



FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE BUENOS AIRES

Propuesta de Trabajo Profesional

MUSSA

**Generador de Planes de Carrera  
personalizados utilizando  
Programación Lineal Entera**

*Jennifer Andrea Woites*

Tutor: Lic. Rosa Wachenchauzer

Co-Tutor: Ing. Diego Essaya

Septiembre, 2017

## Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Motivación . . . . .	2
<b>2. Alcance</b>	<b>3</b>
<b>3. Características del Trabajo</b>	<b>3</b>
3.1. Login / SigIn . . . . .	3
3.2. Perfil de usuario . . . . .	3
3.3. Carreras de la facultad . . . . .	4
3.4. Excepciones de correlatividades . . . . .	4
3.5. Simultaneidades de carreras . . . . .	5
3.6. Encuestas . . . . .	6
3.7. Búsqueda de Materias . . . . .	6
3.8. Algoritmos para la generación del plan de estudios personalizado	6
3.9. Armado y Visualización del Plan de Carrera . . . . .	8
<b>4. Tecnologías y Herramientas</b>	<b>9</b>
4.1. Librería: PuLP . . . . .	9
4.2. Herramientas . . . . .	9
<b>5. Plan de Trabajo</b>	<b>9</b>
5.1. Equipo de Trabajo . . . . .	9
5.2. Metodología . . . . .	9
5.3. Estimaciones . . . . .	10
5.4. Cronograma . . . . .	10
<b>6. Referencias y Material consultado</b>	<b>11</b>

## 1. Introducción

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

El objetivo del proyecto es aplicar los conocimientos adquiridos en la carrera. MUSSA - Generador de Planes de Carrera personalizados utilizando Programación Lineal Entera”.

El objetivo general del presente trabajo, será 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. Cabe destacar, que el corazón de este trabajo reside en la modelización y generación del algoritmo que permita la generación del plan de carrera personalizado; para ello, se generará un modelo de resolución con Programación Lineal Entera.

### 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 las 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 cursar 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 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.

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 de este trabajo contempla:

1. Desarrollar una página web responsive que conste de los siguientes:
  - Login / Sign In
  - Almacenamiento y edición de datos personales, carreras, materias aprobadas
  - Excepciones de correlatividades
  - Simultaneidad de carreras
  - Encuestas de cursos / materias
  - Búsquedas de materias con filtros
2. Modelar y desarrollar el algoritmo para la generación del plan de estudios personalizada

## 3. Características del Trabajo

### 3.1. Login / SigIn

Para el Login / Sign In se contará con un usuario y contraseña que servirá como identificación. El padrón será un dato no obligatorio, de forma tal que pueda ser utilizado por aquellos alumnos que aún se encuentran en el CBC y no cuentan con padrón. Sin embargo, en caso de introducir el padrón, este deberá ser único ya que no puede haber dos alumnos con el mismo padrón.

Se debe permitir además, el recupero de contraseña a través de el mail proporcionado por el usuario, generando una clave random y enviándosela por mail. Esta clave deberá estar marcada como caducada el usuario tendrá que modificarla en cuanto vuelva a entrar al sistema.

Para la primer implementación, no se realizará bloqueo y desbloques de usuarios.

### 3.2. Perfil de usuario

Cada usuario deberá tener sus datos personales (nombre, apellido, mail, padrón, etc), pero además, sus datos académicos que deberán incluir:

- Carreras en las que se encuentra inscripto

- Materias aprobadas
- Materias en curso
- Materias con final pendiente

Se debe permitir además, que el alumno pueda descargar en pdf el listado de materias que tiene aprobadas (este listado suele ser solicitado en la mayoría, sino todos, los trámites de la facultad).

### 3.3. Carreras de la facultad

Las diferentes carreras de la facultad están compuestas por una serie de materias obligatorias, un trabajo profesional, 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 cargarán y modelarán 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 contenidos mínimos, los diferentes cursos disponibles con sus docentes y horarios, la cantidad de créditos y sus correlatividades.

Es posible modelar las diversas materias en un grafo dirigido no pesado, en el cual las aristas equivalen a las correlatividades. Es decir, si la materia A es correlativa de B y C, entonces habrá una arista de A hacia B, y de A hacia C respectivamente.

Los horarios de los cursos de las materias deberán mantener un histórico, y avisar en caso de que sea un curso nuevo (en cuyo caso se deberán repetir los horarios indicando en el cuatrimestre al que se hayan copiado los horarios que podrían variar), se debe identificar las materias que se dictan solo el 1º o 2º cuatrimestre.

La obtención de los horarios del cuatrimestre actual se realizará a través del pdf publicado por la facultad cada cuatrimestre, basándose en un parser desarrollado por el LUGFI[7].

### 3.4. Excepciones de correlatividades

Se deberá contar con una sección en la que sea posible ingresar pedidos de excepciones de correlatividad. Al hacer un pedido, se debe indicar la materia que se desea cursar y con qué materia se solicita la excepción, además del estado de la misma (si aún no fue cursada o si tiene el final pendiente). Adicionalmente podrá ingresar un texto con mayor detalle.

Al generar la solicitud de excepción, se guardará la misma en el historial de pedidos de excepciones en un estado 'No resuelta'. Además, se descargará en un pdf la nota modelo correspondiente ya completa con los datos necesarios para poder ser presentada en la facultad.

Desde el historial de excepciones será posible cambiar el estado a 'Aceptada' o 'Rechazada' y subir como adjunto la resolución correspondiente (esto permitirá tener en un único lugar todas las copias de las resoluciones que deben ser

presentadas cuando se realiza el trámite del título).

En caso de que el alumno no haya modificado el estado de la solicitud de excepción, cuando marque la materia como aprobada y deba completar la encuesta, se le añadirá una pregunta que requiera indicar si hizo la materia con la excepción o no.

El manejo de las excepciones es importante para este sistema ya que las materias del plan de estudio de una determinada carrera tienen una correlatividad que podemos llamar "fuerte", es decir, no es posible cursarlas si no se tiene aprobada su correlativa; si se los coloca en un grafo, sería su predecesor. Lo que se desea modelar en ese caso es que si una excepción de correlatividad fue entregada con frecuencia con la correlativa no cursada aún (por ejemplo, las aprobadas son de al menos el doble que las rechazadas o similar), esa materia tendría una correlatividad débil con la materia con la cual se pidió excepción, por ello, se podría quitar el predecesor (con una alerta que avise que es requerido el pedido de excepción antes de cursar la materia).

En la visualización de las materias, se incluirá el porcentaje de gente que la hizo con excepción de correlatividad con fines estadísticos.

### 3.5. Simultaneidades de carreras

Para los casos de simultaneidades de carrera se deberá formular un único plan combinado o permitir que seleccione el armado de un plan por carrera (por ejemplo si decidió terminar todas las materias de una carrera y luego terminar la otra), para este último se deberá indicar si comienza una a continuación de la otra o si comienza en el siguiente período lectivo anual.

Además, se deberán tener en cuenta:

1. Las equivalencias automáticas de materias: En caso de que se curse una materia con equivalencia, la equivalente debe darse por aprobada y no tenerse en cuenta para la distribución horaria en el plan de carrera.
2. Equivalencias por resolución: Se debe permitir ingresar un pedido de equivalencia entre materias y/o por créditos. Luego, se deberá indicar si fue otorgada o no, anexando la correspondiente resolución e indicando los créditos o la materia dada por equivalencia, además de texto libre en caso de requerirlo. Si se cursa una materia que fue disparadora de una resolución por equivalencia, se deberá generar la alerta correspondiente en el cuatrimestre que corresponda, para poder generar automáticamente una nota de solicitud de equivalencia con las resoluciones históricas correspondientes. No se deberá tener en cuenta horarios de cursada de dicha materia pero contará para los créditos requeridos en el armado del plan. Se debe tener en cuenta que no es posible tener una equivalencia de una materia que fue otorgada por equivalencia (es decir si con la materia A dan B, y con la materia B dan C no es válido que solo con A se tengan B y C; solo se tendrá B).
3. Algunas materias de mismo código tienen correlatividades diferentes según la carrera, por lo que deberán agregarse como materias diferentes en el

armado del plan, y añadir la restricción de que, en caso de que se curse esta materia, solo se podrá hacer una de ellas, es decir o se cursa con las correlatividades de la carrera A o con las de la B, pero solo se hará una.

### 3.6. Encuestas

Se deberá contar con una sección de encuestas a las que pueda puntuar a cada materia con el horario real de cursada que tuvo la materia, la cantidad de horas extras además de la cursada que requiere la materia por semana, un puntaje del curso, la dificultad de los temas, la dificultad de los TPs y exámenes, preguntas acerca de la materia, de los profesores y comentarios personales.

Además, se añadirá una sección de Tema Principal para las materias de carácter de "electiva", con el que se podrá clasificar la materia para determinar si está relacionada con el Data Mining, Robótica, Gestión, etc de forma que pueda ser utilizado como preferencia para priorizar una materia electiva por sobre otra en el armado del plan. Respecto de esto, se busca que la cantidad de materias electivas que se seleccionen automáticamente en el plan correspondan a porcentajes de "afinidad" que establezca el alumno con cada uno de estos temas.

Adicionalmente, se podrán indicar hasta 3 palabras claves (tags) con las que el alumno podría identificar la materia. Las 3 palabras más elegidas, serán las utilizadas para la agilización de búsquedas de materias puntuales.

Cada alumno tendrá su listado de encuestas realizadas y podrá contestar una sola vez la combinación [curso + cuatrimestre + año].

Los resultados de las encuestas serán públicos sin mostrar quién fue el usuario que respondió la encuesta, de forma tal que los docentes puedan hacer uso de la página para poder evaluar sus propios cursos.

Los valores numéricos de las encuestas serán tomados en promedio para evaluar la calidad del curso y la dificultad, parámetros que podrán ser utilizados luego como parte de las preferencias de armado del plan.

### 3.7. Búsqueda de Materias

Desde la vista pública, se podrá acceder al buscador de materias. En él se podrá buscar por lo siguiente:

- Carrera
- Código de materia
- Nombre (o parte del nombre) de la materia
- Palabras clave (tags más frecuentes de cada materia)

Para cada materia se podrán acceder a los datos generales de la misma y al historial de encuestas por cada cuatrimestre.

### 3.8. Algoritmos para la generación del plan de estudios personalizado

Se analizó la posibilidad de realizar fuerza bruta con poda en primera instancia, pero la cantidad de árboles generados para analizar crecía con mucha

rapidez. Para poder generar los árboles, se partían de las materias disponibles (todas aquellas para las cuales las correlativas habían sido aprobadas) y de la cantidad de materias máxima que se permitía cursar por cuatrimestre. Para cada materia, se consideraba que tenía un único horario y que, sin importar con qué materias compartiera el cuatrimestre, sus horarios iban a ser compatibles.

Luego, se toma una de las materias al azar y se generaban dos árboles posibles, el primero con esta materia incluida en el cuatrimestre que se estaba intentando completar, el segundo sin esta materia y marcándola como no disponible para ser utilizada en este cuatrimestre. Luego, se proseguía con ambos árboles, manteniendo las materias disponibles por separado.

Los árboles que desencadenaban en una imposibilidad de continuar por no quedar materias disponibles (ej, todas fueron marcadas como no válidas para ese cuatrimestre) eran descartados.

De esta forma, para pocas materias, era posible conseguir en poco tiempo los planes de estudio posibles, y luego, elegir de estos el de menor duración en cuatrimestres. Ahora bien, para una cantidad de materias de la magnitud de un plan de estudios general, no es posible realizar por fuerza bruta con toda los posibles planes de estudio en un tiempo pequeño, ya que la cantidad de ramificaciones se incrementaba en  $2^n$ .

Otra posibilidad que se analizó es la generación a través de un algoritmo Greedy, que sencillamente elija las materias en base a lo que, según números como la cantidad de créditos que añade, la cantidad de correlatividades que libera, etc, seleccione las materias más convenientes para el cuatrimestre que se esté analizando. El problema con esto, es que no realiza un análisis del problema global sino localizado, y si bien puede tener una buena solución, no necesariamente será la óptima, ya que no dista mucho de lo que hoy realizan los alumnos intuitivamente para decidir su cuatrimestre.

En base a la investigación, se encontró que en algunas universidades han desarrollado sistemas para determinar las aulas y los horarios de los cursos que se dictarán utilizando «Programación Lineal Entera», que es la línea que se seguirá en este trabajo.

Básicamente, se plantean restricciones que llamaremos "fuertes" que consisten en restricciones que sí o sí deben cumplirse (por ejemplo, no es posible estar en dos lugares al mismo tiempo, o que todas las materias obligatorias deban cumplirse), y por otro lado, restricciones que son "deseables", es decir, que cumplirlas hace que la solución elegida sea mejor, pero si no pueden cumplirse, es posible relajarlas pero puntuando negativo a la solución. De esta forma, se permite flexibilidad al problema. Una posible restricción deseable es que el puntaje del curso sea mayor a un determinado valor.

Así, se generan algunas de las siguientes ecuaciones básicas<sup>1</sup>:

Sean:

$Y_{ij}$ : La materia  $i$  se realiza en el cuatrimestre  $j$  [Boolean]

$C_i$ : Numero de cuatrimestre en que se realiza la materia  $i$  [Integer]

$M$ : Constante. Máxima cantidad de cuatrimestres posibles.

---

<sup>1</sup>Se han seleccionado solo las ecuaciones básicas del problema para ilustrar el trabajo a realizar. Falta el desarrollo de las ecuaciones de horarios, restricciones deseables, etc.



Tenemos algunas de las siguientes restricciones fuertes que comienzan a modelar el problema:

- La materia  $i$  obligatoria, debe cursarse en algún cuatrimestre. Además, este cuatrimestre debe ser único.

$$\sum_{j=1}^M Y_{ij} = 1; \forall i \in [Materias]$$

- Se debe conocer el número de cuatrimestre en que es cursada la materia  $i$ .

$$C_i = \sum_{j=1}^M j * Y_{ij}; \forall i \in [Materias]$$

- Si la materia  $B$  tiene como correlativa a  $A$ , entonces, la materia  $A$  debe realizarse primero. Lo que quiere decir, que el cuatrimestre de  $A$  debe ser menor que el número de cuatrimestre de la materia  $B$ .

$C_A < C_B$ ; o en general:

$C_i < C_j$ ;  $\forall j$  que tenga como correlativa a la materia  $i$

- La cantidad de materias por cuatrimestre no puede superar un valor máximo preestablecido por el usuario. (cantidad de materias máxima que el alumno está dispuesto a cursar por cuatrimestre).

$$\sum_{i \in MATERIAS} Y_{ij} \leq \text{Max\_cant\_materias\_cuatrimestre}; \forall j \in [1, M]$$

- En el caso básico la función objetivo que se busca minimizar es la cantidad de cuatrimestres totales. Para ello, se busca el valor más grande de los números de cuatrimestres. Posteriormente, esta función objetivo se modificará para ser penalizada por otras restricciones.

$$Z_{min} = \text{Max}(C_i); \forall i \in [Materias]$$

### 3.9. Armado y Visualización del Plan de Carrera

Para permitir el armado del plan, se deberán poder establecer las preferencias del alumno. Estas preferencias serán guardadas y precargadas cada vez que el alumno desee volver a diseñar el plan por algún motivo (por ejemplo, no completó las materias que se proponía cursar y requiere ajustar las materias que realmente puede cursar el cuatrimestre siguiente, o que comenzó a trabajar y por ende, tiene menos disponibilidad horaria, etc).

Para ello clasificaremos las preferencias en fuertes [F] (la restricción debe cumplirse) o deseables [D] (si no se cumplen, se pondera negativamente):

- Cantidad de cuatrimestres máximos de duración del plan [F]
- Máxima cantidad de horas de cursada por semana [F]
- Máxima cantidad de horas de trabajo extra además de la cursada, por semana [F]
- Días y horarios que el alumno tiene disponibles para cursar [F]
- Preferencias de topics de las materias electivas [D]: Seleccionar los porcentajes deseados para cada Topic.
- Puntuación del curso mayor a un determinado valor [D]

## 4. Tecnologías y Herramientas

### 4.1. Librería: PuLP

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.

### 4.2. Herramientas

Categoría	Herramientas
Backend	Python, Flask
Frontend	JavaScript, React
Base de Datos	?
Entornos de Desarrollo	Sublime, Ninja IDE
Control de Versiones	GitHub
Administración y Control del Proyecto	GDocs, Trello
Elaboración de Documentos	LaTeX
Librerías Principales	PuLP

## 5. Plan de Trabajo

### 5.1. Equipo de Trabajo

El equipo de trabajo está compuesto por:

- Tutor: Lic. Rosa Graciela Wachenchauser
- Co-tutor: Ing. Diego Essaya
- Desarrollador: Jennifer Andrea Woites

### 5.2. Metodología

Para el desarrollo de este trabajo se seguirá con una metodología mixta con Kanban y Scrum.

Por un lado, se tendrá un tablero con las tareas principales como 'Pendientes', 'En curso', 'Desarrollo Terminado', 'En Testing', 'Finalizadas', y se las hará mover a través del tablero según corresponda; para poder tener una visión global del proyecto.

Cada una de estas tareas principales será explotada en varias subtareas o tareas secundarias. Cuando todas las tareas secundarias de una tarea hayan finalizado, se podrá mover la tarea principal a 'Desarrollo Terminado' de forma que después pueda ser tomada para un testing integrado y completo de las misma.

Cada una de las tareas secundarias se corresponderá con un issue en gitHub para facilitar el trackeo de los commits con respecto a la funcionalidad que se estaba desarrollando.

Por otro lado, se establecerá un período de tiempo (como un sprint) en el que se establecerán las tareas secundarias deseadas a desarrollar durante ese lapso. Al finalizar el sprint se realizará la entrega del avance que haya sido completado. Los sprints durarán un tiempo no mayor a dos semanas. En caso que durante el sprint se detecte que se debe modificar alguna tarea, se realizará el ajuste correspondiente y se intercambiará por otras menos prioritarias de la misma duración que las nuevas que se incorporan.

A medida que se vayan avanzando en los sprints se realizarán los ajustes que se consideren necesarios en base a lo que se establezca con el tutor y el cotutor en las correspondientes reuniones.

### **5.3. Estimaciones**

### **5.4. Cronograma**

A continuación se hace un cronograma de entregables tentativo al finalizar cada una de las iteraciones. Las mismas pueden ser modificadas en caso de verse necesario, por el tutor, el cotutor o el desarrollador. La fecha de cada una será determinada en conjunto con el tutor y el cotutor.

## 6. Referencias y Material consultado

- [1] «*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*»  
RODRIGO HERNANDEZ, JAIME MIRANDA P. Y PABLO A. REY  
[http://www.dii.uchile.cl/ris/RISXXII/horariosUDP\\_RISVersion%20FINAL.pdf](http://www.dii.uchile.cl/ris/RISXXII/horariosUDP_RISVersion%20FINAL.pdf)
- [2] «*Create responsive websites with Django, REST and AngularJS*»  
HANNES HAPKE  
<https://www.slideshare.net/hanneshapke/create-responsive-websites-with-django-rest-and-angularjs>
- [3] «*Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia*»  
ENRIQUE CASTILLO, ANTONIO J. CONEJO, PABLO PEDREGAL, RICARDO GARCÍA Y NATALIA ALGUACIL  
<http://www.dia.fi.upm.es/~jafernán/teaching/operational-research/LibroCompleto>
- [4] «*PuLP Classes*»  
<https://www.coin-or.org/PuLP/pulp.html>
- [5] «*Linear Programming with Python and PuLP*»  
BEN ALEX KEEN  
<http://benalexkeen.com/linear-programming-with-python-and-pulp/>
- [6] «*Optimization with PuLP*»  
STUART MITCHELL, ANITA KEAN, ANDREW MASON, MICHAEL O'SULLIVAN, ANTONY PHILLIPS  
<https://pythonhosted.org/PuLP/>
- [7] «*Parser de Materias para el Organizador de Horarios*»  
LUGFI  
<https://github.com/lugfi/organizador-fiuba/blob/master/tools/parser.js>