



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

REDISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECuento DE UNIDADES
DOCENTES PARA LA FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

WILLY ADOLFO MAIKOWSKI CORREA

PROFESOR GUÍA:
PATRICIO POBLETE OLIVARES

MIEMBROS DE LA COMISIÓN:
JAVIER VILLANUEVA GONZÁLEZ
LUIS MATEU BRULE

SANTIAGO DE CHILE
NOVIEMBRE 2015

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN
POR: WILLY ADOLFO MAIKOWSKI CORREA
FECHA: NOVIEMBRE 2015
PROF. GUÍA: SR. PATRICIO POBLETE OLIVARES

REDISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECuento DE UNIDADES DOCENTES PARA LA FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

*"... Gracias a los que me han dado el apoyo necesario, sigo,
gracias a ti por estar conmigo (por querer),
y a tantos más..."*

Agradecimientos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Tabla de Contenidos

Introducción	1
1. Descripción del Proyecto	2
1.1. Antecedentes	2
1.2. Motivación	6
1.3. Objetivos	6
1.4. Metodología	7
2. Marco Teorico	8
2.1. Complejidad	8
2.2. Teoría de Grafos	9
2.3. Optimizacion Combinatorial	11
3. El Problema: Situacion Actual	13
3.1. Combinaciones	13
3.2. Proceso Presente	16
4. Solución: Modelo y Rediseño	18
5. Implementación	23
6. Validación	25
Conclusión	25

Índice de Tablas

6.1. Tabla 1	27
------------------------	----

Índice de Ilustraciones

1.1.	Procesos necesarios para la obtención de un Título Profesional	3
1.2.	Árbol de planes y ramos con sus respectivas reglas y unidades	4
1.3.	Planes de la carrera de Ingeniería Civil en Computación (v3)	5
2.1.	Grafico con las curvas de los ordenes de complejidad en una escala logaritmica	9
2.2.	Ejemplo de grafo dirigido	10
2.3.	Calculos hechos en serie de fibonacci siguiendo perspectiva top-down.	12
3.1.	Estadísticas de peticiones por día relacionadas al recuento de UD's.	16
4.1.	19
4.2.	20
4.3.	21
4.4.	22
6.1.	Logo de la Facultad	26

Introducción

Hace algún tiempo en la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, en una etapa previa del proceso de titulación se debía realizar un recuento manual de los ramos y sus respectivos créditos (Unidades Docentes o UDs). Esto quiere decir que se debía corroborar los ramos aprobados con el respectivo plan y título que se quería obtener. Este trabajo que parecía sencillo, resultó ser arduo y complejo, lo que implicaba que una solicitud de recuento demorara meses en ser calculada.

A partir del año 2013 el Área de Infotecnologías (ADI) implementó un sistema de recuentos automático para apoyar esta labor manual, el que implicó una mejora considerable, permitiendo que la misma labor pasara de realizarse de meses a segundos.

El problema consiste en verificar si un alumno específico cumple con un plan de estudios. Cuando hay más de una manera posible de que el plan se cumpla, se busca maximizar la nota con la que debería egresar. La tarea de comprobar el cumplimiento del plan actualmente no está siendo abarcada en su totalidad. El sistema puede calcular que un alumno no cumple con un plan cuando realmente sí lo hace (falso negativo). Lo anterior se considera crítico.

También hay problemas respecto del promedio de titulación que entrega el sistema. Como es posible que un plan de estudios se cumpla a través de distintas combinaciones de cursos y a la forma en que está implementada la solución, se alcanzan a examinar sólo algunos resultados posibles, pero no necesariamente se encuentra el óptimo global.

Aunque institucionalmente no es parte del reglamento el que se deba maximizar la nota del avance curricular, esto es de gran interés para el alumnado, por lo que es parte de lo que se pretende mejorar.

Las preocupaciones descritas anteriormente provocan que este recurso, si bien es ampliamente ocupado, se utilice sólo como información referencial, debiéndose revisar posibles errores. Todo lo antes descrito motivó la propuesta de este tema para mejorar el actual sistema.

Capítulo 1

Descripción del Proyecto

A continuación se presentará los conceptos básicos necesarios para la comprensión de la memoria. Primero, se introducirá la plataforma en la que esta alojada la solución actual para, luego, explicar paso a paso el modulo y sus elementos.

Después, se profundizará en el elemento principal, Los planes. Para ello se mostrará su estructura, reglas y alcances. Además, se ejemplificará una carrera para mostrar como se realiza la comprobación de satisfacción y el calculo del promedio.

Posteriormente, se profundizará en los elementos que motivaron esta memoria y la relación que poseen en conjunto con los objetivos.

Finalmente, se mencionará concisamente la metodología seguida para afrontar el problema y la implementación de la solución.

1.1. Antecedentes

El Área de Infotecnologías (ADI) es un equipo de personas dedicadas a dar soluciones tecnológicas a la Universidad de Chile. Entre las soluciones que ha presentado se destacan dos plataformas: U-Cursos y U-Campus.

La primera, esta orientada al apoyo docente siendo un nexo entre el alumno y el profesor. Es decir, permite que existan foros para relacionarse, secciones de material docente para compartir documentos, un lugar para publicar las notas que va obteniendo el alumno en sus evaluaciones, entre otros.

En cambio, U-Campus esta orientado a lo administrativo. Hay variados procesos que se dan en la universidad en el que existe interacción entre un alumno y los funcionarios no académicos. Un ejemplo de ello era la entrega de certificados en donde un alumno se dirigía donde un funcionario para su solicitud.

Aun así, una amplia cantidad de estos procesos han pasado por un periodo de automatización de manera tal de facilitar la parte operativa y orientar la administración en velar por el cumplimiento estratégico de la universidad. Entre estos procesos se encuentra la titulación. Ésta consiste en variadas etapas que distintas áreas de la facultad deben verificar de manera tal de chequear finalmente si un alumno esta habilitado para obtener un titulo profesional.

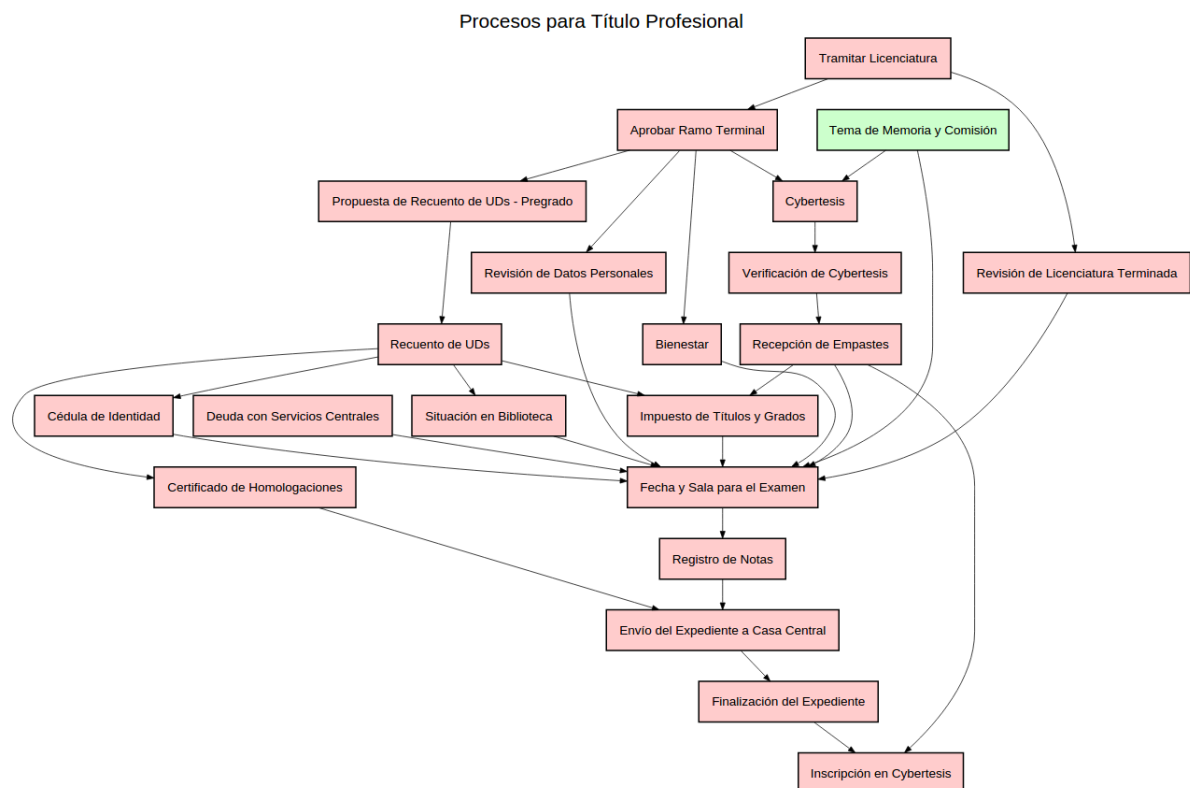


Figura 1.1: Procesos necesarios para la obtención de un Título Profesional

Una de las etapas de verificación es el chequeo de si un alumno posee el avance académico suficiente. Este procedimiento es conocido como el “Recuento de UD’s” debido a que un alumno debe ir aprobando cursos que son medidos en ¿Unidades Docentes¿ (unidad de medida de demanda horaria) de tal manera que se cumpla una serie de reglas. Estas reglas antiguamente consistían netamente en sumar una cantidad específica de UD’s y de allí nació la palabra recuento.

En la antigüedad, este procedimiento se hacia manualmente y tomaba meses en ser calculado. Anteriores memorias intentaron lograr que este procedimiento fuera cada vez mejor pero no fueron fructíferas. Por ejemplo, en [5] solo se logró el entendimiento en profundidad del problema proponiendo una solución y en [2] se logro llegar a un sitio web que solo reemplazaba lo que antiguamente se hacia en papel por un formulario web sin modificar en nada el proceso.

Finalmente, hace algunos años en U-Campus se agregó un modulo dedicado para ello que centralizaba toda la información del alumno con sus cursos aprobados. El modulo acogió el nombre de “Recuento de UD’s” y logró minimizar los tiempos de meses a segundos.

Para lograr evaluar el avance académico de un alumno se debió modelar un título profesional en planes. Un plan es un conjunto de ramos y/o subplanes con una regla específica. Dado que poseen subplanes y estos, a su vez, también pueden poseer subplanes se empieza a armar un estilo de árbol en donde el plan principal sería la cúspide siendo el último en ser evaluado debido a que para cumplirlo se debería concluir previamente los de más bajo nivel, por ende a este plan se le conoce como la carrera (si es que hablamos de título profesional).

Las reglas mencionadas previamente pueden ser tres: regla contar, UD's o todo. La regla contar quiere decir que para concluir el plan se debe aprobar cierta cantidad de ramos. Si un plan con esta regla posee, a su vez, un subplan con la misma regla, los ramos aprobados del subplan son adicionados al contador del plan principal. La regla UD's es similar pero en vez de contar cantidad de ramos aprobados lo que se hace es sumar las unidades docentes de los ramos aprobados.

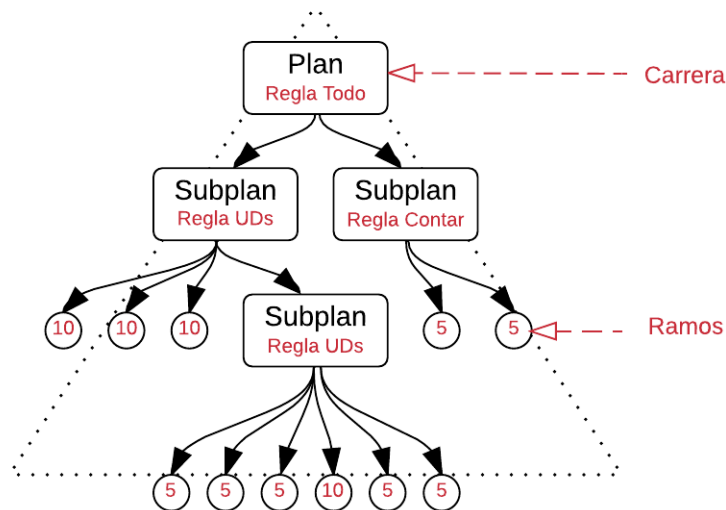


Figura 1.2: Árbol de planes y ramos con sus respectivas reglas y unidades

En la imagen se puede observar el árbol de planes y ramos con sus reglas y unidades, respectivamente. Para el caso del plan de más bajo nivel que posee regla UD's, si esta fuera sumar veinticinco se podría tomar una combinación de ramos que dé esa cantidad, por ejemplo, todos los ramos de cinco unidades. Para el plan con regla contar, si esta fuera de contar una unidad se debe aprobar al menos uno de ellos sin importar las unidades que poseen los ramos.

Dado que las reglas son de contar o sumar alguna unidad, existe la forma de no cumplir un subplan. Si el plan del segundo nivel que posee regla UD's tuviera que sumar treinta unidades, estas podrían sumarse a partir de solo los ramos que posee, sin necesidad de sumar las unidades del subplan de él. De esta manera el subplan de más bajo nivel se daría por cubierto aunque no se cumpla su regla.

Aun así, existe la regla "todo", que no permite el comportamiento anterior debido a que obliga a cumplir todos los elementos (ramos y subplanes) que pertenecen a un plan. En la imagen anterior, la carrera posee la regla todo y, por ende, los dos subplanes que contiene deben cumplirse de manera obligatoria.

En la Facultad de Ciencias Físicas y Matemáticas el modulo recuento de UD's es ampliamente ocupado y las carreras que han sido modeladas poseen variedad de combinaciones de las reglas antes descritas. Además, se poseen versiones debido a que en ciertas generaciones los cursos que se dictan son distintos, siendo un hecho natural debido al avance académico y a las innovaciones que se realizan a la malla curricular año tras año.

Un cambio de versión implica referencias complejas dentro de los ramos debido a que dos ramos en una versión pueden unirse en uno a la fase siguiente implicando diferencias en la cantidad de UD's, la nota que debe ser puesta, entre otros. Es decir, un alumno que poseía esos dos ramos antiguos aprobados y pasa a la nueva versión del plan tendrá inconsistencias en esos valores. Por lo anterior, el manejo de versiones en los casos complejos no se considera dentro del alcance de esta memoria pero se abarcará las equivalencias uno a uno que puedan poseer los ramos.

Si observamos como ha sido modelada la carrera ¿Ingeniería Civil en Computación¿ tendremos un ejemplo claro del diseño estándar dentro de la facultad. A continuación se muestra un diagrama con la ultima versión donde se evidencia la tenencia de una linea de cursos obligatorios y otra de cursos electivos.

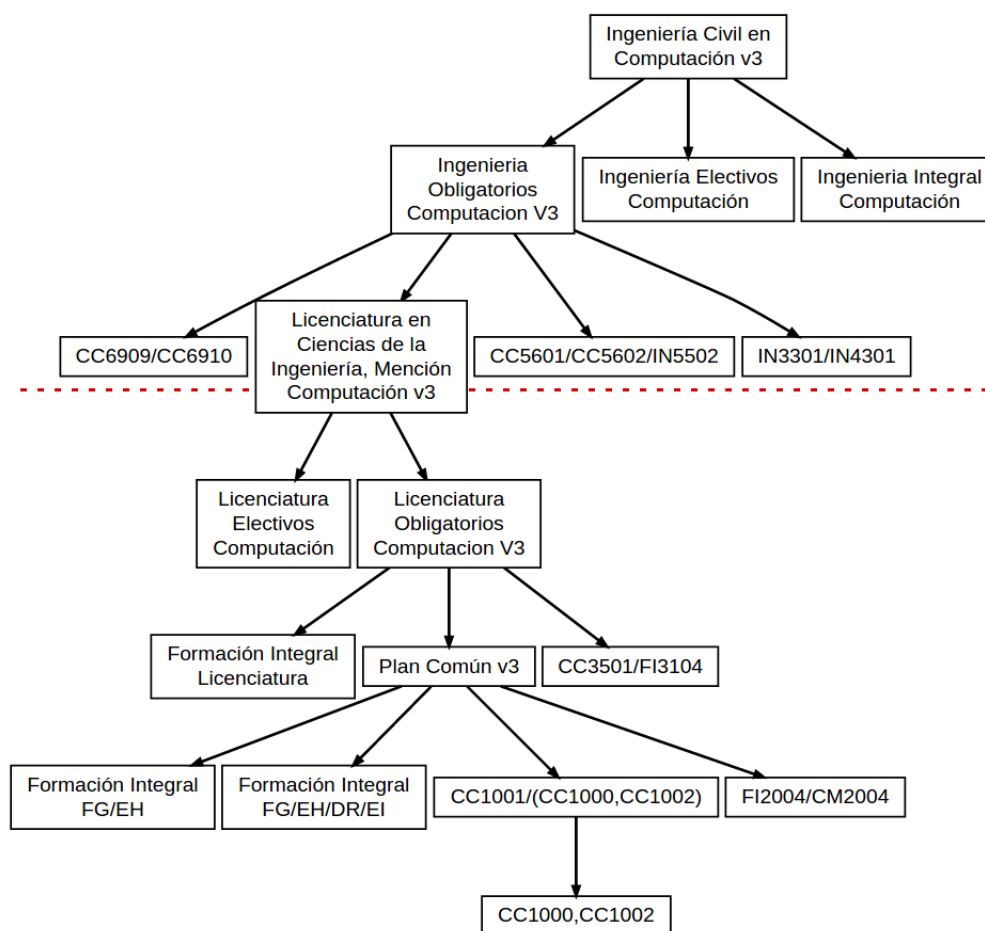


Figura 1.3: Planes de la carrera de Ingeniería Civil en Computación (v3)

Un punto importante de la titulación es la nota con la que un alumno egresa. Para realizar

ese calculo se debe tener en consideración el tipo de titulo que se obtiene debido a que es posible promediar todos los ramos o, si es titulo profesional, promediar solo los ramos de los planes de carrera ([1]). En el caso de la imagen anterior, se tendría un promedio con las notas de los planes que están superior a la linea roja punteada pero en el caso de otro tipo de titulo se deberían promediar todos los ramos.

Por todo lo antes descrito, una correcta solución debe velar por satisfacer dos condiciones: determinar si un alumno esta en momento de titularse y encontrar la mejor nota para ello. La solución actual no confirma del todo estas dos condiciones ya que se tienen falsos negativos (puede mencionar que un alumno no puede titularse siendo que si posee todo el avance curricular necesario) y no se tiene garantía de que entregue la mejor nota.

1.2. Motivación

Si bien actualmente se tiene una mejora considerable de lo que es la titulación, en lo que respecta a la propuesta de recuento de unidades docentes, aun se sigue con un procedimiento de chequeo manual. Durante el tiempo ésta se ha ido acortando cada vez más pero un desarrollo mayor en esa área significaría el punto culmine logrando obtener resultados confiables y en un tiempo considerable.

Además, el desafío es de gran envergadura lo que incentiva solucionarlo. Como se mencionó en los Antecedentes, hubieron dos memorias en donde se intento llegar a una mejora automática a este proceso. En [5] **explicar que se hizo en esa memoria.**

Por otra parte, en [2] se tenia presente la anterior memoria por lo que se logro avanzar en la automatización pero a partir de una perspectiva de apoyo netamente y no de reemplazar parte del proceso. Esto quiere decir que se construyó un sitio en el que la persona encargada debía seguir los mismo pasos que si lo hiciera con papel. Si bien se podía centralizar datos y se contaba con una base de datos donde se guardaban los registros de recuento, no logró mayor mejora.

Finalmente, el problema (que se abordará de manera mas profunda posteriormente) consiste en asignar ramos dentro de planes de manera de encontrar una combinación que satisfaga las dos condiciones mencionadas en los antecedentes. Este tipo de problemas, de calcular múltiples combinaciones, puede tomar mucho tiempo. Aun así, dado el entendimiento del caso que se ha hecho y los intentos previos de automatización, se puede lograr una reducción ad-hoc que permita lograr el desafío y por ende motiva aun mas la búsqueda.

1.3. Objetivos

Los objetivos de este trabajo de memoria se describen a continuación:

General:

- Rediseñar e implementar el actual sistema de recuento de UD's para asegurar la confiabilidad de los resultados, tanto desde el punto de vista del cumplimiento del plan de estudios como del cálculo del promedio de titulación, de manera tal que al ser utilizado se resuelva en un tiempo razonable para un sitio web (menos de 10 segundos).

Específicos:

1. Enunciar el problema, sus distintas extensiones y restricciones.
2. Buscar posibles algoritmos de solución en el estado del arte .
3. Analizar la potencial factibilidad para tomar una decisión respecto a un método.
4. Seleccionar un método para la verificación del cumplimiento del plan de estudios.
5. Seleccionar a lo sumo dos métodos para maximizar el promedio del alumno, de manera de acotar el desarrollo futuro.
6. Desarrollar el modelo y la implementación de los métodos seleccionados.
7. Establecer los casos tipo que sirven para validar la resolución del problema.
8. Evaluar los algoritmos con los casos tipo.

1.4. Metodología

El ADI posee un equipo de profesionales de distintas aptitudes que siguen una metodología. Inicialmente se realiza un estudio del lugar donde se encuentra la oportunidad de mejora, logrando un entendimiento profundo tanto de la situación actual como de las necesidades futuras. Luego, se replantean procesos y se diseñan las herramientas de apoyo de manera que, finalmente, se implemente una solución ad-hoc.

Gran parte del entendimiento ya se poseía debido a las anteriores memorias y a la implementación previa que poseía el equipo. Por lo tanto, los pasos que se debían seguir serían el lograr entender el problema a partir de lo ya obtenido, comprender la implementación actual y, luego, rediseñar e implementar una nueva solución.

Las plataformas desarrolladas por el ADI poseen un ambiente de desarrollo. Esto quiere decir que se posee una copia de la herramienta pero en un servidor distinto al público en donde pueden realizarse pruebas con datos reales o simulados. Además, están estructuradas en módulos administrables que permiten el manejo de permisos y aislación de manera tal que el desarrollo de una persona no afecte a otra.

Capítulo 2

Marco Teorico

Gran parte de los conceptos utilizados en esta memoria no son necesariamente de conocimiento general. Es por ello que a continuación se explicara de manera breve los elementos mas utilizados.

2.1. Complejidad

Cuando se deben comparar procedimientos o algoritmos se debe utilizar alguna unidad de medida. Una unidad importante es el tiempo (cuanto tiempo demora uno con respecto al otro) pero este es relativo a la cantidad de datos que se procesan, por ejemplo, es distinto intentar ordenar cien números a ordenar un millón. Por lo anterior, para comparar dos procedimientos, comúnmente se define una función $T(n)$ donde n es la variable que define el tamaño de los datos.

A partir de estos hechos y del procesamiento de los datos que se realiza en el algoritmo, se va describiendo el comportamiento de la función. Por ejemplo, si el algoritmo visita dos veces cada uno de los datos la función estaría definida por $T(n) = 2 \cdot n$.

Hay que tener en consideración, además, que esta definición es independiente de la tecnología utilizada para procesar. Es decir, si utilizamos una maquina con mayor capacidad se procesará una cantidad de datos determinada mucho mas rápido pero seguirá teniendo la misma función, por lo tanto es común enunciar la tecnología utilizada.

Usualmente no es utilizada directamente la función $T(n)$ y se usan asintotas que determinan el mejor y peor caso en que se puede comportar el algoritmo. Esto permite un análisis mas rápido debido a reducciones que se pueden utilizar. En esta memoria se trabajará con el peor caso que se denota con una letra O . Debido a la descripción de esta ultima función se puede tener distintos tipos de complejidades, por ejemplo cuando se tiene una función $O(n^2)$ se dice que esta tiene un comportamiento cuadrático para hacer referencia al polinomio de segundo grado o cuando se tiene una función $O(3^n)$ se dice que es de orden exponencial debido a que la variable se encuentra en el exponente.

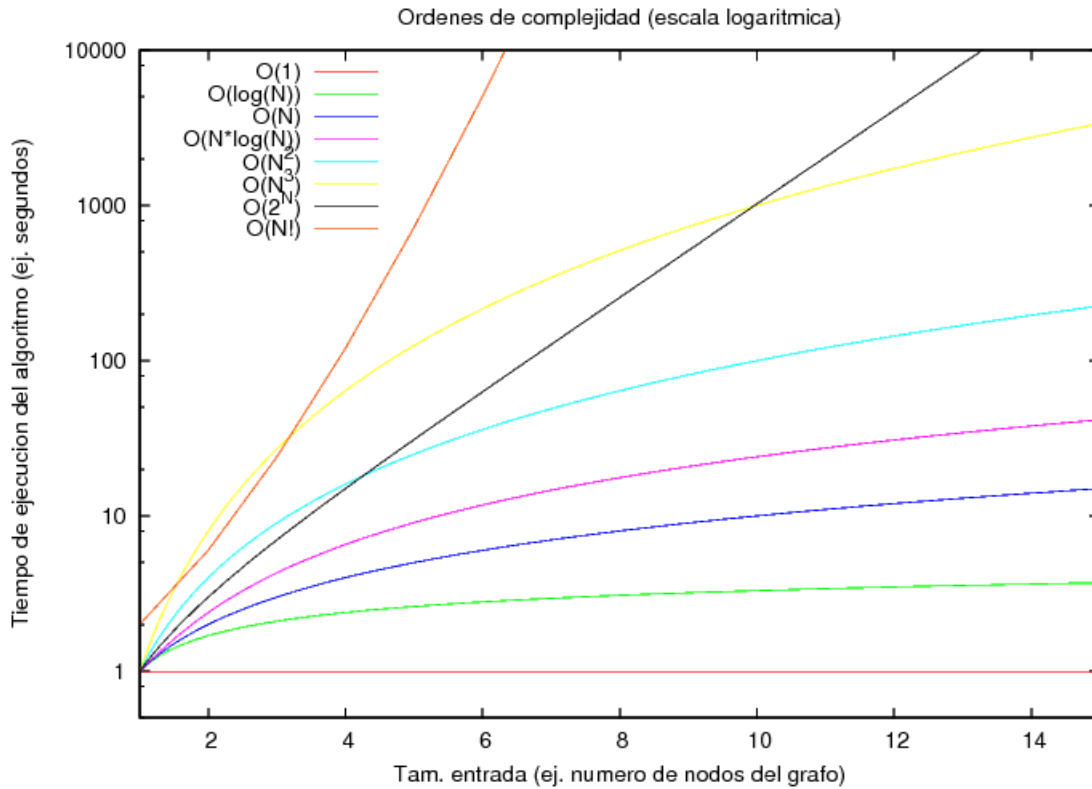


Figura 2.1: Grafico con las curvas de los ordenes de complejidad en una escala logarítmica

En la figura 2.1 se pueden ver los tiempos que demora en procesar un algoritmo una cantidad de datos determinada en una escala logarítmica donde mientras más arriba diverge la función más difícil o más tiempo toma su calculo. En computación se tienen distintos tipos de clases de complejidad para clasificar los problemas según la mejor solución descubierta por el momento siendo las mas conocidas la clase P, los problemas que pueden ser resueltos en tiempo polinomial, y NP, los problemas que pueden ser resueltos en tiempo polinomial pero solo por una maquina no determinista. Estos últimos poseen un subconjunto de problemas conocidos con una complejidad mucho mayor y que son reducibles entre ellos, es decir, un problema puede ser representado o transformado a una variante equivalente de otro problema dentro de ese subconjunto.

Otra unidad de medida diferente al tiempo es la unidad de espacio. Con esto se evalúa la cantidad de memoria necesaria que requiere una solución. En esta tesis se tendrá énfasis en la unidad de tiempo pero en la sección ¿Validación¿ se mencionará brevemente los cuidados que hay que tener con el espacio.

2.2. Teoría de Grafos

Un campo que ha tomado gran importancia en la computación es la teoría de grafos. Los grafos son una estructura que esta compuesta por dos elementos: los nodos y las aristas. Las

aristas pueden tener dirección y en ese caso se habla de un grafo dirigido.

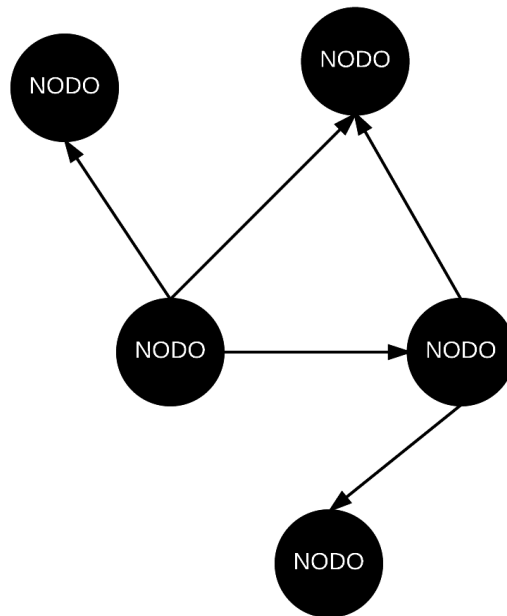


Figura 2.2: Ejemplo de grafo dirigido

La forma que posee esta estructura ha sido de utilidad debido a que muchas cosas pueden representarse mediante ella. Por ejemplo, posiciones dentro de un mapa pueden ser representadas por nodos y las aristas son los caminos entre ellas teniendo la habilidad de agregarle valores (la arista tiene un valor mayor mientras más alejados están los nodos). A partir de esto y una representación matemática de los elementos de un grafo pueden resolverse problemas teóricos con variada utilidad en la práctica. Siguiendo el mismo ejemplo anterior, la solución teórica del camino más corto entre dos nodos podría representar el camino más corto entre dos lugares geográficos.

Existen variados tipos de grafos entre ellos los grafos conexos que se definen como los grafos donde para todo par de nodos se tiene un camino que los une. Cuando no se posee un grafo totalmente conexo se pueden tener subconjuntos que si lo sean a los que llamaremos componentes conexas.

Otro caso particular de grafo son los árboles. Este es un grafo conexo en que la unión de todo par de vértices es única, es decir, se posee solo un camino entre un nodo y otro. En este caso, si bien las aristas en un árbol no poseen dirección, se puede armar un estilo de jerarquía donde el nodo desde donde se originan las primeras aristas es conocido como la raíz. Además, es común en este tipo de estructuras describir los nodos como padres e hijos de acuerdo al nivel que poseen y desde cual nodo se originan. Los nodos que ya no poseen nodos hijos son conocidos como las hojas del árbol.

Dentro de la teoría de grafos y las componentes conexas, existe un concepto útil de conocer para esta memoria conocido como ¿la componente gigante? (Giant component en inglés). Si

bien es un estudio que implica variadas cosas y utiliza conceptos de probabilidades, uno de los elementos importantes es que si se posee un grafo de n nodos y se van adicionando aristas aleatorias, si la cantidad de aristas adicionadas supera aproximadamente $n/2$ con alta probabilidad se tendrá una componente conexa gigante. **Quizas no sea tan importante en la memoria y deba sacarse**

2.3. Optimizacion Combinatorial

La optimización es un área de las matemáticas donde se busca determinar la mejor manera de realizar una actividad bajo criterios extraídos del modelamiento del problema. Un ejemplo es maximizar los ingresos de alguna institución o disminuir los tiempos de ejecución de una tarea.

En el caso particular, la optimización combinatorial involucra encontrar la solución dentro de un conjunto discreto. Comúnmente, en los problemas relacionados se tiene la opción de encontrar la solución con una búsqueda exhaustiva pero debido a la magnitud del conjunto el tiempo necesario para llegar a la confirmación de que se esta en presencia de la mejor solución se vuelve muy grande y por lo tanto el método no es factible.

Usualmente, un problema de optimización combinatorial puede representarse como un árbol de decisión. Es decir, un grafo donde cada nodo diverge hacia otros nodos según la cantidad de opciones que se tiene sobre una variable, de esta manera para cada variable que se debe definir se tendrá una representación visual de alguna elección. Ejemplificando, si tenemos un problema de dos preguntas que se pueden responder con si o no, tendremos un nodo donde se decida sobre la primera pregunta y se tendrá dos aristas debido a las dos opciones de respuesta. Luego, en los nodos hijos se representara la decisión de la segunda pregunta pero teniendo como dependencia la decisión ya tomada, por lo tanto las hojas de este árbol de decisión representan las distintas combinaciones que se pueden obtener del problema conjunto.

Variados problemas de optimización combinatorial presentan una complejidad de clase NP y muchos de ellos pueden ser representados con grafos. Además, como no se logra encontrar una solución rápida, los problemas son estudiados en profundidad documentándose distintos avances con algoritmos y heurísticas. Estos pueden tener mejoras en casos particulares y es importante conocer algunos de ellos que se parezcan al caso de la presente memoria.

Casos conocidos son el problema de la mochila o el de mudanza. En ellos es necesario poner objetos dentro de algún contenedor de tal manera de maximizar la cantidad de objetos que se pueden llevar, maximizar la utilidad de esos objetos, minimizar la cantidad de contenedores necesarios u otra optimización. Martello Toth [3, 4] propusieron algoritmos para resolver este tipo de problemas siendo una de ellos la estrategia de ramificación donde el ordenamiento de los objetos se realiza por valor decreciente (tamaño, peso o importancia dependiendo del objetivo). De esta manera, para cada nodo de decisión se tiene el objeto de mayor valor y es ubicado en el contenedor de la más próxima factibilidad.

Cuando no se posee una solución exacta adecuada se puede incurrir en reducciones del conjunto de combinaciones con relajaciones de las restricciones. Aun así se debe tener presente conceptos como el de programación dinámica para no incurrir en cálculos reiterados de un subconjunto. Para ejemplificar lo anterior podemos intentar resolver la sucesión de fibonacci la cual esta dada por:

$$F(n) = F(n - 1) + F(n - 2), F(0) = 0, F(1) = 1$$

Para calcular el valor de $F(5)$, de acuerdo a la secuencia, se debe calcular el valor de $F(4)$ y $F(3)$. Aun así, para calcular el valor de $F(4)$ se debe calcular nuevamente el valor de $F(3)$ y el valor de $F(2)$. Si se siguiera lo anterior el calculo de $F(3)$ se realizaría repetitivamente siendo que no es necesario si ya fue calculado una vez, por lo que se podrían realizar dos mejoras a la forma en que fue orientada la solución al problema: tener en memoria los cálculos ya hecho para consultarlos posteriormente o seguir una estrategia bottom up (comenzar desde $F(1)$ hasta llegar a $F(n)$).

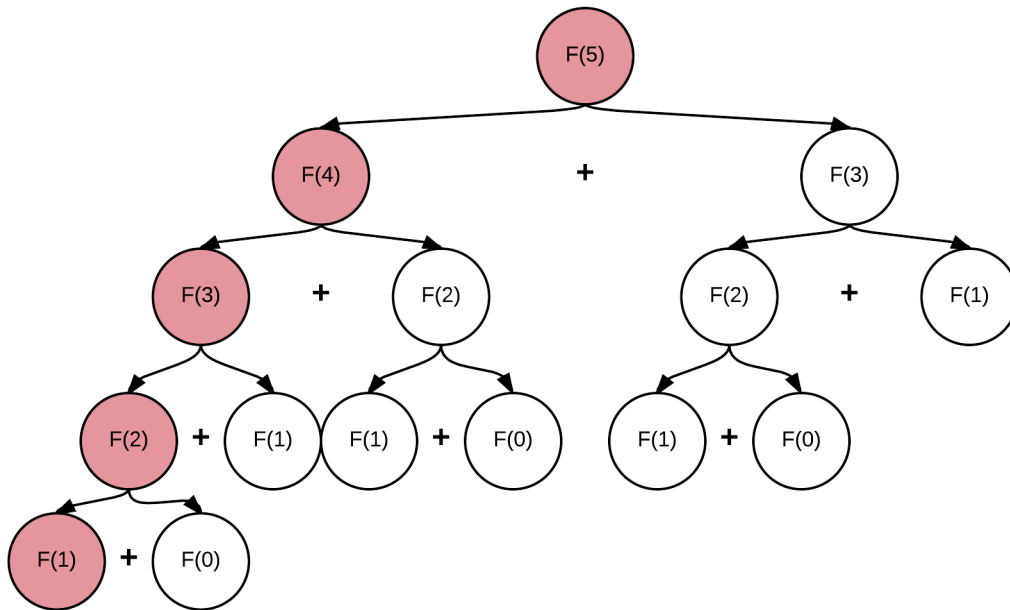


Figura 2.3: Cálculos hechos en serie de fibonacci siguiendo perspectiva top-down.

Capítulo 3

El Problema: Situación Actual

El recuento de créditos inicialmente era un proceso netamente manual. Estaba constituido por un solo actor que a partir de la experiencia podía manejar las diferentes condiciones que se generaban. Ante eventualidades, como casos extraños en el que faltaban UD¿s en cierto plan, el mismo actor tenía la potestad de flexibilizar las reglas para chequear si realmente un alumno cumplía con la carrera.

Como se mencionó en la descripción del proyecto, este proceso fue mejorando poco a poco logrando que existieran componentes automáticas en el procedimiento, tratando de suplir todo el conocimiento generado del actor mencionado en el párrafo anterior. Aun así, la parte automatizada posee errores de tal manera que se debe realizar un chequeo manual posterior.

Los errores se deben a la complejidad de las reglas que se deben cumplir y a la naturaleza del problema. Por ende, se ahondará en ello en la sección ¿Combinaciones¿ explicando el modelo e implementación actual.

Finalmente, en la sección del proceso actual se abordará el procedimiento realizado por el SGD para suplir la falta de certeza de la heurística en ciertos casos y las estadísticas de ocurrencia de los errores.

3.1. Combinaciones

Para entender el problema de forma mas detallada podemos realizar un simplificación del recuento y reducirlo a un problema de mudanza (bin packing). Para ello debemos asimilar los planes como si fueran cajas, y los ramos como si fueran elementos que deben ser guardados dentro de estas cajas para poder ser trasladados.

Para este problema la solución trivial seria ir intentando distintas combinaciones posibles de tal manera de lograr que todos los elementos sean guardados. Esto seria la solución a fuerza bruta del problema entregando la mejor solución pero el tiempo necesario para realizar todas combinaciones posibles es muy grande. No se posee un tiempo ilimitado y, por ende, lo mas

natural sería comenzar a privilegiar algunos elementos guardando inicialmente los de mayor valor para luego ir rellenando con los de menor valor o los que se podrían dejar olvidados (Como lo realizado por Martello y Toth descrito en el "Marco Teórico").

En el caso de la solución presente del recuento de UD's se utiliza la anterior heurística. Se consideran los ramos de valor aquellos que poseen mayor nota y son de nivel mas alto (se debe privilegiar poner los ramos de mejor nota en la carrera debido a que ellos son considerados para el promedio final). Finalmente, esto iría mediante backtracking visitando todas las soluciones posibles pero teniendo en un inicio ramos de importancia fijos.

Matemáticamente, se puede definir que un ramo r puede pertenecer a cierto conjunto de planes P_r de tamaño n_r , y que un plan se ve cubierto si y solo si tiene asignado una cantidad determinada de ramos mayor o igual a L_p . Por lo anterior, el determinar el cumplimiento del plan y la nota se trata de un problema de asignación en el que se puede usar el ramo en alguno de sus planes. Si consideramos $X_{rp} \in \{0, 1\}$ como la variable de decisión que determina si se usa el ramo r en el plan p , el modelamiento del problema de calcular si se cumple el plan quedaría determinado por:

$$\begin{aligned}
& \max \sum_r \sum_p X_{rp} \\
& \text{s.a.} \quad \sum_r X_{rp} > L_p \quad \forall p \in \text{planes} \\
& \sum_p X_{rp} \leq 1 \quad \wedge \quad \sum_p X_{rp} \leq 0 \quad \forall r \in \text{ramos}
\end{aligned}$$

De esta forma, el orden de complejidad de este modelo vendría dado por la pitatoria $\prod_r n_r$. Es decir, en el caso hipotetico en que m ramos tuvieran tres opciones donde ser ubicados se tendría $\prod_r 3 \Rightarrow O(3^m)$, lo que equivale a un orden de complejidad exponencial. A partir de esto se descarta, a priori, la posibilidad de un problema mucho mayor en que al ser combinaciones de todos los ramos contra todos los planes se obtendría un orden factorial de complejidad.

Como se posee poco tiempo en un sitio web, este modulo es detenido a los quince segundos y por lo tanto puede suceder que hasta ese tiempo no se haya visitado una combinación que diga que todos los planes se cumplen siendo que quizás mas adelante podría encontrarla. En el caso del calculo de la nota se sigue el mismo comportamiento debido a que se realiza en conjunto con la comprobación del cumplimiento del plan principal.

Si bien se asegura que se tiene una buena distribución porque se asignaron los ramos de importancia, no se tiene certeza de la correctitud en el punto donde es detenido el algoritmo. Aun así, se privilegió este procedimiento porque estos casos generan un falso negativo, es decir, el error que puede producir es que un alumno que puede titularse el sistema le diga que aun no puede. Para estos sucesos se siguen los pasos que se explicaran en la sección de "proceso presente".

Aun así, lo descrito anteriormente solo es una relajación del problema. Si se consideran todas las reglas específicas de cada plan se puede armar un modelo como sigue:

$$\begin{aligned}
& \max \quad \text{usar}_{carr, carr} \quad : \quad carr \in \text{planes} \\
& \text{s.a.} \quad \sum_p^{planes} \text{usar}_{op} \leq 1 \quad \forall o \in \text{objetos} \setminus o \neq p \\
& \text{usar}_{pp} \leq \text{usar}_{op} \quad \forall p \in \text{planes}, \forall o \in \text{objetos} \setminus \text{regla}_p = \text{todo} \wedge \text{pertenece}_{op} = 1 \\
& \text{usar}_{pp} \cdot \text{cantidad}_p \leq \text{contar}_p \quad \forall p \in \text{planes} \setminus \text{regla}_p = \text{contar} \\
& \text{usar}_{pp} \cdot \text{cantidad}_p \leq \text{uds}_p \quad \forall p \in \text{planes} \setminus \text{regla}_p = \text{ud} \\
& \text{usar}_{pq} \leq \text{contar}_p \quad \forall p, q \in \text{planes} \setminus \text{regla}_p = \text{contar} \wedge \text{pertenece}_{pq} = 1 \\
& \text{usar}_{pq} \leq \text{uds}_p \quad \forall p, q \in \text{planes} \setminus \text{regla}_p = \text{ud} \wedge \text{pertenece}_{pq} = 1 \\
& \text{usar}_{pq} \leq \text{usar}_{pp} \quad \forall p, q \in \text{planes} \setminus \text{regla}_p = \text{todo} \wedge \text{pertenece}_{pq} = 1
\end{aligned}$$

En este modelo se generaliza lo que son los ramos y los planes debido a que se comportan de manera parecida. Es decir, como un plan puede poseer tanto ramos como subplanes y además un ramo puede verse como un plan debido a que puede poseer equivalencias, finalmente pueden modelarse como un único conjunto llamado objetos ($\text{ramos} \cup \text{planes}$). Además, puede utilizarse una variable que en el modelo le llamamos “Usar” que implica que un elemento “i” esta siendo usado en un elemento “j” de tal manera que esto implica los objetos considerados dentro de un plan. Un objeto estaría siendo usado en si mismo si y solo si es un ramo aprobado o es un plan donde se cubren todas las reglas relativas a él, y por ende de esta manera se podría verificar cuando una carrera se cumple.

De esta manera las restricciones vendrían a explicar reglas parecidas al modelo simplificado como que un objeto puede ser usado una única ocasión, que cuando se tiene una regla ¿todo"se deben usar todos los elementos dentro de él y que cuando se tiene regla ¿contar¿ o ¿uds¿ se debe utilizar una cantidad mayor o igual a la determinada.

La forma que esta implementado actualmente el sistema es mediante backtracking. Es decir, lo que se realiza es tomar un ramo de una lista, fijar el plan donde puede ser utilizado (escogiendo el primero de los candidatos para ese ramo) y continuar con el ramo siguiente. En cada paso se va evaluando si la solución que se tiene en ese instante es mejor que una guardada en una variable global y si lo fuese se actualiza esa variable. Luego, cuando todos los ramos se asignaron, el algoritmo debe ir devolviéndose de manera tal de ir probando los siguientes candidatos de cada ramo pero volviendo a calcular las combinaciones de los ramos siguientes porque cambió la situación.

Ejemplificando, si se poseen tres ramos A, B y C, se deben asignar los dos primeros y calcular todas las opciones para C. Luego, se debe retroceder un paso, cambiar el candidato de B y volver a calcular las opciones de C. Así de manera sucesiva hasta lograr calcular todas las opciones de B y A, respectivamente. Como se dijo anteriormente, no se posee un tiempo infinito y, por ende, el algoritmo debe detenerse, por lo tanto es probable que no se logre retroceder hasta A y solo se quede con el calculo de las mejores opciones de B y C. Es por

ello que los ramos determinados en un principio deben asignarse de la mejor manera posible para que el costo de dejar el algoritmo inconcluso se minimice.

3.2. Proceso Presente

Inicialmente, gran parte del recuento de UD's pertenecía a ¿Secretaria de Estudios¿ ya que ellos verificaban el cumplimiento de los requisitos académicos del proceso de titulación. Aun así, por la complejidad del proceso y las fallas encontradas en el modulo automático la Subdirección de Gestión Docente (SGD) debió ser parte del proceso para generar una propuesta inicial.

Este proceso inicia con la aprobación de un ramo terminal. Es decir, al cumplir un ramo que esta cercano al termino de la carrera se puede confirmar que un alumno va necesitar la comprobación de sus antecedentes académicos. Inicialmente, eso si, es necesario procesar la licenciatura asociada a la carrera del alumno para proseguir normalmente.

Luego, el SGD valida el resultado entregado por el modulo del recuento de UD's verificando que no se este en presencia de un falso negativo. Esto se evidencia ya que el sistema menciona que el alumno no cumple el plan y existen ramos no usados que con un re-ordenamiento de los ramos utilizados pueden disponerse y lograr una combinación válida.

La búsqueda de una combinación puede ser lenta y por ello se pueden recibir notificaciones de parte del alumno que busca que su proceso continúe como es debido. Además, él posee mucho mayor conocimiento de su avance curricular y por ende puede ayudar a la búsqueda de la solución. Dado esto, existe un modulo en U-Campus que permite la interacción entre alumnos e integrantes del SGD.

Estadísticas

Threads Recibidos por Día (866 - promedio: 2)

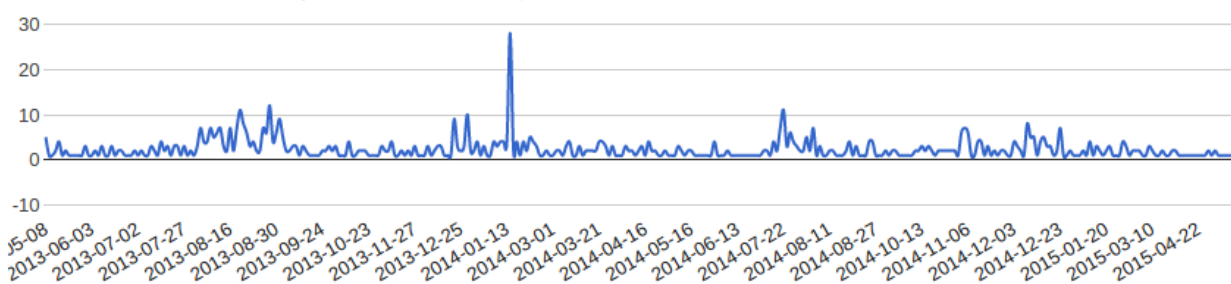


Figura 3.1: Estadísticas de peticiones por dia relacionadas al recuento de UD's.

Se posee un promedio de dos consultas por dia relacionadas con el recuento. Estas son resueltas comúnmente en algunos días pero existen ocasiones en que toman mayor tiempo (meses) debido a que involucran normas adicionadas en los cambios de planes de estudios. Esto esta íntimamente relacionado con las versiones de los planes explicadas en la descripción del proyecto.

Finalmente, cuando ya se posee una combinación valida o cuando el alumno aprueba lo necesario para el cumplimiento del plan, se envía el resultado del sistema con observaciones de los cambios necesarios para obtener la combinación valida si se requiere. Esta es recibida por la Secretaria de Estudios quien valida la propuesta donde la acepta confirmando el cumplimiento de los requisitos académicos o la rechaza para que el SGD justifique de mejor manera las observaciones agregadas.

Otros de los procesos que tiene a cargo el SGD son la inscripción académica, el catalogo de cursos, el calendario académico, entre otros. Muchos de ellos presentan alta carga y con tasas de errores dependientes del tiempo involucrado en ellos para una correcta planificación. Si se posee una mejora en la certeza del recuento de UD;s se disminuirían los reclamos por parte del alumnado y la cantidad de rechazos por parte de Secretaria de Estudios.

Capítulo 4

Solución: Modelo y Rediseño

Dentro de las opciones mencionadas en un problema de optimización combinatorial se encuentra la reducción del espacio de soluciones. Es decir, un buen entendimiento y modelamiento del problema deberían implicar un espacio mínimo pero no necesariamente, y es por ello que pueden existir aun formas de abordar de esta manera el rediseño.

En un recuento de UD¿s un alumno que esta cercano a titularse posee aproximadamente 70 ramos. Un plan de titulo profesional posee alrededor de 18 subplanes los cuales poseen distintas reglas y por ende los ramos antes descritos pueden utilizarse o no, dependiendo estas normas. Aun así, pensando en la flexibilización inicial que se hizo en la sección ¿Problema¿, si los ramos poseen en promedio 3 lugares donde ser ocupados tendríamos un numero cercano a 3^{70} combinaciones que recorrer para decidir cual seria la solución correcta. Si se eliminara un ramo se tendría 3 veces menos combinaciones.

En la imagen anterior se observan las reglas de los planes de la tercera versión de la carrera de ingeniería civil en computación. Han sidos dispuestos como un árbol de acuerdo a la dependencia que tienen entre ellos ya que, recordando, un plan puede tener subplanes dentro de él. Si desde la raíz del árbol se va bajando de nivel se puede observar que se sigue una linea donde la regla de los planes es ¿todo", es decir, se deben cumplir obligatoriamente todos los elementos pertenecientes a esa linea. Por ende, todos los ramos que se encuentran dentro de esos planes podrían verificarse en un inicio reduciendo la cantidad de ramos que pasarían posteriormente por el algoritmo.

Si bien se puede pensar que esta es una reducción considerable ya que se puede sacar del calculo cerca del 50% de los ramos, de todas formas no es del todo significativa ya que comúnmente estos ramos son considerados obligatorios dentro del modelamiento de los planes y por ende poseen a lo mas dos lugares donde ser ocupados. Además, se debe tener precaución debido a que, por ejemplo, los ramos dentro del plan de regla ¿todo"que esta en el nivel inferior poseen un plan padre que tiene regla ¿contar¿ y podría cumplirse sin la necesidad de aprobar los ramos del subplan.

Siguiendo la misma linea pueden reducirse los ramos que poseen un lugar donde ser ocupados debido a que si se agregaran al procesamiento del algoritmo, este considerará que pueden

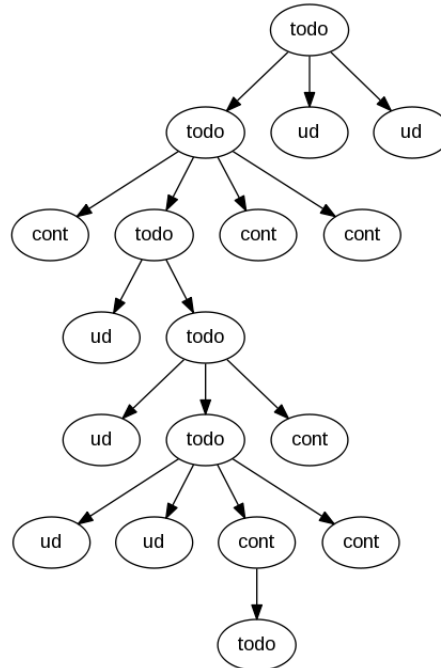


Figura 4.1:

ocuparse o dejarse fuera del calculo, y por ende no se estaría en presencia de tan solo un lugar donde ser ocupado, si no que dos. Eso sí existe una diferencia entre este tipo de ramos que con los con planes con regla ¿todo", ya que estos realmente podrían no ocuparse. Aun así, esta reducción podría considerarse dentro de las opciones porque para aprobar una carrera se podría ocupar la máxima cantidad de ramos posibles siendo perjudicada la nota pero se realizaría un post-procesamiento para mejorar aquello.

Lamentablemente, como se mencionó, la implementación actual ya realiza una reducción descartando poner un ramo dentro de un plan si este ya está cubierto. Si se realizará la reducción del párrafo anterior, se agregarían ramos con menor nota y el orden de poner los ramos de mejor nota inicialmente se vería entorpecido, provocando que la reducción de la implementación actual funcione incorrectamente. De todas formas se implementarían ambas y se realizaría una comparación para decidir cual es mejor dentro de ellas pero, a priori, debido a que los ramos reducidos de la implementación actual pueden tener más que una opción donde ser ubicados, implicaría una reducción mayor y sería una mejor decisión.

Yendo a la búsqueda de mejora dentro de la heurística ya implementada, se debe encontrar una perspectiva distinta para ver el problema. Si se modela éste como un grafo podríamos encontrar formas dentro de la teoría que podrían servir como mejora. Por ende, si se consideran los ramos como si fueran nodos unidos solo si poseen un plan en común dentro de sus opciones, se tendría una imagen como la que está a continuación.

Se observa un grupo de ramos de un alumno de la carrera de ingeniería civil en computación, en donde el nodo posee el nombre del ramo en su interior y entre ellos están unidos por aristas de distintos colores. El color indica el plan por el cual están unidos (detallado en la leyenda) existiendo ramos donde se origina un solo color debido a que solo tienen un

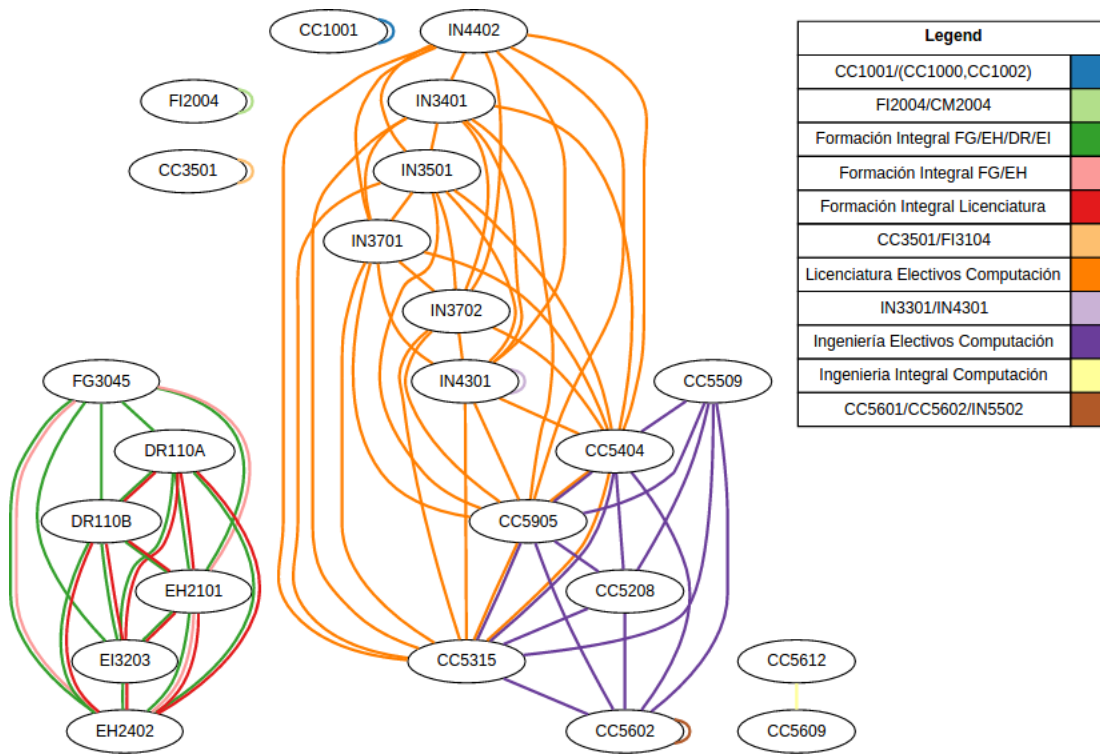


Figura 4.2:

candidato donde ser ubicados y pueden no estar asociados a ningún otro nodo.

Además, como se observa, existen componentes conexas dentro del grafo. Si bien esto a priori puede no significar mucho, si se contrasta con la implementación actual toma sentido un tipo de mejora hablada anteriormente como lo es la programación dinámica. Si se toma un ramo A y se puede gastar en los planes P o Q, sea cual sea la decisión con ese ramo esto no va a afectar la decisión de un ramo B que se puede gastar en los planes R o S. Esto debido a que no comparten ningún plan y pueden tomarse como problemas independientes. Se volverían dependientes uno sobre el otro si se tuviera un ramo C que pueda gastarse en Q o R (ambas opciones de A y B) debido a que si se gasta en Q implicaría menos espacio para A o si se gasta en R significaría menos espacio de solución para B. La solución actual esta considerando todos los ramos y sus decisiones dependientes de los demás y lo que significan las componentes conexas dentro del grafo es que existen subproblemas independientes.

De todas manera, inicialmente se consideró una dependencia completa debido a la forma de árbol de dependencia de los planes. Por lo mismo, se debe aun considerar reglas que no están siendo visualizadas dentro del grafo pero se ven en la visualización de reglas como lo son la posibilidad de no cumplir un plan debido a que el plan padre se cumple.

En el caso de la imagen, dado el calculo de un grafo de ramos de la segunda versión de la carrera ingeniería civil electricista se pintaron dos componentes conexas (azul y rojo) en el arbol de niveles de los planes. Si se sigue el razonamiento de uno de los párrafos anteriores los arboles de decisión de estas componentes conexas deberían estar separados pero dado que un plan no necesariamente debe cumplirse ya que puede cumplirse el padre estas deberían estar juntas. Podemos aprobar los planes azules y solo un plan rojo (el que no tiene un padre azul)

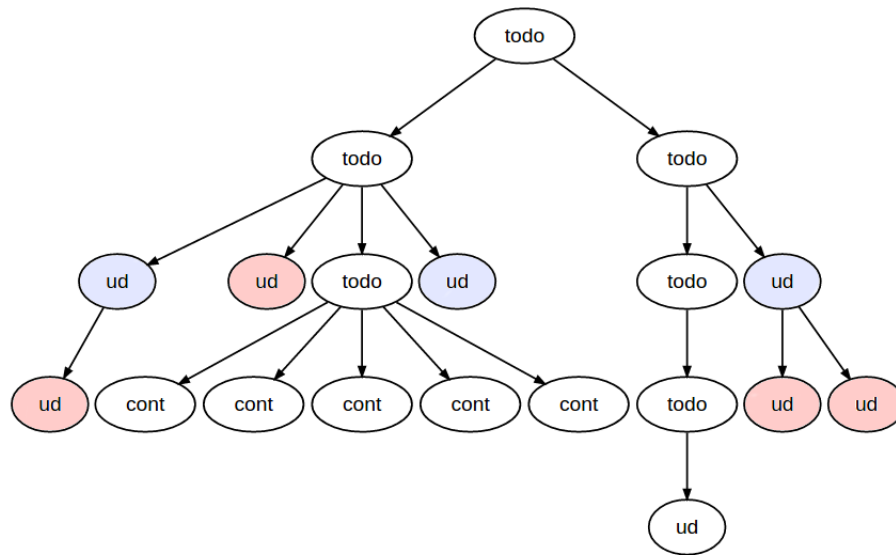


Figura 4.3:

y estaríamos cumpliendo todas las reglas pero si lo consideramos problemas independientes en la componente conexas roja se hará esfuerzo por completar los planes de nivel bajo aunque no sea necesario.

Teniendo todas estas cosas en consideración se lograría reducir el problema en varios problemas pequeños. Si anteriormente se tenían 3^{30} combinaciones, con esta mejora se lograría una reducción a problemas independientes y el calculo se haría con sumas, es decir, si se pudiera dividir en tres componentes conexas con diez ramos cada uno se tendría la adición $3^{10} + 3^{10} + 3^{10}$ lo que es igual a 3^{11} combinaciones.

Si se vuelve a ver la imagen donde esta el grafo con las componentes conexas, se puede ver que la componente con mayor cantidad de ramos posee aristas de color naranja y morado. Los ramos que están uniendo estos dos colores son tres (CC5315, CC5905 y CC5404) y si se toma inicialmente la decisión sobre ellos podrían formarse dos componentes conexas separadas. Es decir, decidir sobre un ramo podría formar nuevas componentes reduciendo aun más el espacio de soluciones. Aun así, el calculo de qué ramos tendrían mayor probabilidad de ocasionar esta separación y el ordenamiento según ello podría significar un coste mayor debido a que se estaría rompiendo el orden inicial de mayor valor (al igual que una de las reducciones que se pensó). Por lo tanto, se puede implementar el calculo recursivo de componentes conexas y evaluar los dos posibles ordenamientos para ver cual ocasiona una reducción mayor.

Finalmente, una mejora final considerada en esta memoria es el cache. [Explicar brevemente lo que es un cache]. Si bien con el calculo de componentes conexas se estaría eliminando gran parte de los elementos repetidos, también pueden existir cálculos repetidos por la formación de nuevas componentes conexas que ya se habían formado anteriormente.

Si se tuviera un ramo que al asignarse genera tres componentes conexas, el ramo podría haberse asignado dentro de esas tres opciones solamente. Si evaluamos paso a paso como se comportaría el algoritmo al asignar este ramo podremos evidenciar el calculo repetido.

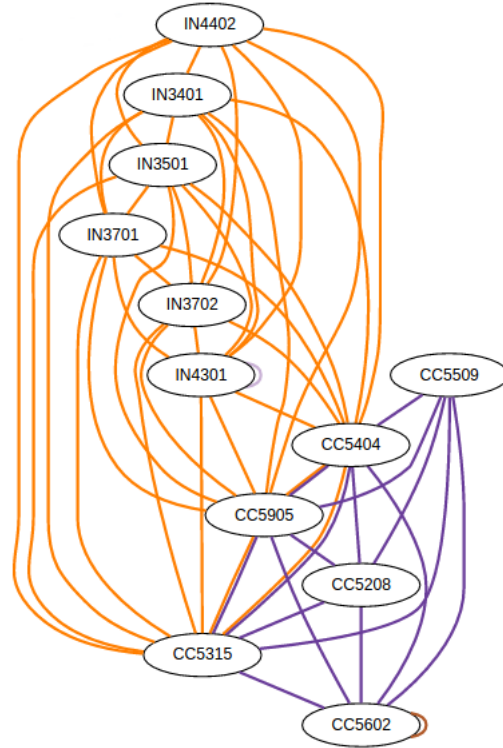


Figura 4.4:

Al inicio, el ramo será asignado en la primera componente calculándola considerando el ramo dentro de su solución. En cambio las otras dos componentes se calcularán sin considerar el ramo. Una vez hechos esos cálculos, el algoritmo debe devolverse y calcular que pasaría si el ramo fuera asignado en la segunda componente conexas. Allí, la segunda componente tendrá probablemente una solución distinta porque en esta ocasión debe considerar el ramo, la primera componente también deberá hacer su calculo nuevamente porque ahora no debe considerar el ramo (anteriormente si lo había hecho) pero la tercera componente volverá a calcularse sin considerar el ramo lo que podría guardarse dentro de un cache y no necesitar calcularla.

Lo anterior, debido al calculo del algoritmo, solo se produciría con ramos que generan mas de dos componentes conexas lo que es de baja probabilidad. Aun así se implementara dentro de esta memoria porque puede significar mejoras considerables si la componente conexas guardada fuese de un volumen grande.

Capítulo 5

Implementación

Para la implementación se utilizó como base lo hecho anteriormente. Lo anterior se dividía en cuatro funciones importantes destacando dos de ellas: la inicial y la recursiva. La función inicial realiza la petición de los ramos y planes a partir del rut y carrera de un alumno, limpia estos datos, realiza asociaciones como las relaciones entre ramos de distintas versiones, llama a la función recursiva y, finalmente, retorna los resultados. Por otra parte, la recursiva realiza todo el procesamiento de la heurística hablada llamándose a si misma por cada asignación de ramo hecha. Además, debe convocar a las otras dos funciones donde se propaga la elección de un ramo en un plan sobre todos los demás planes y donde se evalúa si la solución actual es mejor que la guardada. Esta ultima también posee el contador de tiempo donde si se exceden los quince segundos retorna la solución guardada por el momento y hace retornar los resultados desde la recursiva hacia la función inicial, para que esta los devuelva al controlador web que los solicitó.

En este caso se utilizó la misma forma de programación preocupándose que todas las funciones tuvieran una firma similar entre ellas respetando que siempre reciban la lista de ramos y planes, y retornen la mejor solución si corresponde. Además, como no estaba dentro del alcance se reutilizó la implementación hecha en la funcion inicial de la implementación anterior. Por lo tanto se poseen seis funciones donde tres de ellas se corresponden con las explicadas en el parrafo anterior exceptuando la recursiva debido a que se cambio el comportamiento del algoritmo.

Una de las funciones es el calculo de las componentes conexas donde a partir de los ramos se dividen en distintos arreglos debido a los planes que comparten. A esta función siempre se le entrega la lista de todos los ramos y planes pero puede adicionarse una tercera variable opcional que dice que subconjunto de esas listas deben ser considerados para el calculo y cuales elementos deben quedar fuera.

Otra función es el calculo de la mejor solución a partir de un arreglo de soluciones. En ella se recibe un arreglo y en una variable auxiliar se va guardando la mejor mientras se recorre el arreglo. Al termino retorna aquella variable.

La función recursiva de la implementación antigua se cambio por dos funciones nuevas. La

primera es el paso intermedio donde se calculan las componentes conexas y por cada una de ellas se inicia el calculo de las distintas opciones de un ramo. La segunda es la que realiza este calculo, elige un ramo y calcula todas sus opciones (llamando a la primera función) generando una arreglo de posibles soluciones el cual es pasado al método explicado en el párrafo anterior donde se calcula cual es mejor.

Finalmente se posee la función que realiza un recorrido con orden posterior (post order traversal) del árbol de planes para propagar las elecciones hechas en cada paso del segundo método explicado en el párrafo anterior.

Capítulo 6

Validación

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Definición 6.1 (ver [?]) *Definición definitiva*

$$\frac{d}{dx} \int_a^x f(y) dy = f(x).$$

Conclusión

Mauris ac ipsum. Duis ultrices erat ac felis. Donec dignissim luctus orci. Fusce pede odio, feugiat sit amet, aliquam eu, viverra eleifend, ipsum. Fusce arcu massa, posuere id, nonummy eu, pulvinar ut, wisi. Sed dui. Vestibulum nunc nisl, rutrum quis, pharetra eget, congue sed, dui. Donec justo neque, euismod eget, nonummy adipiscing, iaculis eu, leo. Duis lectus. Morbi pellentesque nonummy dui.

Aenean sem dolor, fermentum nec, gravida hendrerit, mattis eget, felis. Nullam non diam vitae mi lacinia consectetur. Fusce non massa eget quam luctus posuere. Aenean vulputate velit. Quisque et dolor. Donec ipsum tortor, rutrum quis, mollis eu, mollis a, pede. Donec nulla. Duis molestie. Duis lobortis commodo purus. Pellentesque vel quam. Ut congue congue risus. Sed ligula. Aenean dictum pede vitae felis. Donec sit amet nibh. Maecenas eu orci. Quisque gravida quam sed massa.

Nunc euismod, mauris luctus adipiscing pellentesque, augue ligula pellentesque lectus, vitae posuere purus velit a pede. Phasellus leo mi, egestas imperdiet, blandit non, sollicitudin pharetra, enim. Nullam faucibus tellus non enim. Sed egestas nunc eu eros. Nunc euismod venenatis urna. Phasellus ullamcorper. Vivamus varius est ac lorem. In id pede eleifend nibh consectetur faucibus. Phasellus accumsan euismod elit. Etiam vitae elit. Integer imperdiet nibh. Morbi imperdiet orci euismod mi.



Figura 6.1: Logo de la Facultad

Donec tincidunt tempor metus. Aenean egestas cursus nulla. Fusce ac metus at enim viverra lacinia. Vestibulum in magna non eros varius suscipit. Nullam cursus nibh. Mauris neque. In nunc quam, convallis vitae, posuere in, consequat sed, wisi. Phasellus bibendum consectetur massa. Curabitur quis urna. Pellentesque a justo.

In sit amet dui eget lacus rutrum accumsan. Phasellus ac metus sed massa varius auctor. Curabitur velit elit, pellentesque eget, molestie nec, congue at, pede. Maecenas quis tellus non lorem vulputate ornare. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam magna arcu, vulputate egestas, aliquet ut, facilisis ut, nisl.

Donec vulputate wisi ac dolor. Aliquam feugiat nibh id tellus. Morbi eget massa sit amet purus accumsan dictum. Aenean a lorem. Fusce semper porta sapien.

Campo 1	Campo 2
Valor 1	Valor2

Tabla 6.1: Tabla 1

Curabitur sit amet libero eget enim eleifend lacinia. Vivamus sagittis volutpat dui. Suspendisse potenti. Morbi a nibh eu augue fermentum posuere. Curabitur elit augue, porta quis, congue aliquam, rutrum non, massa. Integer mattis mollis ipsum. Sed tellus enim, mattis id, feugiat sed, eleifend in, elit. Phasellus non purus sed elit viverra rhoncus. Vestibulum id tellus vel sem imperdiet congue. Aenean in arcu. Nullam urna justo, imperdiet eget, volutpat vitae, semper eu, quam. Sed turpis dui, porttitor ut, egestas ac, condimentum non, wisi. Fusce iaculis turpis eget dui. Quisque pulvinar est pellentesque leo. Ut nulla elit, mattis vel, scelerisque vel, blandit ut, justo. Nulla feugiat risus in erat.

[5, 2, 3, 4, 1]

Bibliografía

- [1] Universidad de Chile. Reglamento de estudios de la facultad de ciencias físicas y matemáticas (plan 2007). <http://escuela.ing.uchile.cl/normas-y-reglamentos/reglamento-de-estudios-plan-2007>. [En línea].
- [2] José Miguel Garrido. Diseño y construcción de un sistema de apoyo a la verificación del cumplimiento del plan de estudio en la facultad de ciencias físicas y matemáticas. Master's thesis, Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Santiago, 2005.
- [3] Silvano Martello, Paolo Toth. *Knapsack Problems, Algorithms and Computer Implementations*. J. Wiley Sons, New York, 1990.
- [4] Silvano Martello, Paolo Toth. Lower bounds and reduction procedures for the bin packing problem. In *Discrete Applied Mathematics*, pages 59–70. Elsevier Science Publishers B.V., North Holland, 1990.
- [5] Nelson Valdivia. Diseño de un sistema inteligente de recuento de unidades docentes para la escuela de ingeniería y ciencias. Master's thesis, Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Santiago, 2003.