

Algoritmo de Função de Fitness

Com base nas [Premissas da Grade de Horários](#), foi definido o seguinte algoritmo para a função de *Fitness* do projeto:

```
algoritmo Funcao_Fitness(grade)
    violacoes = 0;
    para cada (horario pertence a grade) faça

        //A mesma sala não poderá ter mais de uma aula ao mesmo tempo;
        salas_ocupadas = selecionaSalasOcupadas(grade, horario);
        para cada (sala_x pertence a salas_ocupadas) faça
            conta_sala = 0;
            para cada (sala_y pertence a salas_ocupadas) faça
                se (sala_x = sala_y) então
                    conta_sala = conta_sala + 1;
            fim-se
        fim-para
        //Existe mais de uma aula em uma sala ao mesmo tempo?
        se (conta_sala > 1) então
            violacoes = violacoes + 1;
        fim-se;
    fim-para;

    // 1) A mesma aula não poderá acontecer simultaneamente em salas diferentes;
    // 2) Um professor não poderá dar 2 ou mais aulas ao mesmo tempo;
    // A premissa 2) já engloba a premissa 1) pois cada disciplina
    // tem sempre o mesmo professor

    professores_em_aula = selecionaProfessoresEmAula(grade, horario);
    para cada (professor_x pertence a professores_em_aula) faça
        conta_professor = 0;
        para cada (professor_y pertence a professores_em_aula) faça
            se (professor_x = professor_y) então
                conta_professor = conta_professor + 1;
        fim-se;
    fim-para;
    //Existe professor em duas aulas ao mesmo tempo?
    se (conta_professor > 1) então
        violacoes = violacoes + 1;
    fim-se;
    fim-para;

    // Caso a carga horária de uma aula seja maior que um 1 horário por semana,
    // é desejável que as aulas não sejam na sequência imediata;
    // Como é uma premissa desejável e não obrigatória o peso da violacao é 0.1

    se (horario < ultimo_horario_ocupado_da_grade) entao
        disciplinas_atuais = selecionaDisciplinas(grade, horario);
        proximas_disciplinas = selecionaDisciplinas(grade, horario+1);
        para cada (disciplina_x pertence a disciplinas_atuais) faça
            conta_disciplina = 0;
            para cada (disciplina_y pertence a proximas_disciplinas) faça
                se (disciplina_x = disciplina_y) então
                    conta_disciplina = conta_disciplina + 1;
            fim-se;
        fim-para;
        //Existem duas aulas da mesma disciplina seguidas
        se (conta_disciplina > 0) então
            violacoes = violacoes + 0.1*conta_disciplina;
        fim-se;
    fim-para;
    fim-se;

    // É desejável que não existam buracos (horários/salas sem aula)
    // na grade de horário
    // Considera-se o número total de salas = 5
    // Como é uma premissa desejável e não obrigatória o peso da violacao é 0.1
    se ((horario < ultimo_horario_ocupado_da_grade) e
        (salas_ocupadas < numero_total_de_salas) então
```

```
        salas_livres = numero_total_de_salas - salas_ocupadas;
        violacoes = violacoes + 0.1*salas_livres;
    fim-se;

fim-para;

// 0 fitness será :
// Melhor - quanto mais se aproximar de 1 (menos violacoes)
// Pior - quanto mais se aproximar de 0 (mais violacoes)
fitness = 1 / (1 * violacoes + 1);

fim-algoritmo;
```

Obs.: A definição do algoritmo acima não reflete necessariamente a forma como a lógica proposta por ele será implementada.