inferior actual. Esto significa que el proceso de enumeración no necesita comenzar desde cero cada vez. Así mismo, cuando se resuelve un nodo, la descripción de la solución del politopo es mejorada por todos los nodos de la búsqueda del árbol al mismo tiempo.
 - Dado que las restricciones son removidas del tableau PL, pueden necesitar ser regeneradas luego en la búsqueda del árbol. También, las restricciones

pueden ser removidas demasiado temprano, en cuyo caso el límite inferior puede no ser tan alto como podría haber sido. La ventaja de remover las restricciones es que el tamaño del problema PL es reducido, y con ello, el tiempo para hallar una solución y los requerimientos de memoria son reducidos drásticamente.

- Cuando un nodo hijo es removido de la lista y el límite inferior es generado, no hay garantía de que el límite inferior generado vaya a ser al menos igual al límite inferior guardado cuando el nodo hijo estaba situado en la lista. Esto sucede porque las restricciones en el problema PL son probablemente diferentes de cuando el nodo estaba guardado. Sin embargo, esto no será el caso usual, ya que las restricciones adicionales de un nodo hijo, comparadas con las del nodo padre, por lo general son más restrictivas.
- Para problemas grandes, el número de variables en el problema PL pueden ser lo suficientemente grandes para hacer el tiempo de solución prohibitivo y con ello, se reduce la efectividad del procedimiento de Branch & Cut. Consecuentemente, reducir el número de variables 0-1 a través de un preprocessamiento suele ser requerido.

7.3.3. Algoritmo Desarrollado

Para el desarrollo del algoritmo, se genera un archivo escrito en Python que hace uso de la librería PuLP para la resolución del problema; en este archivo se modela el problema, agregando las definiciones de las variables, de la función objetivo y de las restricciones que se deben cumplir.

Luego, estos datos son verificados, se identifican las variables igualadas a 0 y se las reemplaza en las ecuaciones, de forma de eliminar todas aquellas variables para las cuales ya se conoce el valor. Luego, todas aquellas restricciones que sean de una variable mayor o igual a cero o una variable menor igual a infinito, son eliminadas ya que son restricciones que se cumplen por la misma naturaleza de la variable.

Una vez que el código está limpio, es ejecutado con el solver CBC. El algoritmo puede determinar tanto que fue compatible (y tener una solución) o ser incompatible.

Desarrollo

Se definen las variables que van a ser utilizadas. Los valores que podrán tomar, dependerán de los parámetros de entrada.

Los cuatrimestres pueden tomar valores entre 1 y MAX_CUATRIMESTRES.

Los ids de las materias y los cursos, serán los ids de las materias y cursos disponibles para realizar el plan.

Los días disponibles son LUNES, MARTES, MIERCOLES, JUEVES, VIERNES y SABADO. Se pueden haber eliminado algunos días dependiendo de los parámetros de entrada.

Las franjas de horarios van entre FRANJA_MINIMA y FRANJA_MAXIMA, que fueron modificadas en base a los horarios de los cursos disponibles. El valor default para ellos son 1 y 33 respectivamente.

- Y_{ij} ($Y_{\{\text{id materia}\}_{-}\{\text{cuatrimestre}\}}$): La materia con id i se realiza en el cuatrimestre j.

$$Y_{ij} \in \{1,0\}$$

- $Y_{\text{TP_FINAL}_{ijk}}$ ($Y_{\text{TP_FINAL}_{\{\text{id materia}\}_{-}\{\text{codigo materia}\}_{-}\{\text{cuatrimestre}\}}}$): La materia de trabajo final con id i y código j se realiza en el cuatrimestre k. Es necesario identificar el código de la materia ya que fue doblada en parte A y B y ambas tienen igual id.

$$Y_{\text{TP_FINAL}_{ijk}} \in \{1,0\}$$

- C_i ($C_{\{\text{id materia}\}}$): Número de cuatrimestre en que se realiza la materia con id i. Si la materia no se realiza vale 0.

$$C_i \in \mathbb{Z}$$

- $C_{\text{TP_FINAL}_{ij}}$ ($C_{\text{TP_FINAL}_{\{\text{id materia}\}_{-}\{\text{codigo materia}\}}}$): Número de cuatrimestre en que se realiza la materia de trabajo final con id i y código j.

$$C_{\text{TP_FINAL}_{ij}} \in \mathbb{Z}$$

- TOTAL_CUATRIMESTRES: Total del cuatrimestres requeridos para completar el plan.

$$\text{TOTAL_CUATRIMESTRES} \in \mathbb{Z}$$

- CRED_i ($\text{CRED}_{\{\text{numero de cuatrimestre}\}}$): Cantidad de créditos obtenidos al final del cuatrimestre i.

$$\text{CRED}_i \in \mathbb{Z}$$

- H_{ijk} ($H_{\{\text{id materia}\}_{-}\{\text{id curso}\}_{-}\{\text{cuatrimestre}\}}$): La materia con id i se cursa en el curso con id j en el cuatrimestre k.

$$H_{ijk} \in \{1,0\}$$

- dia_{ij} ($\{\text{dia}\}_{-}\{\text{franja}\}_{-}\{\text{cuatrimestre}\}$): El día (LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO) en la franja horaria i, en el cuatrimestre j se está cursando (en algún curso, no importa cuál).

$$\text{dia}_{ij} \in \{1,0\}$$

- R_{mcdfk} ($R_{\{\text{id materia}\}_{-}\{\text{id curso}\}_{-}\{\text{dia}\}_{-}\{\text{franja}\}_{-}\{\text{cuatrimestre}\}}$): El horario para la materia con id m y el curso con id c, para ese día y esa franja horaria está habilitado (Solo valen 1 aquellos horarios en los que se dicta el curso y sólo en el caso de que se dicte en el cuatrimestre indicado).

$$R_{mcdfk} \in \{1,0\}$$

Luego, se define la función objetivo a minimizar. Por el momento, sólo se busca minimizar el total de cuatrimestres, que se encuentra representado con la variable entera: TOTAL_CUATRIMESTRES.

La primer restricción que tenemos es que la materia se debe cursar en un único cuatrimestre. Además, si la materia es OBLIGATORIA, debe cursarse sí o sí. Por lo que se desprenden dos ecuaciones, una para los ids de materias obligatorias (3) y otra para los ids de materias electivas (4).

La ecuación (3) se debe realizar para toda materia i obligatoria.

$$\sum_j Y_{ij} = 1; \forall j \in \{cuatrimestres\} \quad (3)$$

La ecuación (4) se debe realizar para toda materia i electiva.

$$\sum_j Y_{ij} \leq 1; \forall j \in \{cuatrimestres\} \quad (4)$$

Se debe establecer el número de cuatrimestre en que será cursada la materia, esto es, se deben sumar las variables Y_{ij} multiplicándolas por el cuatrimestre j, de forma tal que el número de cuatrimestre i sea igual a ese valor j cuando la variable Y sea positiva (lo cual solo ocurrirá a lo sumo una vez). En el caso de que la materia no se curse, el cuatrimestre dará 0. Esta ecuación se debe realizar para toda materia i.

$$C_i = \sum_j j * Y_{ij}; \forall j \in [1, MAX_CUATRIMESTRES] \quad (5)$$

Cuando la materia se cursa, los cuatrimestres de las correlativas deben ser menores. Como no todas las materias se cursan, es necesario agregar un ajuste para cancelar la ecuación cuando la materia no será cursada.

Para toda materia con id i disponible y toda materia m que tiene como correlativa a la materia i:

- Si la materia i es obligatoria y la materia m que la tiene de correlativa es obligatoria (6):

$$C_m \geq C_i + 1 \quad (6)$$

- Si la materia i es obligatoria y la materia m que la tiene de correlativa es electiva (9):

$$ajuste_electiva_no_cursada = INFINITO * (1 - \sum_{\forall j} Y_{mj}) \quad (7)$$

$$j \in [1, MAX_CUATRIMESTRES] \quad (8)$$

$$C_m + ajuste_electiva_no_cursada \geq C_i + 1 \quad (9)$$

- Si la materia i es electiva (12):

$$ajuste_electiva_no_cursada = INFINITO * (1 - \sum_{\forall j} Y_{mj}) \quad (10)$$

$$j \in [1, MAX_CUATRIMESTRES] \quad (11)$$

$$C_m + ajuste_electiva_no_cursada \geq C_i + \sum_{\forall j} Y_{ij} \quad (12)$$

$$j \in [1, MAX_CUATRIMESTRES] \quad (13)$$

Para calcular los créditos acumulados, se deben considerar los créditos de las materias obligatorias, electivas y las del trabajo profesional (si corresponde).

El valor de $CRED_0$ será el valor de los créditos acumulados con las materias aprobadas antes de invocar al algoritmo.

Para cada cuatrimestre $j \in [1, MAX_CUATRIMESTRES]$, con $i \in \{id\}$ de materia} y k código de materia de trabajo profesional asociada a i (La sumatoria de las materias del trabajo final solo es agregada si existen materias de este tipo) (14):

$$\sum_{\forall i} Y_{ij} * creditos_i + \sum_{\forall(i,k)} Y_TP_FINAL_{ikj} * creditos_{ik} + CRED_{(j-1)} = CRED_j \quad (14)$$

Para todas las materias que requieren una cantidad de créditos mínimos para poder ser cursadas, se pide que los créditos acumulados en el cuatrimestre anterior al que se hace la materia sean mayores que los requeridos por la misma.

Así, \forall materia con id i que requiera créditos mínimos para ser cursada, $\forall j \in [1, MAX_CUATRIMESTRES]$ (15):

$$creditos_minimos_aprobados_i * Y_{ij} \leq CRED_{(j-1)} \quad (15)$$

Se posee una restricción para la cantidad máxima de materias que se pueden realizar por cuatrimestre. Para cada cuatrimestre $j \in [1, MAX_CUATRIMESTRES]$, con $i \in \{id\}$ de materia} y k código de materia de trabajo profesional asociada a i (La sumatoria de las materias del trabajo final solo es agregada si existen materias de este tipo) (16):

$$\sum_{\forall i} Y_{ij} + \sum_{\forall(i,k)} Y_TP_FINAL_{ikj} \leq MAX_CANT_MATERIAS_CUATRIM \quad (16)$$

Se restringe además, la máxima cantidad de horas de extra de cursada (18) de la misma manera:

$$horas_extras_tp_final = \sum_{\forall(i,k)} Y_TP_FINAL_{ikj} * horas_extras_{ik} \quad (17)$$

$$\sum_{\forall i} Y_{ij} * horas_extras_i + horas_extras_tp_final \leq MAX_HORAS_EXTRA \quad (18)$$

Para restringir la máxima cantidad de horas de cursada (19) se restringe en base a la variable H que muestra si se está cursando el curso o no. Para cada cuatrimestre $j \in [1, MAX_CUATRIMESTRES]$. Con $i \in \{id\}$ de materia} y k id de los cursos asociados a la materia i:

$$\sum_{\forall i} \sum_{\forall k} H_{ikj} * horas_cursada_{ik} \leq MAX_HORAS_CURSADA \quad (19)$$

Para el cálculo del total de cuatrimestres, que es la función a minimizar se calcula como el máximo de todos los números de cuatrimestres de las diferentes materias (20). Con $i \in \{\text{id de materia}\}$ y k código de materia de trabajo profesional asociada a i (La sumatoria de las materias del trabajo final solo es agregada si existen materias de este tipo):

$$TOTAL_CUATRIMESTRES = \max(C_1, C_2, \dots, C_i, C_TP_FINAL_{ik}) \quad (20)$$

Relacionados a los cursos seleccionados para las materias, se desprenden varias ecuaciones.

Si se elige un curso de una materia, se debe elegir su horario completo (esto es, todas sus franjas horarias) (21). Para cada cuatrimestre j ; para cada i id de materia; para cada k id de curso que es dictado en el cuatrimestre (algunas cursos de las materias solo se dictan en cuatrimestres pares, otras solo en impares y otras en ambos); para cada día d ; para cada franja horaria f entre FRANJA_MIN y FRANJA_MAX:

$$H_{ikj} = R_{ikdfj} \quad (21)$$

No se puede cursar en más de un lugar a la vez (22). Para ello, se toma la variable que indica si se está cursando en el día en determinada franja y se lo iguala a la sumatoria de las variables que indican si el curso se está tomando en dicha franja horaria ese día. Como la primera solo puede valer 0 o 1, solo un curso podrá ser cursado en dicha franja. Para cada cuatrimestre j ; para cada día d ; para cada franja horaria f entre FRANJA_MIN y FRANJA_MAX. Con $i \in \{\text{id materia}\}$, y $k \in \{\text{id curso para la materia } i\}$:

$$\sum_{\forall i} \sum_{\forall k} R_{ikdfj} = d_{fj} \quad (22)$$

Si la materia no se cursa en ese cuatrimestre, entonces no debe cursarse en ninguno de sus cursos para ese cuatrimestre (23). Para cada cuatrimestre j ; para cada $i \in \{\text{id materia}\}$, y $k \in \{\text{id curso para la materia } i\}$:

$$Y_{ij} \geq H_{ikj} \quad (23)$$

La materia deberá ser cursada en un único curso (24). Para cada cuatrimestre j ; para cada $i \in \{\text{id materia}\}$. Con $k \in \{\text{id curso para la materia } i\}$:

$$\sum_{\forall k} H_{ikj} = Y_{ij} \quad (24)$$

No todas las materias se dictan ambos cuatrimestres (25). Por ello, si la materia no es dictada en el cuatrimestre indicado, se iguala a 0 la variable para que no pueda ser elegida. Para cada cuatrimestre j ; para cada $i \in \{\text{id materia}\}$; para $k \in \{\text{id curso para la materia } i\}$, cuando la combinación no sea válida ese cuatrimestre:

$$H_{ikj} = 0 \quad (25)$$

Se deben cumplir una cantidad de créditos mínimos en materias electivas (26). Además, se restringirán las mismas para que no se hagan materias extras

(27). Para ello, se limitarán a un máximo de 6 créditos extras (una materia en términos generales) debido a que se pueden elegir algunas materias electivas de 8 créditos o combinaciones similares que sean mejores debido a sus horarios. Solo se agregarán materias si es requerido y no aumenta la cantidad de cuatrimestres, ya que el total de cuatrimestres es el valor que se está minimizando en la función objetivo. Con j cuatrimestre entre 1 y MAX_CUATRIMESTRES; con $i \in \{\text{id materia}\}$:

$$\sum_{\forall j} \sum_{\forall i} Y_{ij} * creditos_i \geq CREDITOS_MINIMOS_ELECTIVAS \quad (26)$$

$$\sum_{\forall j} \sum_{\forall i} Y_{ij} * creditos_i \leq CREDITOS_MINIMOS_ELECTIVAS + 6 \quad (27)$$

No todas las materias son comptaibles (28), por ejemplo como sucede con el caso de las materias de idiomas. En este caso, si se elige una materia i que es incompatible con otras materias m , todas las materias m no deben ser cursadas. Para cada materia con id i que tiene materias incompatibles. Con j cuatrimestre entre 1 y MAX_CUATRIMESTRES; Con m id materia incompatible con i :

$$\sum_{\forall j} (Y_{ij} + \sum_{\forall m} Y_{mj}) \leq 1 \quad (28)$$

En el caso de que el alumno tenga materias con final pendiente, establecerá en los parámetros en qué cuatrimestre puede ser considerada como aprobada. Con ello, se restringirán a todas las materias con id i que la hayan tenido como correlativa a que su cuatrimestre sea mayor o igual al cuatrimestre posterior a que la correlativa fue aprobada (29) (cuatrimestre mínimo de la materia i).

$$C_i \geq cuatrimestre_minimo_i \quad (29)$$

En el caso de que exista un trabajo final para la carrera, se añaden las restricciones para el mismo.

Cada parte del trabajo final debe cursarse en un único cuatrimestre (30). Para cada combinación de id i del trabajo final y el código k asociado a éste. Con j cuatrimestre entre 1 y MAX_CUATRIMESTRES:

$$\sum_{\forall j} Y_TP_FINAL_{ikj} = 1 \quad (30)$$

El número de cuatrimestre en el que es cursada cada parte del trabajo final se calcula de la siguiente manera (31). Para cada combinación de id i del trabajo final y el código k asociado a éste. Con j cuatrimestre entre 1 y MAX_CUATRIMESTRES:

$$\sum_{\forall j} Y_TP_FINAL_{ikj} * j = C_TP_FINAL_{ik} \quad (31)$$

Para poder cursar las partes del trabajo final, se requieren un mínimo de créditos (32). Para cada combinación de id i del trabajo final y el código k asociado a éste; para todo j cuatrimestre entre 1 y MAX_CUATRIMESTRES:

$$creditos_minimos_aprobados_{ik} * Y_TP_FINAL_{ikj} \leq CRED_{(j-1)} \quad (32)$$

Por último, la parte 1 y 2 de la materia del trabajo final i deben realizarse en cuatrimestres consecutivos (33):

$$C_TP_FINAL_{i_PARTE_1} + 1 = C_TP_FINAL_{i_PARTE_2} \quad (33)$$

7.4. Algoritmo Greddy

Para este algoritmo se utiliza una combinación de Greedy y Fuerza Fruta. Para el Greedy, se ordenan las materias por un determinado criterio y se las selecciona en orden si éstas encajan con las anteriores. En el caso de que las combinaciones de cursos disponibles sean relativamente pocas, se busca la mejor combinación posible a través de la fuerza bruta con poda. De esta forma se elige la mejor combinación que además permita tener mayores opciones en el cuatrimestre siguiente.

La selección Greedy genera un cuatrimestre válido con las materias disponibles mientras que no se haya finalizado el plan o superado la cantidad máxima de cuatrimestres. Las materias disponibles son una concatenación entre las materias obligatorias, las materias electivas primarias y las electivas secundarias.

Las materias electivas primarias son aquellas que suman créditos para las temáticas solicitadas. Las materias electivas secundarias son aquellas que no suman créditos para las materias de las temáticas elegidas. Si no hay temáticas elegidas, o los créditos por temática han sido cumplidos, todas las materias electivas son primarias.

Las materias obligatorias son ordenadas según:

- Menor Horario de finalización
- Mayor cantidad de materias que la tienen de correlativa: Cuántas materias libera porque la tienen de correlativa
- Mayor cantidad de franjas ocupadas
- Mayor punaje
- Mayor cantidad de créditos
- Menor código de materia

El horario de finalización entre dos materias no es trivial ya que cada materia puede ser dictada en diferentes días y en más de un día, por lo que una materia puede tener menor horario que la otra para algun dia de la semana pero la otra tener más cantidad de horarios que terminen antes, o que se dicten en diferentes cantidades de días, etc. Se decidió que la comparación de horarios sería realizada de la siguiente forma:

La máxima franja de un curso en todos sus días de cursada comparada con la máxima del otro. Por ejemplo si el curso A se cursa Lunes en las franjas 1 a

15 y Viernes en las franjas 3 a 17 su máxima franja será 17. Si el curso B de cursa los sabados en las franjas 13 a 18, su máxima franja será 18. En este caso el curso A es menor que el curso B, aunque A tenga curso varios días.

Si ambos tuviesen la misma franja máxima, entonces un curso será menor que otro si se cursa menos días. Por ejemplo, si A y B hubiesen tenido la misma franja máxima, B hubiese sido menor que A ya que se cursa un solo día.

Si cursan la misma cantidad de días, entonces será menor aquel que para la mayor cantidad de días tenga menor horario de finalización.

Las materias electivas son ordenadas según:

- Menor Horario de finalización
- Mayor cantidad de créditos en temáticas (suma sus créditos tantas veces como a temáticas aporte)
- Mayor cantidad de créditos
- Menor cantidad de horas de cursada
- Mayor cantidad de materias que la tienen de correlativa: Cuántas materias libera porque la tienen de correlativa
- Mayor puntaje
- Menor código de materia

Luego, de las materias disponibles se verifica si encaja en los horarios libres. Si encaja (y no es otro curso de una materia ya colocada, no es incompatible con otra ya elegida, etc), se agrega la materia al cuatrimestre actual y se ocupan las franjas horarias correspondientes. Cuando el cuatrimestre está completo porque ya se tienen el máximo de materias u horas establecidas, o ya no hay materias disponibles, se cierra el cuatrimestre y se lo agrega al plan. Para el cuatrimestre siguiente se vuelven a obtener las materias disponibles considerando a las materias agregadas en el último cuatrimestre como aprobadas.

En caso de que se hayan cumplido los créditos requeridos en electivas, las materias de este tipo son eliminadas del conjunto total pendiente para reducir los tiempos de los recorridos en las materias, ya que éstas ya no serán requeridas.

Cuando las combinaciones posibles de los cursos son pocas (en este momento se está utilizando un valor de 15 cursos con 5 materias máximo ($15 * 14 * 13 * 12 * 11$)), se realiza el algoritmo de Fuerza Bruta con poda.

Este algoritmo toma las materias disponibles y realiza las combinaciones considerando que se agrega la materia y considerando que no se agrega. Para cada materia disponible se toman las combinaciones anteriores y se hace lo propio.

Cabe destacar que no todas las materias generarán una combinación, ya que no todas pueden ser añadidas por ser incompatibles con las materias agregadas a la combinación anteriormente, por lo que la cantidad de casos se reduce.

De todas las combinaciones generadas se elige la mejor para el cuatrimestre y se la agrega al plan. La mejor combinación se decide en base a:

- Mayor cantidad de materias

- Mayor cantidad de materias obligatorias que libera
- Mayor cantidad de créditos en temáticas cubiertos
- Mayor cantidad de créditos en electivas
- Mayor cantidad de créditos totales
- Menor cantidad de horas de cursada
- Menor cantidad de horas extra

Durante el desarrollo se realizó el profile del código del algoritmo de generación del plan Greedy, obteniéndose un promedio de 150 segundos para el plan completo de licenciatura y un total de 76093034 llamadas a funciones ejecutadas. Las funciones que más tiempo estaban demorando eran las funciones que realizaban copias de los parámetros para la ejecución de la fuerza bruta con poda, éstos parámetros son modificados durante la ejecución del programa por lo que se requiere la copia.

Revisando el código se observó que no solo se copiaban las clases que eran modificadas (los parámetros y las materias con sus correlativas) sino que además se estaban copiando los cursos y sus horarios. Estos datos no requieren copia ya que no serán modificados, por lo que basta con hacer una copia de la lista de referencias y no del objeto en sí.

Eliminando la copia de esos objetos innecesarios, se obtuvo un promedio de 35 segundos para la misma ejecución del plan de Licenciatura completo, con un total de 12922025 llamadas a funciones ejecutadas.

7.5. Pruebas

Para las pruebas se aplicó tanto el algoritmo Greedy como el algoritmo de PLE sobre un set de pruebas realizado con el plan completo de Licenciatura en Análisis de Sistemas (Plan 1986) en el cual se iban dando por aprobadas las materias sugeridas por el plan de estudios oficial. Se excluyó la sección del CBC ya que ésta es tratada de forma independiente del algoritmo. Sobre cada cuatrimestre, se restringió a un máximo de 5, 4, 3 y 2 materias por cuatrimestre respectivamente. Las horas máximas de cursada y de horas extras, se las dejó fijas en 28hs para no afectar a la variable de máxima cantidad de materias por cuatrimestre.

El set de pruebas, fue ejecutado en la misma máquina con la que se hizo el desarrollo de los algoritmos y del sistema, la cual posee las siguientes características. Esta máquina la rotularemos como M1.

- CPU: Intel(R) Core(TM) i3-2330M CPU @ 2.20GHz
- RAM: 10 GB
- 2 Core

En particular, para la configuración de la cantidad de threads para el algoritmo de PLE fue seteada en 2 ya que es la cantidad de Cores que posee.

Los resultados de la ejecución pueden verse en la siguientes tablas:

Total cuatrimestres	Tiempo de generación	Cantidad de Materias disponibles	Max. materias por cuatrimestre
2	0.69 seg	21	2
4	1.99 seg	25	2
7	5.41 seg	30	2
10	13.91 seg	35	2
12	18.91 seg	39	2
13	21.59 seg	42	2
15	32.49 seg	45	2
2	2.07 seg	21	3
3	4.79 seg	25	3
5	15.16 seg	30	3
7	47.31 seg	35	3
8	56.28 seg	39	3
9	58.43 seg	42	3
10	41.74 seg	45	3
2	0.04 seg	21	4
3	0.38 seg	25	4
5	0.89 seg	30	4
5	0.56 seg	35	4
7	1.87 seg	39	4
8	1.80 seg	42	4
9	1.26 seg	45	4
1	0.01 seg	21	5
3	0.02 seg	25	5
4	0.03 seg	30	5
5	0.03 seg	35	5
8	0.17 seg	39	5
9	0.11 seg	42	5
10	0.14 seg	45	5

Cuadro 1: Ejecucion algoritmo GREEDY (M1)

Total cuatrimestres	Tiempo de generación	Cantidad de Materias disponibles	Max. materias por cuatrimestre
2	13.93 seg	21	2
4	21.56 seg	25	2
7	28.50 seg	30	2
9	1.0 min, 1.55 seg	35	2
11	2.0 min, 15.63 seg	39	2
13	2.0 min, 44.04 seg	42	2
14	11.0 min, 6.16 seg	45	2
2	16.22 seg	21	3
3	20.57 seg	25	3
5	38.09 seg	30	3
6	46.32 seg	35	3
8	1.0 min, 44.95 seg	39	3

Total cuatrimestres	Tiempo de generación	Cantidad de Materias disponibles	Max. materias por cuatrimestre
9	2.0 min, 45.50 seg	42	3
10	9.0 min, 16.35 seg	45	3
1	15.28 seg	21	4
3	16.23 seg	25	4
4	30.50 seg	30	4
5	57.21 seg	35	4
6	1.0 min, 6.32 seg	39	4
7	49.0 min, 17.57 seg	42	4
7	7.0 min, 31.78 seg	45	4
1	19.96 seg	21	5
2	17.25 seg	25	5
3	35.20 seg	30	5
4	1.0 min, 2.48 seg	35	5
5	1.0 min, 7.96 seg	39	5
6	4.0 min, 1.08 seg	42	5
7	4.0 min, 53.44 seg	45	5

Cuadro 2: Ejecucion algoritmo PLE (M1)

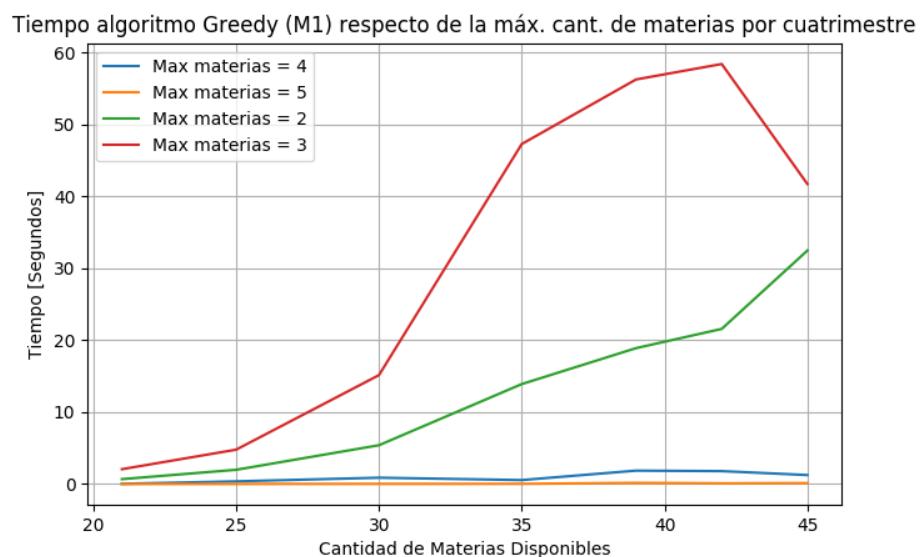


Figura 45: Comparación de tiempos respecto del máximo de materias por cuatrimestre - Algoritmo Greedy - Máquina M1

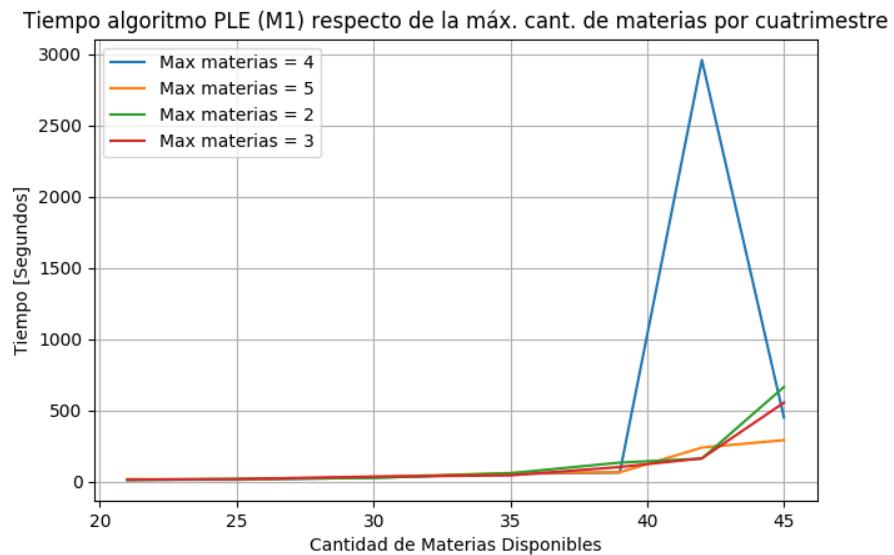


Figura 46: Comparación de tiempos respecto del máximo de materias por cuatrimestre - Algoritmo PLE - Máquina M1

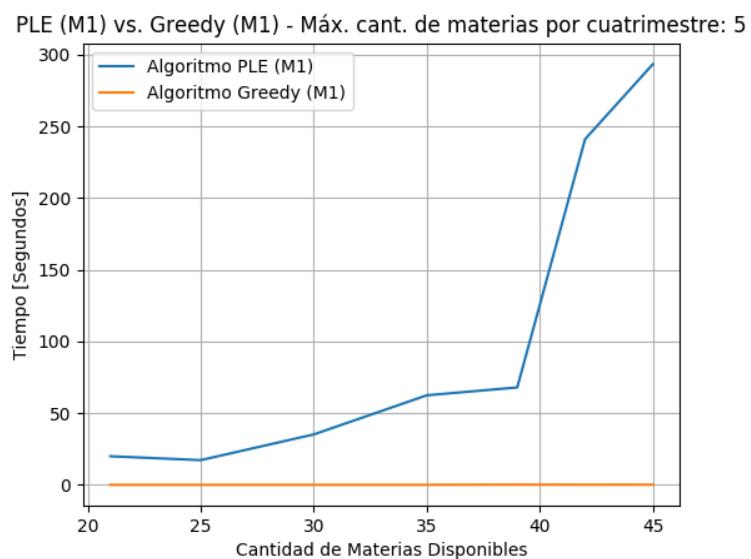


Figura 47: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 5 - Máquina M1

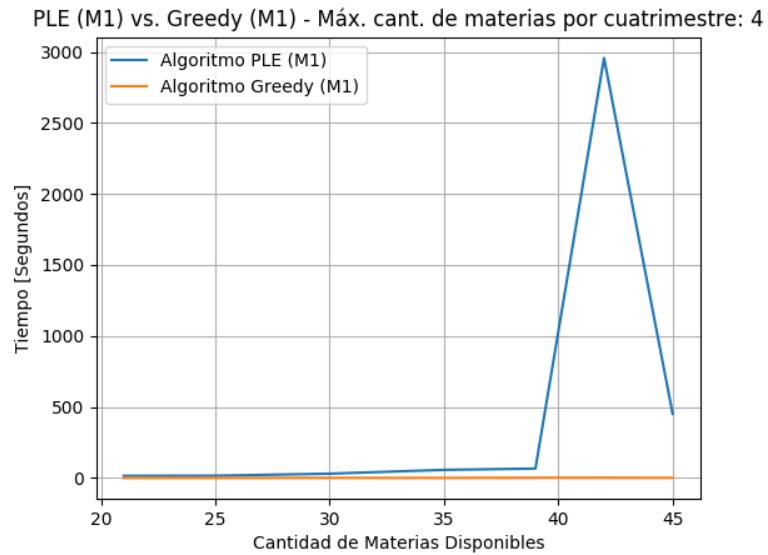


Figura 48: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 4 - Máquina M1

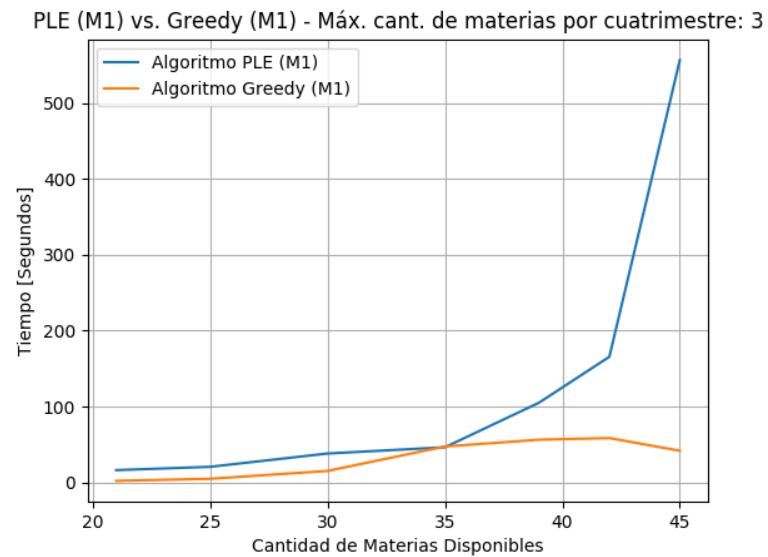


Figura 49: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 3 - Máquina M1

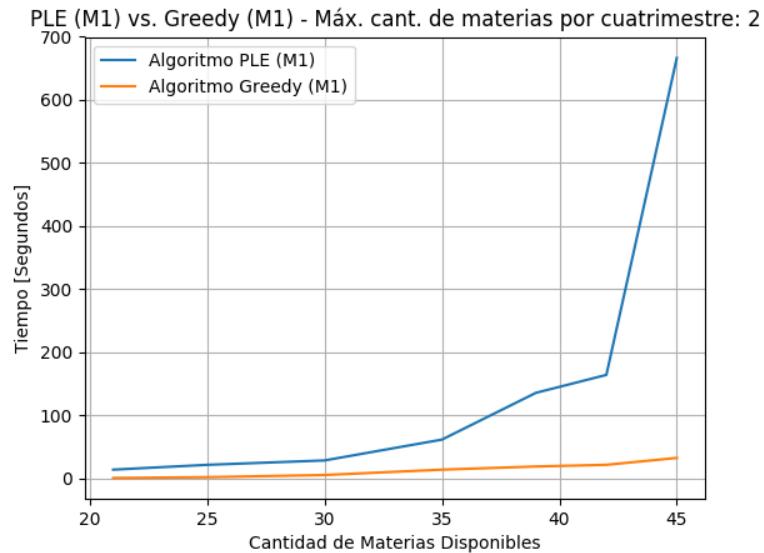


Figura 50: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 2 - Máquina M1

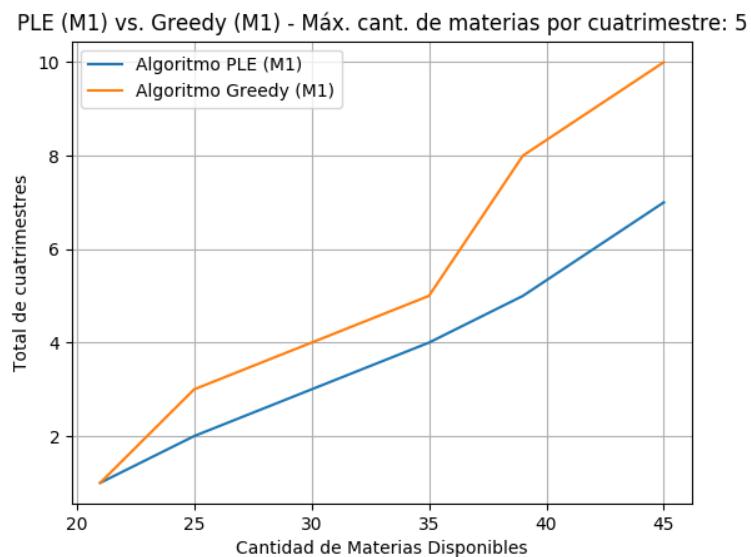


Figura 51: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 5 - Máquina M1

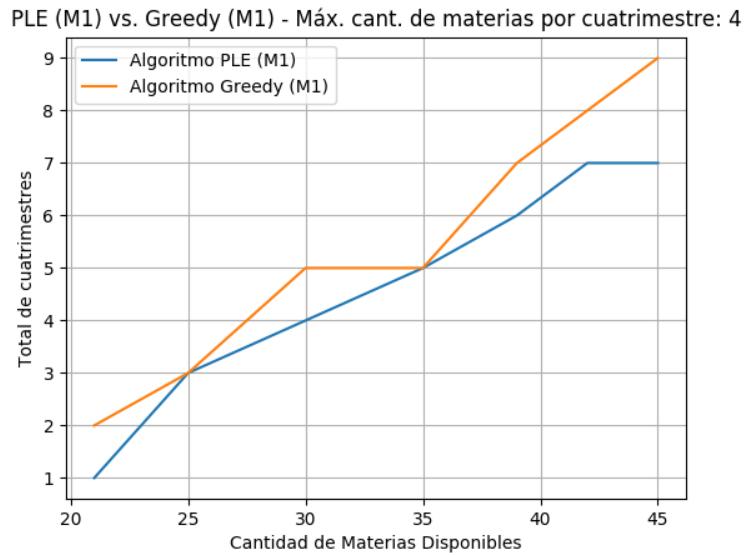


Figura 52: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 4 - Máquina M1

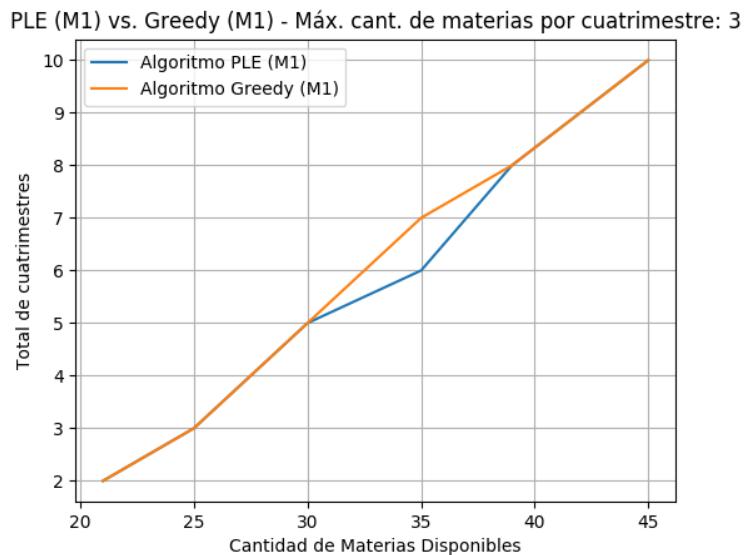


Figura 53: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre:3 - Máquina M1

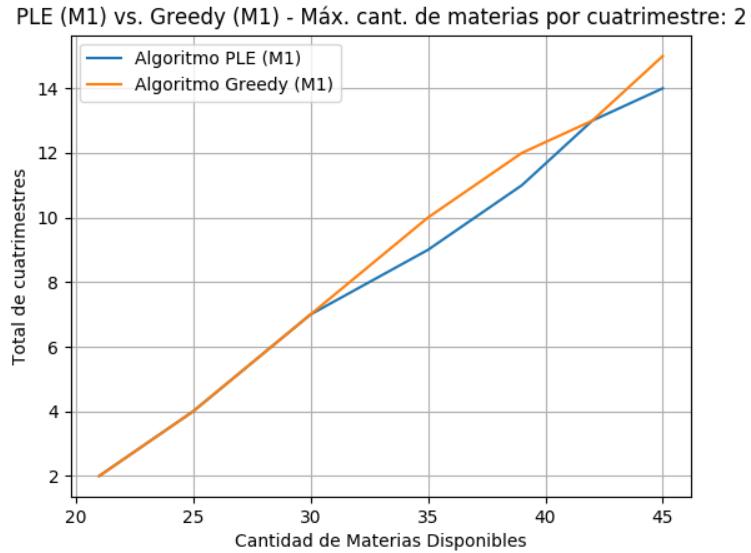


Figura 54: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 2 - Máquina M1

Como se puede observar en los gráficos anteriores, el algoritmo se comporta en términos generales, tanto en el de PLE y en el algoritmo Greedy, demorando más cuanto mayor cantidad de materias por cuatrimestre hay disponibles. Se produce en ambos algoritmos una rareza cuando está el plan completo por completar vs el plan sólo con un cuatrimestre completo, que no se logró determinar a qué se debe. Sin embargo, se pudo establecer que no fue un ruido estadístico, sino que es algo que dcede de forma determinística para ese valor, ya que se realizaron varias ejecuciones de los algoritmos y siempre arrojó resultados similares.

Luego, se ejecutaron los algoritmos en una máquina virtual con un servidor dedicado con las siguientes características. Esta máquina la rotularemos como M2.

- CPU: Intel Xeon Processor (Skylake) @ 2.10GHz
- RAM: 16 GB
- 4 Core

En particular, para la configuración de la cantidad de threads para el algoritmo de PLE fue seteada en 4 ya que es la cantidad de Cores que posee.

Los resultados de la ejecución pueden verse en la siguientes tablas:

Total cuatrimestres	Tiempo de generación	Cantidad de Materias disponibles	Max. materias por cuatrimestre
2	0.46 seg	21	2
4	1.29 seg	25	2
7	3.88 seg	30	2

Total cuatrimestres	Tiempo de generación	Cantidad de Materias disponibles	Max. materias por cuatrimestre
10	10.10 seg	35	2
12	13.39 seg	39	2
13	15.42 seg	42	2
15	23.19 seg	45	2
2	1.29 seg	21	3
3	3.43 seg	25	3
5	10.65 seg	30	3
7	33.85 seg	35	3
8	41.13 seg	39	3
9	42.37 seg	42	3
10	29.84 seg	45	3
2	0.03 seg	21	4
3	0.32 seg	25	4
5	0.66 seg	30	4
5	0.37 seg	35	4
7	1.62 seg	39	4
8	1.30 seg	42	4
9	0.86 seg	45	4
1	0.00 seg	21	5
3	0.01 seg	25	5
4	0.01 seg	30	5
5	0.01 seg	35	5
8	0.11 seg	39	5
9	0.06 seg	42	5
10	0.06 seg	45	5

Cuadro 3: Ejecucion algoritmo GREEDY (M2)

Total cuatrimestres	Tiempo de generación del plan	Cantidad de Materias disponibles	Max. materias por cuatrimestre
2	9.12 seg	21	2
2	9.58 seg	21	2
4	11.63 seg	25	2
4	11.62 seg	25	2
7	18.11 seg	30	2
9	32.60 seg	35	2
9	33.11 seg	35	2
11	1.0 min, 11.55 seg	39	2
13	50.26 seg	42	2
14	3.0 min, 10.80 seg	45	2
2	9.39 seg	21	3
3	11.76 seg	25	3
5	18.74 seg	30	3
6	31.46 seg	35	3
8	53.88 seg	39	3

Total cuatrimestres	Tiempo de generación del plan	Cantidad de Materias disponibles	Max. materias por cuatrimestre
9	1.0 min, 17.01 seg	42	3
10	4.0 min, 11.13 seg	45	3
1	9.43 seg	21	4
3	10.27 seg	25	4
4	17.97 seg	30	4
5	32.43 seg	35	4
6	34.83 seg	39	4
7	12.0 min, 2.54 seg	42	4
7	18.0 min, 36.93 seg	42	4
7	2.0 min, 48.80 seg	45	4
7	3.0 min, 34.30 seg	45	4
1	9.88 seg	21	5
2	9.98 seg	25	5
3	17.82 seg	30	5
4	33.74 seg	35	5
5	35.61 seg	39	5
6	1.0 min, 36.87 seg	42	5
7	3.0 min, 14.85 seg	45	5
7	3.0 min, 10.57 seg	45	5

Cuadro 4: Ejecución algoritmo PLE (M2)

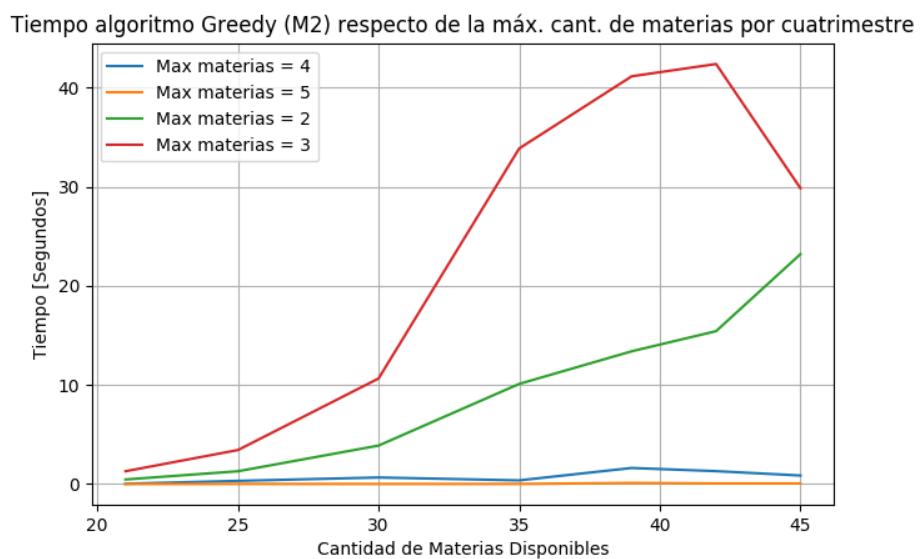


Figura 55: Comparación de tiempos respecto del máximo de materias por cuatrimestre - Algoritmo Greedy - Máquina M2

Tiempo algoritmo PLE (M2) respecto de la máx. cant. de materias por cuatrimestre

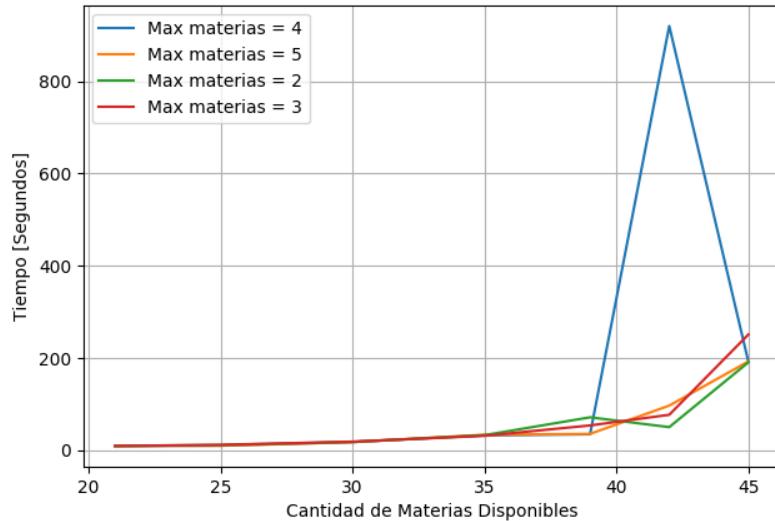


Figura 56: Comparación de tiempos respecto del máximo de materias por cuatrimestre - Algoritmo PLE - Máquina M2

PLE (M2) vs. Greedy (M2) - Máx. cant. de materias por cuatrimestre: 5

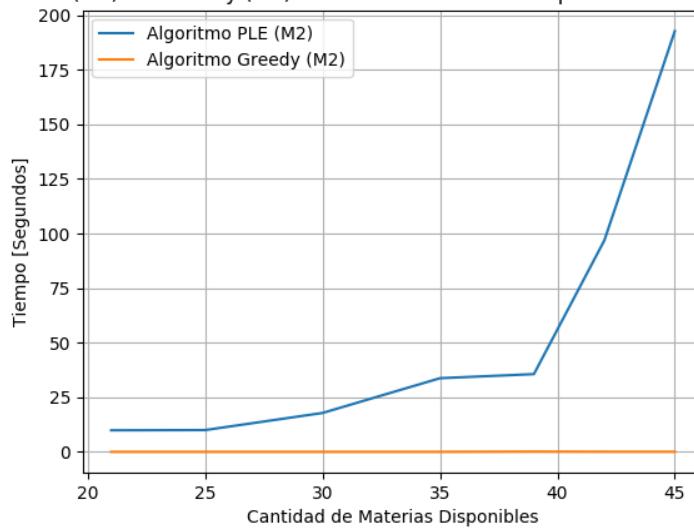


Figura 57: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 5 - Máquina M2

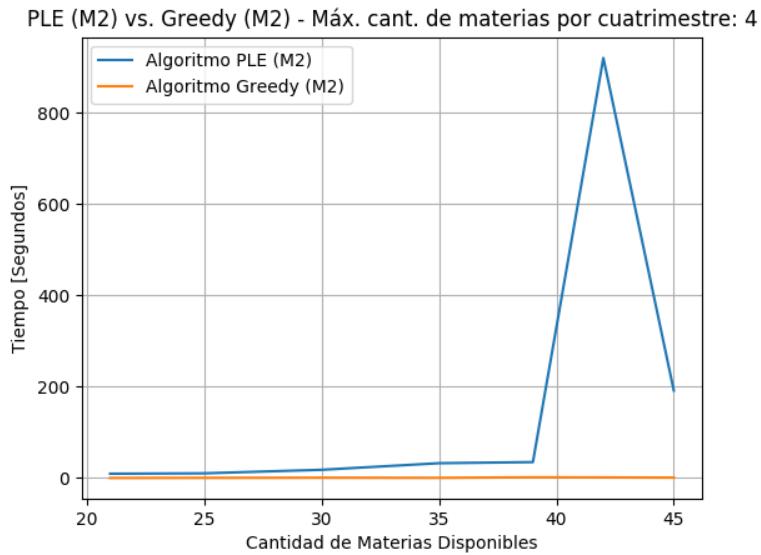


Figura 58: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 4 - Máquina M2

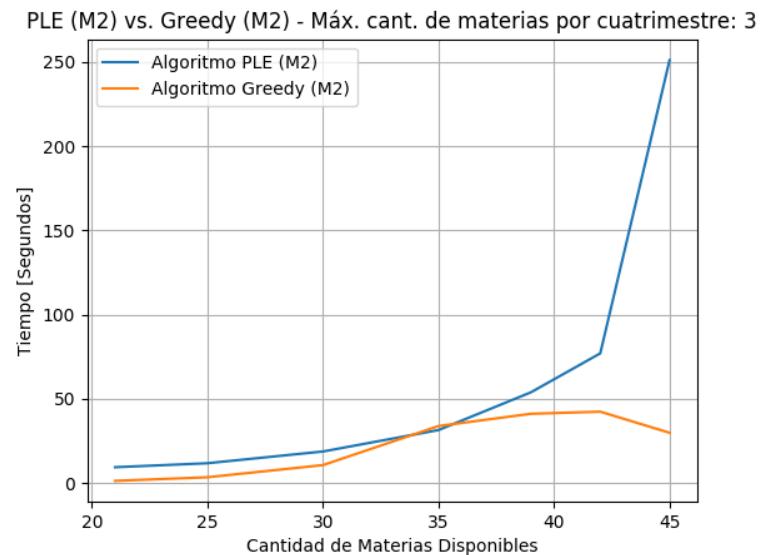


Figura 59: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 3 - Máquina M2

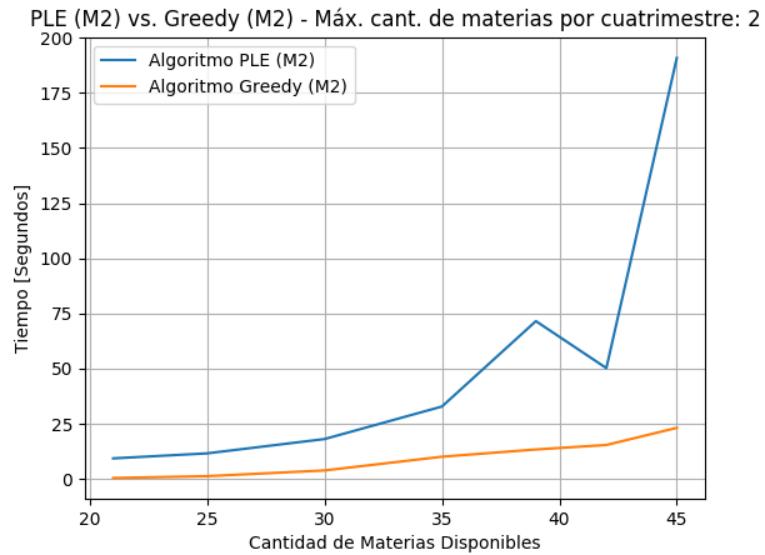


Figura 60: Comparación de tiempos Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 2 - Máquina M2

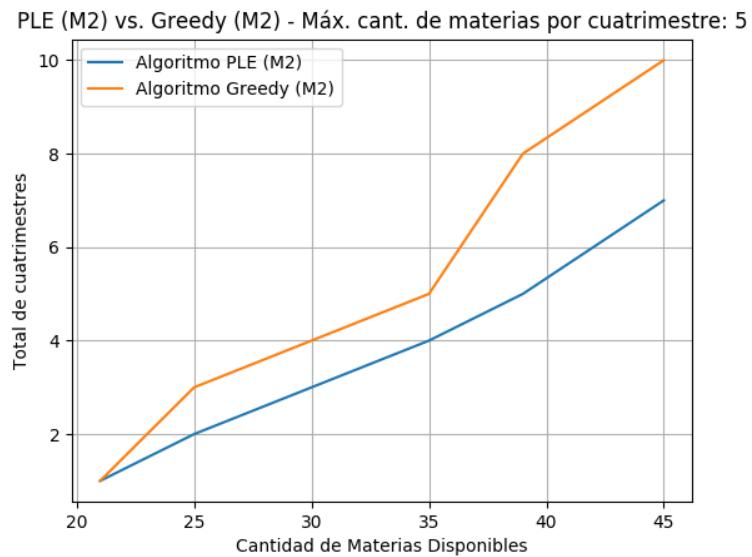


Figura 61: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 5 - Máquina M2

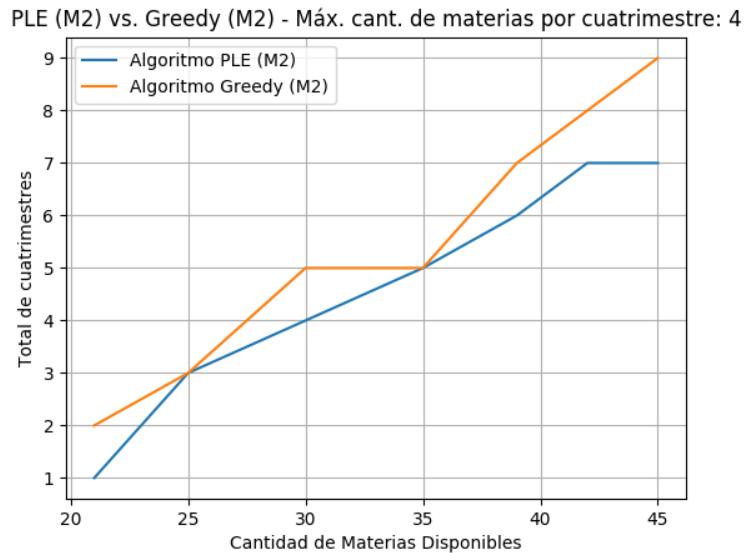


Figura 62: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 4 - Máquina M2

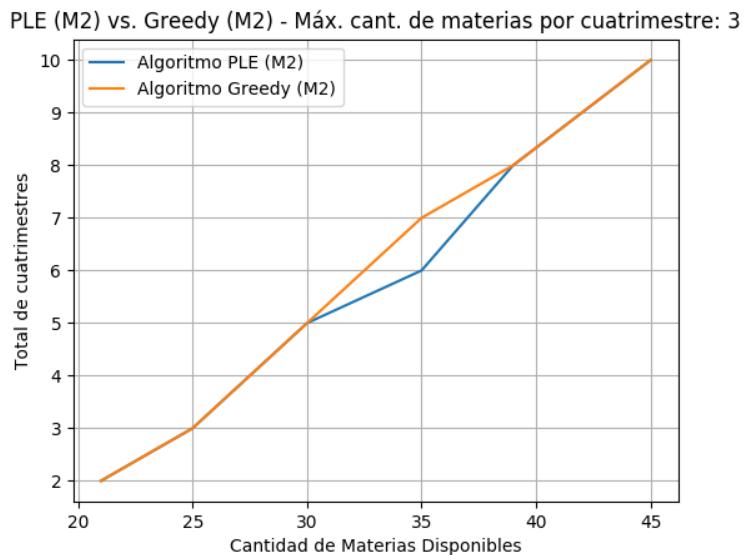


Figura 63: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre:3 - Máquina M2

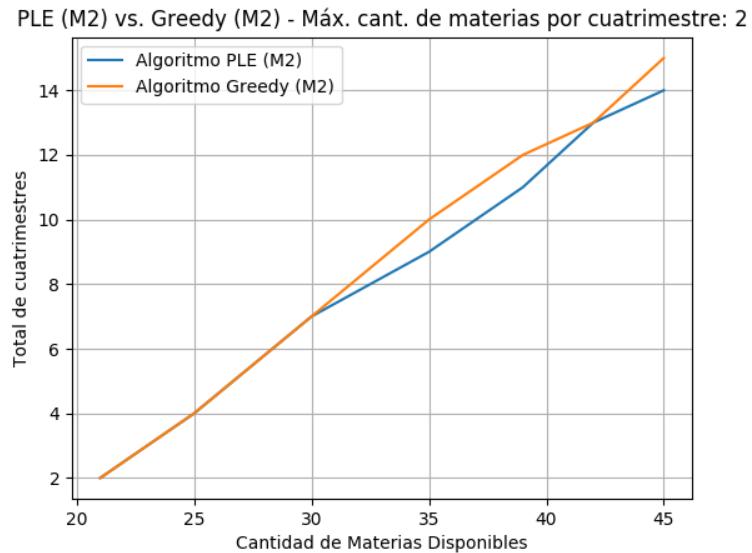


Figura 64: Comparación de resultado de cantidad de cuatrimestres Algoritmo PLE vs. Greedy - Máx. cantidad de materias por cuatrimestre: 2 - Máquina M2

Se pueden comparar también los gráficos de tiempos de la ejecución del algoritmo en ambas máquinas (no se realiza la comparación sobre la cantidad de cuatrimestres ya que en ambos casos dan los mismos valores). En ellos, se puede observar que la relación de las líneas del gráfico se mantienen pero que la máquina M2 siempre tiene mejores tiempos que la máquina M1.

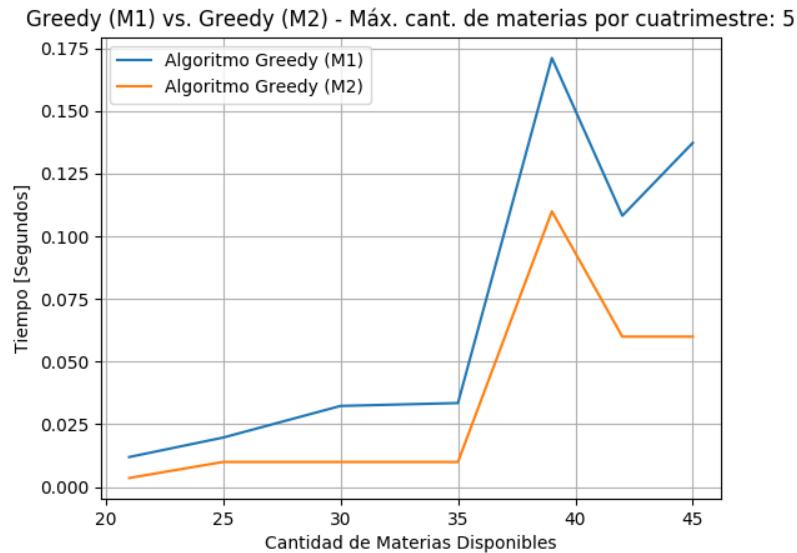


Figura 65: Comparación de tiempos Algoritmo Greedy - Máx. cantidad de materias por cuatrimestre: 5 - Máquina M1 vs. Máquina M2

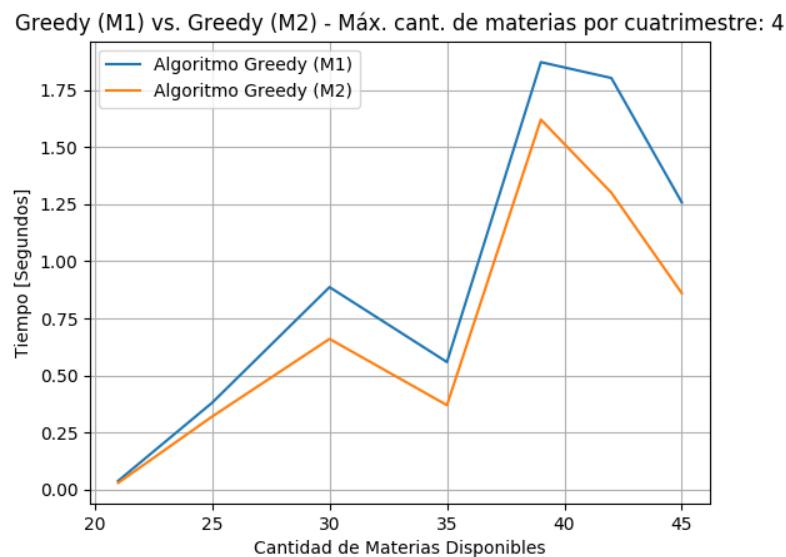


Figura 66: Comparación de tiempos Algoritmo Greedy - Máx. cantidad de materias por cuatrimestre: 4 - Máquina M1 vs. Máquina M2

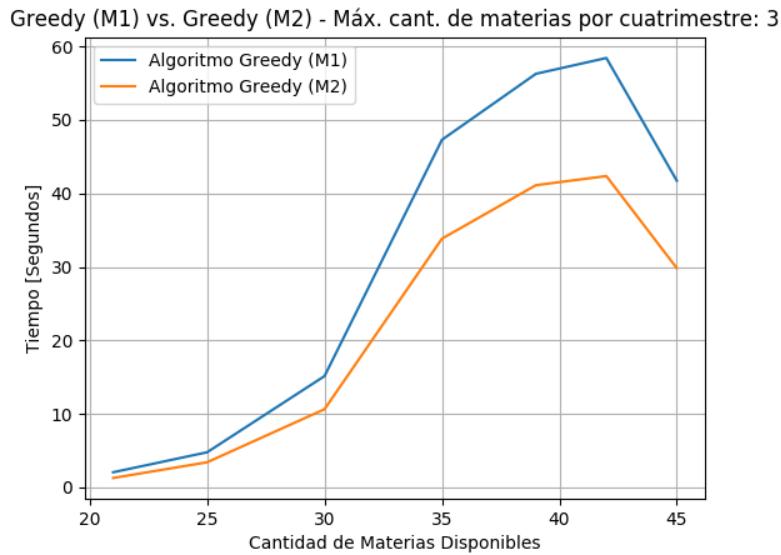


Figura 67: Comparación de tiempos Algoritmo Greedy - Máx. cantidad de materias por cuatrimestre: 3 - Máquina M1 vs. Máquina M2

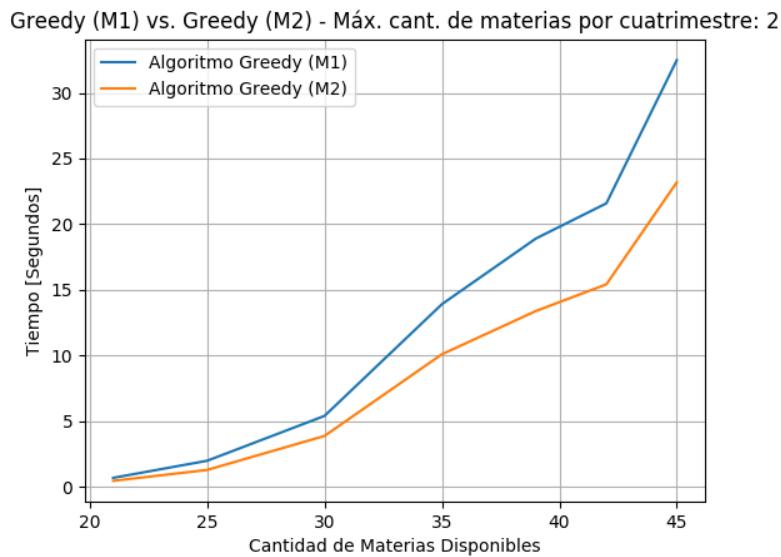


Figura 68: Comparación de tiempos Algoritmo Greedy - Máx. cantidad de materias por cuatrimestre: 2 - Máquina M1 vs. Máquina M2

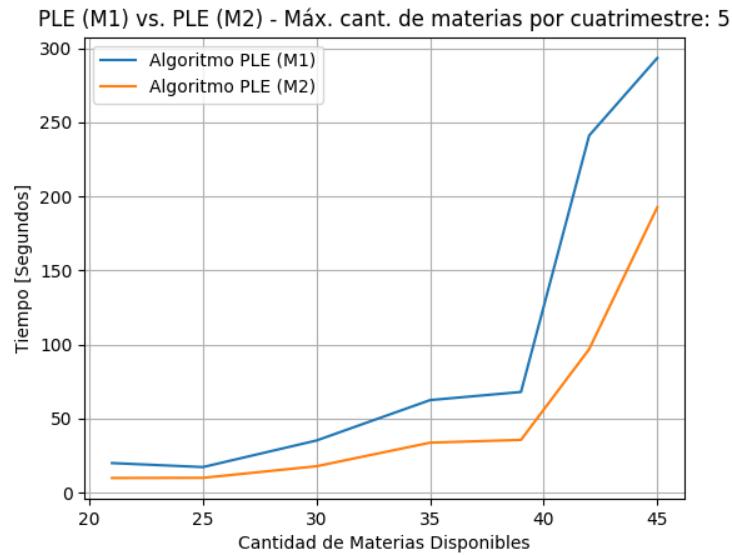


Figura 69: Comparación de tiempos Algoritmo PLE - Máx. cantidad de materias por cuatrimestre: 5 - Máquina M1 vs. Máquina M2

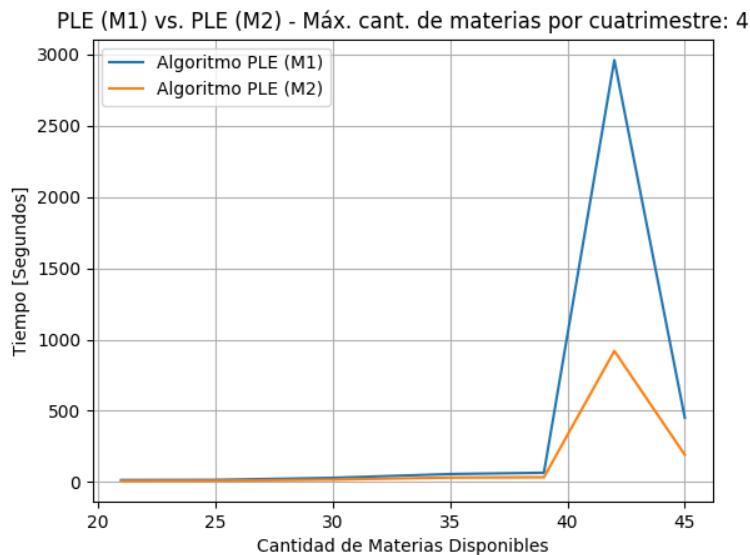


Figura 70: Comparación de tiempos Algoritmo PLE - Máx. cantidad de materias por cuatrimestre: 4 - Máquina M1 vs. Máquina M2

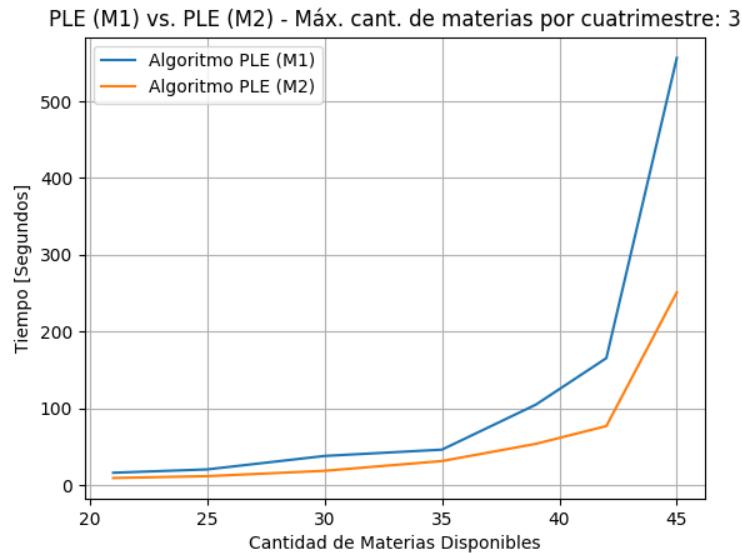


Figura 71: Comparación de tiempos Algoritmo PLE - Máx. cantidad de materias por cuatrimestre: 3 - Máquina M1 vs. Máquina M2

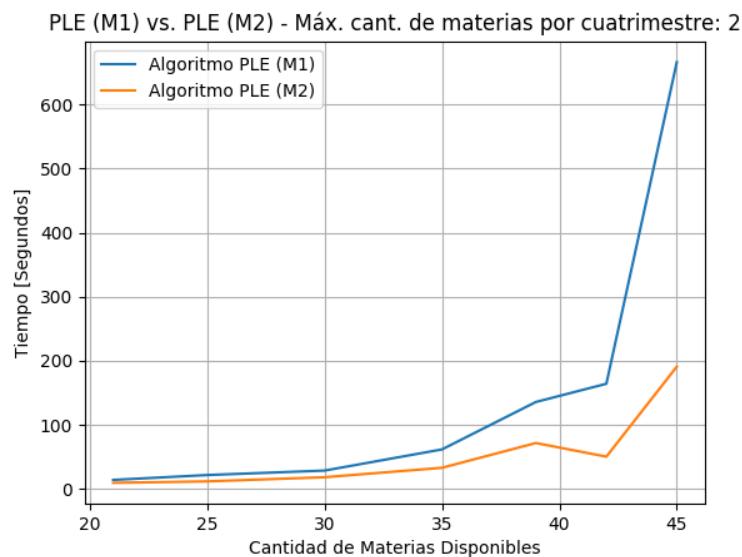


Figura 72: Comparación de tiempos Algoritmo PLE - Máx. cantidad de materias por cuatrimestre: 2 - Máquina M1 vs. Máquina M2

Para ambos algoritmos, se tuvieron los siguientes máximos en ambas máquinas. Las únicas diferencias fueron respecto de los tiempos que requirieron.

Algoritmo Greedy

Cantidad de materias Disponibles	42
Max cantidad de materias por cuatrimestre	3
Duración del plan (total de cuatrimestres)	9
Tiempo Máquina M1	58.43 segundos
Tiempo Máquina M2	42.37 segundos

Algoritmo PLE

Cantidad de materias Disponibles	42
Max cantidad de materias por cuatrimestre	4
Duración del plan (total de cuatrimestres)	7
Tiempo Máquina M1	49 minutos, 17.57 segundos
Tiempo Máquina M2	15 minutos, 19.74 segundos

De lo observado anteriormente, se puede ver que el algoritmo Greedy demora menos tiempo que el algoritmo de Programación Lineal Entera. Sin embargo, a diferencia del Greedy, el de Programación Lineal Entera tiene siempre el valor óptimo.

A pesar de que el algoritmo Greedy no tiene el valor óptimo el algoritmo es en términos generales (resultado vs. tiempo) muy bueno, ya que en muchos casos da el valor óptimo y en los que no suele tener un cuatrimestre de diferencia. La máxima diferencia de cuatrimestres que da el Greedy respecto del algoritmo de PLE, ha sido de dos cuatrimestres y se da en el caso de todo el plan a completar (o casi todo el plan) y muchas materias por cuatrimestre. Esto se explica porque el algoritmo Greedy no tiene en cuenta los cuatrimestres a futuro en esos casos ya que hay muchas combinaciones posibles de materias. En los casos donde el Greedy realiza fuerza bruta con poda, de todas formas la condición sobre el ‘futuro’ es solo el cuatrimestre inmediato siguiente, y no todos los cuatrimestres futuros.

8. ¿Integrando con el SIU Guarani?

Por un lado, la situación ideal es que ambas aplicaciones se fusionen en una, de forma tal que la resultante incluya las inscripciones, el manejo de formularios y solicitudes académicas, estado de carrera, etc. De esta forma, ni alumnos ni docentes requerirían tener espejados los datos en dos sistemas independientes.

Un caso un poco menos ideal, pero que aún no ha sido posible, es el caso de que el SIU provea servicios al que MUSSA pueda acceder para realizar sincronizaciones de datos (al menos de lectura de datos).

En el caso de que en algún futuro la integración de este sistema sea posible, se deberían realizar las siguientes:

- Obtención de horarios definitivos: El PDF que publica la facultad contiene los horarios para el cuatrimestre. Sin embargo, en las primeras semanas de publicado y antes de las inscripciones, suelen modificar esos horarios debido a errores y/o cambios de último momento. Estos cambios son reflejados en el SIU pero no en el PDF. Para evitar tener que hacer las modificaciones de los cursos manualmente, deberían poder obtenerse los horarios con un servicio y actualizarlos en MUSSA cada vez que se producen cambios.
- Los docentes deberían estar completos para cada curso en el SIU. Además, sería muy útil que tengan un servicio para obtener el listado completo de docentes y que al enviar el horario, el docente del curso venga indicado por su ID. De esta forma se podría evitar el problema de docentes que dictan más de una materia y que desde el PDF es imposible saber si es el mismo docente o no.
- Inscripción de Materias: Al inscribirse un alumno a las materias desde el SIU, se debería agregar un servicio que envíe los datos de la inscripción, indicando el curso, la materia y la carrera a la que se inscribió.
- Envío de calificaciones de exámenes finales / coloquios: Cuando el alumno rinde un examen final, la calificación es cargada en el SIU junto con el acta del examen. Se requiere un servicio que envíe este dato a MUSSA, de forma tal que el alumno no requiera actualizar la materia manualmente. Incluso, si el alumno hace la actualización manual, se le podría indicar usando estos mismos datos, cuáles difieren de la información oficial ya cargada.
- Encuestas: Actualmente las encuestas del SIU no están funcionando correctamente (una vez completada, al clickear para guardarla la misma no se guarda (o al menos no queda registro de ello) y se reinicia indicandote que aún tenés la misma encuesta por completar. Estas encuestas además, no son públicas. Si se decide usar MUSSA para las encuestas, se deberían agregar las preguntas que estén en la encuesta oficial y que sean importantes como para que las encuestas de MUSSA puedan ser consideradas como 'oficiales' para ser usadas por los departamentos, curriculares, concursos y todo aquél que las requiera.

El SIU actualmente tiene botones para generación de documentación y/o pedidos (como por ejemplo el certificado de alumno regular) que devuelven una respuesta de que se ha iniciado el trámite, pero estos trámites no se generan

realmente sino que deben hacerse personalmente a través del departamento de alumnos. Se debe cambiar esto, no solo porque es confuso para quienes no saben que realmente no funciona, sino porque no aporta las funcionalidades que dice tener y que son muy necesarias.

Si se permite el uso de servicios, hay dos posibilidades para la integración entre ambos sistemas.

Por un lado, que el SIU al realizar modificaciones de datos, notifique a MUSSA a través del servicio. Esta manera, requiere modificaciones en ambos sitios, por lo que es poco probable que se realice. De todas maneras, es la solución en la que los datos estarían siempre sincronizados con el último valor y no requeriría más invocaciones que las mínimas necesarias.

Por otro lado, una solución más sencilla (para quienes desarrollan el SIU) implica solo exponer servicios de consulta con los datos actuales. Allí, el sistema MUSSA deberá consultar periódicamente el dato actual y si se producen diferencias con los datos que éste tiene, actualizar los registros correspondientes. Que el dato que se observe en MUSSA esté actualizado o no dependerá de la frecuencia con la que se sincronicen los datos. Una opción, para que siempre el dato esté actualizado es realizar sincronizaciones completas de forma periódica con períodos de tiempo espaciados (por ejemplo, una vez por día en la noche), y por otro, realizar una sincronización de un dato específico (por ejemplo, el estado de un curso) on-demand; es decir cuando se está queriendo acceder a este dato. De esta forma, el dato observado siempre estará actualizado.

9. Mejoras y Nuevas Funcionalidades Futuras

Al ser un proyecto con miras a que los alumnos puedan hacer uso de él y que sea de desarrollo continuo, se tienen varias mejoras y funcionalidades futuras que se desean desarrollar.

Asociado a esto, se ha tenido contacto con algunos alumnos en particular que se han mostrado interesados en poder desarrollar alguna de las funcionalidades un poco más adelante.

Algunas de las funcionalidades y mejoras a futuro son:

- Carga de otras carreras de la facultad: En principio se debe evaluar y establecer si requieren ajustes los modelos de datos ya que no todas las carreras mantienen la misma estructura (por ejemplo, algunas carreras deben cumplir con X créditos en materias de otra facultad). Luego, se deberán generar los datos de entrada para poder dar de alta en la base de datos los planes de estudios asociados a ellas.
- Simultaneidades de carreras: Muchos alumnos obtienen simultaneidades de carreras. En el plan viejo de licenciatura y el plan de ingeniería en informática es algo muy común ya que comparten muchas materias. Se desea poder agregar las tablas de equivalencias para que al aprobar una de por aprobada la equivalente. Se debe tener en cuenta también este dato para las materias habilitadas para cursar y para la generación del plan con dos o más carreras en simultáneo.
- Excepción de Correlatividades: Poder solicitar las excepciones (generar la nota automática). Llevar registro de las excepciones solicitadas, negadas y/u otorgadas, de manera que las correlatividades tornen en flexibles para aquellas correlativas a las que se haya dado la excepción en la mayoría de los casos.
- Incorporar un calendario con parciales y otras fechas importantes: El calendario deberá tener fechas importantes como elecciones, inscripciones, etc que sean agregadas por un administrador. Por otro lado, también se deberá permitir a los alumnos/docentes subir las fechas de parciales, trabajos prácticos, etc de un curso. En el caso de que sean los alumnos quienes lo agreguen, se realizará una validación por los otros alumnos del curso. Si obtiene un porcentaje suficiente de validaciones, la fecha será agregada. Las fechas importantes deberán ser notificadas al usuario permitiendo configurar el tiempo y el mecanismo utilizado para la comunicación.
- Superposiciones Horarias: Permitir la verificación de forma sencilla sobre la superposición horaria acorde al plan de estudios diseñado por la facultad, emitiendo el listado de las materias que se superponen en cada cuatrimestre e indicando una solución que sea compatible con los cuatrimestres de horas carreras donde se dicte la materia.
- Contacto y Mensajería: Permitir la mensajería a través de la plataforma entre los diferentes usuarios. Permitir contactar a los alumnos que se encuentren cursando en el mismo curso que el alumno está inscripto para el cuatrimestre actual, de forma que puedan compartirse notas o hacerse preguntas.

- Incorporar un rol para usuarios docentes de forma que puedan modificar sus datos personales y agregar contenido/links relacionados a las materias que dictan.
- Incorporar un rol administrador de departamento para que cada departamento pueda modificar los horarios que le corresponden a sus cursos asociados y hacer actualizaciones de docentes de manera más efectiva.
- Permitir agregar calificaciones de trabajos prácticos y parciales a las materias del alumno.
- Permitir agregar materias no contempladas en el plan (por ejemplo, cuando cursan una materia en otra facultad) y la asignación de créditos (por ejemplo, por materias de intercambio) junto a la resolución correspondiente.
- Permitir que el usuario agregue su prioridad actual y en base a los cursos que finalmente se ha inscripto, estimar la máxima prioridad con la que alguien se inscribió al curso (estos datos no serán completamente fieles a menos que todos los alumnos con prioridad alta utilicen la aplicación).
- Hacer un parser de los datos del SIU Guaraní (solicitando usuario y password del mismo cada vez que el usuario quiera hacer uso de esta funcionalidad) para poder sincronizar las materias que ya tiene cargadas en el sistema (y no tener que cargarlas manualmente) y/o para informar en el caso de que hayan materias aprobadas / desaprobadas cargadas en MUS-SA y que aún no aparezcan en el SIU.
- Agregar incompatibilidades conocidas de materias que no se suelen cursar juntas en un mismo cuatrimestre debido a su exigencia y/o dificultad (por ejemplo Organización de Datos y Taller de Programación no suelen ser elegidas para cursarse juntas). Colocar como un parámetro opcional la selección de dichas incompatibilidades.

10. Conclusiones

Durante el desarrollo de este trabajo se utilizaron tecnologías, frameworks y herramientas con las que nunca se había trabajado como Flask, Celery, Redis y PuLP.

El uso de Flask y PuLP si bien requirió tiempo de aprendizaje, no fue un reto en sí. En cambio, la utilización de Celery y Redis si lo fue, no tanto por la implementación en sí, sino en entender el concepto teórico y aplicarlo en el uso ya que no se habían trabajado con este tipo de arquitecturas anteriormente. El desconocimiento de las mismas requirió que se realizaran modificaciones luego para poder hacer que el software soporte mayor flujo de usuarios sin que eso derive en que se ‘caiga’ el servidor.

El desarrollo de la página web fue extenso y se requirieron varias pantallas, y si bien esta parte era ‘sencilla’ (por más que requiera tiempo), la modelización de los datos y las interrelaciones requirieron una gran cantidad de esfuerzo y atención. Se diseñó el modelo de entidades que se utilizaría inicialmente, pero cuando se agregaron las encuestas hubo necesidad de modificarlo ya que algunos datos dependían de los otros (en especial en las bajas) y se requería determinar cuándo se deseaba salvaguardar dichos datos (e indicar que la relación entre los mismos no existía) vs. cuándo se deseaba eliminarlos de las tablas definitivamente.

El uso de Flask-User como aplicación inicial facilitó mucho la curva de aprendizaje al tener código de referencia y una estructuración previa. Sin embargo, esto fue un problema cuando se decidió tener una API REST independiente ya que la autenticación no podía vincularse con Flask-User y requería quitarlo y generarla en la API o solucionar el problema del Cross Domain. Por el momento, se mantiene Flask-User (con intenciones de quitarlo luego) y la API se separa ‘lógicamente’ pero no físicamente de la APP.

Respecto de las estimaciones se debe destacar que hubo una gran variación (alrededor de un 42 % extra) respecto de los valores inicialmente estimados. Algunos de los motivos por los cuales se produjo esta variación fueron: retrabajo (muchas cosas se desarrollaron con los conocimientos que se poseían de antemano y tiempo después, al conversar y comentar sobre estos desarrollos con colegas, hicieron notar que no eran correctos y hubo que corregirlos); tareas no incluidas en la estimación inicial (no se consideró el uso de redis/celery para los algoritmos, no se consideró la escritura de este informe, etc); y tareas estimadas por menos horas que las requeridas (un caso particular de ello, son las horas de testing que requirieron muchas más horas que las indicadas inicialmente).

Sobre los algoritmos desarrollados, el algoritmo Greedy presenta una velocidad mucho mayor que el de PLE, y si bien en muchos de los casos éste presenta resultados similares al de PLE, no siempre es óptimo pudiendo agregar 1 o 2 cuatrimestres (en tiempos de cursada dos cuatrimestres es mucho tiempo) respecto del óptimo.

La utilización de programación lineal entera, tema principal de la materia de Modelos y Optimización I (71.14), fue una gran novedad ya que nunca había tenido la oportunidad de utilizar programaciones alternativas en un problema

real. La generación del algoritmo completo desde cero presentó un gran reto.

El algoritmo de PLE lo máximo que ha tardado es 15 minutos con una máquina con buenas características (o casi 50 minutos con una de características bastante pobres), por lo que se considera que es un tiempo más que aceptable. Igualmente, para todo algoritmo generado anteriormente, se mantendrá el hash de forma que este tiempo sea reducido al tiempo que toma replicar los valores en la base de datos.

El algoritmo Greedy, si bien no es óptimo, para ser un Greedy se puede decir que da ‘buenos’ resultados ya que en la mayoría de los casos coinciden con el valor óptimo, y cuando no, por lo general la distancia es de 1 cuatrimestre, aunque la distancia máxima presentada para algunos casos puntuales ha sido de 2 cuatrimestres.

El desarrollo de este trabajo, como experiencia en sí, fue muy enriquecedor ya que permitió pasar por todas las instancias de un proyecto (desde la concepción y las estimaciones iniciales, a la puesta en funcionamiento en un servidor web). En particular, se aprendieron muchas herramientas y detalles que se deben tener en cuenta a la hora de montar un proyecto de los cuales no tenía conocimiento. De todas formas, creo que hubiese sido mucho más productivo poder desarrollar este trabajo (con un alcance más amplio) en un equipo de al menos dos personas para poder realizar discusiones productivas y contrastar ideas antes de desarrollarlas.

11. Agradecimientos

A mi familia, que estuvo presente desde la distancia cuando comencé la carrera. En especial, a mi hermana Lorena que me trajo a Buenos Aires cuando comencé a estudiar en la universidad manejando ambas por la ruta durante esos 2000 kilómetros. A mi hermana Verónica que muchas veces vino a Buenos Aires a visitarme a pesar de que estábamos tan lejos; y a mi hermana Mariel que me dió casa cada vez que podía viajar yo y se levantaba a las 3 de la mañana para buscarme en el aeropuerto.

A mi nueva familia adquirida, que me han acompañado durante este último tramo, en especial a Nico que nos hizo mil y un favores relacionados a la facultad desde que nos mudamos a Irlanda.

A mis amigos y compañeros de facultad, con los que siempre compartí muy buenas experiencias y muchas veces tuvimos que dejar cosas de lado porque ‘había que estudiar’. En especial a Nicolás, Martín, Ezequiel, Flor y Javier con los que compartí la mayor parte de mi carrera y nos ayudamos mutuamente a avanzar, sé muy bien que sin ellos la carrera no hubiese sido igual. Con mucho cariño también, a Mariano Lapenna, fuiste un gran amigo y compañero y luego de tantos años, aún se te extraña.

A todos mis grupos de trabajo profesional antes de este, que se armaron y desarmaron por uno u otro motivo, pero que han sido pasos necesarios para poder finalmente llegar a este punto.

A mis tutores, Rosita y Diego que me acompañaron durante este trabajo. En especial a Diego que siendo su primer tutoría me acompañó y me alentó mucho con sus comentarios y recomendaciones.

A la curricular de Licenciatura, de la cual formé parte durante muchos años, por permitirme crecer y ayudar en lo que podía.

A los chicos (y no tan chicos) de Casa Informática, por luchar conmigo estos años y demostrar que si queremos que las cosas cambien, tenemos que dejar de estar con los brazos cruzados.

A todos los Wachencholdiers los que están y los que estuvieron. Con ustedes compartí miles de experiencias y locuras. Aprendí mucho, me divertí mucho. Me sentí muy acompañada y todos los días los extraño un poco. Estoy orgullosa de haber podido ser parte de este gran equipo, y aún me considero parte aunque ya no pueda estar tan cerca...

A mis alumnos, que me enseñaron a enseñar, a tener paciencia, a tener mejor predisposición y a encontrar mil y un maneras de decir lo mismo. Muchas veces siento que las cosas las aprendí realmente al tener que explicarlas a ustedes.

Y de manera especial, agradezco a mi marido, Ariel Wainer, que ha estado conmigo todo este tiempo, sosteniéndome en mis tristezas y compartiendo conmigo mis alegrías. Gracias por haberme mostrado tantas herramientas nuevas y por retarme cuando hacía las cosas mal para que las mejorara, sé que no soy fácil pero me encanta que me enseñés y expliques lo que vos sabes. Esto no hubiese sido posible sin vos.

12. Referencias y Material consultado

- [1] Rodrigo Hernandez, Jaime Miranda P. y Pablo A. Rey (2008). Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad Diego Portales Mediante un Enfoque de Programación Entera. *Revista Ingeniería de Sistemas, Volumen XXII*, páginas 121-141. Recuperado de: http://www.dii.uchile.cl/ris/RISXXII/horariosUDP_RISVersion%20FINAL.pdf
- [2] Enrique Castillo, Antonio J. Conejo, Pablo Pedregal, Ricardo García y Natalia Alguacil (2002). Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia. Páginas 461-483. Recuperado de: <http://www.dia.fi.upm.es/~jafernан/teaching/operational-research/LibroCompleto>
- [3] Jorge A. Baier y Sheila A. McIlraith (2008). Planning with Preferences. *Association for the Advancement of Artificial Intelligence, Volumen 29 (4)*, páginas 25-36. <https://doi.org/10.1609/aimag.v29i4.2204>
- [4] Shirin Sohrabi, Jorge A. Baier y Sheila A. McIlraith (2009). HTN Planning with Preferences. *Proceedings of the 21st international joint conference on Artificial intelligence*, páginas 1790-1797. Recuperado de: <https://www.ijcai.org/Proceedings/09/Papers/298.pdf>
- [5] Nan Li, William Cushing, Subbarao Kambhampati y Sungwook Yoon (2014). Learning Probabilistic Hierarchical Task Networks as Probabilistic Context-Free Grammars to Capture User Preferences. *ACM Transactions on Computational Logic, Volumen V*, páginas 1-32. Recuperado de: <http://rakaposhi.eas.asu.edu/nan-tist.pdf>
- [6] Nan Li, William Cushing, Subbarao Kambhampati y Sungwook Yoon (2013). Learning Probabilistic Hierarchical Task Networks to Capture User Preferences *Preprint submitted to Artificial Intelligence*, páginas 1-30. Recuperado de: <https://arxiv.org/pdf/1006.0274.pdf>
- [7] Ramón Sala - Garrido (2005). Programación Lineal Entera. *Clase 14*, páginas 1-16. Recuperado de: <https://www.uv.es/~sala/Clase14.pdf>
- [8] Gunnar Klaus (2008). *Solving the ILP using branch-and-cut*. Recuperado de: http://www.mi.fu-berlin.de/wiki/pub/Main/GunnarKlaup1winter0708/discMath_klau_ILP_I.pdf
- [9] L. Caccetta - S. P. Hill (1999). Branch and Cut Methods for Network Optimization *PERGAMON. Mathematical and Computer Modelling*, páginas 517-530. Recuperado de: <https://www.sciencedirect.com/science/article/pii/S0895717700002582>
- [10] *Qué es la programación Lineal Entera*
<https://www.gestiondeoperaciones.net/programacion-entera/que-es-la-programacion-entera/>

- [11] «*Linear Programming with Python and PuLP*»
BEN ALEX KEEN
<http://benalexkeen.com/linear-programming-with-python-and-pulp/>
- [12] «*Optimization with PuLP*»
STUART MITCHELL, ANITA KEAN, ANDREW MASON, MICHAEL O'SULLIVAN, ANTONY PHILLIPS
<https://pythonhosted.org/PuLP/>
- [13] «*Flask*»
ARMIN RONACHER
<http://flask.pocoo.org/>
- [14] «*The Flask Mega-Tutorial*»
MIGUEL GRINBERG
<https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>
- [15] «*SQLAlchemy*» *Python SQL toolkit and Object Relational Mapper*
<https://www.sqlalchemy.org/>
- [16] «*Flask-User*»
LING THIO
<https://github.com/lingthio/Flask-User>
- [17] «*Bootstrap*» *Framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web, permitiendo crear de forma sencilla webs de diseño adaptable.*
<https://getbootstrap.com/>
- [18] «*MySQL*» *Sistema de gestión de bases de datos relacional*
<https://www.mysql.com/>
- [19] «*CELERY*» *Cola de tareas asíncrona open source basada en el pasaje de mensajes distribuido*
<http://www.celeryproject.org/>
- [20] «*REDIS*» *Almacén de estructura de datos de valores de clave en memoria rápido y de código abierto*
<https://redis.io/>