```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Aluno: Thales Carl Lavoratti
Matricula: 15100656
Data: 27/11/2019

Funções auxiliares
"""

from CoolProp.CoolProp import PropsSI

class SolutionContext:
    def __init__(self):
        self.L = []
        self.N = []
        self.T = []
        self.P = []
        self.h = []
        self.s = []
        


def getGasFraction(fluid,G,P1,P2,x_1):
     h_1 =       PropsSI('H','P',P1,'Q',x_1,fluid)
     h_liq_1   = PropsSI('H','P',P1,'Q',0.0,fluid)
     h_vap_1   = PropsSI('H','P',P1,'Q',1.0,fluid)
     h_liq_2   = PropsSI('H','P',P2,'Q',0.0,fluid)
     h_vap_2   = PropsSI('H','P',P2,'Q',1.0,fluid)
     rho_1     = PropsSI('D','P',P1,'Q',x_1,fluid)
     rho_liq_1 = PropsSI('D','P',P1,'Q',0.0,fluid)
     rho_vap_1 = PropsSI('D','P',P1,'Q',1.0,fluid)
     rho_liq_2 = PropsSI('D','P',P2,'Q',0.0,fluid)
     rho_vap_2 = PropsSI('D','P',P2,'Q',1.0,fluid)
     
     a = 0.5 * G * G * (1/rho_vap_2 - 1/rho_liq_2)**2
     b = (h_vap_2 - h_liq_2) + G * G * (1/rho_vap_2 - 1/rho_liq_2)/rho_liq_2
     c = (h_liq_2 - h_1) + 0.5 * G * G * (1/rho_liq_2)**2 - 0.5*(G/rho_1)**2
     
     x1=(-b+(b**2-4*a*c)**0.5)/(2*a)
     x2=(-b-(b**2-4*a*c)**0.5)/(2*a) 
     if x1>=0 and x1<=1:
         return x1 
     else: 
         if x2>=0 and x2<=1:
             return x2
         else:
             print('Erro: Título fora do intervalo [0,1])')
             return float('nan')

def getMu(x,fluid,P):
     mu_liq = PropsSI('VISCOSITY','P',P,'Q',0.0,fluid)
     mu_vap = PropsSI('VISCOSITY','P',P,'Q',1.0,fluid)
     return (x/mu_vap + (1-x)/mu_liq)**-1


def getRho(x,fluid,P):
    rho_liq = PropsSI('D','P',P,'Q',0.0,fluid)
    rho_vap = PropsSI('D','P',P,'Q',1.0,fluid)
    return (x/rho_vap + (1-x)/rho_liq)**-1

def getS(x,fluid,P):
    s_liq = PropsSI('S','P',P,'Q',0.0,fluid)
    s_vap = PropsSI('S','P',P,'Q',1.0,fluid)
    return s_liq +x*(s_vap - s_liq)

def getH(x,fluid,P):
    h_liq = PropsSI('H','P',P,'Q',0.0,fluid)
    h_vap = PropsSI('H','P',P,'Q',1.0,fluid)
    return h_liq +x*(h_vap - h_liq)

def getGasFractionUsingTemperature(fluid,G,T1,T2,x_1):
    h_1 =       PropsSI('H','T',T1,'Q',x_1,fluid)
    h_liq_1   = PropsSI('H','T',T1,'Q',0.0,fluid)
    h_vap_1   = PropsSI('H','T',T1,'Q',1.0,fluid)
    h_liq_2   = PropsSI('H','T',T2,'Q',0.0,fluid)
    h_vap_2   = PropsSI('H','T',T2,'Q',1.0,fluid)
    rho_liq_1 = PropsSI('D','T',T1,'Q',0.0,fluid)
    rho_vap_1 = PropsSI('D','T',T1,'Q',1.0,fluid)
    rho_liq_2 = PropsSI('D','T',T2,'Q',0.0,fluid)
    rho_vap_2 = PropsSI('D','T',T2,'Q',1.0,fluid)
    
    a = 0.5 * G * G * (1/rho_vap_2 - 1/rho_liq_2)**2
    b = (h_vap_2 - h_liq_2) + G * G * (1/rho_vap_2 - 1/rho_liq_2)/rho_liq_2
    c = (h_liq_2 - h_1) + 0.5 * G * G * (1/rho_liq_2)**2 - G/rho_liq_1
    
    x1=(-b+(b**2-4*a*c)**0.5)/(2*a)
    x2=(-b-(b**2-4*a*c)**0.5)/(2*a)
    if x1>=0 and x1<=1:
        return x1
    else:
        if x2>=0 and x2<=1:
            return x2
        else:
            print('Erro: Título fora do intervalo [0,1])')
            return float('nan')
```