

Sergio Leandro Ramos Villena

```
class matriz:

    def __init__(self,row,col,valor=0):

        self.row=row

        self.col=col

        self.lista=[]

        for i in range(self.row):

            a=[]

            for j in range(self.col):

                a.append(valor)

            self.lista.append(a)


    def __str__(self):

        cad=""

        for i in range(self.row):

            for j in range(self.col):

                cad = cad +str(self.lista[i][j])+"\t"

            cad += "\n"

        return cad


    def set_valores(self,m):

        if(self.col*self.row ==len(m)):

            a=0

            for i in range(self.row):

                for j in range(self.col):

                    self.lista[i][j]=m[a]
```

```
a+=1
```

#1. Cree una nueva clase "matrizBooleana" la cual es una matriz cuadrada y hereda de la clase "matriz" e implemente su constructor. Note El constructor debe además verificar que el parámetro valor sea únicamente cero o uno.

#2. Implemente dos métodos para la clase que "matrizBooleana" que devuelvan True si las propiedades asimétrica y reflexiva se cumplen, caso contrario que devuelva False.

```
class matrizBooleana(matriz):
    def __init__(self, row, col, valor=0):
        if valor == 0 or valor == 1:
            matriz.__init__(self, row, col, valor)

    def asimetria(self):
        for i in range (self.row):
            for j in range(self.col):
                if self.lista[i][j] != self.lista[j][i]:
                    return True
            else:
                return False

    def reflexiva(self):
        for i in range (self.row):
            for j in range(self.col):
                if self.lista[i][i] == 0 or self.lista[i][i] == 1:
                    return True
            else:
                return False

x = matrizBooleana(3,3)
x.set_valores([1,2,3,4,1,6,7,8,1])
print(x)
print(x.asimetria())
print(x.reflexiva())
```