

USO DE PROGRAMAÇÃO POR RESTRIÇÕES PARA A GERAÇÃO AUTOMÁTICA DO PLANO INDIVIDUAL DE TRABALHO DOCENTE

Instituto Federal de Mato Grosso do Sul - Campus Nova Andradina
Aluna: Larissa Sevilha de Souza
Orientador: Prof^o.Dr. Lucas Hermann Negri
Coorientadora: Dr^a.Yujuan Wang

Introdução

De forma a reduzir o trabalho manual dos docentes da instituição, propomos um método para construir de forma otimizada o plano de trabalho semanal docente por meio do paradigma de programação por restrições. O método foi implementado utilizando a biblioteca Google OR-Tools e se baseia em restrições obrigatórias e opcionais que surgem das leis trabalhistas e das preferências descritas pelo docente.

Método

- Estudo sobre o paradigma de programação por restrições;
- Estudo sobre a ferramenta Google Or-Tools para Python 3;
- Divisão do horário docente em períodos de 45 minutos;
- Definição da entrada e saída de dados com o formato JSON;
- Levantamento das regras para montagem do horário docente:

Desenvolvimento

- Separação das restrições obrigatórias (RO) e não obrigatórias (NO);
Exemplo de RO:
 - Descanso de 11 horas entre os dias de trabalho.
- Exemplo de NO:
 - Preferência de turno, caso o docente prefira ministrar as atividades em determinado período do dia.

Referencias

- BARTÁK, Roman. Constraint programming: In pursuit of the holy grail. In: Proceedings of the Week of Doctoral Students(WDS99). Prague: MatFyzPress, 1999. p. 555-564.
- CHERRI, Luiz Henrique. PROGRAMAÇÃO POR RESTRIÇÕES: UM BREVE TUTORIAL. Pesquisa Operacional para o Desenvolvimento, v. 10, n. 1, p. 1-8, 2018.
- DE SÁ, Claudio Cesar; NEGRI, Lucas Hermann; SILVA, Alexandre Gonçalves. Picat: uma Linguagem para Planejamento em Pesquisa Operacional., Anais do XLIX SBPO, 2017.

```
2     "nome": "Professor Sergio",
3     "objetivo": 170.0,
4     "atividades": [
5         {
6             "nome": "PAT- Ensino",
7             "horarios": [
8                 {
9                     "dia": 1,
10                    "horario": 0
11                },
12                {
13                    "dia": 1,
14                    "horario": 1
15                },
16                {
17                    "dia": 1,
18                    "horario": 2
19                },
20                {
21                    "dia": 1,
22                    "horario": 3
23                },
24                {
25                    "dia": 1,
26                    "horario": 4
27            ]
28        }
29    ]
30 }
```

Imagem 1:

Resultado

O programa gera o PIT de maneira automática, em um tempo de aproximadamente 10 segundos, dependendo da complexidade do problema, e mostra resultados coerentes via arquivo JSON.

***Na imagem 1 temos um exemplo de saída no formato json**