

## ▼ Trabalho Prático 1

### ▼ Ex.1 : StartUp

Pretende-se construir um horário semanal para o plano de reuniões de projeto de uma “StartUp” de acordo com algumas restrições:

```
from ortools.linear_solver import pywraplp
#Variáveis S-salas, D-Dias, T-Tempos, P-Projetos
S = 2
D = 2
T = 2
P = 2
#Cada projeto tem associado o seu id, o lider, o número de reuniões e uma lista dos colabo
projetos = {1:[1,2,4],2:[2,2,5]}
# Criar a instância do solver para definir o horário
horario = pywraplp.Solver.CreateSolver('SCIP')
```

Declaramos a matriz de alocação (x) e colaboradores, como um dicionário de modo a guardar as variáveis p(ld do projeto), d(Dia), t(Tempo) e s(Sala).

```
#Criar a matriz x para atribuir projeto, dia, tempo e sala da reunião
x = {}

for p in range(1,P+1):
    x[p] = {}
    for d in range(1,D+1):
        x[p][d] = {}
        for t in range(1,T+1):
            x[p][d][t] = {}
            for s in range(1,S+1):
                x[p][d][t][s] = horario.BoolVar('x[%i][%i][%i][%i]' % (p,d,t,s))

#Criar a matriz Colab para armazenar os colaboradores associados a cada projeto
colaboradores = {}
for p in range(1,P+1):
    LC = projetos[p][2]
    colaboradores[p] = {}
    for d in range(1,D+1):
        colaboradores[p][d] = {}
        for t in range(1,T+1):
            colaboradores[p][d][t] = {}
            for c in range(1,LC+1):
                colaboradores[p][d][t][c] = {}
                for s in range(1,S+1):
                    colaboradores[p][d][t][c][s] = horario.BoolVar('x[%i][%i][%i][%i][%i]'
```

#Funções para o auxilio da resolução

```

#funções para o auxílio da resolução
def X(p,d,t,s):
    return x[p][d][t][s]
def Colab(p,d,t,c,s):
    return colaboradores[p][d][t][c][s]

#Matriz completa

```

Restrições:

O líder do projeto participa em todas as reuniões do seu projeto. Sendo que  $nr$  é o número de reuniões associado a cada projeto.

$$\forall p. \sum_{d \leq D, t \leq T, s \leq S} X_{p,d,t,s} \leq nr$$

$$\forall p. \sum_{d \leq D, t \leq T, s \leq S} Colab_{p,d,t,lider,s} \leq nr$$

```

for p in range(1,P+1):
    lider = projetos[p][0]
    nr = projetos[p][1]
    #Verificar que cada projeto se tem o número de reuniões correto
    horario.Add(sum([X(p,d,t,s) for d in range(1,D+1) for t in range(1,T+1) for s in range
    #Verificar se o lider de cada projeto está presente em todas as reuniões
    horario.Add(sum([Colab(p,d,t,lider,s) for d in range(1,D+1) for t in range(1,T+1) for

```

A sala não pode ser usada por mais de uma reunião ao mesmo tempo.

$$\forall s \leq S. \forall d \leq D. \forall t \leq T. \sum_{p=0}^P x_{p,d,t,s} \leq 1$$

```

#Cada sala não pode ter mais que uma reunião ao mesmo tempo
for d in range(1,D+1):
    for t in range(1,T+1):
        for s in range(1,S+1):
            horario.Add(sum([X(p,d,t,s) for p in range(1,P+1)])) <=1)

```

Cada colaborador não pode estar em mais de uma sala ao mesmo tempo.

$$\forall p \leq P. \forall d \leq D. \forall t \leq T. \sum_{s=0}^S X_{p,d,t,s} \leq nr$$

$$\forall p \leq P. \forall d \leq D. \forall t \leq T. \forall c \leq C. \sum_{s=0}^S Colab_{p,d,t,c,s} \leq 1$$

```

for p in range(1,P+1):
    C = projetos[p][2]
    for c in range(1,C+1):
        for d in range(1,D+1):

```

```

for t in range(1,T+1):
    #Um projeto não pode ter reuniões em mais de uma sala ao mesmo tempo
    horario.Add(sum([X(p,d,t,s) for s in range(1,S+1)]) <= nr)
    #Um colaborador não pode estar em mais de uma sala ao mesmo tempo
    horario.Add(sum([Colab(p,d,t,c,s) for s in range(1,S+1)]) <= 1)

```

Cada reunião tem de ter presente, no mínimo, 50% dos colaboradores do projeto em questão.

$$\forall_{p \leq P} \cdot \forall_{d \leq D} \cdot \forall_{t \leq T} \cdot \forall_{s \leq S} \cdot \sum_{c=0}^C Colab_{p,d,t,c,s} \geq X(p,d,t,s)$$

```

#Tem de estar no mínimo 50% dos colaboradores em cada reunião
for p in range(1,P+1):
    C = projetos[p][2]
    for d in range(1,D+1):
        for t in range(1,T+1):
            for s in range(1,S+1):
                horario.Add(X(p,d,t,s) <= sum([Colab(p,d,t,c,s) for c in range(1,C+1)]) / C)

#Invocar o solver para o horario
status = horario.Solve()
#Se possível imprimir o horário
if status == pywraplp.Solver.OPTIMAL:
    for p in range(1,P+1):
        C = projetos[p][2]
        print('Projeto Nrº:',p)
        for s in range(1,S+1):
            for d in range(1,D+1):
                for t in range(1,T+1):
                    for c in range(1,C+1):
                        if round(Colab(p,d,t,c,s).solution_value()) == 1 and round(X(p,d,t,s).solution_value()) == 1:
                            print('Colaborador:',c,'Sala:',s,'Dia:',d,'Tempo:',t)
else:
    print("oops...")

```

```

Projeto Nrº: 1
Colaborador: 1 Sala: 1 Dia: 1 Tempo: 1
Colaborador: 2 Sala: 1 Dia: 1 Tempo: 1
Colaborador: 3 Sala: 1 Dia: 1 Tempo: 1
Colaborador: 4 Sala: 1 Dia: 1 Tempo: 1
Colaborador: 1 Sala: 1 Dia: 1 Tempo: 2
Colaborador: 2 Sala: 1 Dia: 1 Tempo: 2
Colaborador: 3 Sala: 1 Dia: 1 Tempo: 2
Colaborador: 4 Sala: 1 Dia: 1 Tempo: 2
Projeto Nrº: 2
Colaborador: 1 Sala: 1 Dia: 2 Tempo: 1
Colaborador: 2 Sala: 1 Dia: 2 Tempo: 1
Colaborador: 3 Sala: 1 Dia: 2 Tempo: 1
Colaborador: 4 Sala: 1 Dia: 2 Tempo: 1
Colaborador: 5 Sala: 1 Dia: 2 Tempo: 1

```

```
Colaborador: 1 Sala: 1 Dia: 2 Tempo: 2  
Colaborador: 2 Sala: 1 Dia: 2 Tempo: 2  
Colaborador: 3 Sala: 1 Dia: 2 Tempo: 2  
Colaborador: 4 Sala: 1 Dia: 2 Tempo: 2  
Colaborador: 5 Sala: 1 Dia: 2 Tempo: 2
```

