**Guía de Clase. (11 de Octubre de 2019)**

**Profesor: Julian David Sanchez**

# ARRAYS EN MINIZINC

Defina según su experiencia:

Array: colección de datos bajo un solo nombre, es manipulable con punteros de un mismo tipo, este ataca problemas que tienen que ver con agrupación de datos. sirve para agrupar datos de un mismo tipo.

¿Para que tipo de problemas es bueno usar arrays?
Se usa para la organización de datos. Es bueno usarlo para problemas donde se deban agrupar datos.

En Minizinc el empleo de arrays es muy común debido a la capacidad de modelamiento que estos nos permiten, en problemas o planteamientos complejos. Un array puede ser tanto un parámetro como una variable de decisión de acuerdo con el tipo de uso que se le quiera dar.

Definición de un array (Sintaxis):

array[expr] of [var] type : name_array;

Donde expr es un rango de elementos, que define el tamaño del vector y sus posiciones, type son los tipos de variables que ya hemos definido (float, int …) y que serán los datos almacenados en el array y name_array es el nombre del elemento.

Ejemplo de definición de un array

array[0..9] of int: alumnos; %Define un parámetro tipo array de enteros de 10 elementos enumerados entre 0 y 9 que se llama alumnos.
array [0..9] of int:alumnos;

Defina una variable tipo array de 20 posiciones de floats que se llame notas:

array[0..19] of var float:notas;

Defina un arreglo de 100 posiciones de enteros entre -20 y 30 llamado xyz, este arreglo contendrá los valores resultado de una maximización.

array[0..99] var -20..30 :xyz;

Con los conocimientos que ha adquirido, realice la definición de un array x de tamaño n, donde n es un valor ingresado por el usuario.

array[0..n-1] of int:X;

**RECORRER UN ARRAY:**

Para recorrer un array en minizinc se hace necesario conocer la estructura de control forall.

Sintaxis:

forall( variable in [rango de valores entre los que itera] ) (Instrucción);

Con esta definición validemos un array en que todos sus elementos sean igual a 1:

constraint forall(i in 0..9)(x[i] = 1);

Construya un constraint que valide que cada valor en un vector x sea mayor que el anterior.

constraint forall (i in 1..9)(X[i]>X[i-1]);