# UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

**Colegio de Ciencias e Ingenierías**

# Modelo de programación entera para la asignación de materias a las aulas de la USFQ

**Sistematización de experiencias prácticas de investigación e intervención**

.

**Jose Luis Alvear Calero**
**Carlos Iván Sandoval Vargas**

**Ingeniería Industrial**

Trabajo de titulación presentado como requisito
para la obtención del título de
Ingeniero Industrial

Quito, 19 de mayo de 2017

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO DE CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN**

**Modelo de programación entera para la asignación de materias a las aulas de la USFQ**

**Jose Luis Alvear Calero**

**Carlos Iván Sandoval Vargas**

Calificación:

Nombre del profesor, Título académico: Galo Eduardo Mosquera Recalde, MSc

Firma del profesor ______________________________

Quito, 19 de mayo de 2017

# Derechos de Autor



Firma del estudiante: ________________________________________

Nombres y apellidos: Jose Luis Alvear Calero

Código: 00111476

Cédula de Identidad: 1726307117

Firma del estudiante: ________________________________________

Nombres y apellidos: Carlos Iván Sandoval Vargas

Código: 00110890

Cédula de Identidad: 1719635318

Lugar y fecha: Quito, mayo de 2017

# RESUMEN

En la actualidad, la creación de horarios en la universidad es una tarea bastante compleja y que consume mucho tiempo. Se crea un modelo de programación entera para la asignación de horarios a las aulas de la USFQ, el cual es un problema que cae dentro de la categoría de NP-hard y NP-complete, lo cual significa que el tiempo de resolución crece exponencialmente conforme se incrementa el número de variables. El modelo permite asignar clases teóricas, laboratorios, ejercicios y clases de computación. Además, evita todo tipo de conflictos de horario entre clases, profesores, aulas, y entre cursos que deben ser tomados por los mismos estudiantes. También se consideran los requerimientos de cada clase en cuanto a infraestructura y equipamiento. Se tiene la opción de que los profesores indiquen las horas en las que prefieren dictar clases, y se asegura que los profesores tengan un horario balanceado. Para resolver el modelo se crea una heurística que permite superar las limitaciones computacionales producidas por la complejidad del problema. El modelo fue validado con datos de dos colegios de la USFQ permitiendo crear horarios exitosamente, sin ningún tipo de conflictos y con las características señaladas. Finalmente, con los datos obtenidos se verifica que el problema es del tipo NP-Hard por lo que su tiempo de resolución crece exponencialmente conforme aumenta el número de variables.

Palabras clave: programación entera, horarios universidad, NP-hard, heurística, AMPL

# ABSTRACT

Nowadays, the creation of the schedules is a hard and time-consuming task in the internal processes of the university. An integer programming model is used in order to solve the timetabling problem in USFQ. The model allows assigning lectures, laboratories, exercise classes, and computational laboratories. Moreover, it avoids time conflicts between subjects, professors, rooms and courses that have to be taken by the students. Furthermore, equipment and infrastructure requirements are also considered. In addition, each professor shows their schedule preferences, so they are assured to have a balanced agenda. To solve this model an heuristic is created in order to overcome the computational limitations that result of the problem complexity. The model was validated with the data of the School of Engineering and Business Administration School, allowing to solve the timetabling problem successfully. There were not conflict detected and all the specifications were achieved. Finally, it was verified that the problem solved was NP-Hard so the resolution time grows exponentially when the variables number increases.

*Key words*: Integer programming, timetable problem, ampl, NP-Hard, heuristic.

# TABLA DE CONTENIDO

# Modelo de programación entera para la asignación de materias a aulas en la Universidad San Francisco de Quito

Jose Luis Alvear, Carlos Sandoval, Galo Mosquera

Universidad San Francisco de Quito, Quito, Ecuador
jose.alvear@estud.usfq.edu.ec, carlos.sandoval.vargas@estud.usfq.edu.ec, gemosquera@usfq.edu.ec

En la actualidad, la creación de horarios en la universidad es una tarea bastante compleja y que consume mucho tiempo. Se crea un modelo de programación entera para la creación de horarios en la USFQ. El modelo permite asignar clases teóricas, laboratorios, ejercicios y clases de computación. Además, evita todo tipo de conflictos de horario entre clases, profesores, aulas, y entre cursos que deben ser tomados por los mismos estudiantes. También se consideran los requerimientos de cada clase en cuanto a infraestructura y equipamiento. Se tiene la opción de que los profesores indiquen las horas en las que prefieren dictar clases, y se asegura que los profesores tengan un horario balanceado. Para resolver el modelo se crea una heurística que permite superar las limitaciones computacionales producidas por la complejidad del problema. El modelo fue validado con datos de dos colegios de la USFQ permitiendo crear horarios exitosamente, sin ningún tipo de conflictos y con las características señaladas. Finalmente, con los datos obtenidos se verifica que el problema es del tipo NP-Hard por lo que su tiempo de resolución crece exponencialmente conforme aumenta el número de variables.

*Palabras Clave:* programación entera, horarios universidad, ampl, NP-Hard, heurística

## 1. Introducción:

La asignación de horarios y clases es una tarea muy importante en todas las instituciones educativas, pero a su vez es muy complicada debido a que se deben tomar en cuenta todas las restricciones operacionales de la organización y además las preferencias y demás limitaciones propias de una institución educativa. Lo más importante es que los horarios satisfagan a todos los involucrados y esto sólo se consigue incluyendo factores que se relacionan con las expectativas de los usuarios (Daskalaki et al., 2004, p.117).

La Universidad San Francisco de Quito (USFQ) cada semestre se enfrenta a la decisión sobre la asignación de aulas a los distintos horarios y dado el crecimiento de la universidad, esta tarea es cada vez más complicada. Según Galo Valencia, quien se desempeña como registrador de la universidad el proceso de asignación actual se realiza mediante un sistema de prueba-error utilizando un software denominado MS, el cual tiene como objetivo maximizar el número total de cursos asignados, sin considerar medidas de calidad para el modelo tales como utilización de las aulas .

Adicionalmente, cerca del 10% de cursos son asignados manualmente cada semestre porque el software actual no logra asignar algunos paralelos (Valencia, 2016). La USFQ sigue la filosofía de las artes liberales por lo que cada estudiante (de segundo semestre en adelante) elige los cursos que desee, en los horarios que desee, basando su elección en la malla curricular de su carrera (ver lista de carreras en Anexo 1), siempre y cuando las clases estén disponibles (abiertas) para el período deseado. Por ello es muy importante que cada clase sea asignada a un horario que no cree conflictos para el estudiante ni para los profesores. Como se mencionó anteriormente, la asignación de clases en la USFQ es actualmente un proceso manual que requiere arduo trabajo y cuyo resultado final no siempre satisface las expectativas de profesores y estudiantes (Valencia, 2016).

Según Diego Gabela, Director de Admisiones de la USFQ, el número de estudiantes en la USFQ crece cada año a una tasa de aproximadamente 3%, y siempre se busca llenar los cupos disponibles para cada carrera (Gabela, 2016). Este crecimiento hace que la cantidad de clases impartidas crezca tanto en número como en capacidad, lo cual genera complicaciones operacionales y por ende malestar en algunos estudiantes ya que se genera problemas como insuficiente número de pupitres, aulas de clase demasiado pequeñas para el número de estudiantes, falta de cupos, entre otros. Es por esto que la necesidad de mejorar y resolver este tipo de situaciones es una prioridad.

Por lo tanto, se propone encontrar soluciones a los problemas previamente señalados mediante un proceso sistemático de asignación de materias a aulas de clases, desarrollado en base de un modelo de optimización. Tomando en cuenta requerimientos, especificaciones y restricciones operacionales, con el afán de llevar cada clase al horario preciso de una manera eficiente.

# 2. Objetivo General:

Asignar las clases a las aulas de la universidad, mediante el uso de programación entera cumpliendo las restricciones operacionales y requerimientos de la Universidad San Francisco de Quito.

### 2.1 Objetivos específicos:

1. Determinar las restricciones a considerar al momento de asignar horarios y aulas.
2. Crear un modelo matemático para la asignación de horarios que maximice el uso de los recursos y cumpla con restricciones operacionales, requerimientos y especificaciones de las autoridades de la universidad.
3. Encontrar la solución óptima para el modelo matemático de ser posible, caso contrario buscar la solución con menores resultados infactibles.

### 2.2 Metas y actividades:

1. Determinar el número de materias que imparte la USFQ cada semestre.
2. Determinar los requerimientos de cada una de las materias sobre todo en infraestructura para entregar una solución que sea aplicable.
3. Identificar las materias que deben tener preferencias en cuestión de horarios y aulas.
4. Resolver el modelo de optimización mediante el lenguaje de modelado AMPL en busca de una solución que optimice el uso de recursos.

# 3. Revisión Literaria

El objetivo del problema de asignación de horarios de los cursos de la universidad (UCTTP por sus siglas en inglés) tiene como meta la asignación de eventos a un número predeterminado de horas y aulas disponibles, cumpliendo las restricciones operacionales establecidas (Babaei et al., 2015, p.43). Por esta razón son considerados problemas combinatorios ya que implican asignar un número discreto de eventos satisfaciendo ciertas restricciones. Los problemas combinatorios son de los más interesantes que existen debido a que la dificultad crece exponencialmente con el tamaño del problema

El problema de asignación de horarios y aulas es utilizado por las diferentes instituciones educacionales con el fin de optimizar los recursos limitados que poseen. Actualmente, el uso inadecuado de horarios y cursos conlleva a una pobre utilización de instalaciones y espacios. Los espacios a analizar son sumamente diversos y pueden incluir las siguientes instalaciones: aulas, teatros, auditorios, oficinas, entre otros. Para esto dichos espacios deben ser identificados con sus respectivos usos, tiempos en los cuales pueden ser utilizados y las diferentes restricciones pertinentes que ayuden a resolver el respectivo problema de asignación (Esraa & Ghada, 2016, p316)

Como se ha mencionado anteriormente, cada autor le da un tratamiento distinto al problema, de modo que se ajuste a las necesidades, requerimientos y sobre todo a la realidad de cada organización. Por ejemplo, de acuerdo con Méndez et al. (2016) el objetivo de los proyectos de este tipo en el ámbito académico puede ser de tres tipos: asignar horarios y aulas para examinaciones (Examination Timetabling - ETP), asignación de cursos en aulas y horas luego de que los estudiantes eligen los cursos (Post-enrollment based course timetabling PECTP), y asignación de eventos en horarios y aulas en base a las mallas curriculares (Curriculum based course timetabling CCTP).

La idea de utilizar métodos matemáticos para la asignación de horarios viene desde los años setenta (Al Husain et al., 2011, p.157). En los últimos años se ha incrementado notablemente la atención brindada a este problema, lo cual es notorio por ejemplo con la creación de conferencias bianuales denominadas Theory and practice of automated timetabling (PATAT), la creación de grupos de investigación sobre el tema como el EURO Working group for Automated Timetabling, y sobre todo por la gran cantidad de tesis de postgrado investigando al respecto (Al-Yakoob & Sherali, 2015, p.57). Adicionalmente, según Méndez et al. (2016) el crecimiento del interés en este tema se evidencia por la organización de las competiciones internacionales denominadas International Timetabling Competition, las cuales han sido llevadas a cabo en los años 2002, 2007 y 2011. Estos autores manifiestan que las competiciones han permitido que se hayan desarrollado numerosos métodos y algoritmos para resolver el problema.

Actualmente, en el medio ecuatoriano no se ha encontrado evidencia de ningún estudio relacionado a la implementación de asignación de horarios por medio de métodos de optimización aplicados a sectores educativos de educación superior.

### *3.1. Definiciones básicas*

Los problemas de asignación de horarios se pueden tratar desde diferentes puntos de vista, pero en general Babaei et al. (2015) consideran que los conceptos básicos que se identifican son los siguientes: *eventos* que son entidades o actividades que pueden ser asignadas tales como un curso, un profesor o un estudiante; *intervalos de tiempo* en que los eventos pueden ser asignados; los *recursos* a los cuales los eventos son asignados como aulas y horas; *personas* que van a ser asignadas tales como profesores y estudiantes; *restricciones* generalmente operacionales o de preferencia; y *conflictos* que deben evitarse tales como la asignación de un curso a dos aulas.

Las restricciones en estos problemas se clasifican en dos tipos: duras y flexibles. Las restricciones duras son aquellas que se deben cumplir obligatoriamente como por ejemplo: los profesores no pueden dar dos cursos al mismo tiempo, un curso no puede estar en dos aulas al mismo tiempo, cada curso debe ser asignado a una hora y aula específicos con un solo profesor determinado, etc. Las restricciones flexibles no se deben cumplir necesariamente, pero se relacionan con la función objetivo por lo que la meta es maximizar el número de restricciones satisfechas. Ejemplos de restricciones flexibles son: peticiones de profesores para ciertas clases y aulas en particular, ciertos cursos con prioridad para ser asignados a ciertas horas, utilización de la capacidad de las aulas, etc. (Babaei et al., 2015, p.44).

*3.2. Métodos de resolución utilizados*

El problema de asignación de horarios y aulas es bastante complejo desde el punto de vista matemático y computacional (Phillips et al., 2014, p.42). Por esta razón se encuentra dentro de la categoría de problemas de tipo NP-hard que significa que el tiempo de resolución crece exponencialmente cuando se incrementa la complejidad del modelo, es decir añadiendo más estudiantes, materias y detalles (Babaei et al., 2015, p.56). Daskalaki et al. (2004) también clasifican al problema como NP-complete lo cual significa que si todas las posibilidades son examinadas, el tiempo de resolución crece dramáticamente. Esta creciente complejidad ha hecho que gran parte de los autores hayan implementado el modelo solamente en departamentos o facultades y no en toda la universidad ya que el modelo se complica cuando se comparten instalaciones entre las distintas facultades (Babaei et al., 2015, p.44). De acuerdo con Burkle et al. (2008), para algunas formulaciones de problemas prácticos no se ha podido todavía obtener una solución óptima. Por esta razón, las investigaciones en este tema se han enfocado en desarrollo de diferentes algoritmos, métodos y heurísticas para resolver las dificultades matemáticas y computacionales y obtener soluciones óptimas o aproximadamente óptimas.

Existen diferentes acercamientos con respecto a las metodologías utilizadas que podemos encontrar en diferentes artículos de investigación científica con respecto a las metodologías de resolución para ETP. Carter (1986) nos habla de la aplicación de la teoría gráfica de los ETPs con un enfoque especializado en el método graph coloring. Schaerf (1999) primero realiza una diferenciación entre los diferentes tipos de problemas de asignación de horarios, problemas de asignación de horarios de cursos de universidad (UCTPs) y (ETPs), con esto muestra heurísticas directas, método de graph-coloring, simulated annealing, algoritmo genético, entre otros.

Una revisión más completa de los métodos de resolución se muestra en el Anexo 2.

*3.3. Indicadores de la calidad del modelo*

Cuando se crea un modelo para la asignación de horarios, es evidente que se debe obtener medidas del desempeño de dicho modelo para poder evaluarlo y sobre todo para compararlo con otros modelos utilizados. Existen varias métricas que se han creado para la evaluación de los modelos, algunas de las cuales incluso se integran dentro del modelo como restricciones flexibles. Phillips et al. (2014) proponen ciertas métricas las cuales permiten evaluar la asignación y pueden ser incluidas en la función objetivo de acuerdo a los intereses de la organización, las medidas propuestas son las siguientes: *Horas de los eventos (EH)*, que se refiere a maximizar el número de eventos asignados de entre todos eventos; *Horas sentadas de los estudiantes (SH*) que se refiere a maximizar el número cursos y por ende estudiantes asignados a un horario y aula, generalmente se utiliza cuando se quiere priorizar la asignación de cursos grandes; *Utilización de asientos (SH)* para maximizar la relación entre la cantidad de alumnos en un aula con la capacidad del aula; *Preferencia de aula (RP)* para maximizar el cumplimiento de las solicitudes especiales para ciertas aulas; *Estabilidad aula-curso,* para minimizar el número de aulas distintas asignadas a un mismo curso; y *Robustez de la asignación*, para minimizar el impacto en el modelo cuando cambia la cantidad de alumnos inscritos.

De la misma manera existen otras medidas de desempeño que nos permiten determinar que tan bueno es un modelo con respecto a otro. Otra medida de desempeño es la tasa de utilización del aula, el cual depende de otras dos tasas para su respectivo cálculo: la tasa de frecuencia y ocupación. La tasa de frecuencia refiere a que tanto el aula es utilizada con respecto a las horas disponibles que la misma puede ser utilizada y la tasa de ocupación mide que tanto el aula está ocupada con respecto a su capacidad máxima. La tasa de utilización estará dada por el producto de las medidas de desempeño mencionadas con anterioridad (Esraa & Ghada, 2016, p316).

### *3.4. Softwares utilizados*

Diferentes matemáticos han concordado que los problemas de clasificación de horarios de clases son un problema definido como NP-hard. Esta es una medida que muestra la dificultad computacional de resolución, la clase P son aquellos problemas que pueden ser resueltos en un tiempo polinomial, mientras que las clases NP son aquellas que tienen un tiempo polinomial no determinístico lo cual quiere decir que el tiempo en el cual el programa estará corriendo crecerá exponencialmente a medida que el tamaño del problema incrementa lo cual es un factor determinante en la dificultad computacional (Esraa & Ghada, 2016, p.315-316).

Woumans (2016) en su estudio muestra que para sus pruebas se utilizó un sistema con un AMD Phenom II 4x965 CPU, temporizado a 3.4 GigaHertz con 32 GigaBytes en RAM. El solver utilizado para dicho estudio fue un IBM ILOG CPLEX Optimization Studio. Es decir, que la complejidad computacional que representan dichos modelos y el tiempo polinomial no determinístico de los mismos abren la necesidad de utilizar sistemas de resolución computacional avanzados y de la misma manera el hardware necesario para correr un sistema de esta magnitud.

### *3.5. Uso de Programación Entera para este problema*

Varios autores han utilizado métodos de programación entera para tratar el problema de asignación de horarios en universidades, entre los trabajos que se pueden mencionar se encuentran los siguientes: Saldaña et al. (2007) en la Universidad de Concepción de Chile; Schimmelpfeng & Helber (2006) también utilizaron programación entera en Hannover University; Kumar R. (2014) quien presenta un modelo de programación entera en base a hojas de cálculo como modelo piloto para el Colegio de Negocios de California State University: Bakir & Aksop (2008) que utilizaron programación entera binaria para los horarios del Departamento de estadística de Gazi University; Al-Housain et al. (2011) propusieron un método de programación entera en 3 etapas para los horarios de un departamento en Kuwait University; Philliphs et al. (2014) quienes proponen un método que permite mantener la trazabilidad en caso de problemas grandes en base a sus experiencias en University of Auckland; entre otros autores que también se han enfocado en desarrollar y perfeccionar metodologías en base a programación entera.

Según Babei et al. (2015) es difícil obtener la solución óptima para el problema de asignación de horarios mediante programación entera por lo que muchos autores han combinado programación entera con otros algoritmos y heurísticas para facilitar la aproximación a las soluciones óptimas. El tamaño de la universidad influye directamente la la aplicación de metodologías basadas en programación entera.

Babei et al. (2015) realizaron un análisis y comparación de los métodos de resolución mencionados en la Tabla 1 utilizando datos proporcionados en las International Timetabling Competitions para de esta manera concluir objetivamente sobre el desempeño de los distintos métodos. Estos autores afirman que los métodos basados en programación entera no son eficientes, y comprueban que caen en la categoría de NP-hard (tiempo de solución crece exponencialmente). Sin embargo, afirman que si el tiempo de resolución y el espacio no son un problema, se puede utilizar estos métodos ya que su implementación es simple y funcionan bastante bien combinados con otros algoritmos.

Una estrategia para mejorar la eficiencia de estos modelos consiste en dividir el problema original en etapas, por ejemplo primero asignando horarios y luego asignando aulas. Esto hace que los modelos se reduzcan y que sean mucho más fáciles de resolver desde el punto de vista computacional y matemático. Por ejemplo, Al Husain et al. (2011) proponen una metodología basada en 3 etapas utilizando programación entera en donde el output de cada etapa es utilizado como input de la siguiente para de esta manera llegar a una solución global óptima. La primera etapa fue la asignación de cursos y profesores, la segunda etapa la asignación de horas para los cursos y finalmente se asignan las aulas para cada curso. Estos autores afirman que dividir el problema por partes tiene la ventaja de reducir considerablemente el tiempo de ejecución permitiendo llegar a la solución satisfactoria rápidamente.

### *3.6. Selección del método de resolución*

De entre los cuales la metodología que se procedió a escoger para el caso de asignación de horarios a aulas fue el de programación entera, esto se debe debido a que el modelo seleccionado satisface las necesidades del problema a resolver. Esto pese a que según Babei et al. (2015) es difícil obtener la solución óptima para el problema de asignación de horarios mediante programación entera, sobre todo en casos grandes ya que el tamaño de la universidad influye directamente la la aplicación de metodologías basadas en programación entera. Sin embargo, los mismos autores, luego de analizar todos los métodos presentados en el Anexo *2*, afirman que si el tiempo de resolución y el espacio no son un problema, se puede utilizar estos métodos ya que su implementación es simple y funcionan bastante bien combinados con otros algoritmos. Por lo tanto, el enfoque que hemos decidido utilizar será el de programación entera.

# 4. Metodología

Al igual que la mayor parte de problemas prácticos de optimización, el problema de asignación de aulas y horarios no es netamente un problema matemático, sino que requiere completar ciertas fases para su exitosa terminación. Hiller & Lieberman (2010) consideran que un estudio de investigación de operaciones tiene las siguientes fases:

*4.1. Definición del problema de interés y recolección de datos relevantes.*

*4.2 Formulación de un modelo matemático que represente al problema.*

*4.3 Obtención de soluciones a partir del modelo.*

*4.4 Pruebas, validación y mejoramiento del modelo*

*4.5 Preparación para la aplicación del modelo*

*4.6 Implementación de los resultados*

La descripción de las actividades que implica cada una de las fases de la metodología se muestran en el *Anexo 3*.

# 5. El modelo de programación entera para la asignación de aulas a las materias de la USFQ.

*Consideraciones:* La población objetivo de la investigación se puede ver sujeta a cambios. Pese a que el modelo debe funcionar con cualquier tamaño de población, debido al tiempo de resolución y dificultad computacional, inicialmente el modelo se validará con las clases del colegio de Ciencias e Ingeniería de la USFQ. Las medidas o indicadores de calidad del modelo, utilizadas como los coeficientes de la función objetivo pueden estar sujetos a cambios, para ajustarse a los intereses de las autoridades de la USFQ. Para ajustarse a la realidad de la universidad, este modelo no asignará los profesores a las distintas materias, ya que los encargados de realizar esta tarea son los coordinadores de los departamentos. Sin embargo, el modelo tiene la capacidad de realizar dicha asignación de ser necesario. Como ha sido mencionado con anterioridad el modelo será descrito como un modelo de programación entera con las siguientes características:

**Conjuntos:**

$\mathrm{IIN}_{100}, \mathrm{IME}_{100}, \mathrm{IEE}_{100}..$:Conjuntos de clases de primer año (por carrera)
$\mathrm{C}_1$:Conjunto de clases de primer año

$\mathrm{IIN}_{200}, \mathrm{IME}_{200}, \mathrm{IEE}_{200}..$:Conjuntos de clases de segundo año (por carrera)
$\mathrm{C}_2$:Conjunto de clases de segundo año

$IIN_{300}, IME_{300}, IEE_{300}..$: *Conjuntos de clases de tercer año (por carrera)*
$\mathrm{C}_3$:Conjunto de clases de tercer año

$\mathrm{IIN}_{400}, \mathrm{IME}_{400}, \mathrm{IEE}_{400}..$:Conjuntos de clases de cuarto año (por carrera)
$\mathrm{C}_4$:Conjunto de clases de cuarto año

$\mathrm{IIN}_{500}, \mathrm{IME}_{500}, \mathrm{IEE}_{500}..$:Conjuntos de clases de quinto año (por carrera)
$\mathrm{C}_5$:Conjunto de clases de quinto año

C:Conjunto de todas las clases
LABS: Conjunto de todas las clases de laboratorio
CMP: Conjunto de todas las clases de computación
EJERC: Conjunto de todas las clases de ejercicios

LU: Conjunto para el día Lunes
MA: Conjunto para el día Martes
MI: Conjunto para el día Miércoles
JU: Conjunto para el día Jueves
D: Conjunto de todos los días

$P_{lab}$: Conjunto de días para laboratorios
$P_{normal}$: Conjunto de periodos para clases normales
P: Conjunto de todos los períodos de horas
$P_1$, $P_2$, $P_3$, ... , $P_{12}$: Subconjuntos de períodos de acuerdo a la hora
AM: Conjunto de períodos de la mañana
PM: Conjunto de períodos de la tarde
AM: Conjunto de *períodos de la mañana*
Pe: Conjunto de períodos extremos de clases normales
Pl: Conjunto de períodos extremos de clases de laboratorio

$R_{normal}$: Conjunto de aulas normales disponibles
$R_{labs}$: Conjunto de laboratorios disponibles
R : Conjunto de todas las aulas disponibles

T: Conjunto de profesores disponibles

RT : Conjunto de tipos de laboratorios disponibles
RC : Conjuntos de posibles características de aulas (infraestructura)

**Parámetros:**

$capacidad_R$:Capacidad del aula A
$demanda_C$: Demanda de la clase C
$profesor_{c,t}$: Profesor de la clase C
$aula_{R, RC}$:Características de cada aula en infraestructura
$req\_clase_{C, RC}$:Requerimientos de clase en infraestructura
$TA_{RLABS,RT}$:Tipo de aula de laboratorio
$TC_{LABS,RT}$:Tipo de clase de laboratorio
$TCA_{CMP,RT}$:Tipo de clase de computación
$horas_{T,P}$:Períodos preferidos para clases de un profesor

**Variables:**

$x_{CRTPD}$=variable que vale 1 si la clase c, es asignada a la hora t, en el aula r, en el periodo p,en el dia d.

$aux_{PCRT}$=variable auxiliar para clases de 2 días

**Función Objetivo:**

*maximizar*

$$\frac{\sum_{D=1}^{d} \sum_{P=1}^{p} \sum_{C=1}^{c} \sum_{R=1}^{r} \sum_{T=1}^{t} x_{dpcrt} * \frac{demanda_R}{capacidad_R}}{2*n(NORMAL)+2*n(CMP)+n(LABS)+n(EJERC)} \quad (4)$$

**Restricciones:**

$$\sum_{c=1}^{C} \sum_{r=1}^{R} x_{d,p,c,r,t} \leq 1 \quad \forall\ p \in P, t \in T, d \in D \quad (5)$$

$$\sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} \leq 2 \times profesor_{c,t} \quad \forall\ c \in C, t \in T \quad (6)$$

$$\sum_{p=1}^{PM} \sum_{c=1}^{C} \sum_{r=1}^{R} x_{d,p,c,r,t} - \sum_{p=1}^{AM} \sum_{c=1}^{C} \sum_{r=1}^{R} x_{d,p,c,r,t} + 1 \geq 0 \quad \forall\ t \in T, d \in D \quad (7)$$

$$\sum_{p=1}^{PM} \sum_{c=1}^{C} \sum_{r=1}^{R} x_{d,p,c,r,t} - \sum_{p=1}^{AM} \sum_{c=1}^{C} \sum_{r=1}^{R} x_{d,p,c,r,t} \leq 0 \quad \forall\ t \in T, d \in D \quad (8)$$

$$\sum_{p=1}^{Pe} \sum_{r=1}^{R} \sum_{c=1}^{C} x_{d,p,c,r,t} \leq 1 \quad \forall\ d \in D, t \in T \quad (9)$$

$$\sum_{p=1}^{Pl} \sum_{r=1}^{R} \sum_{c=1}^{C} x_{d,p,c,r,t} \leq 1 \quad \forall\ d \in D, t \in T \quad (10)$$

$$\sum_{c=1}^{C} \sum_{r=1}^{R} x_{dpcrt} \leq horas_{t,p} \quad \forall\ t \in T, p \in P, d \in D \quad (11)$$

$$x_{d,p,c,r,t} \times demanda_{c} \leq capacidad_{r} \quad (12)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} \times aula_{r,rc} \geq 2 \times req\_clase_{c,rc} \quad \forall\ c \in CNORMAL, rc \in RC \quad (13)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} \times aula_{r,rc} \geq req\_clase_{c,rc} \quad \forall\ c \in EJERC, rc \in RC \quad (15)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} \times aula_{\ r,rc} \geq 2 \times req\_clase_{c,rc} \quad \forall\, c \in CMP, rc \in RC \quad (16)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} \times aula_{\ r,rc} \geq req\_clase_{c,rc} \quad \forall\, c \in LABS, rc \in RC \quad (14)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{Plabs} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in CNORMAL \quad (17)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{Pnormal} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in LABS \quad (18)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{Pnormal} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in EJERC \quad (19)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{Plabs} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in CMP \quad (20)$$

$$\sum_{t=1}^{T} \sum_{r=1}^{Rlabs} \sum_{d=1}^{D} \sum_{p=1}^{P} x_{d,p,c,r,t} * tipo\_aula_{(r,rt)} = tipo\_clase_{(c,rt)} \quad \forall\, c \in LABS, rt \in RT \quad (21)$$

$$\sum_{t=1}^{T} \sum_{r=1}^{Rlab} \sum_{d=1}^{D} \sum_{p=1}^{P} x_{d,p,c,r,t} * tipo\_aula_{(r,rt)} = 2 * tipo\_clase_{(c,rt)} \quad \forall\, c \in CMP, rt \in RT \quad (22)$$

$$\sum_{t=1}^{T} \sum_{c=1}^{C} x_{d,p,c,r,t} \leq 1 \quad \forall\, p \in P, r \in R, d \in D \quad (23)$$

$$\sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{t=1}^{T} x_{d,p,c,r,t} \leq 1 \quad \forall\, d \in D, c \in C \quad (24)$$

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \sum_{c=1}^{IIN100} x_{d,p,c,r,t} \leq 1 \quad \forall\, d \in D, p \in P \quad (25)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{Plabs} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in CNORMAL \quad (26)$$

$$\sum_{t=1}^{T} \sum_{p=1}^{Pnormal} \sum_{r=1}^{R} \sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in LABS \quad (27)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{Plabs}\sum_{r=1}^{R}\sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in EJERC \quad (28)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{Plabs}\sum_{r=1}^{R}\sum_{d=1}^{D} x_{d,p,c,r,t} = 0 \quad \forall\, c \in CMP \quad (29)$$

$$\sum_{p=1}^{P1}\sum_{r=1}^{R}\sum_{c=1}^{C} x_{d,p,c,r,t} \leq 1 \quad \forall\, t \in t, d \in D \quad (30)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P1}\sum_{r=1}^{R}\sum_{c=1}^{IIN100} x_{d,p,c,r,t} \leq 1 \qquad \forall\, d \in D \quad (31)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{D} x_{d,p,c,r,t} \leq 2 \quad \forall\, c \in CNORMAL \quad (32)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{D} x_{d,p,c,r,t} \leq 2 \quad \forall\, c \in CMP \quad (33)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{D} x_{d,p,c,r,t} \leq 1 \quad \forall\, c \in LABS \quad (34)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{D} x_{d,p,c,r,t} \leq 1 \quad \forall\, c \in EJERC \quad (35)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{LU} x_{d,p,c,r,t} + \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{MA} x_{d,p,c,r,t} \leq 1 \quad \forall\, c \in CNORMAL \quad (36)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{LU} x_{d,p,c,r,t} + \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{JU} x_{d,p,c,r,t} \leq 1 \quad \forall\, c \in CNORMAL \quad (37)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{MA} x_{d,p,c,r,t} + \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{MI} x_{d,p,c,r,t} \leq 1 \quad \forall\, c \in CNORMAL \quad (38)$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{MI} x_{d,p,c,r,t} + \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{d=1}^{JU} x_{d,p,c,r,t} \leq 1 \quad \forall\, c \in CNORMAL \quad (39)$$

$$\sum_{d=1}^{D} x_{d,p,c,r,t} = 2 \times aux_{cprt} \quad \forall\, c \in CNORMAL, r \in R, t \in T, p \in P \quad (40)$$

**Conjuntos:**

Los conjuntos son uno de los pilares del modelo presentado. Las clases que se quieren asignar se declaran en varios conjuntos separados por año y por departamento o carrera. Esta división permite que las clases de un mismo año y carrera no se crucen, ya que se asume que todas las clases declaradas en un mismo conjunto tienen que ser tomadas por el mismo grupo de estudiantes. Esta división sistemática también puede ser utilizada para evitar que clases de distintos departamentos no se crucen. Luego de declarar todos los conjuntos de clases, mediante operaciones de conjuntos se crea el conjunto *C*, que representa el conjunto de todas las clases a asignarse. Dentro del conjunto *C* se crean los subconjuntos *LABS, CMP,* y *EJER*. Dentro de estos subconjuntos se deben declarar las clases que son laboratorios, clases de ejercicios y que son clases normales pero que deben dictarse en laboratorios de computación. Nuevamente, mediante operaciones entre conjuntos utilizando el comando `symdiff`, el cual obtiene la diferencia entre la unión y la intersección de dos conjuntos (IBM, 2010).
Los días también son declarados en conjuntos, se crean cuatro conjuntos, uno para cada día de la semana y mediante la operación de unión se forma el conjunto *D* que contiene todos los días. Los períodos de clase se declaran en dos conjuntos: un conjunto de períodos de una hora y media (declarados con h minúscula), y otro conjunto con períodos de dos horas (declarados con H mayúscula). Mediante unión se crea también el conjunto *P* de todos los períodos. Adicionalmente se crean 12 subconjuntos de períodos, en los que se declaran los períodos que se encuentran en el mismo horizonte de tiempo. Por ejemplo, el subconjunto $P_1$ contiene los períodos 7H00, 7h00 y 8h30. El resto de subconjuntos funcionan de la misma manera y sirven para evitar conflictos entre clases que duran una hora y clases de dos horas. Los subconjuntos de horas permiten enlazar las clases que duran una hora y media con las que duran dos horas para evitar conflictos durante la asignación.
Las aulas también son declaradas en conjuntos separados. Inicialmente se declaran todas las aulas en el conjunto *R*, y luego se declara el subconjunto *Rlab* de aulas de laboratorio. Mediante diferencia de conjuntos se obtiene el conjunto *Rnormal* que contiene el resto de aulas, las cuales son categorizadas como normales. Adicionalmente se crean dos conjuntos auxiliares; el conjunto RC que contiene los diferentes tipos de laboratorio que pueden existir y el conjunto RT que contiene las opciones de equipamiento que puede tener un aula tales como infocus, accesibilidad, aire acondicionado, etc.

**Parámetros:**

Los parámetros de los que se alimenta este modelo son varios, pero algunos de ellos pueden adquirir valores por default para hacer más rápida la declaración de datos. Se necesita saber la *capacidad* aulas y la *demanda* de cada clase. Ambos parámetros deben ser determinados con mucha certeza pues el modelo basa su asignación en dicha característica.
Este modelo tiene la capacidad de asignar profesores aleatoriamente, solo cumpliendo las restricciones impuestas para cada uno. Sin embargo, este tipo de asignación estaría bastante alejada de la realidad de la USFQ, en donde los coordinadores de cada departamento son quienes asignan una clase determinada a un profesor específico. Es por ello que se ha creado un parámetro *profesor* el cual especifica el profesor (o los profesores) que pueden dictar cada clase.

Otros parámetros que se declaran tienen relación con las aulas. El parámetro *aula* sirve para caracterizar a cada aula de acuerdo a su infraestructura o a otras características declaradas en el conjunto *RC*. De la misma manera, el parámetro *req_clase* sirve para declarar los requerimientos de infraestructura que debería tener cada clase. También se tienen parámetros que caracterizan a los laboratorios, especificando el tipo de laboratorio que puede llevarse a cabo en cada aula. Esta función es realizada por el parámetro *tipo_aula*, en el cual se debe especificar si se trata de un aula para laboratorios de física, química, biología, etc. Por otra parte, el parámetro *tipo_clase* especifica el tipo de laboratorio que una clase necesita. Para las clases de dos días a la semana y que deben ser dictadas en aulas de computación se ha especificado el parámetro *tipo_clasecmp*. Finalmente, se ha creado el parámetro horas, el cual especifica las horas en las que un profesor está dispuesto a dictar sus clases. Este parámetro hace que el modelo para la asignación de horarios de la USFQ se apegue completamente a la realidad pues en la actualidad muchos profesores no están dispuestos a dictar una clase a cualquier hora por lo que es importante considerar su opinión. Para declarar fácilmente este parámetro es recomendarlo establecer que su valor predeterminado sea 1, y declarar con valor 0 las horas en las que ciertos profesores no están dispuestos a dar clases.

**Función Objetivo:**

En la revisión de literatura se encontró que en su mayoría se utilizan indicadores de la calidad del modelo para su optimización, tales como los propuestos por Phillips et al. (2014) o por Esraa & Ghada (2016). La función objetivo depende de cada autor, o de los requerimientos de la universidad, pero en este caso se desea maximizar la ocupación de las diferentes aulas de la universidad. Por ello, se crea un indicador que mide el porcentaje de utilización de las aulas. El indicador se muestra en la ecuación (4) y básicamente relaciona la cantidad de asientos requeridos por cada clase con la capacidad del aula en que se asignó. Luego calcula el promedio de dichas relaciones para obtener una medida global de la utilización de las aulas de la universidad. El denominador de la función objetivo es el número total de asignaciones realizadas, valor que se calcula determinando la cardinalidad de los distintos conjuntos de clases.

**Restricciones:**

Con el fin de acoplar el modelo a la realidad de la Universidad San Francisco de Quito, se crearon 29 restricciones que pueden ser clasificadas en 5 grupos de acuerdo a su función:

- ❏ Restricciones de profesores
- ❏ Restricciones de aulas
- ❏ Restricciones de cursos
- ❏ Restricciones de periodos
- ❏ Restricciones de días

Dentro de las restricciones de profesores se encuentran las ecuaciones desde la (5) hasta la (11). La ecuación (5) permite que un mismo profesor solo pueda dar una clase a la vez en cada período y día, por su parte la ecuación (6) hace que cada profesor dicte sólo las clases que previamente le fueron asignadas por el coordinador. Dentro de la normativa de la universidad se sugiere que cuando los profesores tengan dos o más clases en un mismo día, se deberá tener al menos una clase en la mañana y una en la tarde, lo cual se controla por medio de las ecuaciones (7) y (8). De la misma manera, los profesores no pueden tener clases en 2 periodos extremos durante un mismo día lo cual es regulado por las ecuaciones (9) para profesores de clases normales y mediante la ecuación (10) para profesores de laboratorio. Finalmente, la ecuación (11) genera horarios en los cuales no se les puede asignar a los profesores según sea su preferencia.

Dentro de las restricciones de aulas se encuentran desde las ecuaciones (12) hasta la (22). Para controlar que no se exceda la capacidad de un aula en una asignación se utiliza la restricción detallada en la ecuación (12). Por otra parte, se crean restricciones para que cada clase asignada cumpla con los requerimientos de infraestructura que se solicitó mediante el parámetro RC. La ecuación (13) hace que las aulas a las que se asignaron las clases normales cumplan con sus requerimientos. De la misma manera, la ecuación (14) hace que los laboratorios cumplan con los requerimientos, la ecuación (15) hace lo mismo con los ejercicios y la ecuación (16) para las clases de computación. Las restricciones de la (17) a la (20) sirven para asignar un tipo clase a su tipo de aula, y funcionan haciendo que la suma de todas las clases asignadas en el conjunto de aulas a las que no deben ser asignadas (laboratorios) sea igual a 0. La ecuación (17) hace que las clases normales sean asignadas a aulas que se encuentran en el conjunto de "normales". Igualmente, la ecuación (18) hace que los laboratorios sean asignados a aulas de laboratorio, la ecuación (19) hace que las clases ejercicio sean asignadas a aulas normales y la ecuación (20) hace que las clases de computación sean asignadas a aulas de computación. Adicionalmente se ha creado la restricción (21) que hace que los laboratorios sean asignados a aulas de laboratorio al tipo de laboratorio del que se trata la clase, como por ejemplo que un laboratorio de física sea asignado a un aula para laboratorios de física. La restricción (22) hace lo mismo pero solamente con clases que deben funcionar en un laboratorio de computación.

Dentro de las restricciones de cursos se encuentran las ecuaciones (23), (24 y (25), donde la ecuación (23) controla que en cada día, periodo y aula solo sea asignado un curso. Complementariamente la ecuación (24) permite que un mismo curso solo sea dictado una vez por día. Adicionalmente, la ecuación (25) evita que cursos pertenecientes a un mismo grupo sean asignados el mismo día a la misma hora. Esto con el fin de permitir que los alumnos cursando un cierto año de una carrera puedan tomar todas las clases que indica su malla curricular, dicha ecuación se repetirá por cada año y por cada carrera. La restricción (25) evita conflictos entre todas las clases declaradas en un conjunto, por ende dicha restricción se debe repetir n veces dependiendo del número de conjuntos en que se están declarando las clases. Se debe tomar en cuenta que el número máximo de clases declaradas en cada conjunto depende de la cantidad de aulas y períodos que se establezcan. Se debe controlar que el número de clases declaradas no sobrepase la cantidad de períodos disponibles con todas las aulas pues esto podría llevar al software a declarar el modelo como infactible. Debe notarse también que las restricciones del tipo (25) no son necesarias si existen muchos paralelos para las materias de un mismo año y carrera, ya que los conflictos que puedan darse no impedirán que los estudiantes tomen alguna clase.

Dentro de las restricciones de periodos se encuentran las ecuaciones de la (26) a la (31). Las ecuaciones (26) a (29) especifican la duración de las horas de clase, ya sea 1h30 para las clases normales y de computación y de 2h00 para los laboratorios y ejercicios. Las ecuación (26) asigna una duración de 1h30 a las clases normales, clases de ejercicios (28) y laboratorios de computación (29). La restricción (27) por su parte, hace que la duración de las clases de laboratorio sea de 2 horas. Debido a la diferencia de periodos entre laboratorios y clases normales mencionada anteriormente se crearon los conjuntos $P_1$ a $P_{12}$, los cuales agrupan las horas de laboratorio con las horas normales para cada horizonte de tiempo. Por esta razón se añade la ecuación (30), la cual impide que un mismo profesor tenga conflictos de horas si dicta clases normales y laboratorios. La ecuación (30) debe ser declarada para todos los conjuntos de horas desde $P_1$ hasta $P_{12}$. De la misma manera se debe evitar que las clases de laboratorio y de ejercicios que fueron inicialmente declaradas en un mismo conjunto se crucen. Esto se consigue mediante restricciones del tipo (31), las cuales permiten que máximo 1 clase de cada grupo sea asignada en cada horizonte de tiempo. Restricciones como la (31) deben ser creadas para cada grupo en el que se pueda tener conflictos entre laboratorios y ejercicios y para los 12 conjuntos de tiempo creados.

Dentro de las restricciones de días se encuentran las ecuaciones (32) a (37). La asignación de cuántas veces por semana se dicta una clase está controlada por las ecuaciones (32) y (35), las cuales asignan a clases normales y clases de computación dos veces por semana mientras que los ejercicios y laboratorios se asignan una vez por semana. Por otro lado las ecuación (36) a (39) controla que cuando se asignen una clase dos veces por semana esta sea solamente lunes y miércoles o martes y jueves. Por último, la ecuación (40) controla que cuando se asignen una clase dos veces por semana esta sea asignada los dos días a la misma hora, en la misma aula y con el mismo profesor. Para esto la restricción (40) hace uso de una variable binaria auxiliar.

# 6. Análisis de la solución

El modelo planteado en la sección anterior, se implementó en AMPL y se resolvió mediante el solver Gurobi. Como se ha mencionado en varias ocasiones, este tipo de problemas son del tipo NP-hard y NP-complete (Babaei et al., 2015, p.56), lo cual hace que el número de variables, la dificultad computacional y por ende el tiempo de resolución crezcan exponencialmente conforme aumenta el número de cursos a ser asignados. Durante la resolución de este modelo, las dificultades mencionadas en la literatura pudieron ser experimentadas. El modelo funciona perfectamente cuando el tamaño del problema es pequeño, es decir cuando se tienen que asignar aproximadamente 50 cursos. Sin embargo, cuando crece el tamaño del problema, la dificultad computacional disponible no es suficiente. Por ejemplo, inicialmente se resolvió para el departamento de Ingeniería Mecánica, que con 43 clases normales, 15 laboratorios, 14 profesores, y 5 aulas representan un total de 335161 variables y 305252 restricciones.
Es evidente que resolver directamente el modelo con todos los datos de la USFQ o del Colegio de Ciencias e Ingeniería es una tarea demasiado complicada para computadores comunes pues el número de variables y restricciones tendería al infinito. No fue posible resolver el modelo para asignar 80 cursos al mismo tiempo, pues luego de varios minutos de resolución se obtiene un mensaje de AMPL afirmando que se ha utilizado demasiada memoria del computador. Para resolver el modelo se utilizó un computador con RAM de 8 GB.

Por esta razón, se decidió crear un método de resolución tipo heurística que divide el problema global en pequeñas partes. Inicialmente se declara el modelo, luego se resuelve el modelo para el primer departamento, se borra el archivo de datos (por ende se libera memoria), se vuelve a declarar otro archivo de datos para otro departamento y así sucesivamente hasta asignar los cursos de todos los departamentos que se están tomando en cuenta. El esquema de la heurística se muestra en la *Figura 1*.

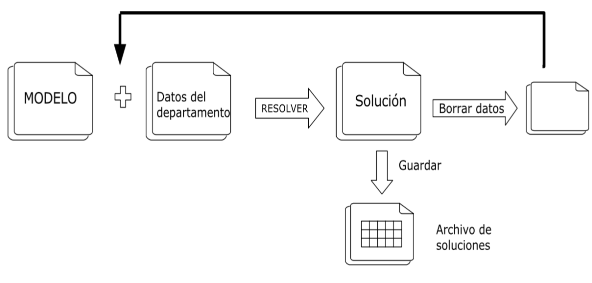


*Figura 1:* Método de resolución

El inconveniente que trae el método de resolución presentado es que si no se controlan las aulas que se declaran en cada archivo de datos, pueden crear conflictos y cruces de horarios. Para evitar esto, en cada archivo de datos se debe declarar un grupo de aulas que no deben estar en otros archivos de datos. Esta heurística hace que la solución encontrada no sea la óptima global, sino que se encuentran óptimos locales cada vez que se resuelve el modelo para cada departamento. Sin embargo, las soluciones encontradas son totalmente válidas y aplicables a la realidad de la USFQ.

El modelo descrito anteriormente se validó utilizando los datos del segundo semestre 2016-2017 sobre las clases de todos los departamentos del Colegio de Ciencias e Ingeniería y del Colegio de Administración y Economía de la USFQ. En total se utilizaron 18 departamentos con un total de 684 cursos asignados que representan aproximadamente la tercera parte de todos los cursos que se asignan en un período de la USFQ.

El modelo funciona exitosamente con los datos ingresados y permite crear horarios sin ningún tipo de conflictos tanto para profesores, aulas, y en los departamentos. Ejemplos de horarios creados por el modelo se pueden encontrar en el *Anexo 5*.

En la *Tabla 1*, se tabulan los resultados de la resolución del modelo para los distintos departamentos mediante la heurística mencionada.

| ***Archivo*** | ***FO*** | ***Variables*** | ***Restricci.*** | ***Normal*** | ***Labs*** | ***Ejer.*** | ***CMP*** | ***Tiempo*** | ***Cursos*** |
|---|---|---|---|---|---|---|---|---|---|
| *QUI* | *76,82%* | *558605* | *481258* | *13* | *23* | *21* | *0* | *0,411* | *57* |
| *FIS* | *72,76%* | *859955* | *770261* | *27* | *38* | *0* | *0* | *0,43* | *65* |
| *CMP-ING* | *93,43%* | *676205* | *687164* | *2* | *0* | *0* | *67* | *2,33* | *69* |
| *AG_AL_IA* | *61,00%* | *399005* | *384737* | *35* | *14* | *1* | *0* | *2,93* | *50* |
| *IIN-INQ* | *67,86%* | *333905* | *315027* | *31* | *17* | *5* | *0* | *13,65* | *53* |
| *ICV-IEE* | *71,39%* | *739205* | *703197* | *41* | *16* | *3* | *0* | *4,32* | *60* |
| *IME* | *57,61%* | *335165* | *306280* | *25* | *28* | *4* | *0* | *1,45* | *57* |
| *MAT-A* | *73,51%* | *635045* | *646741* | *56* | *0* | *0* | *0* | *7,82* | *56* |
| *MAT-B* | *76,54%* | *478805* | *488910* | *57* | *0* | *0* | *0* | *1,99* | *57* |
| *ADM* | *68,04%* | *827755* | *843051* | *55* | *0* | *0* | *0* | *61,86* | *55* |
| *ECN-FIN* | *59,88%* | *828805* | *843165* | *64* | *0* | *0* | *0* | *86,69* | *64* |
| *CON-MKT-SEG* | *65,70%* | *717505* | *733536* | *41* | *0* | *0* | *0* | *16,36* | *41* |
| ***Promedio*** | ***70,38%*** | ***7389960*** | ***7203327*** | ***447*** | ***136*** | ***34*** | ***67*** | ***Total*** | ***684*** |

***Tabla 1:*** *Resultados obtenidos para dos colegios de la USFQ*

Como se puede observar en la Tabla 1, el modelo desarrollado consigue una utilización bastante alta de las aulas de la universidad. Cada archivo de datos (departamento) se corrió con un promedio de 60 clases, lo cual no crea problemas con la capacidad computacional disponible. Se observa también que algunos departamentos como IIN-INQ fueron corridos en el mismo archivo de datos, pues su número de clases no es muy elevado. Por otro lado, los departamentos muy grandes como Matemáticas, fueron divididos en dos partes para poder resolverlos. Analizando los resultados, se puede notar que en promedio se tiene un 70% de utilización de las aulas de la universidad lo cual es bastante bueno dadas las fluctuaciones que experimenta la demanda por las clases durante las primeras semanas de cada semestre. La mejor utilización se consiguió para las clases de CMP, dado que en este departamento la cantidad de cupos que se abren es limitada de acuerdo al tamaño de las aulas. La peor utilización se consiguió con Ingeniería Mecánica, lo cual puede ser explicado por la cantidad de alumnos registrados en cada laboratorio. Sin embargo, la utilización está determinada por las aulas que sean asignadas a cada archivo de datos el cual es un proceso manual y que puede producir sesgos en el valor de la función objetivo. Pese a esto, el objetivo final de modelo es crear un horario para las clases, por lo que el valor de la función objetivo en este caso es simplemente una medida del desempeño de la asignación. Cuando se analizan los datos de la *Tabla 1*, también se pueden extraer conclusiones importantes que permiten verificar lo encontrado en la literatura acerca de este tipo de problemas. En la *Figura 2*, se observa la gráfica entre el número de cursos y el número de variables y restricciones. Se observa la forma en la que crece el problema de acuerdo al número de cursos que se van a asignar. Se esperaría que la relación sea lineal, es decir que a mayor cantidad de cursos, más grande sea el problema. Se puede observar que los datos efectivamente siguen esta tendencia, pero que no es una relación perfectamente lineal. Esto se puede explicar por la diversidad de cada departamento, pues en algunos casos se tienen clases de laboratorios y ejercicios y en otros solamente se tienen clases teórica. Esto significa que no solamente el número de cursos determina el tamaño del problema; sino también la estructura del modelo planteado y el tipo de clases en el departamento son determinantes del tamaño del problema y por ende en su complejidad de resolución.

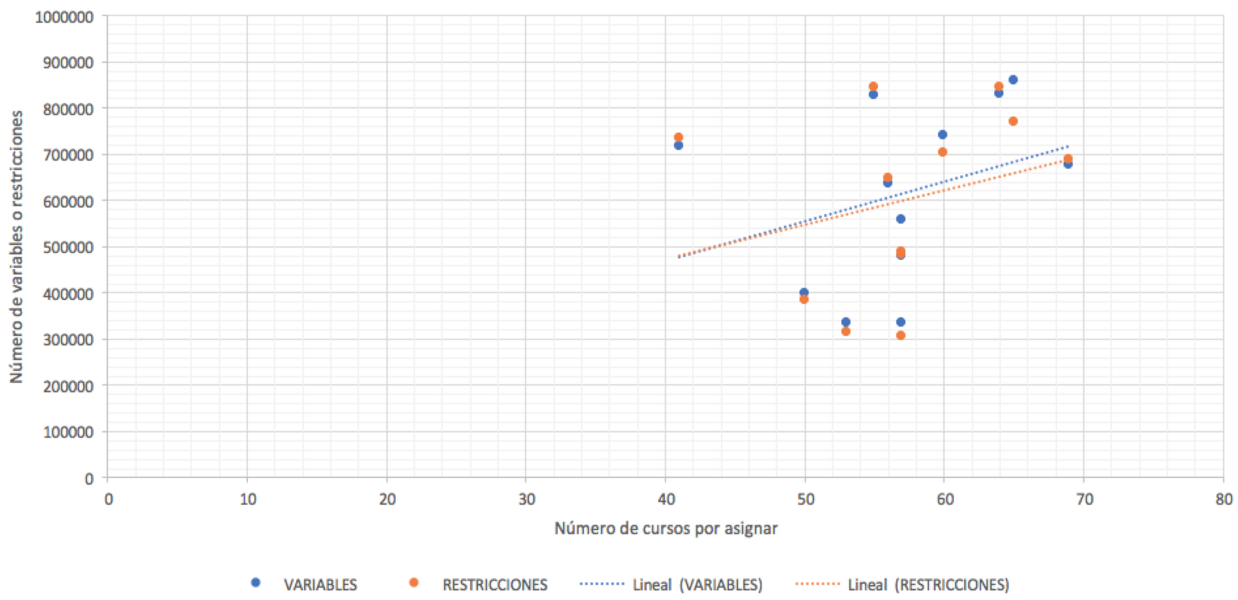


***Figura 2:*** *Cursos a asignar vs. Tamaño del problema*

También se va a analizar la relación que existe entre el tamaño del problema y el tiempo de resolución. Para ello se creó la *Figura 3*:

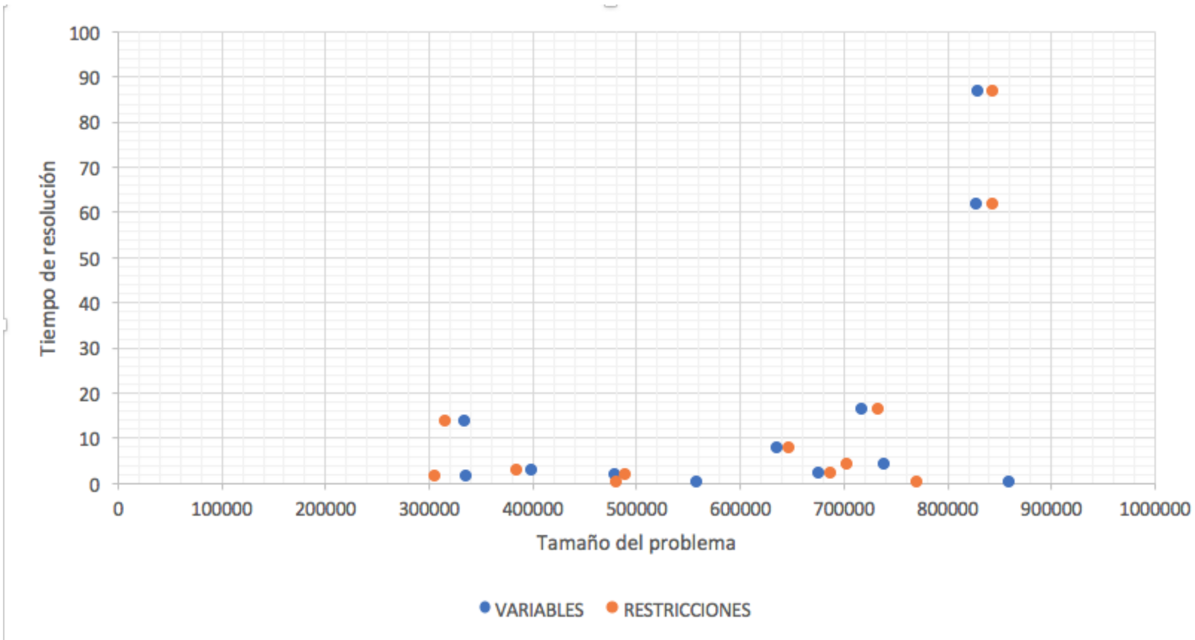


***Figura 3:*** *Tamaño del problema vs. Tiempo de resolución*

En la *Figura 3*, se puede observar la relación que existe entre el tiempo y el tamaño del problema. Se observa que el crecimiento del tiempo de resolución sigue una tendencia exponencial, tal y como se esperaba al tratarse de un problema del tipo NP-Hard y NP-complete. Por ende, dividir la resolución por departamentos fue útil para no superar las restricciones computacionales y para disminuir el tiempo de resolución drásticamente.

# 7. Conclusiones y recomendaciones

Existen diferentes organismos internacionales y concursos a nivel global que están constantemente buscando métodos de resolución para los problemas de asignación de horarios debido a la complejidad de los mismos. Dicha complejidad se da por la gran cantidad de restricciones y variables a considerar, por lo que matemáticos alrededor del mundo han coincidido en que estos problemas deben ser clasificados como NP-hard. Debido a esto existen diferentes metodologías aplicables tanto puras como mixtas que se ajustan dependiendo del modelo. El enfoque que se utilizó para crear el modelo fue programación entera. El modelo creado permite asignar tanto clases teóricas, así como clases de laboratorio, ejercicios y de computación. Evita conflictos de horario entre clases, profesores, aulas, horas y entre cursos que deben ser tomados por los mismos estudiantes. También se consideran los requerimientos de cada clase en cuanto a infraestructura y equipamiento requerido por las clases. Se tiene también la opción de que los profesores señalan las horas en las que prefieren dictar clases, y se asegura que las clases asignadas a un profesor estén en horario balanceado. Para resolver el modelo se crea una heurística que permite superar tanto las limitaciones computacionales, así como la complejidad del problema. El modelo fue validado con datos de dos colegios de la USFQ permitiendo crear horarios exitosamente sin ningún tipo de conflictos y con las características señaladas. Finalmente, con los datos obtenidos se verifica que el problema es en efecto del tipo NP-Hard y que su tiempo de resolución crece exponencialmente con el tamaño del problema.

Para futuras investigaciones se recomienda utilizar este modelo cuando se disponga de mayor capacidad computacional pues constituye el mayor limitante del modelo. También se recomienda trabajar con otras heurísticas que permitan encontrar una solución más cercana a la óptima global. Finalmente, se recomienda realizar investigaciones variando la estructura del modelo, para de esta manera reducir la complejidad y tamaño del problema.

# 8. Referencias:


1. Al Husain et al. (2011
2. ) A Sequential Three-Stage Integer Goal Programming (IGP) Model for Faculty-Course-Time-Classroom Assignments,35, 157-164.
3. Al-Yakoob, S. M., & Sherali, H. D. (2015). Mathematical models and algorithms for a high school timetabling problem. Computers & Operations Research, 61, 56–68. doi:10.1016/j.cor.2015.02.011
4. Bakir M. & Aksop C. (2008). A 0-1 INTEGER PROGRAMMING. APPROACH TO A UNIVERSITY. TIMETABLING PROBLEM, 37, 41-55
5. Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. Computers & Industrial Engineering, 86, 43–59. doi:10.1016/j.cie.2014.11.010
6. Burke EK, Mareček J, Parkes AJ, Rudová H. Uses and abuses of MIP in course timetabling, Poster at the workshop on mixed integer programming. MIP2007, Montréal; 2008. Available online at 〈http://cs.nott.ac.uk/jxm/timetabling/ mip2007-poster.pdf.
7. Birbas, T., Daskalaki, S., & Housos, E.. (2004). An integer programming formulation for a case study in university timetabling. European Journal of Operational Research, 153, 117-135.
8. Carter, M. W. (1986). A survey of practical applications of examination timetabling algorithms. Operational Research, 34(10.1287/opre.34.2.193), 193–202.
9. Dimopoulou, M., & Miliotis, P. (2001). Implementation of a university course and examination timetabling system. European Journal of Operational Research, 130, 202–213. doi:10.1016/S0377-2217(00)00052-7
10. Epelman (2012) Introduction to integer programming. University of Michigan. Tomado el 14 de noviembre de 2016 http://www-personal.umich.edu/~mepelman/teaching/IP/Handouts/Handout1.pdf
11. Esraa, A. & Ghada, A. (2016). An information visibility-based university timetabling for efficient use of learning spaces (IVUT). Egyptian Informatics Journal, 17, 315-325
12. Fonseca, G. Santos, H. Carrano, E. (2016). Integrating math heuristics and metaheuristics for timetabling. Computers & Operations Research, 74, 108-117.
13. Hiller F. & Lieberman (2010). Introducción a Investigación de Operaciones. Novena edición. México: McGraw Hill Education
14. IBM (2010) Knowledge center: symdiff. Extraído el 10 de enero de 2017 desde https://www.ibm.com/support/knowledgecenter/en/SS6MYV_3.4.0/ilog.odms.ide.odme.help/Content/Optimization/Documentation/ODME/_pubskel/ODME_pubskels/startall_ODME34_Eclipse1424.html
15. Kumar R. (2014) MODELING A DEPARTMENT COURSE SCHEDULING PROBLEM USING INTEGER PROGRAMMING: A SPREADSHEET-BASED APPROACH. Academy of Information and Management Sciences Journal, 17, 41-53
16. Laporte, G., & Desroches, S. (1984). Examination timetabling by computer. Computers & Operations Research, 11, 351–360. doi:10.1016/0305-0548(84)90036-4
17. Méndez-Díaz, I., Zabala, P., & Miranda-Bront, J. J. (2016). An ILP based heuristic for a generalization of the post-enrollment course timetabling problem. Computers & Operations Research, 76, 195–207. doi:10.1016/j.cor.2016.06.018
18. MirHassani & F. Habibi (2013) Solution approaches to the course timetabling problem. Artif Intell Rev, 39,133–149

19. Parkland College (sf) Project Management Knowledge areas. Tomado el 11 de diciembre de 2016 de http://www2.parkland.edu/businesstraining/documents/KnowledgeAreas.pdf
20. Phillips, A. E., Waterer, H., Ehrgott, M., & Ryan, D. M. (2015). Integer programming methods for large-scale practical classroom assignment problems. Computers & Operations Research, 53, 42–53. doi:10.1016/j.cor.2014.07.012
21. Saldaña et al. (2007) Modelos de programación entera para un problema de programación de horarios en universidades. Revista chilena de ingeniería, 3, 245-259
22. Schaerf, A. (1999). A survey of automated timetabling. Artificial Intelligence Review, 13, 87–127. doi:10.1023/A:1006576209967.
23. Schimmelpfeng K. & Helber S. (2007) Application of a real-world university-course timetabling model solved by integer programming. OR Spectrum, 29, 783–803
24. Woumans, G., De Boeck, L., Belien, J., Creemers, S. (2016). A column generation approach for solving the examination-timetabling problem. European Journal of Operational Research, 253, 178-194.

# 9. Anexos

**Anexo 1:** Colegios y carreras que ofrece la USFQ. Tomado de www.usfq.edu.ec

| COLEGIOS | CARRERAS |
|---|---|
| **CADE** | ADMINISTRACIÓN |
| | ECONOMÍA |
| | MARKETING |
| | FINANZAS |
| **CADI** | ARQUITECTURA |
| | DISEÑO INTERIOR |
| **CHAT** | ADMINISTRACIÓN DE EMPRESAS DE HOSPITALIDAD |
| | ARTE CULINARIO Y ALIMENTOS Y BEBIDAS |
| | ARTE CULINARIO |
| **COM** | MÚSICA CONTEMPORÁNEA |
| | PRODUCCIÓN MUSICAL Y SONIDO |
| **COCIBA** | INGENIERÍA EN PROCESOS BIOTECNOLÓGICOS |
| | LICENCIATURA EN BIOLOGÍA |
| **COCSA** | LICENCIATURA EN NUTRICIÓN HUMANA |
| | MEDICINA VETERINARIA |
| | MEDICINA |
| | ODONTOLOGÍA |
| **COCISOH** | ARTES LIBERALES |
| | EDUCACIÓN |
| | SICOLOGÍA |
| | RELACIONES INTERNACIONALES |

| | |
|---|---|
| **COCOA** | ANIMACIÓN DIGITAL |
| | CINE Y VIDEO |
| | COMUNICACIÓN AMBIENTAL |
| | COMUNICACIÓN PUBLICITARIA |
| | DISEÑO COMUNICACIONAL |
| | PERIODISMO MULTIMEDIOS |
| | INTERACTIVIDAD Y MULTIMEDIA |
| | COM. ORGANIZACIONAL Y RELACIONES PÚBLICAS |
| | ARTES CONTEMPORÁNEA S |
| **JURISPRUDENCIA** | JURISPRUDENCIA |
| **POLITÉCNICO** | INGENIERÍA DE AGROEMPRESA |
| | INGENIERÍA DE ALIMENTOS |
| | INGENIERÍA ELECTRÓNICA |
| | LICENCIATURA EN MATEMÁTICAS |
| | INGENIERÍA AMBIENTAL |
| | INGENIERÍA CIVIL |
| | INGENIERÍA INDUSTRIAL |
| | FÍSICA |
| | INGENIERÍA MECÁNICA |
| | INGENIERÍA QUÍMICA |

| | |
|---|---|
| | INGENIERÍA DE SISTEMAS |

**Anexo 2**: Métodos utilizados para la resolución de este problema

| **Método utilizado** | **Descripción** |
|---|---|
| Programación entera | Cuando a las variables utilizadas sólo pueden adquirir valores enteros y por lo general son variables binarias. |
| Constraint-satisfaction programming | El objetivo es encontrar un conjunto de valores que puedan ser asignados a las variables para satisfacer las restricciones. |
| Graph Coloring | Se busca colorear una red utilizando varios colores donde los vértices adyacentes no repiten colores. |
| Algoritmos genéticos | Siguen etapas definidas: creación de población, mutación, identificación de padres, etc. |
| Optimización de colonización de hormiga | Busca asimilar el comportamiento de las hormigas para encontrar la ruta más corta a la solución. |
| Algoritmos meméticos | El espacio de soluciones posibles es reducido para optimización local de subespacios. |
| Búsqueda Local | El algoritmo se mueve de solución en solución hasta encontrar la solución buscada o hasta que el tiempo pasa. |
| Búsqueda de variable en el vecindario | Se busca aplicar un criterio de aceptación de Monte Carlo para mejorar las exploraciones a través de aceptar las mejores soluciones con una probabilidad determinada. |
| Simulated Annealing | Se repite la búsqueda de soluciones locales hasta que se satisface una condición. |
| Búsqueda del Tabú | A partir de un resultado inicial, el algoritmo busca mejores soluciones en el vecindario de dicha solución. |
| Algoritmos híbridos (meta heurísticas) | Consiste en dividir el problema en fases y utilizar algoritmos diferentes para resolver cada fase y mejorar las soluciones previas. |
| Método fuzzy | Busca separar a los estudiantes en grupos para reducir el número de conflictos de horarios |

| | |
|---|---|
| Sistemas multi-agente | Basan sus métodos en el hecho de que un agente puede recibir información para desenvolverse en el ambiente. |

**Tabla 2:** Métodos y algoritmos utilizados para la solución del problema de asignación de horarios. Fuente: (Babaei et al., 2015, pp.45-52)

## **Anexo 3:** La metodología de Hiller & Lieberman

***Fases de la metodología para la resolución del problema***

### ***5.1. Definición del problema de interés y recolección de datos relevantes.***

Esta fase consiste en la correcta formulación y definición del problema y de los objetivos del estudio de investigación de operaciones. No se pueden obtener soluciones correctas si el problema está mal formulado. En este punto es también importante identificar a todos los involucrados en el problema, tanto las personas que serán afectadas por las decisiones, como los encargados de tomar la decisión final. Al inicio, es común que no existan datos para la formulación del modelo, por lo que los equipos suelen pasar mucho tiempo recolectando información necesaria para entender, plantear y resolver el problema (Hiller & Lieberman, 2010, p.8). Esta es la etapa en la que actualmente nos encontramos.

### ***5.2 Formulación de un modelo matemático que represente al problema.***

Una vez que se ha definido y ha entendido el problema, el siguiente paso consiste en poner el problema en una forma en que se pueda obtener su solución. En investigación de operaciones, esto se realiza formulando un modelo matemático que represente el problema que se quiere resolver. Los modelos matemáticos son representaciones idealizadas de la realidad, que se expresan mediante símbolos y ecuaciones matemáticas. Como se deben tomar decisiones cuantificables, entonces el modelo contiene variables de decisión para las cuales se determinará su valor. La medida de desempeño para el modelo se llama función objetivo y las limitaciones del modelo se denominan restricciones.. Las constantes que acompañan a las variables se denominan parámetros (Hiller & Lieberman, 2010, p.7).

### ***5.3 Obtención de soluciones a partir del modelo.***

En esta etapa se desarrolla un procedimiento generalmente en computadora para obtener las soluciones al modelo planteado. Esta no es ni la parte principal del proyecto ni la más compleja, ya que generalmente consiste en utilizar paquetes de software disponibles. La verdadera complejidad se da en las etapas anteriores y posteriores. El objetivo aquí es obtener los valores de las variables que dan el valor óptimo de la función objetivo. Sin embargo, hay problemas donde esto nos siempre es posible, por lo que la tarea consiste en encontrar soluciones aproximadamente óptimas (Hiller & Lieberman, 2010, p.11).

### ***5.4 Pruebas, validación y mejoramiento del modelo***

Los modelos matemáticos, sobre todo los modelos grandes contienen una gran cantidad de errores, muchos de los cuales nunca se llegan a detectar. Por eso es importante realizar la validación del modelo varias veces, para poder concluir que los resultados que está arrojando son confiables. No existe un procedimiento estándar para realizar la validación del modelo ya que depende en gran parte del tipo de modelo que se utiliza. Uno de los procedimientos recomendados, es comparar el desempeño del modelo con los datos históricos del sistema (Hiller & Lieberman, 2010, p.15).

### *5.5 Preparación para la aplicación del modelo*

Una vez que el modelo ha sido validado, se debe crear un sistema para su posterior aplicación, en donde se debe incorporar el modelo y su respectivo procedimiento de solución. Esto permite que el modelo sea utilizado aún cuando sus creadores no están presentes en la organización. Por lo general involucra una combinación de varias plataformas de software, y una tarea clave es la creación de una interfaz que permita al modelo interactuar con el usuario (Hiller & Lieberman, 2010, p.15).

### *5.6 Implementación de los resultados*

La última etapa en este tipo de proyectos consiste en la implementación real de los resultados del modelo obtenido en los pasos anteriores. Esta es una etapa crucial ya que es donde realmente se tiene que notar los beneficios de la investigación de operaciones. Es importante que el equipo que desarrolló el modelo supervise su implementación, ya que son quienes verdaderamente conocen los cambios que se deben hacer en el sistema (Hiller & Lieberman, 2010, p.16).

**Anexo 4:** Ingreso de datos en el archivo .dat

Dentro del archivo .dat se encuentran 15 conglomerados de datos los cuales se describen a continuación con sus respectivas características para su ingreso de datos.
**Definición de conjuntos:** Esta parte del archivo no se debe modificar ya que es la definición de los parámetros y conjuntos del modelo matemático

**Clases por año:** Este conglomerado de datos está subdividido en 5 subgrupos, uno por cada año. Dentro de cada subgrupo se añadirá las materias que se dictan por cada año, lo cual tiene la finalidad que las materias en cada subgrupo no tengan cruces de horario entre sí. Por lo cual, si se desea ingresar un grupo de datos que se quiera evitar cruces de horario entre sí deben ser ingresados en el mismo subgrupo.

**Set CMP:** En este subconjunto se deben ingresar las clases que se dicten dos días a la semana y se encuentran en aulas de laboratorio de computación.

**Set LABS:** En este subconjunto se enlistan las clases que son laboratorios.

**Set EJERC:** En este subconjunto se enlistan las clases que son ejercicios

**Set Rnormal:** En este subconjunto se especifican las aulas que son normales, es decir sin ninguna especificación especial de laboratorios.

**Set Rlabs:** En este subconjunto se ingresarán los datos de las aulas cuya especificación sea correspondiente a laboratorios.

**Aula y requerimiento de clase:** Son parámetros que únicamente deben ser cambiados se se necesita algún requerimiento especial para el aula como infocus o accesibilidad, se encuentran por default descritas todas las aulas como accesibles y con infocus.

**Tipo de clase:** Aquí se especifica que tipo de clase es, dicha clasificación es requerida únicamente para aulas de laboratorio en la cual se detalla qué tipo de laboratorio se necesita para dicha clase.

**Tipo de clase CMP:** Sirve para especificar los laboratorios de computación es decir clases dictadas dos días a la semana en un aula especificada como laboratorio de computación.

**Lista de profesores:** Se indica los profesores que se tomarán en cuenta para la corrida del modelo, se debe tomar en cuenta que no deben repetirse nombres sin importar el número de clases que de el profesor.

**Demanda:** Refleja cuántos estudiantes serán inscritos por cada clase dictada.

**Profesores:** Se ingresa que profesor dictará cada clase que se va a asign

**Anexo 5:** Ejemplos de los horarios creados por el modelo (Se puede observar que no existen conflictos de ningún tipo).

HORARIO POR DEPARTAMENTO

| | Dia | | | |
|---|---|---|---|---|
| Periodo | L | M | I | J |
| 7h00 | EFICIENCIA OPERACIONAL Y TECNOLOGIAS DE INFORMACION | DISENO DE EXPERIMENTOS<br>DISENO DE PLANTAS INDUSTRIALES | EFICIENCIA OPERACIONAL Y TECNOLOGIAS DE INFORMACION | DISENO DE EXPERIMENTOS<br>DISENO DE PLANTAS INDUSTRIALES |
| 8h30 | Gerencia de Servicios | | Gerencia de Servicios | |
| 10h00 | INGENIERIA ECONOMICA<br>CALIDAD TOTAL Y SEIS SIGMA | INGENIERIA ECONOMICA Cadena de Suministro | INGENIERIA ECONOMICA<br>CALIDAD TOTAL Y SEIS SIGMA | INGENIERIA ECONOMICA Cadena de Suministro |
| 11H00 | | Lab Control Estadist Calidad | | Lab DiseNo Experimentos |
| 11h30 | | Introduccion a Ing. Industrial Logistica | | Introduccion a Ing. Industrial Logistica |
| 13h00 | | CONTROL ESTADISTICO DE CALIDAD | | CONTROL ESTADISTICO DE CALIDAD |
| 14h30 | | Ingenieria de Factores Humanos | | Ingenieria de Factores Humanos |
| 15H00 | Lab Control Estadist Calidad | | Lab DiseNo Experimentos | |
| 16h00 | Desarrollo Tr. Titulacion IIN | Tendencias en IIN 1 Preparacion Tr. Titulacion IIN | Desarrollo Tr. Titulacion IIN | Tendencias en IIN 1 Preparacion Tr. Titulacion IIN |
| 17h30 | SISTEMAS DE MANUFACTURA | | SISTEMAS DE MANUFACTURA | |

## HORARIO POR PROFESOR

| Periodo | L | M | I | J |
|---|---|---|---|---|
| 7h00 | Galo Eduardo Mosquera Recalde (P)<br>N-211<br>Investigacion de Operaciones I | Galo Eduardo Mosquera Recalde (P)<br>N-211<br>DISENO DE PLANTAS INDUSTRIALES | Galo Eduardo Mosquera Recalde (P)<br>N-211<br>Investigacion de Operaciones I | Galo Eduardo Mosquera Recalde (P)<br>N-211<br>DISENO DE PLANTAS INDUSTRIALES |
| 13h00 | Galo Eduardo Mosquera Recalde (P)<br>N-211<br>Investigacion de Operaciones I | Galo Eduardo Mosquera Recalde (P)<br>M-326<br>CONTROL ESTADISTICO DE CALIDAD | Galo Eduardo Mosquera Recalde (P)<br>N-211<br>Investigacion de Operaciones I | Galo Eduardo Mosquera Recalde (P)<br>M-326<br>CONTROL ESTADISTICO DE CALIDAD |

## HORARIO POR AULA

| Periodo | L | M | I | J |
|---|---|---|---|---|
| 7h00 | Ingenieria Geologica y Geotecnica<br>12 | Vias de Comunicacion y Transporte II<br>12 | Ingenieria Geologica y Geotecnica<br>12 | Vias de Comunicacion y Transporte II<br>12 |
| 8h30 | Estructuras de Acero<br>15 | Hormigon Pretensado<br>12 | Estructuras de Acero<br>15 | Hormigon Pretensado<br>12 |
| 10h00 | Preparacion Tr. Titulacion ICV<br>15 | Reinforced Concrete II<br>15 | Preparacion Tr. Titulacion ICV<br>15 | Reinforced Concrete II<br>15 |
| 11h30 | Comunicaciones I<br>14 | Analisis Estructural II<br>15 | Comunicaciones I<br>14 | Analisis Estructural II<br>15 |
| 13h00 | Desarrollo Tr. Titulacion ICV<br>11 | Vias de Comunicacion y Transporte I<br>10 | Desarrollo Tr. Titulacion ICV<br>11 | Vias de Comunicacion y Transporte I<br>10 |
| 14h30 | Control Digital<br>10 | DiseNo de Puentes<br>11 | Control Digital<br>10 | DiseNo de Puentes<br>11 |
| 16h00 | Vias de Comunicacion y Transporte I<br>13 | | Vias de Comunicacion y Transporte I<br>13 | |
| 17h30 | Ingenieria de Costos<br>11 | Sistemas SCADA<br>16 | Ingenieria de Costos<br>11 | Sistemas SCADA<br>16 |