

## Segundo Proyecto

**Ingeniería en Informática**

**Teoría de la Computación**

---

**Integrantes del grupo:**

- Federico Becoña
- Lorena Ferrando
- Rodrigo López
- Valentina Palladino
- Julio Suaya

**Docentes:**

- Bernardo Rychtenberg

<b>Introducción</b>	<b>2</b>
<b>Estado del arte</b>	<b>3</b>
Soluciones académicas:	3
Soluciones comerciales	4
<b>Posibles enfoques</b>	<b>4</b>
<b>NP completo</b>	<b>4</b>
<b>Solución restringida</b>	<b>5</b>
Posible mejora	5
<b>Conclusión</b>	<b>6</b>
<b>Bibliografía</b>	<b>7</b>

# Introducción

Se ha planteado por parte de la cátedra el problema de armado de los horarios (Timetable problem). Este problema involucra una serie de restricciones duras (puntos a cumplir si o si para obtener una malla de horarios coherente) y algunas restricciones blandas (extras).

El problema requiere tener en cuenta las materias a dictarse en el semestre, los docentes que la dictan y la carga horaria correspondiente. También se tiene en cuenta el cupo de salones de la universidad, entre otras restricciones a considerar.

Este trabajo tiene el objetivo de investigar el problema mencionado e intentar buscarle una solución.

La investigación requerirá hacer el estudio del estado del arte. Esto implica analizar qué hace el mundo, o qué intentó hacer, para resolver este problema. Se deberá ver qué soluciones existen, tanto a nivel académico como a nivel comercial.

Con respecto a nuestra solución, se decidirá si restringir el problema y además que posible enfoque tomar.

# Estado del arte

A nivel mundial hay muchos esfuerzos para encontrar formas de automatizar el proceso de armado de horarios. Tanta relevancia tiene que existen competencias a nivel mundial, hay empresas que venden soluciones de armado de horarios.

## Soluciones académicas:

A nivel académico, se han realizado diferentes concursos, investigaciones y desafíos en función de resolver esta problemática planteada.

### **Algoritmos genéticos**

Pacheco, en el año 2000 utilizó una técnica heurística llamada **Algoritmos Genéticos** para optimizar los recursos (humanos y materiales) de un centro de estudios. La organización de horarios consiste en asignar de forma óptima tiempos y lugares a las diferentes materias impartidas en una Universidad, para cumplir con algunas restricciones, como el cupo limitado de salones, de espacios entre un curso y otro, etc.

La restricción de mayor importancia es que no haya colisiones entre los cursos, es decir, que dos o más materias que deben recibir los estudiantes o dictar los profesores no sean asignados a la vez. Luego de describir el algoritmo se procedió a aplicarlo, pero quedó una tarea pendiente en torno al número de materias dictadas en referencia a los espacios disponibles (horarios), por lo que este asunto debió resolverse de otra manera, en la coordinación de programa y mediante la asignación manual de horarios.

### **Algoritmo de búsqueda Tabú**

Se ha demostrado que la búsqueda tabú es eficaz para resolver problemas de horarios de exámenes. Los resultados empíricos también indican que se pueden producir mejores calendarios cuando se diferencian las restricciones duras y blandas. Este trabajo, cuando se pone en práctica, puede ayudar a las universidades a minimizar drásticamente el tiempo dedicado a preparar manualmente el calendario de exámenes, lo que generalmente requiere varios días de trabajo con la solución final todavía a veces insatisfactoria, ya que es un problema difícil de NP. El trabajo futuro incluye ampliar los datos de nuestros experimentos a toda la universidad y formular todas las restricciones posibles. También será interesante comparar el rendimiento de la búsqueda tabú con otras técnicas de vanguardia para la presentación de exámenes universitarios.

## Soluciones comerciales

Existen algunas soluciones comerciales como por ejemplo:

- Peñalara (<https://www.penalara.com/>): es una compañía española con más de 20 años de experiencia, siendo un referente como software de generación de horarios para centros de enseñanza a nivel mundial. Con respecto a su tecnología, aseguran que la fiabilidad y calidad de su producto, su mejora y actualización constante se reflejan en los resultados obtenidos. Desconocemos sus algoritmos utilizados.

## Posibles enfoques

Existen diferentes tipos de enfoques y técnicas para encontrar un resultado que cumpla con los requisitos. Dentro de las técnicas tradicionales, se encuentran los métodos completos recorren todo el espacio de búsqueda de un problema. En este grupo de métodos se encuentran: **programación entera, programación lineal, backtracking**, entre otros.

Las técnicas no tradicionales, no encuentran todas las soluciones posibles, sino que acotan el espacio de búsqueda, por lo cual se les llama métodos incompletos. Algunos ejemplos: **algoritmos evolutivos, búsqueda tabú, algoritmo voraces, redes neuronales, recocido simulado**.

Procederemos a detallar algunos de ellos.

El **backtracking** es una técnica algorítmica para resolver problemas de forma recursiva al tratar de construir una solución de forma incremental, una pieza a la vez, eliminando aquellas soluciones que no satisfacen las limitaciones del problema en cualquier momento (por tiempo, aquí, se hace referencia a el tiempo transcurrido hasta alcanzar cualquier nivel del árbol de búsqueda).

Por ejemplo, el algoritmo genético es un tipo de algoritmo evolutivo.

Los **algoritmos evolutivos** son métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. Todos los Algoritmos Evolutivos operan esencialmente manteniendo una población de soluciones potenciales y de alguna manera 'evolucionando' artificialmente esa población a lo largo del tiempo. Por ejemplo, el **algoritmo genético** es un tipo de algoritmo evolutivo.

## NP completo

Se le llama NP completo al conjunto de problemas para los que aún no se ha hallado una solución en tiempo polinómico. Pueden existir algoritmos para chequear si una determinada solución a un problema NP-completo es correcta en tiempo polinomial(en P). Se considera que este problema es NP-completo porque se puede verificar una solución en P, pero un algoritmo para generar una solución crece de exponencial a medida que agregamos restricciones o aumentamos la cantidad de datos.

# Solución restringida

Para modelar el problema del armado de los horarios de facultad, se consideraron ciertas restricciones (tanto restricciones duras como blandas).

A continuación se detallan las restricciones duras tenidas en cuenta:

1. Las materias a dictarse en el semestre (por año de carrera).
2. Las materias tienen una carga horaria correspondiente.
3. Las materias tienen un único docente.
4. Un docente puede dictar más de una materia (pero no en el mismo horario).
5. Un docente tiene sus restricciones de horario.
6. En el semestre la carga horaria de las materias debe sumar la correspondiente.
7. Cada Universidad define módulos de  $x$  duración.
8. Materias del mismo semestre no pueden quedar solapadas.
9. La Universidad tiene un cupo de salones.
10. No se contempla la capacidad de los salones.
11. Un salón en un horario dado, tiene una sola materia.

Restricciones blandas:

1. Evitado de clase los días sábados.
2. Evitado de horas puente para los alumnos.

Las restricciones blandas se manejaron de forma tal, que el armado de horario de cada materia comienza siempre con la primera hora disponible (de acuerdo a la disponibilidad horaria del profesor y las horas libres de los salones). De esta forma, al agendarse una materia esta cumple con ser el primer horario disponible que cumple los requisitos (hasta el momento).

Respecto a la solución presentada, cabe aclarar que los horarios se modelaron como una lista, entendiendo como la posición 0 de la misma corresponde al primer módulo del lunes, las posiciones 1 y 2 corresponden, respectivamente, con el segundo y tercer módulo del lunes, las posiciones 3, 4 y 5 con los módulos ordenados del martes y así sucesivamente hasta el sábado inclusive.

## Posible mejora

Dada la forma en la que nuestro algoritmo resuelve este problema, hay ciertas consideraciones que habría que tener a la hora de utilizarlo. Consideramos que para mejorar este algoritmo habría que hacer un pre-procesamiento de las entradas (especialmente habría que priorizar en la asignación a aquellos profesores que tengan disponibilidad horaria reducida).

# Conclusión

Se logró la investigación del problema en cuestión, realizando el estudio del estado del arte. Encontramos ciertos intentos de soluciones académicas, y otras soluciones comerciales.

Gracias a esto pudimos aprender sobre los algoritmos genéticos y sobre las búsquedas tabú, y ver su aplicación en el problema.

Profundizamos también sobre los problemas NP completo y entendimos porque este problema en particular forma parte de dicho grupo.

También ideamos una solución restringida al problema, logrando entenderlo más en profundidad. Encontramos algunos puntos a mejorar en nuestra solución y comprendiendo las limitaciones.

Dependiendo fundamentalmente de las restricciones, se podría afirmar que no hay una solución genérica, sino que se halla una solución para ese contexto en específico.

En cuanto a la solución planteada, al utilizar una función random no aseguramos obtener en cada ejecución obtener una misma solución, porque, de hecho, no siempre existe una única solución posible.

Consideramos que fue una buena instancia para investigar y analizar diferentes soluciones, y en base a esto, crear nuestra solución.

# Bibliografía

- Esquivel T, L. L. (2014, 21 septiembre). *MODELO MATEMÁTICO PARA LA PROGRAMACIÓN DE UN HORARIO ESCOLAR CON MULTI-LOCALIZACIÓN DE DOCENTES*. Universidad del Valle. <https://bibliotecadigital.univalle.edu.co/bitstream/handle/10893/8312/CB-0516251.pdf;jsessionid=79290349E1968D48010F9A4DF46A8AF0?sequence=1https://bibliotecadigital.univalle.edu.co/bitstream/handle/10893/8312/CB-0516251.pdf;jsessionid=79290349E1968D48010F9A4DF46A8AF0?sequence=1>
- Suárez, V. F. (2012, 4 septiembre). *Programación de Horarios Escolares basados en Ritmos Cognitivos usando un Algoritmo Genético de Clasificación No-dominada, NSGA-II*. Scielo. [https://scielo.conicyt.cl/scielo.php?script=sci\\_arttext&pid=S0718-07642013000100012](https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642013000100012)
- Corne, D. W. (2018, 28 mayo). *Evolutionary Algorithms*. arXiv.org. <https://arxiv.org/abs/1805.11014>
- *Software generador de horarios para centros educativos*. (s. f.). peñalara. [https://www.penalara.com/es/UY?gclid=CjwKCAiAtK79BRAIEiwA4OskBrfjRUMSRXDy7s36BfnO4VlIAKAI\\_k8xdGEKvc0AdX2iAiIzjhk2PhoCVyMQAvD\\_BwE](https://www.penalara.com/es/UY?gclid=CjwKCAiAtK79BRAIEiwA4OskBrfjRUMSRXDy7s36BfnO4VlIAKAI_k8xdGEKvc0AdX2iAiIzjhk2PhoCVyMQAvD_BwE)
- *UniTime | University Course Timetabling Problem Description*. (s. f.). UniTime. [https://www.unitime.org/uct\\_description.php](https://www.unitime.org/uct_description.php)