

## Práctica 6: Horarios

Se trata de hacer los horarios de un centro de estudios para primer curso. Partimos del número de asignaturas totales ( $n$ ), el número de días con clase por semana (días), y el número máximo de horas por día (horas). Todas estas constantes y las variables que usaremos están en el fichero `horarios.mzn`, que debe completarse y entregarse una vez completado.

Se debe cumplir que

- Todas las clases se dan en bloques de 1 o más horas. Sabemos la cantidad de cada bloques por semana que se deben impartir (array `bloques`). Por ejemplo:  
`bloques = [[0,2 | 2,1 | 2,1 | 4,0 | 4,0 |]`  
indica que de la asignatura 1 se imparten 0 bloques de 1 hora y 2 bloques de dos horas, de la asignatura 2 se imparten 2 bloques de una hora y un bloque de dos horas, y así consecutivamente.
- Una asignatura no puede darse dos veces el mismo día.

El horario se obtendrá en un array:

`array [1..días,1..n] of var 0..maxbloque:horario;`

`horario[i,j]` indica las horas que se imparten el día  $i$  de la asignatura  $j$ . Una posible solución sería:

L	M	X	J	V
0	2	0	0	2
0	0	1	1	2
2	0	1	1	0
1	1	1	1	0
1	1	1	1	0

que indica, por ejemplo, que la asignatura 1 se imparte martes y jueves en tandas de 2 horas.

Añadir al modelo:

1) Un predicado `horasDia` que reciba un horario (array `[1..días,1..n] of var 0..maxbloque`) y asegure que ningún día se superan las horas de clase por día (constante `horas`).

2) Un predicado `nbloques` que recibe como parámetros:

- Un horario, como en el ejercicio anterior
- Una asignatura (número de 1 a  $n$ )
- Un tamaño de bloque `bsize` (de 1 a `maxbloque`)
- Una variable de tipo entero “total”

El predicado debe asegurar que semanalmente se imparte una cantidad “total” de bloques de tamaño bsize de la asignatura.

Aunque no hace falta, si se quiere probar este predicado por separado se pueden añadir las restricciones

`constraint nbloques(horario,4,1,4);`

`constraint nbloques(horario,4,2,0);`

que deben ser satisfactibles.

3) Un predicado **bloquesPorAsignatura** que reciba como parámetros un horario y asegure que para cada asignatura se imparte la cantidad indicada en “bloques” de bloques de cada tamaño. Utilizará el predicado anterior.

4) Constraints que aseguran que se respeta el máximo número de horas por días (pregunta 1) y que todas las asignaturas se imparten en bloques de tamaño adecuado (pregunta 3).

5) Una salida formateada para mostrar el resultado como en este enunciado, con los nombres de los días en primera fila y con cada fila correspondiendo a una asignatura.