/\*

DATOS DEL MODELO

\*/

include "globals.mzn";

/\* Definir qué es una sesión y cuáles hay en total \*/

int: sesiones; % número de sesiones que hay en total

set of int: numSesiones = 1..sesiones;

array[numSesiones, 3] of int: SESIONES; % la fila 1 corresponde a la sesión 1, la fila 2 a la sesión 2, etc. El valor de la columna 1 sirve para determinar a qué tanda pertenece, la columna 2 en minutos es para la duración de la sesión y la columna 3 en días es para la separación ideal respecto a la sesión anterior. Como el enunciado no lo dice y podemos asumir ciertas cosas, o incluso demanadar un formato concreto de input a quienes nos proveen de los datos, asumiremos que si hay una sesión i que pertenece a la tanda j, y es la primera sesión que aparece en el array que pertenezca a la tanda j, entonces esa sesión i es la primera sesión de la tanda j. Del mismo modo, la sesión i+1 que pertenezca a la misma tanda j será la segunda sesión de la tanda j, y la i+2 la tercera de la tanda j, y así sucesivamente para el resto de tandas

/\* Definir los dias que lleva esperando un paciente \*/

int: tandas;

set of int: numTandas = 1..tandas;

array[numTandas] of int: diasEsperando; % la fila 1 corresponde a la tanda 1, la fila 2 a la tanda 2, etc. Cada posición del array tiene un entero en días que indica cuántos días lleva esperando esta tanda a ser planificada

/\* Definir qué son los LINACs y cuántos hay en total \*/

int: m;

set of int: numLINAC = 1..m;

array[numLINAC] of int: LINAC; % cada posición del array corresponde al número de LINAC que es

/\* Definir qué es un día \*/

set of int: DIA; % en nuestro modelo un día es un conjunto de sesiones. Ponemos de tipo “int” porque se refiere al identificador de las sesiones. Si el valor de un elemento de DIA es 1, se refiere a la sesión en la posición 1 del array SESIONES

/\* Definir old planning \*/

set of int: oldNumDias = 1..22; % número calculado con información del enunciado, planning que calculamos el viernes de la semana pasada. Solamente tiene 22 días, porque el viernes pasado si ese mismo día un médico hubiera solicitado una tanda, entonces podrían haber 8 días de espera hasta que empezara la primera sesión de la tanda, más 21 días de lo que puede durar una tanda y 21+8 = 29. Como ya ha pasado una semana desde que hicimos esta planificación (viernes pasado) le tenemos que restar la semana que ya ha pasado, y 29-7 = 22

array[numLinac, oldNumDias] of DIA: OLD\_PLANNING; % se trata de una matriz en la que las filas son los diferentes linacs, y las columnas son los diferentes días que planeamos. Cada posición de esta matriz es un conjunto de sesiones que son las que se realizarán en el correspondiente día y linac. Como es el planning que hicimos el viernes pasado, se llama old\_planning

/\* Definir cómo es la planificación nueva \*/

set of int: numDias = 1..29; % número calculado con información del enunciado. Sabemos que una tanda dura como mucho 3 semanas (21 días) de modo que como mucho planificaremos 21 días en adelante desde hoy. También sabemos que si el médico pide hoy mismo la primera sesión de la tanda, como mucho podrán pasar 8 días hasta que nos den la primera sesión de esta tanda. Así que en total en el “peor” de los casos planificamos 21+8 días en adelante, o sea 29

var array[numLinac, numDias] of DIA: PLANNING; % se trata de una matriz en la que las filas son los diferentes linacs, y las columnas son los diferentes días que planeamos. Cada posición de esta matriz es un conjunto de sesiones que son las que se realizarán en el correspondiente día y linac

/\*  
RESTRICCIONES

1. Para cualquier día y LINAC, la suma de las duraciones de la conjunción de las sesiones que ya estaban planeadas (old\_planning) con las que hemos planeado (planning) no puede superar el tiempo disponible en un día (12 horas).

2. Tenemos que respetar la planificación anterior. Para esto tenemos un dato old\_planning, y cada posición de esta matriz es un día, y un día es un conjunto de sesiones. Del mismo modo funciona para el nuevo planning. Esta restricción dice que la intersección de los dos conjuntos en cada planning tiene que ser igual al del old\_planning. Por ejemplo: old\_planning(LINAC1, día1) = {1, 2} ^ planning(LINAC1, día1) = {1,2,5,6} es igual a {1,2} que es igual a old\_planning(LINAC1, día1)

3. Tenemos que comprobar que para la primera sesión de toda tanda la suma de los días que lleva el paciente esperando (mirar a qué tanda pertenece esta sesión, y mirar diasEsperando para saber cuántos días lleva esperando) + 2 (En sábado y domingo no podemos planear ninguna sesión) + el día (la columna) en la que está planeada la susodicha sesión, esta suma tiene que ser inferior o igual a 8.

4. Una sesión no puede estar en diferentes días, sólo hay un elemento igual a esa sesión.

5. Una sesión no puede estar en más de un LINAC.

6. Se tiene que respetar el orden de sesiones que hay en cada tanda. No puede haber la sesión iésima si antes no ha habido la iésima -1.

7. La última sesión de una tanda tiene que estar como muy tarde 21 días después de la primera sesión.

\*/

/\*

FUNCIÓN OBJETIVO

La función objetivo se tiene que minimizar. Dicha función es el máximo del valor absoluto de la diferencia entre la separación entre dos sesiones consecutivas y la separación ideal.

\*/