Proyecto: Demostrador Booleano

Objetivo Principal: El objetivo de este proyecto es definir en Haskell una función que dada una fórmula de lógica proposicional la clasifica en satisfacible o no satisfacible, y en el caso de que sea satisfacible que indique si la fórmula es válida o no.

Evaluación: La entrega de este trabajo es obligatoria como condición de promoción. Se tomará también en cuenta para determinar la nota final de todos los alumnos. Aquellos alumnos que completen el proyecto satisfactoriamente podrán recibir hasta **un punto** adicional en la nota final. (e.g., un promedio de 5.3 se redondearía a 6).

Entrega: La entrega se realizará al final del cuatrimestre donde cada alumno deberá entregar el código del proyecto debidamente documentado. Se deben entregar dos archivos, proyecto.hs incluyendo todo el código utilizado, y ejemplos.txt incluyendo las corridas indicadas más abajo. Para cada función implementada se debe incluir:

1) Su tipo (en el archivo proyecto-apellido.hs).

2) Una descripción en castellano de su funcionamiento (como comentario haskell en proyecto-apellido.hs).

3) Una corrida con un ejemplo de mediano tamaño (es decir, ni demasiado trivial ni demasiado largo) utilizando el formato de la materia (en corridas-apellido.txt). Se debe hacer el desarrollo de la corrida paso a paso, justificando cada línea con la definición de función correspondiente.

La entrega debe incluir además la especificación de 15 fórmulas (5 satisfacibles y válidas, 5 satisfacibles y no válidas, y 5 no satisfacibles), que deben haber sido corridas con el algoritmo implementado obteniendo los resultados esperados (en el archivo corridas-apellido.txt).

La función principal del proyecto se deberá llamar evaluar y tener el siguiente tipo

```
evaluar :: Form -> IO ()
```

Y debe recibir una fórmula utilizando el tipo de datos

```
data Form = P Int | Neg Form | And Form Form deriving (Eq, Ord, Read, Show)
```

e imprimir por pantalla los mensajes

"Es Satisfacible y Valida"

"Es Satisfacible pero no Valida"

"No es Satisfacible"

según corresponda.

Antes de implementar la funcion evalúa, se deben implementar las siguientes funciones:

1) Una función tablaVerdad :: Int -> [[Bool]], que dado un entero n genera todos los modelos Booleanos posibles con n símbolos de proposición. Esta función debe utilizarse para decidir el status de una fórmula proposicional.

2) Una función check :: Form -> [Bool] -> Bool, que dada una formula y un modelo (lista de Bool) evalúe la fórmula en el modelo y retorne True o False según corresponda.

Finalmente, se recomienda implementar las funciones esSatisfacible :: Form -> Bool y esVálida :: Form -> Bool como pasos previos a la definición de la función principal evaluar.