

---

## Generador de horarios para colegios

# Índice

<b>1. Explicación</b>	<b>3</b>
<b>2. Modelamiento del problema</b>	<b>3</b>
2.1. Datos . . . . .	3
2.2. Variables . . . . .	4
2.3. Restricciones . . . . .	4
2.4. Función objetivo . . . . .	8
<b>3. Implementación</b>	<b>8</b>
<b>4. Programa</b>	<b>8</b>

# 1. Explicación

El proyecto tiene como propósito modelar el problema de generación de horarios como un modelo de optimización, teniendo en cuenta las restricciones que puede tener un problema de este tipo y las que podría querer considerar un colegio al momento de hacer los horarios. Luego, se quiere generar un programa que permita a la persona que quiere generar su horario, personalizar este, imponiendo restricciones y viendo si estas se pueden cumplir. Para la interfaz se propondrán algunas ideas.

## 2. Modelamiento del problema

### 2.1. Datos

- *dias* (**D**): Lista que muestra los días en los cuales se quieren planificar clases en un colegio. Generalmente será de lunes a viernes.
- *modulos* (**M**): Lista que contiene todos los módulos de clases con los que cuenta el colegio.
- *modulos\_consecutivos* (**MC**): Lista que contiene listas de la forma  $[i, j]$  que señala que entre el ramo  $i$  y el ramo  $j$  no hay ninguna pausa (recreo o almuerzo).
- *horarios* [*curso*] [*día*] (**H**): Para cada día de cada curso muestra los módulos de clases que tiene cada curso en cada día de la semana.
- *limitaciones\_profesores* [*profesor*] (**LP**): Para cada profesor entrega el máximo de módulos de clases que puede hacer en una semana como también el máximo de módulos que puede hacer en un día.
- *carga\_profesores* [*profesor*] (**CP**): Para cada profesor entrega, en una lista, el/los curso/s a los que se le hace una clase y el nombre del ramo.
- *profesores\_por\_curso* [*curso*] (**PpC**): Para cada curso se tiene una lista con la lista de los profesores que hacen un ramo y el nombre del ramo.
- *profesores\_a\_definir\_por\_curso* [*curso*] (**PadpC**): Para cada curso se tiene una lista con la lista de los profesores que podrían hacer el ramo y el nombre del ramo. Solo se elegirá uno de los profesores.
- *ramos\_por\_curso* [*curso*] [*ramo*] (**RpC**): Para cada ramo de cada curso indica la cantidad de módulos semanales, la cantidad máxima de módulos diarios y el tipo de módulos (seguidos, disjuntos o sin preferencia).
- *asignaciones* (**A**): Lista que contiene todas las tuplas de la forma curso, profesor, clase.
- *asignaciones\_con\_restricciones\_de\_dias* (**AcR**): Lista que contiene todas las tuplas de la forma curso, profesor, clase que se quiere que tengan módulos consecutivos
- *profesores\_con\_dias\_limitados* (**PDL**): Lista de profesores que solo harán una cantidad de días clases.
- *tope\_dias* [*profesor*] (**TD**): Cantidad máxima de días que cada profesor hace clases. Esto es solo para los profesores que tienen tope.
- *asignaciones\_fijadas* (**AF**): Asignaciones fijadas por el usuario.
- *asignaciones\_vetadas* (**AV**): Asignaciones que el usuario bloquea, es decir, impide que se den en el resultado final.
- *ramos\_en\_distintos\_dias* [*curso*] (**RedD**): Para cada curso indica los ramos que no pueden estar en un mismo día.

- *primeros\_modulos\_exclusivos*[*curso*] (**PME**): Lista de módulos que solo pueden ser un primer módulo de un bloque, es decir, es imposible que un bloque de dos módulos termine en este módulo.
- *segundos\_modulos\_exclusivos*[*curso*] (**SME**): Lista de módulos que solo pueden ser un segundo módulo de un bloque, es decir, es imposible que un bloque de dos módulos parta en este módulo.

## 2.2. Variables

- $x_{cprdj}$ : Toma el valor 1 si el curso  $c$  tiene el ramo  $r$  con el profesor  $p$  el día  $d$  en el módulo  $j$  y 0 en caso contrario. Se usa para módulos de clases sueltos
- $z_{cprdj}$ : Toma el valor 1 si el curso  $c$  tiene el ramo  $r$  con el profesor  $p$  el día  $d$  en el módulo  $j$  y 0 en caso contrario. Se usa para módulos consecutivos y representa el primer módulo de este bloque.
- $y_{cprdj}$ : Toma el valor 1 si el curso  $c$  tiene el ramo  $r$  con el profesor  $p$  el día  $d$  en el módulo  $j$  y 0 en caso contrario. Se usa para módulos consecutivos y representa el segundo módulo de este bloque.
- $s_{pd}$ : Toma el valor 1 si el profesor  $p$  hace clases el día  $d$  y 0 en caso contrario.
- $w_{cpr}$ : Toma el valor 1 si el profesor  $p$  hace el ramo  $r$  al curso  $c$ .
- $n_{pdj}$ : Toma el valor 1 si el profesor  $p$  hace clases el día  $d$  en el módulo  $j$ .

## 2.3. Restricciones

1. Tiene que haber como máximo una clase cada módulo de cada día que forma parte del horario de cada curso:

$$\sum_{\forall ps, r \in PpC[c]} (x_{c,ps[0],r,d,j} + z_{c,ps[0],r,d,j} + y_{c,ps[0],r,d,j}) + \sum_{\forall ps, r \in PadpC[c]} \sum_{\forall p \in ps} (x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j}) \leq 1$$

$$\forall c \in H, \forall d \in H[c], \forall j \in H[c][d]$$

2. Tiene que haber como mínimo un profesor para cada módulo de cada día que forma parte del horario de cada curso:

$$\sum_{\forall ps, r \in PpC[c]} \sum_{p \in ps} (x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j}) + \sum_{\forall ps, r \in PadpC[c]} \sum_{\forall p \in ps} (x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j}) \geq 1$$

$$\forall c \in H, \forall d \in H[c], \forall j \in H[c][d]$$

3. Si más de un profesor hace una clase, entonces estos la hacen simultáneamente (variable x):

$$x_{c,p_1,r,d,j} = x_{c,p_2,r,d,j} \quad \forall c \in PpC, \forall ps, r \in PpC[c] : \text{len}(ps) > 1, \forall p_1, p_2 \in ps, \forall d \in D, \forall j \in M$$

4. Si más de un profesor hace una clase, entonces estos la hacen simultáneamente (variable z):

$$z_{c,p_1,r,d,j} = z_{c,p_2,r,d,j} \quad \forall c \in PpC, \forall ps, r \in PpC[c] : \text{len}(ps) > 1, \forall p_1, p_2 \in ps, \forall d \in D, \forall j \in M$$

5. Si más de un profesor hace una clase, entonces estos la hacen simultáneamente (variable y):

$$y_{c,p_1,r,d,j} = y_{c,p_2,r,d,j} \quad \forall c \in PpC, \forall ps, r \in PpC[c] : \text{len}(ps) > 1, \forall p_1, p_2 \in ps, \forall d \in D, \forall j \in M$$

6. Cantidad de variables x si se tienen módulos seguidos:

$$\sum_{d \in D} \sum_{j \in M} x_{c,p,r,d,j} = RpC[c][r][\text{'modulos\_semanales'}] \% 2$$

$$\forall c \in PpC, \forall ps, r \in PpC[c] : RpC[c][r][\text{'tipo\_modulos'}] = \text{"seguidos"}, \forall p \in ps$$

7. Cantidad de variables z si se tienen módulos seguidos:

$$\sum_{d \in D} \sum_{j \in M} z_{c,p,r,d,j} = \left\lceil \frac{RpC[c][r][\text{'modulos\_semanales'}]}{2} \right\rceil$$

$$\forall c \in PpC, \forall ps, r \in PpC[c] : RpC[c][r][\text{'tipo\_modulos'}] = \text{"seguidos"}, \forall p \in ps$$

8. Cantidad de variables y si se tienen módulos seguidos:

$$\sum_{d \in D} \sum_{j \in M} y_{c,p,r,d,j} = \left\lceil \frac{RpC[c][r][\text{'modulos\_semanales'}]}{2} \right\rceil$$

$$\forall c \in PpC, \forall ps, r \in PpC[c] : RpC[c][r][\text{'tipo\_modulos'}] = \text{"seguidos"}, \forall p \in ps$$

9. Cantidad de variables x si no se tienen módulos seguidos:

$$\sum_{d \in D} \sum_{j \in M} x_{c,p,r,d,j} = RpC[c][r][\text{'modulos\_semanales'}]$$

$$\forall c \in PpC, \forall ps, r \in PpC[c] : RpC[c][r][\text{'tipo\_modulos'}] \neq \text{"seguidos"}, \forall p \in ps$$

10. Cantidad de variables x si se tienen módulos seguidos para profesores no definidos:

$$\sum_{d \in D} \sum_{j \in M} \sum_{p \in ps} x_{c,p,r,d,j} = RpC[c][r][\text{'modulos\_semanales'}] \% 2$$

$$\forall c \in PadpC, \forall ps, r \in PadpC[c] : RpC[c][r][\text{'tipo\_modulos'}] = \text{"seguidos"}$$

11. Cantidad de variables z si se tienen módulos seguidos para profesores no definidos:

$$\sum_{d \in D} \sum_{j \in M} \sum_{p \in ps} z_{c,p,r,d,j} = \left\lceil \frac{RpC[c][r][\text{'modulos\_semanales'}]}{2} \right\rceil$$

$$\forall c \in PadpC, \forall ps, r \in PadpC[c] : RpC[c][r][\text{'tipo\_modulos'}] = \text{"seguidos"}$$

12. Cantidad de variables y si se tienen módulos seguidos para profesores no definidos:

$$\sum_{d \in D} \sum_{j \in M} y_{c,p,r,d,j} = \left\lceil \frac{RpC[c][r][\text{'modulos\_semanales'}]}{2} \right\rceil$$

$$\forall c \in PadpC, \forall ps, r \in PadpC[c] : RpC[c][r][\text{'tipo\_modulos'}] = \text{"seguidos"}$$

13. Cantidad de variables x si no se tienen módulos seguidos para profesores no definidos:

$$\sum_{d \in D} \sum_{j \in M} \sum_{p \in ps} x_{c,p,r,d,j} = RpC[c][r][\text{'modulos_semanales'}]$$

$$\forall c \in PadpC, \forall ps, r \in PadpC[c] : RpC[c][r][\text{'tipo_modulos'}] \neq \text{"seguidos"}$$

14. No se tiene clases en los módulos que no son parte del horario:

$$\sum_{ps, r \in PpC[c]} \sum_{p \in ps} (x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j}) + \sum_{ps, r \in PadpC[c]} \sum_{p \in ps} (x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j}) = 0$$

$$\forall c \in H, \forall d \in H[c], \forall j \in M \setminus H[c][d]$$

15. Los profesores solo pueden hacer una clase al mismo tiempo (sin contar las que se hacen a más de un curso al mismo tiempo):

$$\sum_{cs, r \in CP[p]} (x_{cs[0],p,r,d,j} + z_{cs[0],p,r,d,j} + y_{cs[0],p,r,d,j}) \leq 1 \quad \forall p \in CP, \forall d \in D, \forall j \in M$$

16. Vinculación clases que profesores hacen a un más de un curso simultáneamente (variable x).

$$x_{c_1,p,r,d,j} = x_{c_2,p,r,d,j} \quad \forall p \in CP, \forall cs, r \in CP[p], \forall c_1, c_2 \in cs, \forall d \in D, \forall j \in M$$

17. Vinculación clases que profesores hacen a un más de un curso simultáneamente (variable z).

$$z_{c_1,p,r,d,j} = z_{c_2,p,r,d,j} \quad \forall p \in CP, \forall cs, r \in CP[p], \forall c_1, c_2 \in cs, \forall d \in D, \forall j \in M$$

18. Vinculación clases que profesores hacen a un más de un curso simultáneamente (variable y).

$$y_{c_1,p,r,d,j} = y_{c_2,p,r,d,j} \quad \forall p \in CP, \forall cs, r \in CP[p], \forall c_1, c_2 \in cs, \forall d \in D, \forall j \in M$$

19. Se tiene como máximo la cantidad de módulos diarios definidos:

$$\sum_{\forall j \in M} (x_{c,ps[0],r,d,j} + z_{c,ps[0],r,d,j} + y_{c,ps[0],r,d,j}) \leq RpC[c][r][\text{'maximo_modulos_diario'}]$$

$$\forall c \in PpC, \forall ps, r \in PpC[c], \forall d \in D$$

20. Se tiene como máximo la cantidad de módulos diarios definidos (caso profesores no asignados):

$$\sum_{\forall j \in M} \sum_{\forall p \in ps} x_{c,p,r,d,j} \leq RpC[c][r][\text{'maximo_modulos_diario'}] \quad \forall c \in PadpC, \forall ps, r \in PadpC[c], \forall d \in D$$

21. La misma clase solo puede ocurrir el mismo día en módulos disjuntos si se eligió esa opción:

$$x_{c,p,r,d,j} + x_{c,p,r,d,z} \leq 1 \quad \forall c, p, r \in A : RpC[c][r][\text{'tipo_modulos'}] = \text{"disjuntos"}, \forall d \in D, \forall j, z \in MC)$$

22. Límite a la cantidad de módulos de clases que puede hacer cada profesor:

$$\sum_{cs, r \in CP[p]} \sum_{d \in D} \sum_{j \in M} (x_{cs[0],p,r,d,j} + z_{cs[0],p,r,d,j} + y_{cs[0],p,r,d,j}) \leq LP[p][\text{'maximo_modulos'}] \quad \forall p \in CP$$

23. Límite a la cantidad de módulos de clases que puede hacer cada profesor por día:

$$\sum_{cs,r \in CP[p]} \sum_{d \in D} (x_{cs[0],p,r,d,j} + z_{cs[0],p,r,d,j} + y_{cs[0],p,r,d,j}) \leq LP[p][\text{'maximo\_modulos\_diario'}]$$

$$\forall p \in CP, \forall j \in M$$

24. Introducción de la variable  $s$  para poner un tope a la cantidad de días que los profesores con días limitados pueden hacer clases:

$$M * s_{p,d} \geq \sum_{cs,r \in CP[p]} \sum_{j \in M} (x_{cs[0],p,r,d,j} + z_{cs[0],p,r,d,j} + y_{cs[0],p,r,d,j}) \quad \forall d \in D, \forall p \in PDL$$

25. Uso de la variable  $s$  para limitar la cantidad de días que un profesor hace clases:

$$\sum_{d \in D} s_{p,d} \leq TD[p] \quad \forall p \in PDL$$

26. Implementación de la variable  $w$ :

$$M * w_{c,p,r} \geq \sum_{d \in H[c]} \sum_{j \in H[c][d]} (x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j}) \quad \forall c \in PadpC, \forall ps, r \in PadpC[c], \forall p \in ps$$

27. Máximo un profesor para los ramos de cada curso que no tienen profesor asignado:

$$\sum_{ps,r \in PadpC[c]} \sum_{p \in ps} w_{c,p,r} \leq 1, \quad \forall c \in PadpC$$

28. Las asignaciones fijadas previamente se respetan:

$$x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j} = 1 \quad c, p, r, d, j \in AF$$

29. Las asignaciones vetadas previamente se respetan:

$$x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j} = 0 \quad c, p, r, d, j \in AV$$

30. Algunos ramos no pueden estar en un mismo día:

$$\begin{aligned} & \sum_{j \in M} (x_{c,p_1,r_1,d,j} + x_{c,p_2,r_2,d,j} + z_{c,p_1,r_1,d,j} + z_{c,p_2,r_2,d,j} + y_{c,p_1,r_1,d,j} + y_{c,p_2,r_2,d,j}) \leq \\ & \text{máx}(RpC[c][r_1][\text{'maximo\_modulos\_diario'}], RpC[c][r_2][\text{'maximo\_modulos\_diario'}]) \\ & \forall c \in RedD, \forall ls \in RedD[c], \forall p_1, r_1, p_2, r_2 \in (ls \times ls) \end{aligned}$$

31. Solo se activa una de las variables (x,y,z) a la vez:

$$x_{c,p,r,d,j} + z_{c,p,r,d,j} + y_{c,p,r,d,j} \leq 1 \quad \forall c, p, r \in A, \forall d \in D, \forall j \in M$$

32. Las variables  $z$  e  $y$  se activan en módulos consecutivos:

$$z_{c,p,r,d,j} - y_{c,p,r,d,k} = 0 \quad \forall c, p, r \in A, \forall d \in D, \forall j \in M[i], \forall k \in M[i+1]$$

33. Las variables  $z$  no pueden tomar valor 1 en segundo módulos exclusivos:

$$z_{c,p,r,d,j} = 0 \quad \forall c, p, r \in A, \forall d \in D, \forall j \in SME$$

34. Las variables  $y$  no pueden tomar valor 1 en primeros módulos exclusivos:

$$y_{c,p,r,d,j} = 0 \quad \forall c, p, r \in A, \forall d \in D, \forall j \in PME$$

Solo se usa una de las siguientes restricciones. La primera no permite que un profesor haga más de 4 módulos seguidos. Como es una regla, es posible que los problema planteados con esta restricción no tengan solución.

La segunda implica el uso de la variable  $n$ . Si se minimiza la suma de esta variable, se logra que la menor cantidad de profesores tenga muchos módulos seguidos y los problemas no son infactibles.

35. Ningún profesor hace más de 4 módulos seguidos:

$$\sum_{cs, r \in CP[p]} x_{cs[0],p,r,d,j_1} + x_{cs[0],p,r,d,j_2} + x_{cs[0],p,r,d,j_3} + x_{cs[0],p,r,d,j_4} + x_{cs[0],p,r,d,j_5} \leq 4$$

$$\forall p \in CP, \forall d \in D, \forall j_1, j_2, j_3, j_4, j_5 \in [(i, i+1, i+2, i+3, i+4) \forall i \in M : i+4 \in M]$$

36. Implementación de variable  $n$ :

$$M * n_{p,d,j} \geq \sum_{cs, r \in CP[p]} \sum_{c \in cs} (x_{c,p,r,d,j} + y_{c,p,r,d,j} + z_{c,p,r,d,j}) \quad \forall d \in D, \forall j \in M : j+4 \in M$$

## 2.4. Función objetivo

Para resolver el problema basta con poner una función objetivo igual a 0, dado que esto asegura que se cumplirán las restricciones ya señaladas. Si se quiere hacer uso de la variable  $n$  (intentar que no se tengan muchos bloques de 5 módulos seguidos) se usa la siguiente función objetivo:

$$\min \sum_{p \in CP} \sum_{d \in D} \sum_{j \in M : j+4 \in M} n_{p,d,j}$$

## 3. Implementación

El programa está diseñado para ser usado con los datos mínimos, permitiendo que la persona encargada de hacer los horarios pueda personalizar estos. Existen restricciones que significan mayor dificultad para resolver el problema, como lo son la no-asignación de profesores o la limitación de días para un profesor. Un abuso de estas opciones dificultará la solución del problema e incluso lo podría hacer infactible. Para lograr que siempre se encuentre una solución, se debe implementar una lista con todas las restricciones que no son esenciales para responder el problema (como la cantidad de módulos diarios, que no haya más de una clase al mismo tiempo, etc.). Las restricciones que debieran estar en esta lista son: 21, 22, 23, 25, 28, 29 y 30.

## 4. Programa

La aplicación donde se ingresan los datos debe tener una ventana donde, para cada curso, se escribe el nombre de cada ramo, el nombre del/de los profesor/es que lo dicta, la cantidad de módulos semanales para ese ramo y la cantidad de módulos diarios (1 o 2 generalmente). Si hay más de un módulo diario se debe dar la opción de permitir módulos disjuntos o módulos consecutivos.



También se debe tener la opción de vetar que dos ramos ocurran el mismo día y se debe poder vincular los ramos con otros cursos (señalando que la clase se hace de forma conjunta).

A medida que se van agregando profesores, el programa debe permitir agregar restricciones para cada profesor, en particular, la cantidad de días de trabajo, el máximo de módulos diarios y el máximo de módulos que puede trabajar. Esto último es necesario si es que no se ha asignado algún profesor y se quiere que el modelo elija entre los profesores que se han ingresado.

Tras ingresar estos datos, se tiene definida la cantidad de módulos de clases por curso, por lo que el programa debe solicitar al usuario que este ingrese la forma que tienen los horarios de cada curso, repartiendo la totalidad de módulos que se tiene a disposición. Aquí también se debe determinar qué módulos califican como consecutivos (algunos colegios pueden considerar que un recreo hace que no sean consecutivos y otros podrían pensar que sí).

Luego, se debe dar la opción de agregar cursos a un módulo particular de forma manual, las llamadas asignaciones. Esto se puede hacer desde una vista de los profesores o de una vista de los cursos. Esto puede ser útil para ramos que no pueden ocurrir de forma simultánea. A futuro se podría agregar soporte para considerar salas de clases, la dificultad radica más en la estructura de los datos que en la modelación misma, que en sí no es tan complicada.

Por último, antes de pasarle los datos al solver se tiene que hacer un chequeo simple de que no se estén infringiendo reglas simples (ningún profesor hace clases al mismo tiempo, ningún curso tiene 2 clases simultáneas, etc.). Estos se pueden hacer a medida que el usuario vaya ingresando los datos para ir validando las reglas que pone. Además, se le debe dar la opción al usuario de establecer prioridad para las reglas que no son esenciales para la construcción del horario (mencionadas anteriormente). De forma que, si el problema es infactible, este se pueda hacer factible relajando el problema. Las restricciones debieran ser eliminadas en grupos para que el tiempo de computación sea un poco menor. Verificar la infactibilidad de un problema es bastante rápido, por lo que el programa puede emitir alertas y señalarle al usuario lo que ocurre a medida que va computando.

Una vez el modelo genera un horario, se le debe dar la opción al usuario de modificar el output para así eliminar partes de este que no le hayan gustado. Estas modificaciones también entran en la lista de prioridades anteriormente descrita, dado que puede llevar a infactibilidades.