

Generador de Tablas de Verdad “Manual de Usuario”

PROYECTO DEL PRIMER PARCIAL

A01703455 - LUIS JESUS MORALES JUAREZ
A01250647 - ALBERTO CASTAÑEDA ARANA
A01701302 - SANTIAGO CASTELLANOS PINTO
ITESM CAMPUS QRO. | MATEMATICAS DISCRETAS

GUIA DE USUARIO

El programa es una forma sencilla de generar una tabla de valores de acuerdo a las variables y operadores que le des. Al iniciar el programa te pedirá escribir la expresión proposicional por lo que tu podrás usar cualquier carácter (excepto T,F, ya que estos se usan para los valores booleanos de verdad, y los caracteres de operadores) y junto a ellas el operador que se requiere. El formato en el que se debe ingresar es el siguiente:

(Variable 1)(Variable 2)(Variable 3)(Variable)(Operador 1)(Operador 2)(Operador 3)(Operador)

Siendo las variables las proposiciones representadas en letras, y los operadores los que unen de una variable a otra. Los operadores son los siguientes:

& -----> Conjuncion

| -----> Disyunción

! -----> Negacion

Por ejemplo, para ingresar una proposición de P conjunción Q conjunción R se ingresa lo siguiente al sistema:

PQR&&

Para negar una de las variables, se debe utilizar ! después de haber escrito la variable, ejemplo:

P!Q&

Este se interpreta como No P Conjuncion de Q.

Al confirmar, la tabla de verdad se imprimirá. Podrá ver las variables que se usaron junto a la proposición escrita. En ella se verá el valor booleano de la proposición dependiendo de los diferentes valores de cada una de las variables.

ALGORITMO

```
<!DOCTYPE html><html><head><title>TABLA DE VERDAD</title><script>
var elem,expr,vars;
function esbool(chr){return "&|!^".indexOf(chr)!=-1;}

function tabla(){
    var i,str;
    elem=document.createElement("pre");
    expr=prompt("Expresion booleana:\Ingresa characters singulares (excepto \"T\" y \"F\",
despues, con \"&|!^\".\" ).replace(/s/g,\"");
    vars=[];

    for(i=0;i<expr.length;i++)if(!esbool(expr[i])&&expr[i]!="T"&&expr[i]!="F"&&varsindex(expr[i])!=-1)
vars.push([expr[i],-1]);
    if(vars.length==0)return;
    str="";
    for(i=0;i<vars.length;i++)str+=vars[i][0]+" ";
    elem.innerHTML="<b>"+str+expr+"</b>\n";
    vars[0][1]=false;
    truthpartfor(1);
    vars[0][1]=true;
    truthpartfor(1);
    vars[0][1]=-1;
    document.body.appendChild(elem);
}

function parsebool(){
    var stack,i,idx;
    console.log(vars);
    stack=[];
    for(i=0;i<expr.length;i++){
        if(expr[i]=="T")stack.push(true);
        else if(expr[i]=="F")stack.push(false);
        else if((idx=varsindex(expr[i]))!=-1)stack.push(vars[idx][1]);
        else if(esbool(expr[i])){
            switch(expr[i]){
                case "&":stack.push(stack.pop()&stack.pop());break;
                case "|":stack.push(stack.pop()|stack.pop());break;
                case "!":stack.push(!stack.pop());break;
                case "^":stack.push(stack.pop()^stack.pop());break;
            }
        } else alert("Caracter invalido "+expr[i]+" Expresion impible.");
        console.log(stack);
    }
    return stack[0];
}
```

```

}

function varsindex(chr){
    var i;
    for(i=0;i<vars.length;i++){if(vars[i][0]==chr)return i;}
    return -1;
}

function truthpartfor(index){
    if(index==vars.length){
        var str,i;
        str="";
        for(i=0;i<index;i++){str+=(vars[i][1]?"<b>T</b>":"F")+ " ";
        elem.innerHTML+=str+(parsebool())?"<b>T</b>":"F")+ "\n";
        return;
    }
    vars[index][1]=false;
    truthpartfor(index+1);
    vars[index][1]=true;
    truthpartfor(index+1);
    vars[index][1]=-1;
}

</script></head><body onload="tabla()"></body></html>

```

DIAGRAMA IPO

