A blue and white rectangular object with text

Description automatically generated

Escribir codigo de tres direcciones es una manera conceptual de escribir código en assembler con la diferencia de escribir directamente los demonicos(instrucciones)

A close-up of a computer screen

Description automatically generatedA screenshot of a computer

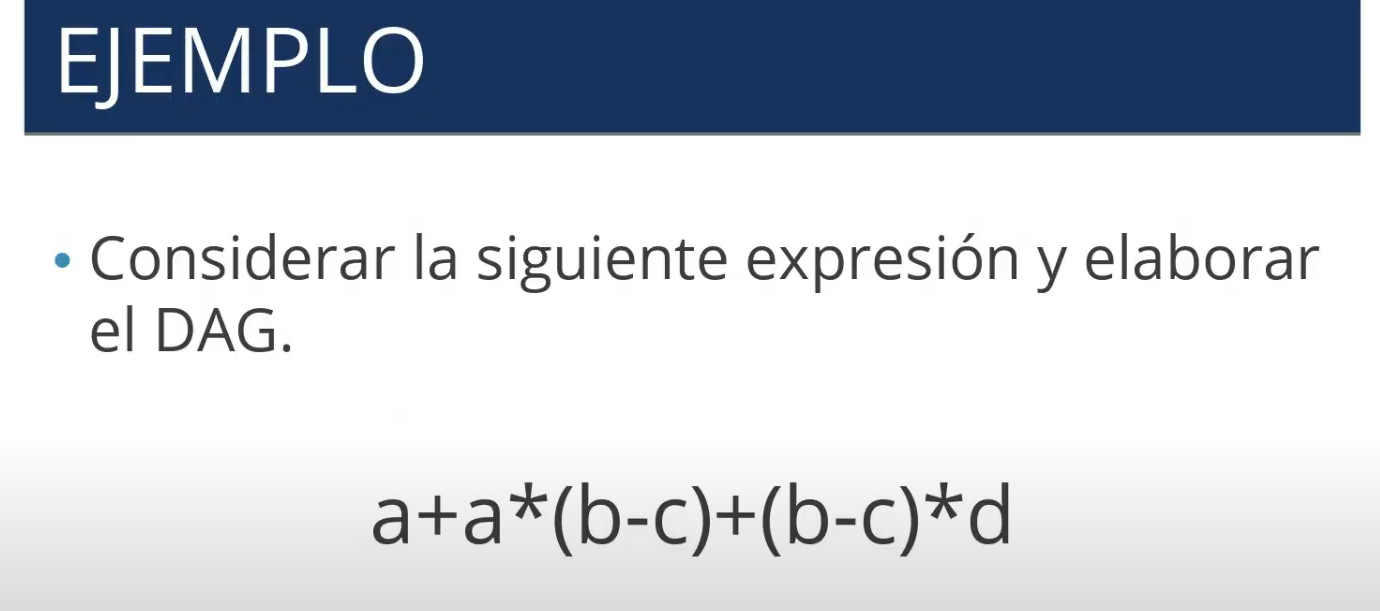
Description automatically generated

Una características de los arboles es que nos permiten identificar ramas repetidas y esto nos permite hacer optimizaciones. Los grafos acíclicos dirigos son una variante optimizada de un AST

Se le llama subexpresiones comunes a los nodos/ramas que se repiten en una expresion

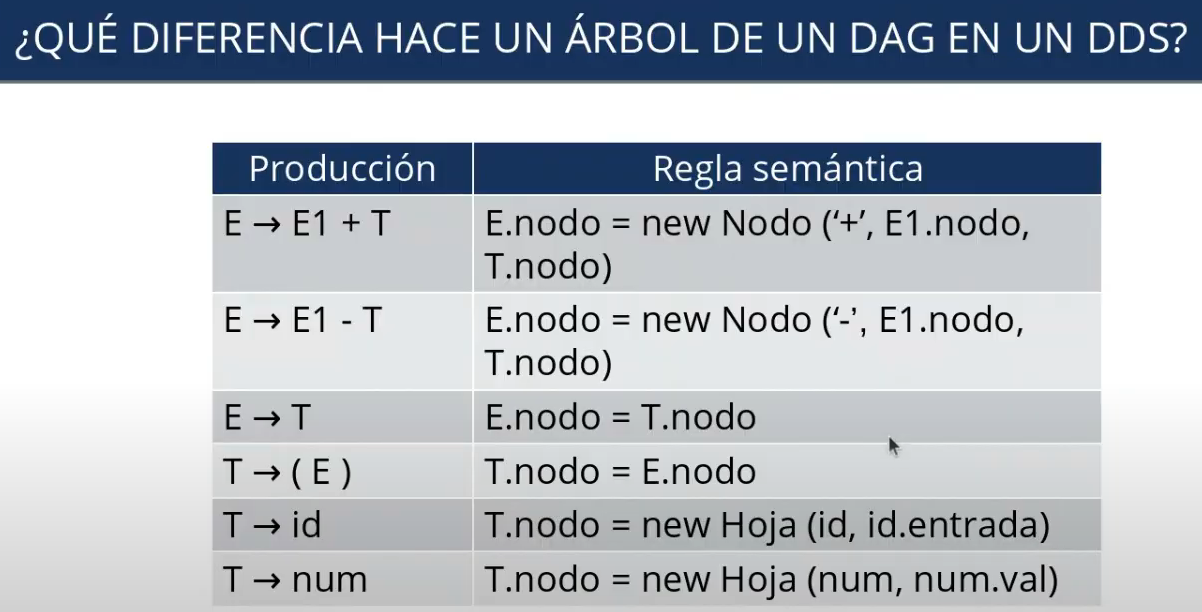
A screenshot of a computer

Description automatically generated

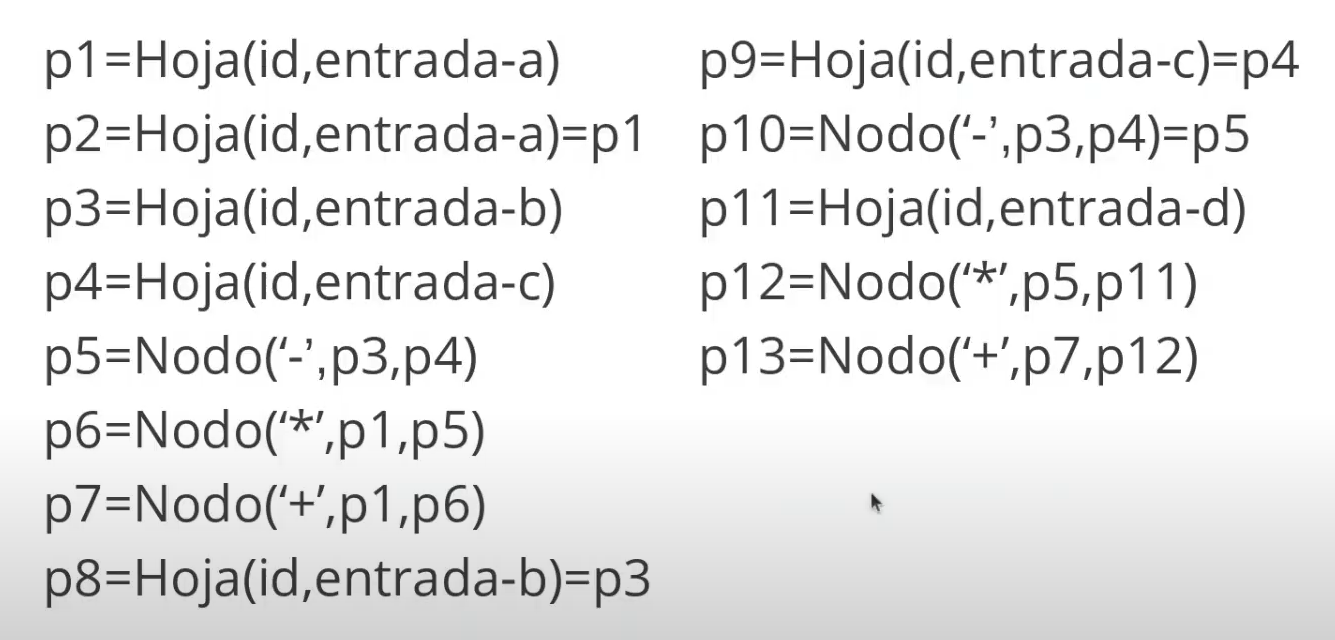


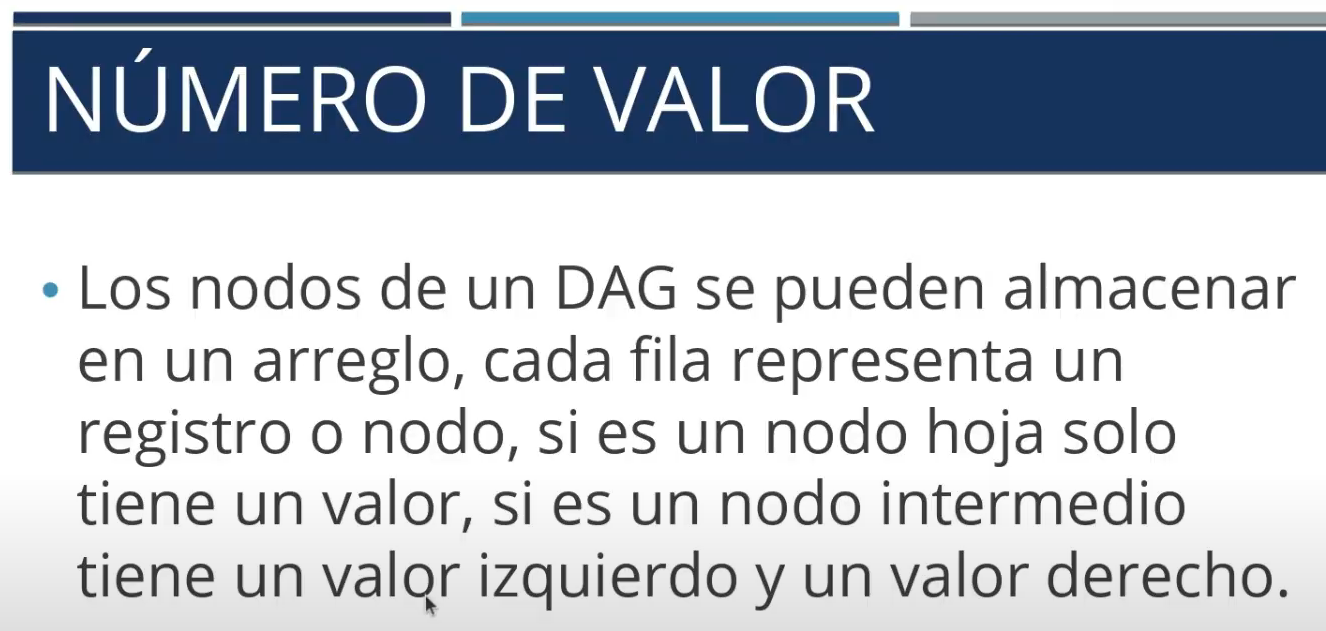
A diagram of a network

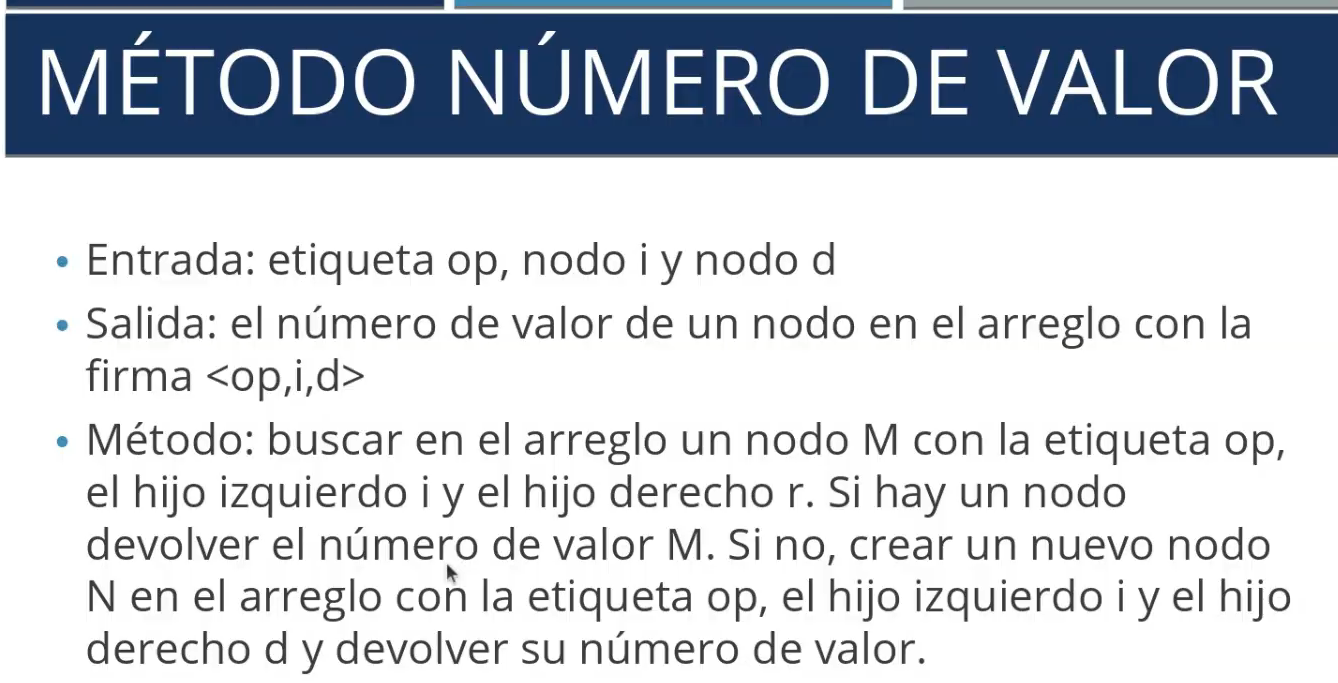
Description automatically generated



Secuencia de pasos para construir el DAG del ejemplo anterior

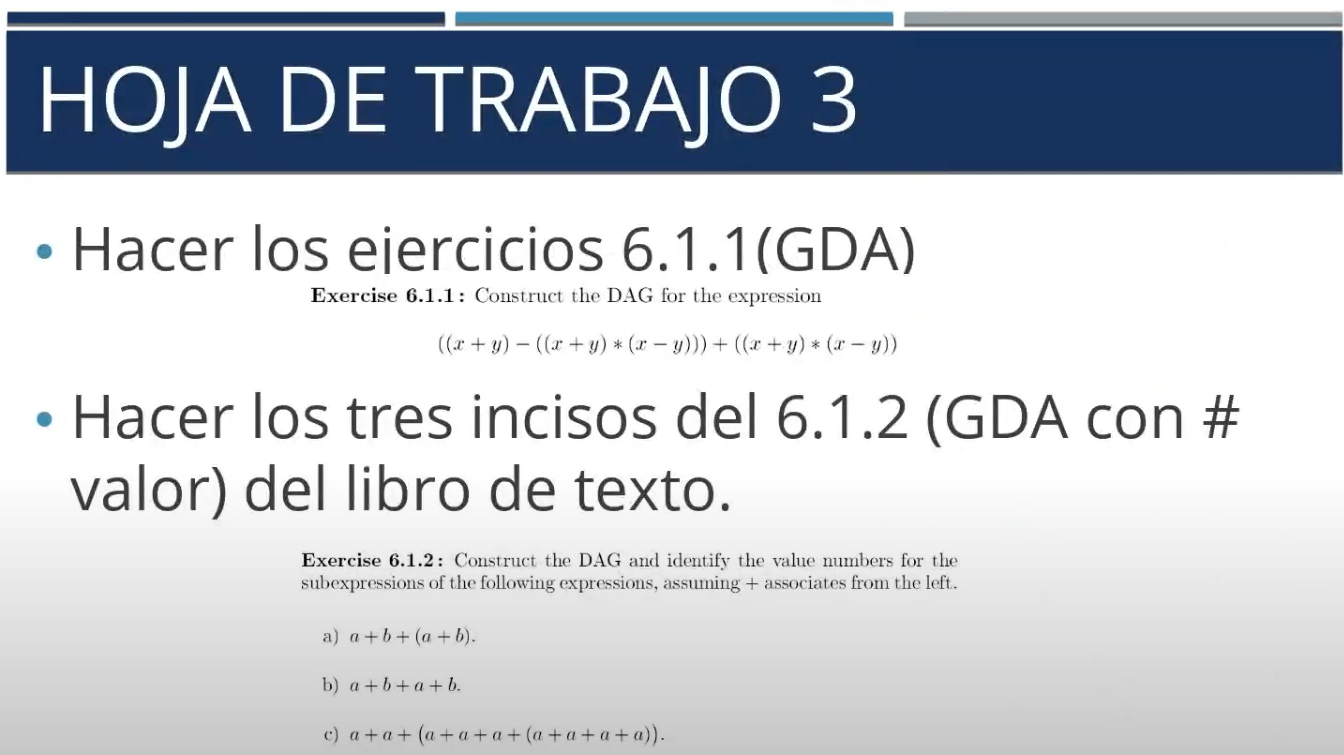






A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Recordar que solo los nodos operadores/intermediarios se imprimen, los nodos hojas no y los nodos operadores generan termporales

T1: b – c

T2: a \* T1

T3: a + T2

T4: T1 \* d

T5: T3 + T4

A diagram of a tree

Description automatically generated

T1: 3 + 5

T2: T1 + 9

T3: 6 + 8

T4: T2 \* T3

Codigo de tres direcciones:

A screenshot of a computer

Description automatically generated

A white rectangular object with black text

Description automatically generated

A white and blue box with black text

Description automatically generated

A math equations and symbols

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A white text on a blue background

Description automatically generated

Se resuelve utilizando etiquetas simbólicas, estas etiquetas se utilizan para hacer saltos, las etiquetas también se utilizan en assembler. Una etiqueta es un ID seguido de dos puntos

El rom mayor no lo hace por índices si no que lo hace por medio direcciones de memoria. El código de tres direcciones maneja el concepto de memoria en estos casos a pesar de ser conceptual porque esta inspirado en assembler ya que en assembler si se necesitan direcciones de memoria, tamaños porque para hacer corrimientos se necesita saber cuanto espacio ocupa cada una de las variables que se usan, el corrimiento que se hace a nivel de memoria se hace con operadores aritméticos. Por ejemplo teniendo un stack pointer que apunta al inicio del stack que se usa de manera temporal que para pedir memoria solo se resta al stack pointer que en este caso es una variable numérica que apunta en una dirección y cuando por ejemplo se quiere guardar un entero se tienere que restar 4 bytes al pointer y cuando ya no se va a usar solo se suma de nuevo al stack pointer

Para resolver este ejemplo hay que saber que i a alto nivel es un índice pero a bajo nivel es una dirección de memoria

En T1 se ejecuta el código que seguirá ejecutándose mientras la condición sea verdadera, se iguala i a T1, en T2 se “asume” solo para este ejemplo que el tamanio de cada uno de los valores en el arreglo mide 8 bytes y por eso se multiplica por el índice y ahora T3 accede al valor del arreglo “a” en la posición de T2(esto ya no es un índice si no una dirección de memoria) y de ultimo la condición

Por que i no se definió como i = i + 1 directamente?

Para responder hay que enfocarse en el AST

A drawing of a diagram with Ice hockey rink in the background

Description automatically generated

Hay un problema con los AST, y es que cada quien lo puede hacer como quiera ya que no hay reglas como tal,

A diagram of a diagram

Description automatically generated with medium confidence

Es para que a la hora de verlo como un árbol las instrucciones sean mas entendibles ya que los dos AST a la derecha están bien también pero pueden ser menos entendibles y pueden tener diferentes interpretaciones y bueno básicamente es porq el = crea un nodo mas al igual que el +. observación, el árbol de la izquierda puede ser levemente menos “eficiente”

A white background with black text

Description automatically generated

En lugar de etiquetas también se pueden usar números de líneas

A white and blue rectangular object with black text

Description automatically generated

Los tres toman el código de tres direcciones y lo codifican. Normalmente en Assembler se utilizan cuadruplos

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated