



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE ENSINO SUPERIOR DO SERIDÓ
DEPARTAMENTO DE CIÊNCIA EXATAS E APLICADAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

MARIA WESLANE DE SOUSA ALMEIDA

**UTILIZAÇÃO DE ALGORITMOS GENÉTICOS PARA MONTAGEM DE
HORÁRIOS ACADÊMICOS COM FOCO NA BLOCAGEM DE
HORÁRIOS**

Caicó – RN

2015

MARIA WESLANE DE SOUSA ALMEIDA

UTILIZAÇÃO DE ALGORITMOS GENÉTICOS PARA MONTAGEM DE HORÁRIOS ACADÊMICOS COM FOCO NA BLOCAGEM DE HORÁRIOS

Trabalho de conclusão de curso apresentado ao curso de graduação em Sistemas de Informação, como parte dos requisitos para obtenção do título de Bacharela em Sistemas de Informação da Universidade Federal do Rio Grande do Norte.

Orientador: Prof. Flavius da Luz e Gorgônio, DSc.

Caicó – RN

2015

Catálogo da Publicação na Fonte
Universidade Federal do Rio Grande do Norte - UFRN

Almeida, Maria Weslane De Sousa.

Utilização de algoritmos genéticos para montagem de horários acadêmicos com foco na blocagem de horários / Maria Weslane De Sousa Almeida. - Caicó: UFRN, 2015.

157f. il.

Monografia Bacharelado em Sistemas de Informação- Universidade Federal do Rio Grande do Norte.

Orientador: Flavius da Luz e Gorgônio.

1. Algoritmos genéticos. 2. Escalonamento de horários acadêmicos. 3. Otimização combinatória. I. Gorgônio, Flavius da Luz e. II. Título.

RN/UF/BS-CAICÓ

CDU 004.8:519.863

MARIA WESLANE DE SOUSA ALMEIDA

**UTILIZAÇÃO DE ALGORITMOS GENÉTICOS PARA MONTAGEM DE
HORÁRIOS ACADÊMICOS COM FOCO NA BLOCAGEM DE
HORÁRIOS**

Trabalho de conclusão de curso apresentado ao curso de graduação em Sistemas de Informação, como parte dos requisitos para obtenção do título de Bacharela em Sistemas de Informação da Universidade Federal do Rio Grande do Norte.

Caicó, 13 de junho de 2015.

Banca Examinadora




Prof. DSc. Flavius da Luz e Gorgônio

Orientador



Prof. MSc. Fabrício Vale de Azevedo Guerra

Examinador



Prof. MSc. João Batista Borges Neto

Examinador



Prof. DSc. João Paulo de Souza Medeiros

Examinador

À minha família pelo apoio recebido, especialmente aos meus pais, Maria das Graças e Neto Cândido, que me deram o suporte necessário para concluir o curso, e aos meus filhos, Israel e Natanael, por serem a minha motivação para chegar até aqui.

AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me ajudado a percorrer esta caminhada, estando ao meu lado, dando forças para enfrentar os obstáculos, suportar os sacrifícios e não me deixar desistir. Entre altos e baixos emocionais, conflitos internos e externos, tantas vezes pensei em desistir, mas sempre teve e tem uma voz a me dizer: "Persista... Você é mais do que imagina ser. Você pode. Você consegue".

Aos meus filhos, *Israel de Sousa Dutra* e *Natanael de Sousa Almeida e Silva*, por serem a motivação principal em recomeçar uma jornada de estudos após um longo período de tempo sem estudar, mas foi por eles que não desisti.

Aos meus pais, *Maria das Graças de Sousa Almeida* e *João Cândido de Almeida Neto*, que mesmo sem ter certeza do propósito do curso, acreditaram em mim e me deram uma nova chance de provar que poderia conquistar o tão sonhado título de graduada.

Obrigada pela educação transmitida desde a infância, o amor e dedicação à família. E, mesmo sem terem formação escolar, sem entenderem a importância do curso nos dias de hoje, me ajudaram, direta e indiretamente, não deixando abster-me da busca por novos conhecimentos, fazendo, até mesmo, o que estava além dos seus alcances, me proporcionando todo o suporte, ajuda e recursos necessários, apesar das turbulências emocionais e problemas enfrentados por mim durante todo o período do curso. Mas que de uma forma única sempre disseram que sou mais do que penso ser e que posso mais do que imagino. Obrigada por nunca terem desistido de mim.

Ao corpo docente do curso de Bacharelado em Sistemas de Informação que sempre me incentivou e acreditou que eu conseguiria. Obrigada por vocês terem um posicionamento não apenas como professores, mas também por tentarem serem amigos dos alunos (os que permitem terem um relacionamento de amizade), ajudando de acordo com a necessidade do aluno, relacionada não apenas às dúvidas nas disciplinas, mas também dando conselhos, incentivando a sermos profissionais qualificados. A vocês, considero-os não apenas como educadores, mas como amigos.

Entretanto, alguns dos docentes proporcionaram maior impacto na minha vida acadêmica, os quais serão especificados a seguir:

Ao meu orientador, *Flavius da Luz e Gorgônio*, por sempre ter me incentivado a continuar no curso. Obrigada pelo papel de educador, psicólogo e amigo, pela confiança demonstrada e orientações dadas a mim ao decorrer não apenas no desenvolvimento deste trabalho, mas durante todo o curso, e por ter acreditado que eu seria capaz de realizar este trabalho e de acreditar que eu tenho potencial de ir além.

Ao professor *João Paulo de Souza Medeiros* não apenas pelas importantes considerações e contribuições dadas a este trabalho, mas também pela transmissão de conhecimento nas disciplinas que ministrou, pela paciência ao sanar dúvidas e auxiliar nas dificuldades encontradas em trabalhos realizados durante o curso. Obrigada por tudo, em especial, por ser um referencial em que me proporciona admiração e inspiração. Ademais, especificamente sobre o presente trabalho, considero-o digno do mérito equivalente ao dado ao orientador.

Ao professor *João Batista Borges Neto* que, por ser um professor exemplar, mesmo de forma indireta, ajudou a me identificar com o curso por meio de uma técnica própria de transmitir conhecimento. Obrigada pela prontidão em ajudar até mesmo quando não era tarefa sua, pela paciência e peculiaridade em explicar os conteúdos de uma forma lúdica e envolvente, resultando numa motivação a continuar mesmo com as dificuldades a serem enfrentadas.

Ao professor *Fabício Vale de Azevedo Guerra* por ter transmitido o espírito de competitividade em equipe, ao mesmo tempo em que incentivava a despertar um caráter investigativo na busca de novos conhecimentos e pela prontidão em ajudar sempre que possível.

À família LABICAN: *Jaqueline Santos, Sâmia Lorena, Narallynne Maciel, Olívia Emanuelle, Pablo Lopes e Hyago Brendoll*; aos amigos: *Graziele Almeida* (uma irmã que a vida me deu), *Jane Cristine, Marcos Túlio, Maycon Jebson, Alexandre Diniz*; e familiares: *Wellington Almeida, Wesley Almeida, Silvana Almeida, Janne Kelly, Diomédio Sousa, Sara Hernandez e Mabel Almeida*, que ajudaram de forma direta e indireta a concluir este trabalho.

Por fim, agradeço também a *Flávio Medeiros de Azevedo* pelo suporte psicológico e por me incentivar a focar na conclusão deste trabalho a fim de dar continuidade na carreira acadêmica e conquista de novos títulos.

De uma forma geral, agradeço a todos pelo apoio, ajuda e dedicação que proporcionaram a mim, através de grandes e pequenos atos que foram feitos ao

meu favor, os eternizaram em mim. Alguns podem até o grau de importância maior (filhos e pais), mas cada um será lembrando por onde eu estiver e enquanto viver.

Que você capacite seu Eu para ser autor de sua história
e gerenciar sua mente.

Se treinar, não tenha medo de falhar.

E se falhar, não tenha medo de chorar.

E se chorar, corrija suas rotas, mas não desista.

Dê sempre uma nova chance para si e para quem ama.

Só adquire maturidade quem usa suas frustrações para
alcançá-la.

Augusto Cury.

RESUMO

A montagem de uma estrutura de horário acadêmico é uma das tarefas mais árduas de planejamento escolar para coordenadores e equipe pedagógica. Em uma proposta de horário bem estruturado é necessário investir tempo e trabalho árduo, devido aos vários fatores envolvidos: a disponibilidade dos professores, disciplinas e suas cargas horárias, alunos, sala. Além disto, cada fator possui restrições associadas a ele, como por exemplo, um professor não pode estar presente em aulas diferentes ao mesmo tempo. A dificuldade relacionada à montagem de uma estrutura de horário é conhecida como o problema de escalonamento de horários. Com base nos trabalhos relacionados é possível afirmar que na maioria das instituições, a solução adotada para esse problema é realizada de forma manual, o que requer muito tempo e esforço por parte do coordenador do curso. O espaço de busca a ser explorado varia de acordo com a quantidade de períodos do curso e a quantidade de disciplinas a serem ofertadas no semestre a ser considerado. Para o problema abordado neste trabalho, existem formas de otimizar a busca por soluções por meio da aplicação de heurísticas de busca que possibilitam encontrar uma solução quase ótima para o problema. É possível encontrar várias propostas de solução usando este tipo de técnica, entretanto elas oferecem soluções visando fatores administrativos, ou seja, consideram apenas os fatores de disciplina, sala e professor, não levando em consideração o corpo discente. Este trabalho difere dos demais por ter como objetivo propor uma solução utilizando algoritmos genéticos que encontre uma solução aceitável para o problema de escalonamento de horários que leve em consideração o corpo discente quanto à blocagem de horários, a fim de favorecer rendimento acadêmico do aluno.

Palavras-chaves: Escalonamento de horários acadêmicos. Otimização combinatória. Algoritmos genéticos.

ABSTRACT

Assembling of an academic timetable structure is one of hardest scholars planning works, in special to the coordinator and pedagogical staff. A proposed schedule well done requires time and hard work, because of some factors involved: professor availability, courses and their workload, students, classrooms. Besides that, each factor has associated a set of restrictions, for example, a professor can not be in two different classes at the same time. The difficulty of this is known as timetable or timetabling problem. Based on related researches, it is possible to claim that in the most of the institutions, the solution is done by manual work, which requires more time and effort from the coordinator course. Because it is a high complexity problem, the search space varies according to the number of periods and the number of disciplines that will be offered in the semester to be considered. To the problem addressed in this work, it has many ways to optimize the search solution by applying search heuristic techniques that make possible to find a *quasi* best solution to the problem. It is possible to find many solution proposals using this kind of technique. However they provide a solution aiming administrative factors, in other words, they consider only the factors related to the disciplines, classrooms and professors, not considering the students. This work differs from the others by having as objective to propose a solution based in genetic algorithms that finds an acceptable solution to the timetable problem which consider the students about the hours blocking, in order to favor academic performance of student.

Keywords: Academic timetable problem. Combinatorial optimization. Genetic algorithms.

LISTA DE FIGURAS

Figura 1: Exemplo de uma estrutura de horário	26
Figura 1: Cruzamento de cromossomos biológicos	31
Figura 2: Cruzamento de cromossomos artificiais	31
Figura 3: Mutação.....	31
Figura 4: Fluxograma de um GA.....	33
Figura 5: Exemplo de cruzamento em um ponto.....	37
Figura 6: Exemplo de cruzamento em dois pontos.....	37
Figura 7: Cruzamento uniforme	38
Figura 8: Cruzamento em maioria.....	38
Figura 9: Cruzamento em ordem	39
Figura 10: Escolha do esquema dominante	40
Figura 9: Mutação em ordem.....	41
Figura 12: Representação de uma turma	48
Figura 13: Matriz de horário por período	49
Figura 14: Representação cromossômica	50
Figura 15: Estrutura geral do cromossomo	50
Figura 16: Exemplo da representação de uma turma	51
Figura 17: Ilustração da representação geral do cromossomo	52
Figura 18: Matriz de seleção / máscara	53
Figura 19: Simulação do cruzamento em ordem.....	54
Figura 20: Mutação – correção de anomalias	55
Figura 21: Mutação – choque de horário	56
Figura 22: Valores médios de aptidão das populações – 2012.1	62
Figura 23: Valores médios de aptidão das gerações – 2012.1	63
Figura 24: Valores médios de aptidão das populações – 2012.2	64
Figura 25: Valores médios de aptidão das gerações – 2012.2	64
Figura 26: Valores médios de aptidão das populações – 2013.1	65
Figura 27: Valores médios de aptidão das Gerações – 2013.1	65
Figura 28: Valores médios de aptidão das populações – 2013.2.....	66
Figura 29: Valores médios de aptidão das gerações – 2013.2	66
Figura 30: Valores médios de aptidão das populações – 2014.1	67
Figura 31: Valores médios de aptidão das gerações – 2014.1	67

Figura 32: Valores médios de aptidão das populações – 2014.2	68
Figura 33: Média das aptidões por gerações – 2014.2.....	69
Figura 34: Valores médios de aptidão das populações – 2015.1	69
Figura 35: Valores médios de aptidão das gerações – 2015.1	70
Figura 36: Valores médios de aptidão das populações – 2015.2.....	71
Figura 37: Média ao longo das gerações – 2015.2	71

LISTA DE TABELAS

Tabela 1: Esquemas.....	35
Tabela 2: Comparação das aptidões das soluções.....	72
Tabela 3: Valores médios de aptidão das populações – 2012.1	136
Tabela 4: Valores médios de aptidão das gerações – 2012.1.....	136
Tabela 5: Valores médios de aptidão das populações – 2012.2	136
Tabela 6: Valores médios de aptidão das gerações – 2012.2.....	137
Tabela 7: Valores médios de aptidão das populações – 2013.1	137
Tabela 8: Valores médios de aptidão das gerações – 2013.....	1 137
Tabela 9: Valores médios de aptidão das populações – 2013.2	137
Tabela 10: Valores médios de aptidão das gerações – 2013.2.....	138
Tabela 11: Valores médios de aptidão das populações – 2014.1.....	138
Tabela 12: Valores médios de aptidão das gerações – 2014.1.....	138
Tabela 13: Valores médios de Aptidão das populações – 2014.2.....	139
Tabela 14: Valores médios de aptidão das gerações – 2014.2.....	139
Tabela 15: Valores médios de Aptidão das populações – 2015.1	139
Tabela 16: Valores médios de aptidão das gerações – 2015.1.....	140
Tabela 17: Valores médios de Aptidão das populações – 2015.2.....	140
Tabela 18: Valores médios de aptidão das gerações – 2015.2.....	140

LISTA DE ABREVIATURAS

AEX	Algoritmos Experimentais
ALG	Algoritmo e Lógica de Programação
ALGL	Álgebra Linear
APM	Aprendizado de Máquina
ARQ	Arquitetura de Computadores
ASR	Administração de Servidores de Redes
BD	Banco de Dados
BSI	Bacharelado em Sistemas de Informação
CAL	Cálculo Diferencial e Integral
CEC	Contabilidade e Custo
CLOUD	<i>Cloud Computing</i>
DIR	Direito e Legislação Social
ED	Estrutura de Dados
EMP	Empreendedorismo em Informática
ES1	Engenharia de <i>Software</i> I
ES2	Engenharia de <i>Software</i> II
ETIC	Ética
FDLD	Fundamentos do Direito e Legislação Digital
FIL	Filosofia
FMAT	Fundamentos de Matemática
FSI	Fundamentos de Sistemas de Informação
GA	<i>Genetic Algorithm</i>
GPS	Gestão de Projeto de <i>Software</i>
IHC	Interação Humano Computador
II	Introdução à Informática
ING	Inglês Técnico
LOG	Lógica
LPT	Leitura e Produção de Texto
MATF	Matemática Financeira
MICRO	Microcontroladores
MIN	Mineração de Dados
MTC	Metodologia do Trabalho Científico
OSM	Organização de Sistemas e Métodos
PABD	Projeto e Administração de Banco de Dados
PDM	Programação para Dispositivos Móveis
EST	Probabilidade e Estatística
POO1/POO2	Programação Orientada a Objetos I / II
PROG	Programação
PV	Programação Visual
PWEB	Programação Web
RED	Redes de Computadores
SAD	Sistemas de Apoio à Decisão
SEG	Tópicos Especiais em Segurança da Informação
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
SO	Sistemas Operacionais
TAP	Tópicos Avançados em Programação
TEES	Tópicos Especiais em Engenharia de <i>Software</i>

TES	Teste de <i>Software</i>
TESBD	Tópicos Especiais em Banco de Dados
TGA	Teoria Geral da Administração
TGS	Teoria Geral dos Sistemas
TSI	Tópicos Especiais em Sistemas de Informação
UFRN	Universidade Federal do Rio Grande do Norte

SUMÁRIO

1.....	INTRODUÇÃO	18
1.1.	Tema	20
1.2.	Contextualização e Problema	20
1.3.	Objetivos da Pesquisa.....	22
1.3.1.....	Objetivo Geral	22
1.3.2.....	Objetivos Específicos	22
1.4.	Delimitação do Estudo	22
1.5.	Motivação e Justificativa	23
1.6.	Apresentação do trabalho	24
2.....	REVISÃO DA LITERATURA	26
2.1.	Escalonamento de Horários	26
2.2.	Algoritmos Genéticos	27
2.2.1.....	História	28
2.2.2.....	Conceitos	29
2.2.3.....	Composição dos GAs.....	32
2.2.4.....	Representação Cromossômica	34
2.2.5.....	Operadores Genéticos	35
2.2.6.....	Parâmetros Genéticos.....	36
2.2.7.....	Tipos de Cruzamento	37
2.2.8.....	Tipos de Mutação.....	39
2.3.	Trabalhos Relacionados.....	41
3.....	METODOLOGIA E ALGORITMO PROPOSTO.....	47
3.1.	Representação Cromossômica Utilizada	48
3.2.	Operadores Genéticos Utilizados.....	52
3.3.	Função de Avaliação.....	57
3.4.	Implementação.....	59
4.....	RESULTADOS E CONCLUSÕES	61
4.1	Geração de Horário para o Semestre 2012.1	62
4.2	Geração de Horário para o Semestre 2012.2	63
4.3	Geração de Horário para o Semestre 2013.1	64
4.4	Geração de Horário para o Semestre 2013.2	66
4.5	Geração de Horário para o Semestre 2014.1	67

4.6	Geração de Horário para o Semestre 2014.2	68
4.7	Geração de Horário para o Semestre 2015.1	69
4.8	Geração de Horário para o Semestre 2015.2	70
4.9	Avaliação.....	71
REFERÊNCIAS.....		75
APÊNDICES		80
APÊNDICE A – Outras Técnicas de Busca.....		81
APÊNDICE B – Horários Propostos pelo Algoritmo		87
APÊNDICE C – Métodos para Geração de um Indivíduo e Operadores Genéticos Utilizados.....		127
APÊNDICE D – Valores de Aptidão Obtidos pelo Algoritmo		135
ANEXOS		141
ANEXO A - Estrutura Curricular do Curso de Bacharelado em Sistemas de Informação		142
ANEXO B - Montagens de Horários Realizadas nos Semestres Anteriores de Forma Manual		147
ANEXO C – Horários Sugeridos pelo <i>Software Asc Timetable</i>		156

1. INTRODUÇÃO

O problema de escalonamento de horários, também conhecido como *timetable problem*, é uma das mais importantes atividades de planejamento do calendário escolar. A necessidade e as dificuldades associadas à geração de uma estrutura de horário escolar o tornaram um problema clássico que vem sendo estudado há muito tempo, podendo ser encontrados trabalhos sobre este tema desde a década de 1960 (LOBO, 2005).

O problema de escalonamento de horários consiste em fixar, em um determinado período de tempo predefinido, um conjunto de aulas, dispostas na forma de uma tabela de horários, que atenda às exigências acadêmicas estabelecidas por certo currículo de estudos para um determinado grupo de disciplinas (GOTTILIEB, 1962).

Tomando como base a definição citada anteriormente, a elaboração de um escalonamento de horários para um curso de nível superior deve considerar diversos fatores que estão diretamente ligados a esta proposta de horário. Estes fatores são: disponibilidade dos horários dos professores, quantidade de disciplinas, alunos, turma, salas, frequência (quanto à quantidade de aulas por semana de uma determinada disciplina) e horários das aulas, etecetera (SILVA, 2014).

Tomando como exemplo os cursos oferecidos no Centro de Ensino Superior do Seridó (CERES), e mais especificamente o curso de Bacharelado em Sistemas de Informação (BSI) da Universidade Federal do Rio Grande do Norte (UFRN), o método mais utilizado para a elaboração de horários é a montagem manual através de tentativa e erro, onde o coordenador do curso elabora manualmente uma proposta e a disponibiliza aos professores e alunos, a fim de obter aprovação ou sugestões sobre a proposta que mais se adeque às partes interessadas. Essa atividade costuma ser uma tarefa repetitiva, árdua e exaustiva, uma vez que o responsável pela mesma costuma ter o intuito de atender aos anseios do maior número de pessoas possível durante a montagem de horário. Ademais, a este conjunto de recursos está associado um conjunto de restrições, o que implica no aumento da complexidade do problema.

Esta atividade, sendo realizada de forma manual, requer tempo e esforço extra por parte do coordenador, até que se obtenha uma solução adequada. Como forma de diminuir a dificuldade e o esforço humano no planejamento de uma

estrutura de horário, surgiram tentativas de otimizar esta atividade através do uso de algoritmos, a fim de diminuir o tempo gasto para realizá-la.

Por se tratar de um problema onde o aumento da quantidade de variáveis envolvidas na alocação (turmas, professores, disciplinas, etcetera) está diretamente relacionado ao aumento da complexidade do problema em uma escala maior, cujo crescimento da complexidade se dá de forma exponencial na busca pela melhor solução, o tempo de execução da busca pela melhor solução cresce rapidamente, a ponto de tornar-se inviável a busca por uma solução ótima em um espaço de tempo aceitável.

Tendo em vista o cenário descrito acima e os fatores que estão associados a este tipo de problema, o problema de decisão associado ao escalonamento de horários pertence a uma categoria de problemas denominada NP-Completo, cuja principal característica é não ter encontrado um algoritmo determinístico que seja capaz de encontrar uma solução ótima em tempo polinomial. Cooper e Kingston (1996) provaram que o escalonamento de horários trata-se de um problema NP-Completo. Dentre os caminhos utilizados como prova desta afirmação, uma das estratégias utilizadas foi quanto à capacidade de em um problema que já tenha sido provado ser NP-Completo ser convertido no problema de decisão associado à montagem de horários.

Neste caso o problema de montagem de horários existe um conjunto R de restrições, por exemplo, intervalos de aulas vagos. Para este exemplo, existe um número x de blocos (posição da matriz) que devem ser preenchidos por y turmas que minimize os intervalos entre eles. Onde x é igual ao tamanho da matriz do horário de determinado período e y é a quantidade de aulas (bloco) das turmas que devem ser alocadas no período.

Ao se deparar com esse tipo de problema, os profissionais da área costumam utilizar heurísticas. Uma heurística consiste em uma estratégia de busca por uma solução para o problema em tempo hábil, a fim de obter uma solução aceitável para o problema de forma mais rápida e eficaz, em comparação com a solução que seria encontrada se a montagem da estrutura de horários fosse realizada de forma manual, buscando a melhor solução dentre as possíveis soluções.

Os algoritmos genéticos (GA, do inglês *Genetic Algorithms*) são um tipo de algoritmo evolutivo que usa uma técnica de busca heurística para encontrar uma

solução em tempo aceitável. Os algoritmos evolutivos fazem analogia à teoria da evolução de Charles Darwin (DARWIN, 1896) e à genética (CASTRO, 2010). Segundo Linden (2012), os GA são técnicas de busca extremamente eficientes no seu objetivo de explorar o espaço de soluções e encontrar soluções próximas da ótima.

Como proposta para encontrar uma solução aceitável, este trabalho investiga a utilização de algoritmos genéticos a fim de encontrar uma solução viável para o problema de escalonamento de horários acadêmicos, respeitando as limitações e as restrições associadas a este problema, além de ter como foco a blocagem de horários (alocações de aulas igualitárias) e sua consequente influência no favorecimento do rendimento acadêmico do corpo discente (GARCIA *et al.* 2014). Devido às diversas estratégias existentes para resolver o problema de escalonamento, este trabalho também enumera outras técnicas igualmente utilizadas em trabalhos semelhantes e que serão apresentadas ao longo do texto para fins de comparação.

1.1. Tema

Montagem automática de horários acadêmicos através da utilização de algoritmos genéticos, com foco na otimização das atividades, visando à blocagem de horários e contribuindo para a melhoria do rendimento acadêmico do corpo discente.

1.2. Contextualização e Problema

Estrutura de horários é um termo comumente utilizado para se referir à construção de uma tabela de horários que atenda a um conjunto de restrições. Nesta tabela está contido o escalonamento de horários de acordo com o tempo e os eventos que ocorrem neste intervalo de tempo.

Problemas de escalonamento de horários são identificados em várias situações, como por exemplo, no escalonamento de voos de uma companhia aérea, no escalonamento de funcionários de uma determinada empresa ou no escalonamento de horários de aulas escolares, sejam de escola secular (nível fundamental e médio) ou acadêmica (nível universitário).

O escalonamento de horários é uma tarefa de fundamental importância para uma instituição de ensino. Ela envolve o agendamento adequado das aulas, o que requer uma boa administração dos horários. Entretanto, uma estrutura de horário escolar não implica apenas no escalonamento de horários em uma disposição qualquer, sua proposta é oferecer uma distribuição de horários adequada, implicando assim na organização sobre a disposição das aulas.

No ambiente acadêmico universitário, durante o período de planejamento que ocorre a cada final de semestre, os coordenadores de cursos e a equipe pedagógica se deparam com o mesmo problema, montar uma estrutura de horários que atenda às necessidades do semestre letivo seguinte. Este problema consiste em selecionar as disciplinas que serão oferecidas e atribuir horários às mesmas, levando em consideração a disponibilidade do professor que a leciona, a quantidade de aulas que cada disciplina deve ser administrada durante a semana e a disponibilidade dos alunos, evitando que o mesmo participante, seja aluno ou professor, esteja alocado em aulas diferentes simultaneamente.

Devida à complexidade associada a esta atividade, a dificuldade que os coordenadores se deparam em encontrar uma solução que esteja adequada às partes interessadas, em via de regra, essas propostas podem ocorrer de possuírem aulas vagas que acabam prejudicando o desempenho dos alunos.

Como forma de otimizar a montagem de horários, técnicas computacionais não determinísticas, baseadas em heurísticas, normalmente são utilizadas para encontrar uma solução para a montagem de estruturas de horários. Os conceitos dessas técnicas podem ser encontrados no APÊNDICE A e pesquisas realizadas nesta área estão descritos nos trabalhos relacionados (Seção 2.3).

O problema abordado neste trabalho é a blocagem de horários através da utilização de uma solução heurística, baseada em algoritmos genéticos, que ofereça suporte à tarefa de criação de uma estrutura de horários em cursos universitários, utilizando uma função de avaliação que priorize a blocagem de horários, evitando a ocorrência de aulas vagas ou intervalos entre aulas para o discente, a fim de favorecer o seu rendimento acadêmico (GARCIA *et al.* 2014).

1.3. Objetivos da Pesquisa

Os objetivos deste trabalho se dividem em objetivo geral e objetivos específicos.

1.3.1. Objetivo Geral

O presente trabalho tem como objetivo geral propor uma solução algorítmica, baseada na utilização de algoritmos genéticos, que auxilie o coordenador do curso na montagem de horários acadêmicos, fixando o foco em impossibilitar a ocorrência de conflitos de horários e uma bloqueio de aulas, a fim de favorecer o rendimento acadêmico do corpo discente.

1.3.2. Objetivos Específicos

- a) Definir uma representação cromossômica que possibilite a descrição de horários acadêmicos;
- b) Definir uma função de avaliação que priorize a bloqueio de horários, evitando aulas vagas e favorecendo o rendimento acadêmico do discente;
- c) Selecionar, dentre os operadores genéticos disponíveis, quais os mais adequados para a tarefa de otimização do horário;
- d) Desenvolver um algoritmo genético que aperfeiçoe a oferta de componentes curriculares, evitando os conflitos de horário, as ocorrências de aulas vagas e/ou intervalos entre aulas;
- e) Avaliar a solução desenvolvida, comparando-a com os métodos convencionais de montagem de horários de forma manual.

1.4. Delimitação do Estudo

De acordo com a sua definição de escalonamento de horários, uma estrutura de horário deve estar de acordo com uma proposta curricular de estudos. Embora o problema de escalonamento de horários seja comum às instituições de ensino, o presente trabalho limita-se a considerar unicamente o curso de Bacharelado em Sistemas de Informação (BSI) da Universidade Federal do Rio

Grande do Norte (UFRN), devido ao fato deste curso fornecer os recursos e restrições necessárias para que a proposta possa ser testada e aplicada.

O curso de BSI possui um corpo docente constituído por dezoito professores (incluindo os docentes da área e de formação complementar) e um corpo discente em torno de 200 alunos (SILVA, 2014). A matrícula é realizada semestralmente pelo aluno nas disciplinas desejadas desde que obedeçam aos seus pré-requisitos. Por definir como foco o rendimento acadêmico do aluno, o presente trabalho não levará em consideração as restrições associadas à alocação de salas/laboratórios e nem o limite de vagas por disciplina, pois o acréscimo dos mesmos aumentaria ainda mais a complexidade do algoritmo.

1.5. Motivação e Justificativa

Por se tratar de um curso na área de exatas, o número de alunos reprovados nas disciplinas básicas do curso, tais como programação e cálculo, chega a 60%, (CAMPOS, 2010; PRIETCH e PAZETO, 2010). Esse fato é investigado em pesquisas realizadas nesta área, que indicam que a maioria dos alunos que desiste do curso possui lacunas relacionadas ao cálculo adquiridas na educação básica (GARCIA *et al.*, 2014). O alto número de reprovados agrega ainda mais dificuldade em gerar uma estrutura de horário por parte do coordenador do curso, pois há inevitavelmente a necessidade de oferta de componentes curriculares para alunos repetentes, tornando o problema do escalonamento de horários, ainda mais crítico.

Conforme citado anteriormente, a atividade de planejamento do horário é complexa e requer muito tempo para ser concluída se realizada de forma manual. Além disso, o número de diferentes possíveis soluções para um horário é muito elevada, dada a natureza do problema, o que impede a análise de todas as soluções possíveis a fim de encontrar a melhor solução. Por fim, pequenos ajustes, por vezes necessários, implicam em grandes alterações na solução proposta, exigindo que boa parte do trabalho seja refeito.

Ademais, as soluções adotadas atualmente, sejam essas manuais ou informatizadas, não levam em consideração a otimização do horário quanto à blocagem de horários visando às características do corpo discente. Desta forma, é possível que a blocagem de horários tenha falhas, ou seja, intervalos longos entre

aulas, o que pode influenciar no desempenho acadêmico do corpo discente, por causar uma possível desmotivação aos alunos ocasionada pelo fato de ficarem ociosos durante um grande intervalo de tempo (equivalente há uma hora-aula ou mais), poderia em contratempo que os alunos estudassem na biblioteca, entretanto ao considerar o curso de BSI e verificar o comportamento do corpo discente, é possível afirmar que este intervalo não é usufruído para estudos por grande parte dos alunos.

Estes motivos justificam a realização deste trabalho, a fim de propor uma solução para este problema, tendo em vista a sua complexidade, o espaço de busca a ser explorado varia de acordo com a quantidade de períodos do curso e a quantidade de disciplinas a ser ofertada no semestre a ser considerado e a necessidade do coordenador em simular alterações nos horários de uma forma mais rápida e eficaz.

Também é possível justificar o uso de algoritmos genéticos, com base nos trabalhos relacionados (Seção 2.3), mais precisamente no trabalho de Coloni, *et al.*(1993), onde os autores fazem comparação entre técnicas de busca e através dos experimentos identificaram que os algoritmos genéticos teve melhor desempenho em relação a outras técnicas de busca, quanto a encontrar uma solução aceitável para esse tipo de problema.

1.6. Apresentação do trabalho

O presente trabalho está estruturado em quatro capítulos: introdução, revisão da literatura, algoritmo proposto e resultados e conclusões. No primeiro capítulo são introduzidas ideias gerais sobre a pesquisa, seus objetivos, delimitação do estudo, motivação e justificativa do mesmo.

O segundo capítulo apresenta os conceitos relacionados ao problema abordado neste trabalho (escalonamento de horários) e os algoritmos genéticos. Quanto aos algoritmos genéticos, além do conceito, também estão dispostos uma breve descrição cronológica de como estes surgiram, sua composição, as possíveis representações que o cromossomo venha a ter, os operadores e parâmetros genéticos e, mais detalhadamente, estão descritos os tipos de cruzamento e mutação. Além disso, ao final do capítulo, são apresentados os trabalhos relacionados.

O terceiro capítulo aborda o algoritmo proposto como solução para o problema de escalonamento de horários de acordo com os objetivos do presente trabalho. Primeiramente foram apresentadas as ideias e adequações do problema aos objetivos do trabalho. Em seguida, é explanada a representação cromossômica e os operadores genéticos que foram utilizados. Ademais, é descrito ainda, como foi realizada a função de avaliação (aptidão) dos indivíduos e posteriormente a implementação do algoritmo.

Por fim, no quarto e último capítulo estão descritos os resultados obtidos durante a pesquisa e suas devidas considerações, além das conclusões e algumas sugestões de trabalhos futuros a serem incorporados ao algoritmo.

2. REVISÃO DA LITERATURA

A presente seção dedica-se a apresentar mais detalhadamente o problema de escalonamento de horários, especificando a técnica e conceito quanto o funcionamento dos algoritmos genéticos. Por fim, são apresentadas referências à pesquisas já realizadas e publicadas, de diversos autores, situando a evolução do assunto e, assim, dando sustentação ao tema, a conceitos e a trabalhos já realizados na área. Esses trabalhos também dão sustentação quanto ao melhor desempenho dos algoritmos genéticos em relação às outras técnicas utilizadas.

2.1. Escalonamento de Horários

Escalonamento de horários é o termo usado pela área da pesquisa operacional para definir problemas de alocação de horários. Em outras palavras, é uma tabela que mostra o escalonamento atividades ou tarefas de um determinado evento, indicando a data (ou período) e o intervalo de tempo em que cada atividade deverá acontecer como mostra o Figura 1.

Figura 1: Exemplo de uma estrutura de horário

Horários	Seg	Ter	Qua	Qui	Sex	Sáb
07:00 - 07:50	---	---	---	---	---	---
07:50 - 08:40	---	---	BSI1203	---	---	---
08:55 - 09:45	---	---	BSI1203	---	---	---
09:45 - 10:35	BSI2203	---	BSI2203	BSI1203	---	---
10:50 - 11:40	BSI2203	---	BSI2203	BSI1203	---	---
11:40 - 12:30	---	---	---	---	---	---
13:00 - 13:50	---	BSI1110	---	---	---	---
13:50 - 14:40	---	BSI1110	---	---	---	---
14:55 - 15:45	BSI2103	---	---	---	---	---
15:45 - 16:35	BSI2103	---	---	---	---	---
16:50 - 17:40	---	BSI2103	---	BSI1110	---	---
17:40 - 18:30	---	BSI2103	---	BSI1110	---	---
18:45 - 19:35	---	---	---	---	---	---
19:35 - 20:25	---	---	---	---	---	---
20:35 - 21:25	---	---	---	---	---	---
21:25 - 22:15	---	---	---	---	---	---

Fonte: SIGAA, 2014.

Este problema é alvo de estudos devido a sua complexidade e o grande número de variáveis, dado o número de possibilidades que devem ser avaliadas

para se verificar qual a melhor solução (ou solução ótima). No caso do problema de escalonamento de horários escolares, para realizar a montagem do horário deve ser levado em consideração os horários, a disponibilidade dos professores, as disciplinas a serem oferecidas, a quantidade de aulas de cada disciplina, as salas de aula disponíveis e os horários dos alunos. Entretanto, não é simplesmente considerar os recursos citados acima. Cada recurso agrega restrições, como por exemplo: nem todos os professores estão disponíveis em tempo integral, há restrições de disponibilidade de uso em algumas salas de aula, como no caso de laboratórios, há problemas de conflito de horários, etcetera.

Para resolver este problema, o gestor escolar deve propor uma solução que atenda total ou parcialmente aos recursos e as restrições referentes a cada recurso, a fim de otimizar o processo de escalonamento de horários e diminuir o número de conflitos existentes.

2.2. Algoritmos Genéticos

A inteligência artificial é uma subárea da Ciência da Computação que envolve a utilização de métodos e técnicas que conseguem, a partir da inspiração na inteligência humana e de outros animais, solucionar problemas complexos. Hoje em dia, há debates sobre a diferença dos conceitos de inteligência artificial e inteligência computacional. A inteligência artificial é a ciência que tenta compreender e emular a inteligência humana como um todo, enquanto que a inteligência computacional procura desenvolver sistemas que tenham comportamento similar a certos aspectos do comportamento inteligente (LINDEN, 2012).

Castro (2010) afirma que “a natureza também pode servir de inspiração para a computação, a computação pode ser utilizada para entender melhor a natureza e a própria natureza pode ser usada para computar”. Dessa forma, inúmeras técnicas computacionais foram criadas a partir da observação de fenômenos da natureza, estando agrupadas sob o rótulo de Computação Bioinspirada.

Dentre essas técnicas inspiradas no comportamento da natureza, estão os algoritmos genéticos, os quais são uma técnica de busca heurística que é baseada na teoria da evolução e seleção natural de Charles Darwin, aliada aos

conceitos da genética (cruzamento, mutação), os quais serão abordados mais detalhadamente nas seções seguintes.

2.2.1. História

A história dos algoritmos genéticos se inicia da década de 1940, quando biólogos e matemáticos de importantes centros de pesquisa influenciados pelas ideias de Charles Darwin e Wallace começam a tentar se inspirar na natureza para criar o ramo da inteligência artificial.

Segundo Bateson (1894), a teoria da evolução através da seleção natural criou o princípio básico da genética, onde a variabilidade dos indivíduos, por meio da reprodução sexuada, é produzida pela mutação e pela recombinação genética. A partir desse período e até o final dos anos 1950, iniciaram as primeiras pesquisas cognitivas e a compreensão dos processos de raciocínio e aprendizado.

Alan Turing publicou em seu trabalho sobre aprendizado de máquina denominado *Computing Machinery and Intelligence* (TURING, 1950) que, para programar uma máquina capaz de simular a mente humana é necessário levar em consideração o processo evolutivo e genético do ser humano como parte do problema.

Rechenberg realizou a tentativa em usar processos evolutivos para resolver problemas (RECHENBERG, 1965), mas em meados de 1970 que John Holland estudou formalmente a evolução das espécies e propôs o modelo heurístico computacional que, quando implementado, poderia oferecer boas soluções. Assim, surgiam, portanto, os algoritmos genéticos.

Holland publicou seu livro *“Adaptation in natural and artificial systems”* (HOLLAND, 1975), no qual expôs seu estudo sobre o processo evolutivo, apresentando os GAs como forma de simular computacionalmente os processos evolutivos. Nesse trabalho, Holland representa os cromossomos de forma binária, o que posteriormente pode ser expandida a forma de representação com estudos de outros pesquisadores.

Na década de 1980, o progresso dos algoritmos evolucionários e sua popularização no meio científico impulsionaram o surgimento do uso destes algoritmos em *softwares* comerciais. Atualmente os algoritmos genéticos são utilizados em várias áreas científicas como: problemas de otimização combinatória,

programação genética, computação evolutiva (onde os programas se adaptam ao meio), gerenciamento de redes, no entendimento do comportamento genético, autômatos auto programáveis, entre outros.

2.2.2. Conceitos

Como parte da inteligência computacional, a computação natural permite a criação de um ambiente digital que simula o ecossistema natural por meio da criação de populações que possuem comportamento análogo ao reino animal, vegetal ou ao comportamento das partículas quando submetido a diferentes temperaturas do reino mineral (CASTRO, 2010). Isto possibilita o desenvolvimento de soluções de problemas complexos do cotidiano das pessoas, das empresas e indústrias, desde a decisão de qual caminho seguir para chegar ao trabalho até qual a melhor combinação de matéria prima para que seja feito um produto de qualidade.

Os algoritmos genéticos como parte dos algoritmos evolucionários por serem fundamentados na teoria de Darwin da evolução das espécies, também agregam conceitos da genética. São algoritmos heurísticos, probabilísticos, que fornecem o mecanismo de busca paralela e adaptativa baseada no princípio da seleção natural dos indivíduos mais aptos e na reprodução sexuada, usando paradigmas computacionais dos processos naturais da evolução como mecanismo para resolver problemas (LINDEN, 2012).

Os GAs se diferem dos métodos tradicionais de busca e otimização principalmente por quatro aspectos (WINSTON, 1992):

- a. GAs trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;
- b. GAs trabalham com uma população e não com um único ponto;
- c. GAs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar;
- d. GAs utilizam regras de transição probabilísticas e não determinísticas.

Segundo a teoria da evolução na natureza, proposta por Darwin (LINDEN, 2009), todos os indivíduos de um determinado ecossistema competem entre si pelo os recursos limitados. Os indivíduos menos adaptados ao meio e que, por este motivo, não obtenham êxito tendem a diminuir sua descendência, através de um processo chamado de seleção natural. À medida que a combinação entre os genes

dos que possuem mais êxito causa modificações positivas ou negativas, trata-se da evolução natural.

Estes princípios são imitados na construção de algoritmos computacionais que buscam uma melhor solução para um determinado problema, através da evolução das populações de soluções codificadas através de cromossomos artificiais (PACHECO, 1999).

Assim como na biologia, onde os cromossomos carregam informações sobre os seres humanos, os cromossomos artificiais da computação são uma estrutura de dados que representa as características de um indivíduo. Analogicamente a representação dos algoritmos genéticos com o sistema natural é representada no Quadro 1.

Quadro 1 – Analogia cromossômica

Natureza	Algoritmos Genéticos
Cromossomo	Palavra binária, vetor, etc.
Gene	Característica do problema
Alelo	Valor da característica
Loco	Posição da palavra, vetor
Genótipo	Estrutura
Fenótipo	Estrutura submetida ao problema
Indivíduo	Solução
Geração	Ciclo, população

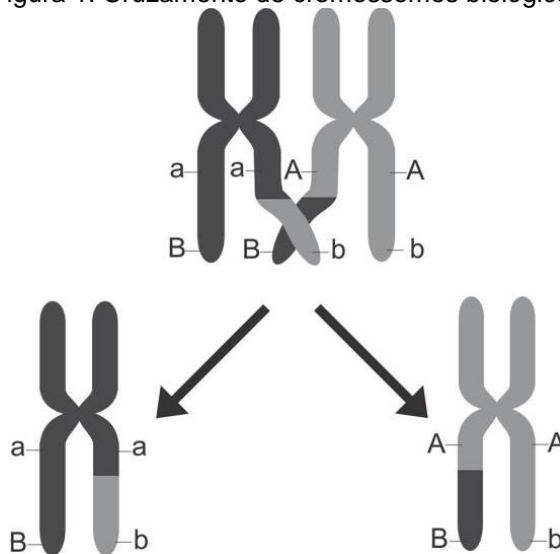
Fonte: adaptada de PACHECO, 1999.

Os algoritmos genéticos tendem a buscar uma solução para o problema que se aproxima da melhor solução (apesar de não haver garantias que esta seja encontrada). Isso se dá através de combinações entre os indivíduos a fim de encontrar os melhores representantes desta população. É uma técnica de busca eficiente em seu objetivo de explorar o espaço de soluções e encontrar soluções próximas da ótima.

Os GAs utilizam uma analogia direta do fenômeno de evolução na natureza, onde cada indivíduo representa uma possível solução e é atribuída uma pontuação de adaptação por meio de uma função de avaliação. Os processos evolucionários que os indivíduos sofrem durante a evolução são:

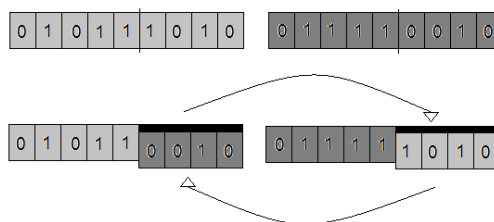
- a. Cruzamento (*crossover*) – biologicamente os cromossomos cruzam com outro para realizar a operação de troca de informações e gerar um novo cromossomo (Figura 1). Na computação o comportamento dos cromossomos é semelhante à natureza, entretanto o cruzamento pode ser feito em mais de um ponto como será visto mais adiante, divergido do que acontece na natureza (Figura 2);

Figura 1: Cruzamento de cromossomos biológicos



Fonte: adaptada de TOLEDO, 2008.

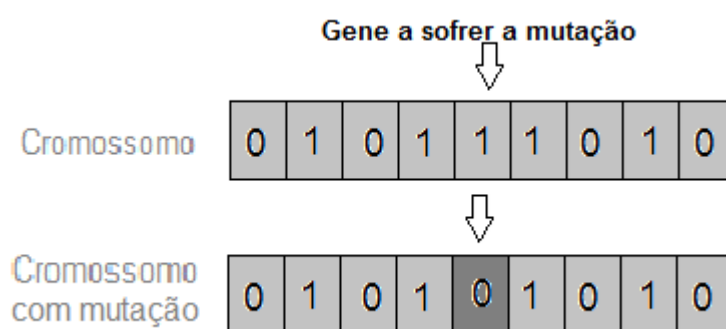
Figura 2: Cruzamento de cromossomos artificiais



Fonte: adaptado de Linden, 2012

- b. Mutação – é importante destacar que o processo de replicação do DNA é extremamente complexo. Pequenos erros podem ocorrer ao longo do tempo gerando modificações do código genético, essas modificações são chamadas de mutação. As mutações também podem ocorrer por fatores externos como a radiação, por exemplo. Essas modificações podem ser boas ou ruins. Como forma de controlar a mutação no cromossomo artificial existe o mecanismo de correção que garante que a taxa de mutação seja muito baixa (Figura 3).

Figura 3: Mutação



Fonte: elaborada pela autora

Desta forma é possível ligar a genética à teoria da evolução, onde os indivíduos mais bem sucedidos tendem a procurar parceiros mais atraentes e também mais adaptados. Os genes dos bons indivíduos, combinados através do *crossover* e da mutação, tendem a gerar indivíduos ainda mais aptos e, assim, a evolução natural caminha, gerando maior complexidade e maior adaptabilidade ao espaço de solução que os indivíduos estão inseridos. (LINDEN, 2009).

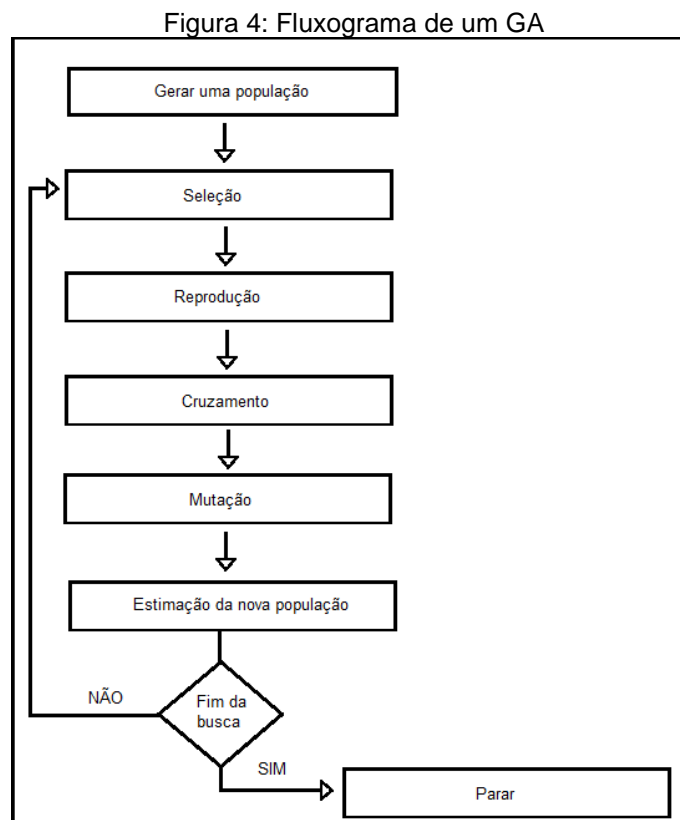
2.2.3. Composição dos GAs

Para encontrar uma solução em problemas de otimização combinatória de grande escala, os GAs funciona da seguinte forma: uma população inicial de cromossomos é gerada (normalmente, criada de forma aleatória); a cada indivíduo é atribuído um valor de adaptabilidade (*fitness*), que está relacionado ao valor da função objetivo a ser otimizada; os indivíduos com maior adaptabilidade possuem mais chances de serem selecionados para participar do processo evolutivo durante a execução do algoritmo; a aplicação dos operadores genéticos irá gerar uma nova população; se o objetivo final for alcançado, o algoritmo para; caso contrário, recomeça no segundo passo (seleção de acordo com a avaliação do indivíduo).

Na versão mais básica dos algoritmos genéticos, que faz uso da representação binária, dois indivíduos são selecionados para serem os pais de duas novas soluções (novos indivíduos) mediante o mecanismo de *crossover*. A operação de *crossover* combina as características dos pais para gerar as configurações binárias dos filhos e se calculam os seus valores de adaptabilidade. Estas soluções substituem dois indivíduos da nova população. O processo se repete quantas vezes for necessário até se obter a convergência ou é interrompido pela quantidade de

vezes que é executado o algoritmo, ou pela quantidade de tempo decorrido pelo algoritmo.

O fluxograma na Figura 4 representa o processo de execução do algoritmo genético.



Fonte: adaptado de Miranda, 2014.

Desde que Holland propôs a ideia dos GAs, tem-se estudado grandes variações do esquema da Figura 4. Independente da sofisticação de um GA existe cinco componentes que devem ser incluídos:

- Uma representação, em termos de cromossomos, da composição do problema;
- Uma maneira de criar a estrutura da população inicial;
- Uma função de evolução que permita ordenar os cromossomos de acordo com a função objetivo;
- Operadores genéticos que permitam alterar a composição dos novos cromossomos gerados pelos pais durante a reprodução;
- Valores dos parâmetros que o GA usa (tamanho da população, probabilidade associadas com a aplicação dos operadores genéticos).

2.2.4. Representação Cromossômica

A representação cromossômica é essencial para o algoritmo genético e consiste em uma maneira de interpretar as informações do problema em uma expressão funcional que possa ser tratada pelo computador. A adequação da representação ao problema implica diretamente na qualidade dos resultados obtidos pelo algoritmo. Uma vez escolhida a representação cromossômica, essa escolha implica diretamente no uso dos operadores genéticos (HOLLAND, 1987).

A definição da representação cromossômica fica a critério do programador, desde que esta esteja adequada ao problema. Entretanto, existem algumas regras gerais que devem ser levadas em consideração (LINDEN, 2009). São elas:

- a. A representação deve ser a mais simples possível;
- b. Se houver soluções proibidas ao problema, elas não devem ter uma representação possível;
- c. Se o problema impuser condições de algum tipo, estas devem estar implícitas dentro da representação.

Existem várias formas de representação, sendo que a mais simples e também a mais usada, a depender do problema, é a forma binária. A representação binária nada mais é que um cromossomo (vetor) com uma sequência de *bits* (cada *bit* é um gene), onde os valores dos *bits* são representados por 1 ou 0.

Outra forma de representação cromossômica que pode ser utilizada para problemas de otimização combinatória é optar pela representação em lista. Neste caso, a solução do problema está relacionada à ordem em que os elementos aparecem nessa lista. Um exemplo de uso desta representação é o problema do caixeiro viajante. Este problema retrata a escolha de uma rota que deve ser percorrida pelo caixeiro nas cidades a serem visitadas por ele com a menor distância possível.

A representação cromossômica para este caso é feita através de uma lista onde cada elemento está associado a uma cidade e a ordem de percurso define qual é a sequência de cidades a serem visitadas, correspondendo a uma possível solução do problema.

A representação numérica é usada quando os cromossomos são representados por números reais, tornando o espaço de busca contínuo e a

representação de forma direta. Linden (2012) afirma que “O uso de cromossomos reais consiste em tornar iguais o genótipo (representação interna) e o fenótipo (valor usado no problema), retirando todos os efeitos de interpretação associados a situações que dois são diferentes”.

Existem várias situações em que as representações citadas anteriormente não se encaixam ao problema. Neste caso, precisa-se de uma representação meio binária e meio numérica. Quanto a isso, não há empecilhos em utilizar as duas representações simultaneamente. Sendo assim, caracteriza-se a utilização de uma representação híbrida, que nada mais é que a combinação de diferentes representações.

Cromossomos similares podem ser representados de forma agrupada, através do conceito de esquemas. De acordo com Vianna (1998), um esquema (ou máscara) é um modelo de representação dos cromossomos parecidos. Um esquema é construído com a inclusão de um símbolo especial (*) no alfabeto de gens. A representação de esquemas pode ser verificada na Tabela 1.

Tabela 1: Esquemas

Esquemas	Indivíduos que representa
1*	10, 11
1*0*1	10001, 10011, 11001, 11011
**0	000, 010, 100, 110

Fonte: Linden, 2012.

2.2.5. Operadores Genéticos

Linden (2012) afirma que “os operadores genéticos consistem em aproximações computacionais de fenômenos vistos na natureza, como a reprodução sexuada, a mutação genética e quaisquer outros que a imaginação dos programadores consiga reproduzir”. Os operadores genéticos transformam a população através de sucessivas gerações, estendendo a busca até encontrar um valor satisfatório, sendo que as características das gerações anteriores tendem a se manter presentes ao longo das gerações. Michelan e Maia (2006) definem os seguintes operadores genéticos:

- Operador de seleção: faz a seleção dos pais mais aptos a fim de reproduzir membros da população que tenham condições de atingir a função objetivo;

- b. Operador de mutação: são necessários para a introdução e manutenção da diversidade genética da população, alterando um ou mais componentes de uma estrutura escolhida. Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de solução nunca seja zero;
- c. Operador de cruzamento: é responsável pela recombinação das características dos pais durante a reprodução, permitindo assim a herança das características dos pais.

2.2.6. Parâmetros Genéticos

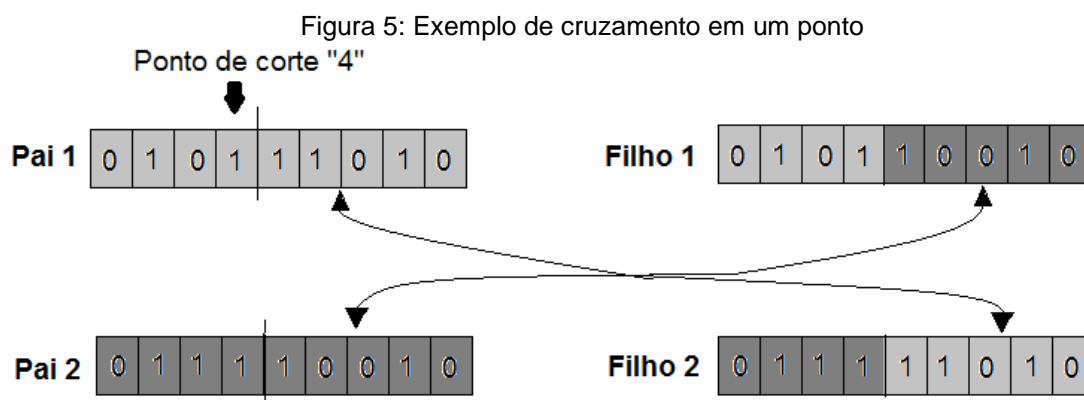
É importante salientar a maneira que alguns parâmetros interferem no comportamento dos GAs, para que se possa estabelecê-los conforme as necessidades do problema e dos recursos disponíveis (MICHELAN e MAIA, 2006).

- a. Tamanho da população – afeta o desempenho real e a eficiência dos GAs. Se a população for pequena oferece um pequeno espaço de busca causando uma queda no desempenho. Uma população maior fornece melhor cobertura e previne a convergência prematura para soluções locais. Entretanto se a população for muito grande o tempo de processamento pode se tornar muito grande, e pode requerer recursos computacionais maiores;
- b. Taxa de cruzamento – quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Entretanto isso pode gerar um efeito indesejado causando a perda de estruturas de alta aptidão. Se o valor for muito baixo o algoritmo pode se tornar muito lento;
- c. Taxa de mutação – uma baixa taxa evita que uma dada posição fique estagnada e possibilita a chegada a qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória, além de possibilitar a perda de características importantes da população;
- d. Intervalo de geração – controla a porcentagem da população que será substituída durante a próxima geração.

2.2.7. Tipos de Cruzamento

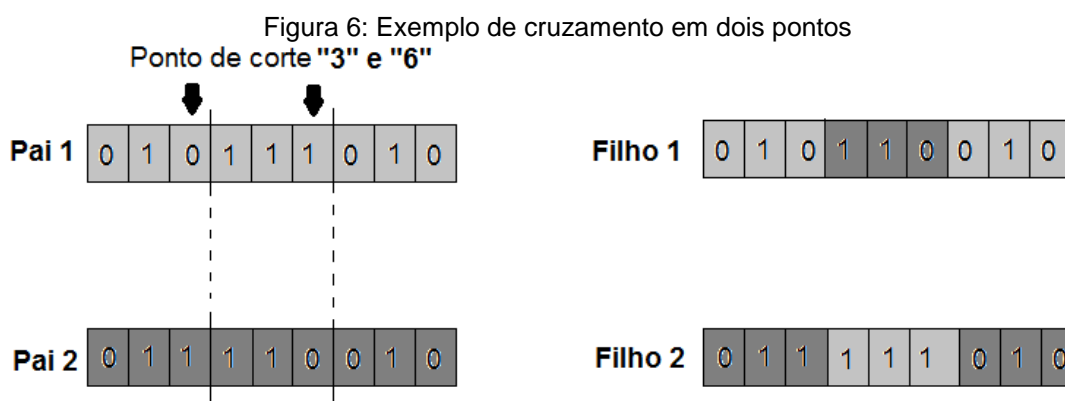
O cruzamento é responsável pela recombinação de informações de indivíduos, após a seleção dos pais, para que haja a reprodução. O cruzamento pode ser realizado de várias formas, a depender do problema. As formas de aplicação dos operadores de cruzamento são:

- Um ponto – um ponto de cruzamento é escolhido e a partir deste ponto as informações genéticas dos pais serão trocadas gerando assim novos filhos (Figura 5).



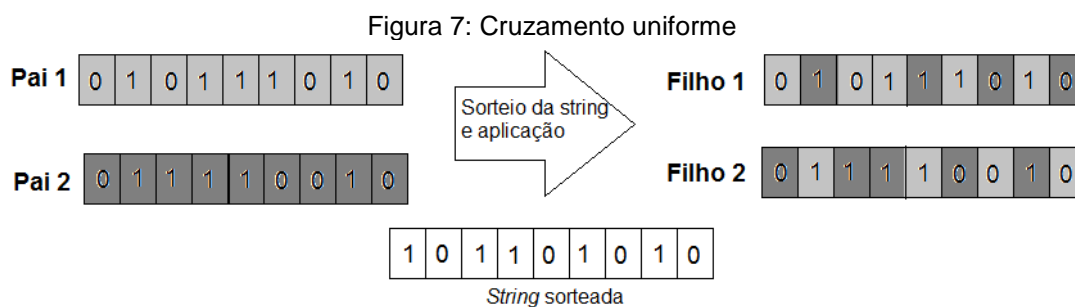
Fonte: adaptado de Michelan e Maia, 2006.

- Multiponto – é semelhante ao cruzamento de um ponto, divergindo deste a partir do momento em que vários pontos podem ser utilizados para troca de informações genéticas (Figura 6).



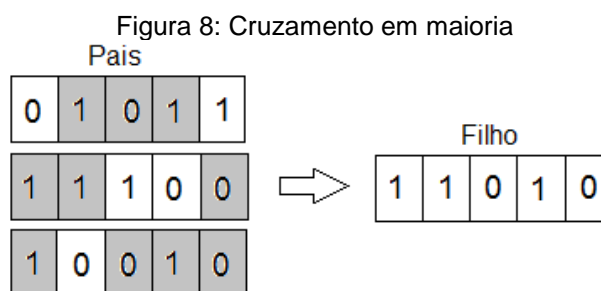
Fonte: elaborada pela autora.

- c. Uniforme – o cruzamento uniforme não utiliza pontes de corte, mas determina através de um parâmetro, qual a probabilidade de cada gene ser trocado. Neste caso é sorteado uma *string* (vetor) com valores de 0 e 1. A troca de informações genéticas é baseada na posição de cada valor da *string* sorteada, se o valor for igual a 1 o filho 1 herda o valor da posição corrente do pai 1 e o filho dois do pai 2, se o valor for igual a 0 o filho 1 recebe o valor do pai 2 da posição corrente e o filho 2 recebe o valor do pai 1 (Figura 7).



Fonte: elaborada pela autora.

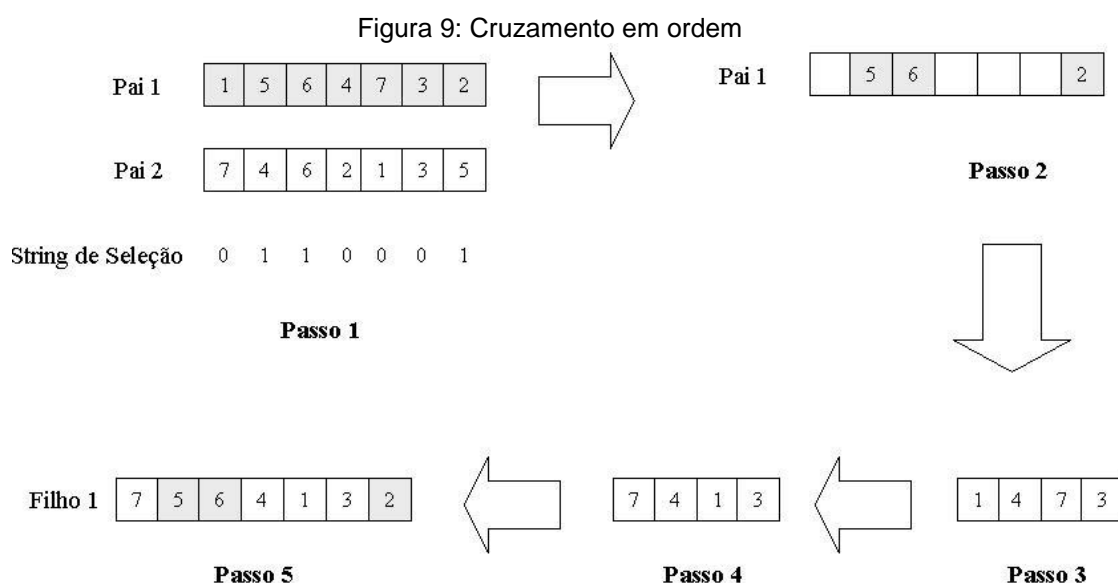
- d. Cruzamento em maioria – esta forma de cruzamento não é muito usada, pois tem a tendência de que o GA convirja rapidamente e não percorra todo o espaço de busca. A operação básica deste *crossover* consiste em sortear vários pais e fazer com que cada bit do filho seja igual ao valor da maioria dos pais selecionados, conforme mostrado na Figura 8.



Fonte: adaptada de Linden, 2012.

- e. Cruzamento baseado em ordem – é uma versão especial do cruzamento uniforme. A diferença está relacionada às características da forma de representação em ordem. Este tipo de cruzamento também utiliza uma *string* de seleção, os valores iguais a 1 indica que o filho herda as características do pai1. O que o diferencia do cruzamento uniforme é o fato de ser utilizada uma

estrutura de dados auxiliar que armazena as características do pai1 que não foram inseridas no filho. Em seguida, essa estrutura auxiliar é colocada na ordem em que as características aparecem no pai2. Por fim, são inseridas no filho as características não utilizadas no pai1, mas na ordem em que aparecem no pai2, o que implica na herança de informações do pai2, como mostra a Figura 9.



Fonte: Linden , 2012.

2.2.8. Tipos de Mutação

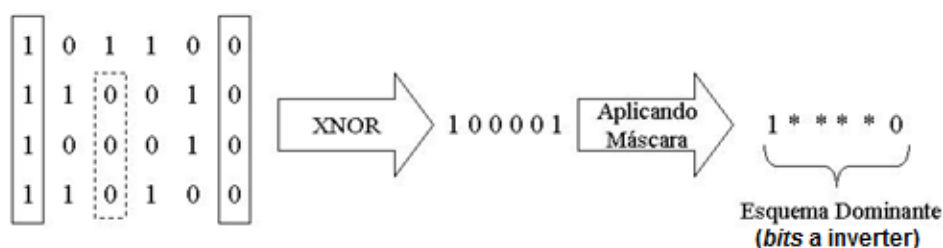
A mutação é fundamental para o GA, pois ela garante a continuidade da diversidade genética e permite que o algoritmo explore pontos diferentes da população. A mutação opera com uma probabilidade associada para identificar se a operação de mutação será realizada ou não. Esta probabilidade influencia no comportamento do algoritmo de forma que se seu valor for muito alto, o algoritmo passa a ter um comportamento aleatório. Entretanto se o valor for muito baixo, a população não terá diversidade depois de certo número de iterações, levando a rápida convergência, mas nem sempre para uma boa solução, devido ao fato do algoritmo poder ficar restrito a um determinado ponto associado a um mínimo ou máximo local no espaço de solução.

A mutação pode acontecer de formas diferentes a depender da representação do problema. Os diferentes tipos de mutação são (LINDEN, 2012):

- a. Tradicional – na mutação tradicional acontece na representação binária, onde é sorteado um valor que indica qual bit será escolhido para aplicar a troca genética. No caso do valor binário, há a troca do valor 0 por 1 e vice-versa..
- b. Mutação dirigida – na mutação tradicional, a probabilidade dos genes serem modificados é a mesma. Sendo assim, na mutação dirigida só ocorrerá depois de certo número de gerações, onde o algoritmo tenta encontrar o esquema dominante dentro de uma codificação binária. Aplica-se o operador lógico XNOR (negação do ou) nos cromossomos para determinar o esquema dominante. O resultado desta operação é atribuir o valor 0 onde todos os indivíduos possuem *bits* iguais, e 1 onde há um elemento diferente. Invertendo este resultado (equivale à operação lógica XNOR), obtém-se uma máscara que equivale ao esquema dominante (Figura 10).

Depois de descoberto o esquema dominante comum às melhores soluções, é realizada a mutação. Isso permite o surgimento de novas soluções com informações genéticas completamente diferentes, permitindo que o GA continue progredindo em soluções ainda não exploradas.

Figura 10: Escolha do esquema dominante



Fonte: adaptada de Linden, 2012.

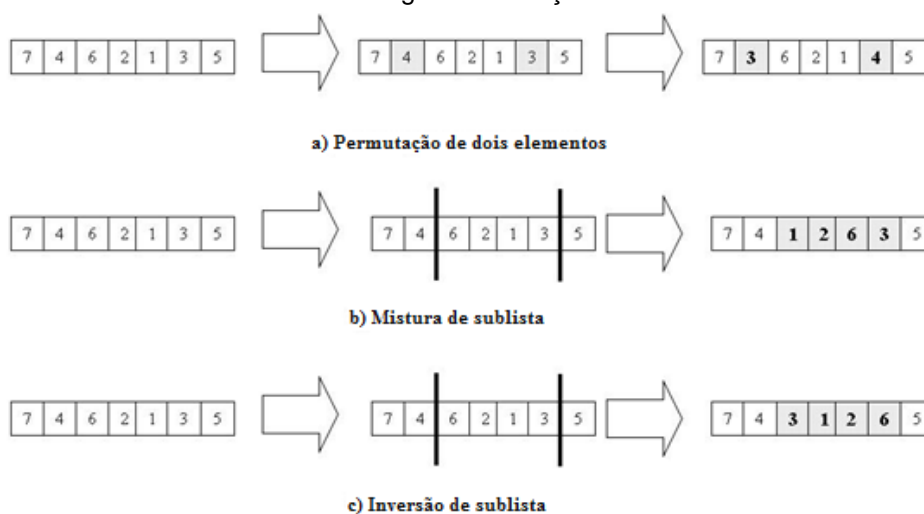
- c. Mutação baseada em ordem – consiste em realizar mudanças locais em cromossomos. No caso de representação em ordem, não há *bits* a inverter e não pode designar valores aleatoriamente. Logo, deve-se operar com vários genes ao mesmo tempo. Existem três maneiras de fazê-lo: a permutação de elementos, a inversão de sublista e a mistura de sublista.

Na permutação de elementos são escolhidos dois elementos ao acaso dentro do cromossomo e troca-se a posição dos elementos, como está ilustrado na Figura 11a.

Na mistura de sublista escolhem-se dois pontos de corte dentro do cromossomo. Para realizar a mutação basta fazer uma permutação aleatória dos elementos da sublista criada entre os pontos de corte (Figura 11b).

O funcionamento da mutação que consiste em inverter a lista sorteada por meio de dois pontos de corte, em vez de realizar uma mistura aleatória dos elementos, a ordem dos elementos é invertida, como mostra a Figura 11c.

Figura 9: Mutação em ordem



Fonte: adaptada de Linden, 2012.

Hoje em dia, os algoritmos genéticos têm sido aplicados em vários tipos de problema e em diversas áreas, o que o torna interdisciplinar. Desta forma, os pesquisadores da área de GA buscam formas que o tornem mais eficientes e inteligentes na resolução de problemas, resultando em variações de GAs por meio de modificações dos métodos anteriormente citados.

2.3. Trabalhos Relacionados

Como já foi citado anteriormente, o problema de escalonamento de horários vem sendo estudado desde a década de 1960. É um problema comum a qualquer instituição de ensino e há vários estudos abordando este assunto, como pode ser visto a seguir.

Pousen (2012), em seu trabalho, oferece um modelo de solução do problema de escalonamento de horários baseado no sistema brasileiro de ensino, visando alocar os professores e as disciplinas que cada professor ministra, e

também as alocações das aulas nas salas. Em seu trabalho, Pousen utiliza a técnica meta-heurística têmpera simulada (*simulated annealing*) a fim de encontrar uma solução com o menor custo possível. O trabalho foi modelado com base nos métodos de pesquisa operacional. O modelo proposto visa reduzir a carga horária dos professores, mas sem infringir as restrições quanto ao número de aulas diárias. Segundo o autor, com a aplicação do modelo foi possível gerar horários, de forma computacional, de qualidade similar às geradas pela escola, que se mostrou ser de ótima qualidade.

No trabalho de Coloni *et al.* (1993), os autores comparam duas versões de algoritmos genéticos para resolver o problema de escalonamento de horários com e sem busca local, avaliando o desempenho de ambos em relação a uma proposta feita manualmente. O trabalho ainda compara o desempenho dos GAs com duas outras propostas feitas por meio das técnicas de têmpera simula e da busca tabu (*tabu search*). A proposta foi aplicada numa escola italiana de ensino médio.

Para encontrar as soluções pelos GAs foram utilizados cruzamentos constantemente e foram definidos operadores de cruzamento e mutação que só retornavam soluções viáveis, aplicando-os para encontrar a melhor solução. Foi levado em consideração a disponibilidade dos professores, os horários das aulas e a duração de cada aula. No experimento, os GAs produziram resultados com melhor desempenho em comparação às demais técnicas, visto que os GAs são flexíveis com a escolha de diferentes escalas de horários. O algoritmo baseado em GA com busca local mostrou melhores resultados, e segundo os autores, foi possível encontrar soluções com número de iterações significativamente menor em comparação com a quantidade de iterações realizadas sem busca local.

Lara (2008) mostra a implementação do algoritmo colônia de abelhas (*bee algorithm*) para resolver o problema de escalonamento de horários escolar. O algoritmo encontra a melhor solução por meio do conceito de vizinhos (apresentados na seção 2.3.3). O autor propõe ainda uma nova maneira para substituir uma população considerando a história evolutiva das abelhas e sua aptidão. O algoritmo foi testado em duas escolas e obteve resultados promissores. Os resultados apresentados no artigo indicam que uma tarefa que antes levava vários dias para ser realizada de forma manual, foi reduzida para aproximadamente 6 horas com a utilização do algoritmo.

Como forma de otimizar e resolver o problema de escalonamento de horários, o trabalho de Birbas *et al.* (2009) utiliza a programação inteira (*integer programming*). O experimento foi realizado em escolas da França e da Alemanha que utilizam o sistema de ensino Hellenic. Os autores observaram que o tamanho da escola influencia positivamente na satisfação dos professores e negativamente na disponibilidade dos professores. Como resultado, o escalonamento obteve êxito de acordo com as restrições (e regras) do sistema escolar, tentando favorecer o máximo possível às preferências dos professores.

Marte (2002) oferece uma forma de solução do problema de escalonamento de horários para escolas alemãs, utilizando a programação com restrições e programação paralela. No trabalho, o autor compara duas soluções por meio da programação paralela, uma para aprofundar-se (*downsize*) no modelo e outra para priorizar o espaço de busca. Para resolver o problema foram utilizados cuidados com performance computacional. Em comparação, o resultado obtido é que as duas formas tiveram desempenho praticamente semelhantes.

Algumas escolas espanholas foram utilizadas como exemplo no trabalho de Pena *et al.* (2008). Neste trabalho, os autores apresentam uma proposta de solução para escalonamento de horários escolar usando o método ascendente não aleatório (*RNA search*) combinado com algoritmos genéticos, como técnica de solução. Os experimentos foram realizados num cenário real e demonstraram que a combinação das duas técnicas obtém resultados mais estáveis.

Ghaemi *et al.* (2007) oferecem uma proposta de solução baseada em algoritmos genéticos em ambiente universitário, a fim de minimizar os conflitos existentes no escalonamento de horários. Foram realizadas duas abordagens, GA modificado (alterações nos processos que implicam na otimização do algoritmo, tornando-o mais adaptado ao problema) e GA cooperativo. Segundo os autores, os resultados mostraram que o GA modificado teve o desempenho melhorado com operadores básicos modificados, visto que operadores inteligentes melhoram o comportamento geral do algoritmo. Além disso, o desempenho é melhorado quando usado o método de genética cooperativa.

Sigl *et al.* (2003) também descreveram em seu trabalho uma solução para o problema de escalonamento de horários escolar através de algoritmos genéticos. Foram feitos testes em pequena e larga escala do problema. O desempenho do algoritmo foi significativo com as modificações dos operadores genéticos. Com o

teste de pequena escala não houve conflitos, entretanto no de larga escala houve 95 conflitos e com operadores inteligentes mostrou resultados melhores, em outras palavras, a convergência do algoritmo acontece de forma mais rápida e diminui a quantidade de conflitos, o que o torna mais eficiente.

No trabalho de Lukas *et al.* (2009), foram consideradas várias condições impostas no problema de escalonamento de horários, como por exemplo, a disponibilidade do conferencista, o grande número de aulas e cursos. Para solucionar este problema foi utilizada a técnica de algoritmos genéticos combinado com a busca heurística. Segundo os autores, após vários testes, os resultados encontrados com a execução do algoritmo foram considerados como a melhor solução para o problema.

Outra proposta de solução para o problema de escalonamento de horários escolar foi apresentado no trabalho de Raghavjee e Pillay (2010). O problema foi deparado nas escolas de ensino fundamental e médio da África do Sul e, como técnica de solução, foram utilizados algoritmos genéticos. Como resultado, foram encontradas soluções viáveis para ambos os casos, apresentando desempenho satisfatório e baixo custo de restrição do problema.

Em outro trabalho, Raghavjee e Pillay (2008) apresentaram uma proposta de solução usando algoritmos genéticos evoluídos. Foram utilizados cinco casos reais para realizar o experimento buscando uma solução iterativamente. Como resultado foram encontradas soluções viáveis para todos os casos. Segundo os autores, o algoritmo convergiu em um minuto. Além disso, o trabalho fez uma comparação de desempenho dos GAs com outras técnicas (têmpera simulada, busca tabu, etc.), de acordo com os autores os GAs obtiveram os melhores resultados.

Outra proposta de solução do problema de escalonamento de horários escolar foi apresentada por Raghavjee e Pillay (2011) com o uso da técnica de algoritmos genéticos. Além disso, os autores compara a performance da execução dos GAs com uma população inicializada aleatoriamente e outra solução usando uma heurística. O experimento foi realizado em seis escolas gregas e obtendo resultados satisfatórios. Segundo os autores, foi possível encontrar boas soluções para o problema e o desempenho do GA usando heurística na população obteve resultados de melhor qualidade, ou seja, menos violação de restrições.

Ahandani e Baghmisheh (2011) comparam duas soluções por meio de algoritmos meméticos para o problema de escalonamento de horários acadêmico. A primeira usa uma heurística de coloração gráfica no cruzamento dos indivíduos. A outra se utiliza de um algoritmo genético híbrido, dando ênfase na busca local. Os resultados encontrados demonstraram que a primeira forma possui melhor performance.

Na proposta apresentada por Chen e Shih (2013), para resolver o problema de escalonamento de horários nas universidades, foi utilizada como técnica para solucionar o problema a otimização por enxame de partículas. A escolha dos autores por esta técnica se deu ao fato dela possuir características de rápida convergência, configuração de poucos parâmetros e pela capacidade de ajuste dinâmico. O algoritmo utiliza heurística para explorar o espaço de solução. Segundo os autores, com a execução do algoritmo foi possível encontrar soluções satisfatórias que atendia as restrições do problema.

No trabalho de Nguyen *et al.* (2011), os autores utilizam o problema de escalonamento de horários nas universidades como exemplo para comparar o desempenho dos algoritmos genéticos e os algoritmos meméticos na solução do problema. O experimento utilizou seis problemas reais e como resultados os autores observaram que os algoritmos meméticos convergem mais rapidamente que os genéticos, ou seja, resolvem o problema em menos tempo.

O trabalho de Derakhshi e Babaei (2012) utiliza o problema de escalonamento de horários classificando as abordagens do problema em três classes: abordagens baseadas em métodos de pesquisa, baseados em meta-heurística e métodos inteligentes. No experimento, os autores observaram que métodos de busca não possuem bom desempenho para resolver este tipo de problema, enquanto que as abordagens baseadas em meta-heurística e métodos inteligentes são mais eficientes em termos de encontrar a solução para o problema.

Jat e Yang (2009) abordaram o problema de escalonamento de horários nas universidades. O trabalho propõe uma solução baseada em algoritmos genéticos com busca guiada e busca local. A combinação das técnicas tem como objetivo encontrar a solução viável que leva em consideração o histórico dos indivíduos da população e melhorar a qualidade destes indivíduos. O experimento mostrou que a proposta produz resultados promissores na solução do problema.

Por fim, no trabalho de Abbaszadeh *et al.* (2012), os autores propõem uma solução para o problema de escalonamento de horários baseada no uso de algoritmos meméticos. Para realizar o experimento, foi adotada uma estrutura diferenciada para os cromossomos, os métodos genéticos foram modificados e foi usada uma busca local. Foram considerados os professores, aulas e informações do curso e as restrições relacionadas a cada um destes fatores. Segundo os autores, o resultado do estudo mostra a alta eficiência do algoritmo proposto comparando com outros algoritmos.

Uma síntese dos trabalhos relacionados, anos da disponibilização das pesquisas e das técnicas utilizadas está dispostos no Quadro 2, na ordem em que foram descritos na seção 2.3.

Quadro 2: Autores e técnicas utilizadas

Autor	Ano	Técnica utilizada
Pousen	2012	Têmpera simulada
Coloni et al.	1993	Algoritmos genéticos
Lara	2008	Colônia de Abelhas
Birbas et al.	2009	Programação inteira
Marle	2002	Programação com restrição e programação paralela
Pena	2008	RNA search e algoritmos genéticos
Ghaemi et al.	2007	Algoritmos genéticos
Sigl et al.	2003	Algoritmos genéticos
Lukas et al.	2009	Algoritmos genéticos
Raghavjee e Pillay	2010	Algoritmos genéticos
Raghavjee e Pillay	2008	Algoritmos genéticos, têmpera simulada e busca tabu
Raghavjee e Pillay	2011	Algoritmos genéticos
Ahandani e baghmisheh	2011	Algoritmos meméticos e algoritmos genéticos híbrido
Chen e Shih	2013	Enxame de partículas
Nguyen et al.	2011	Algoritmos meméticos e algoritmos genéticos
Derakhshi e Babaei	2012	Abordagens baseadas em meta-heurística e métodos inteligentes
Jat e Yang	2009	Algoritmos genéticos com busca guiada e busca local
Abbaaszadeh et al.	2012	Algoritmos meméticos

Fonte: elaborado pela autora.

O Quadro 2 mostra que o problema de escalonamento de horários é alvo de estudos e para encontrar a solução são usadas várias técnicas de busca. Para que melhor detalhamento de como as outras técnicas utilizadas para encontrar uma solução para o problema, são apresentados os conceitos sobre essas técnicas no APÊNDICE A.

3. METODOLOGIA E ALGORITMO PROPOSTO

Conforme o que pôde ser visto na seção anterior, existem várias propostas de solução para o problema de escalonamento de horários baseadas na utilização de algoritmos genéticos como estratégia de solução. Estas propostas também estão disponibilizadas no mercado de *software* na forma de produtos comerciais, gratuitos ou não.

O processo de montagem de horários é um trabalho árduo e desgastante, pois é necessário levar em consideração várias restrições relacionadas às disciplinas, professores, alunos e salas, e pequenas modificações e/ou ajustes que sempre são necessários, por mais simples que sejam, exigem mais algumas horas de trabalho.

As soluções adotadas nos trabalhos relacionados (Seção 2.3) e os *softwares* disponíveis no mercado levam em consideração, principalmente, os aspectos administrativos para montagem de horários. Em outras palavras, consideram as restrições quanto ao corpo docente, estrutura da instituição (salas disponíveis) e as disciplinas que serão oferecidas, o que pode desfavorecer o desempenho acadêmico do corpo discente (GARCIA *et al.* 2014).

Entretanto, o aluno é um ator fundamental no processo de ensino-aprendizagem. Assim, para um bom desempenho do discente, o docente e a administração da instituição precisam desenvolver o que os pedagogos chamam de sequência didática iterativa. Esse processo consiste na ideia de criar maneiras que despertem o interesse do aluno e a participação do mesmo no processo de ensino-aprendizagem. Todavia, a existência de aulas vagas e intervalos de aula na estrutura de horário podem prejudicar o rendimento acadêmico do aluno, visto que o discente pode se sentir desmotivado a aguardar até uma hora e quarenta minutos para assistir a aula seguinte (GARCIA *et al.* 2014).

Partindo-se do princípio que o curso de BSI é oferecido nos turnos matutino e vespertino e considerando-se que o mesmo recebe vários alunos de outros municípios que precisam se deslocar até Caicó a fim de participar das aulas, faz-se necessário um olhar mais criterioso na montagem de horários, considerando o aluno como ator fundamental, e para atender as necessidades dos mesmos, deve ser levada em consideração uma blocagem de horários mais igualitária como critério para montagem de uma estrutura de horário. Os fatores listados acima justificam a

busca por uma solução que foque na otimização do horário do discente, tendo em vista que o melhor rendimento acadêmico do corpo discente requer uma gestão de tempo favorável à construção do conhecimento.

O trabalho aqui proposto consiste em uma solução para o problema do escalonamento de horários acadêmicos baseada na utilização de algoritmos genéticos, que tenha como foco principal a otimização do horário dos discentes, buscando o equilíbrio na distribuição de aulas diárias e a minimização de ocorrência de intervalos entre aulas, com a finalidade de auxiliar o coordenador do curso na montagem de horários acadêmicos.

3.1. Representação Cromossômica Utilizada

Inicialmente, foi definida uma estrutura de dados que viabilizasse a descrição de horários acadêmicos. A representação dos cromossomos utilizada para a montagem de horários é um conjunto de estrutura de dados, que será explanada a seguir.

Primeiramente, foi necessário analisar como se comporta o processo de montagem de horários de forma mais detalhada. Sabe-se que cada horário é composto por um conjunto de disciplinas, que são ministradas por professores e frequentadas (ou cursadas) por alunos, o que remete ao conceito de turma. É importante salientar que uma disciplina pode ser ministrada por mais de um professor, caso seja necessário oferecer a mesma disciplina para turmas diferentes.

A estrutura de dados escolhida para representar uma turma foi um vetor, representado na Figura 12, onde cada posição deste vetor possui informações sobre os componentes citados acima, a exceção dos alunos, além de outras informações necessárias durante a execução do algoritmo.

Figura 12: Representação de uma turma

NOME	DISCIPLINA	PROFESSOR	SEMESTRE LETIVO	AULAS POR SEMANA
------	------------	-----------	--------------------	---------------------

Fonte: elaborada pela autora.

No curso de universitários em geral, as disciplinas estão distribuídas por período, sendo que essas disciplinas estão distribuídas ao longo de vários períodos.

No curso de BSI da UFRN, as aulas são oferecidas nos turnos matutino (M) e vespertino (T), na maioria das vezes em blocos de duas horas-aula juntas (Figura 13).

Figura 13: Matriz de horário por período

							8		
							...	Turno	
							2	Turno	
							1	Turno	
Seg	Ter	Qua	Qui	Sex	Aulas	Turno			M / T
					12				M / T
					34	M / T			
					56				

Fonte: elaborada pela autora.

É importante ressaltar que, apesar da UFRN considerar o sábado como um dia letivo, o curso de BSI, que é utilizado como estudo de caso para o presente trabalho, não possui atividades nesse dia. Sendo assim, não se fez necessário a inclusão de uma coluna que representasse o referente dia.

Além disso, via de regra, as disciplinas vinculadas aos períodos ímpares são oferecidas no primeiro semestre do ano e as disciplinas vinculadas aos períodos pares no segundo semestre do ano. Porém, com alguma frequência, também são oferecidas disciplinas esporádicas relacionadas aos períodos complementares, no caso de turmas de repetentes, de forma que quase sempre há disciplinas de todos os períodos oferecidas a cada semestre.

Devido ao fato da oferta de disciplinas não ser apenas em seus períodos regulares, o corpo docente do curso de BSI decidiu utilizar a seguinte política: no turno vespertino serão oferecidas as disciplinas para alunos regulares (nivelados), e no turno matutino para alunos repetentes.

As disciplinas obrigatórias estão divididas em níveis, que correspondem aos semestres em que devem ser obrigatoriamente oferecidas. Assim, a oferta de disciplina segue um calendário predefinido: quando o semestre corrente é ímpar, as disciplinas oferecidas como regulares no turno vespertino pertencem aos níveis

ímpares (semestres 1, 3, 5 e 7) e no contraturno (matutino) são oferecidas disciplinas dos períodos pares a serem cursadas por alunos repetentes e/ou desnivelados. Quando o semestre corrente é par, ocorre o inverso.

Sendo assim, a montagem de horários é feita com base no período em que as disciplinas estão alocadas (Figura 11), seguindo a determinação existente na estrutura curricular do curso (ANEXO A).

Desta forma, para gerar uma solução para o problema de montagem de horários, é necessário agendar os horários de cada uma das turmas oferecidas em cada um dos períodos. Portanto, a representação cromossômica escolhida para a solução do problema de escalonamento foi um vetor de matrizes (períodos), como apresentado na Figura 14.

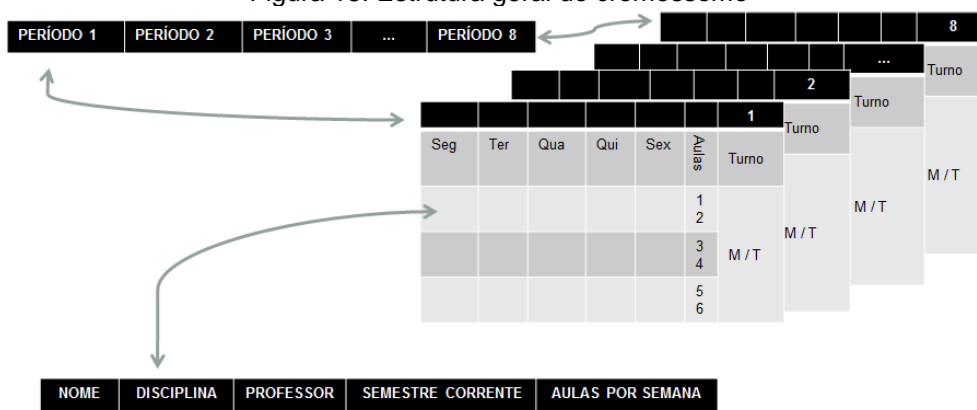
Figura 14: Representação cromossômica

PERÍODO 1	PERÍODO 2	PERÍODO 3	...	PERÍODO 8
-----------	-----------	-----------	-----	-----------

Fonte: elaborada pela autora.

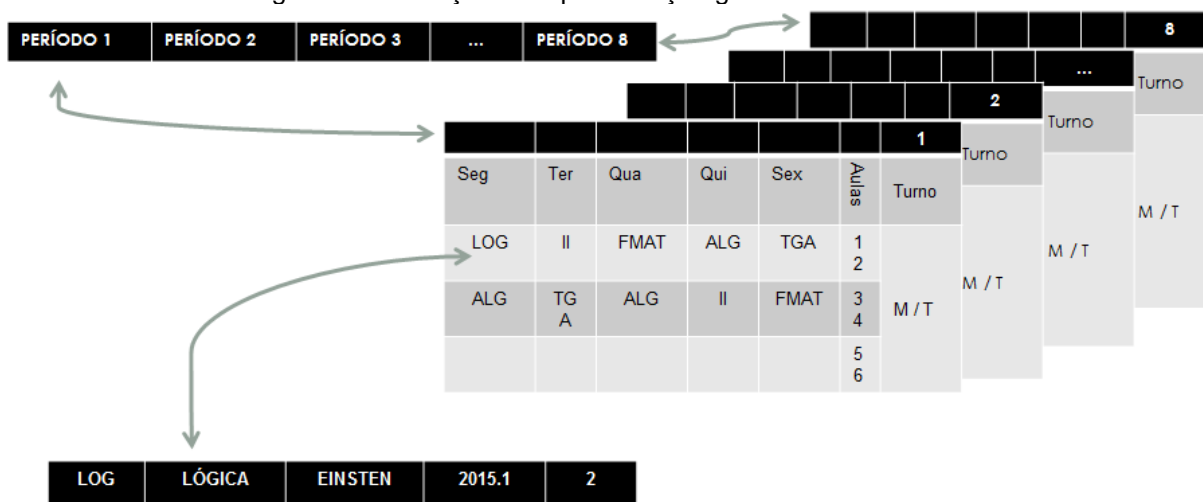
Com base nas informações anteriormente citadas, a representação cromossômica é a junção dessas estruturas alocadas em um vetor. Em outras palavras, a solução do problema é constituída por um vetor, onde cada posição do vetor corresponde a uma matriz e cada posição desta matriz corresponde a um bloco de aula de uma turma (duas horas-aula). É importante destacar que o bloco de aulas da turma pode se repetir duas ou três vezes, dependendo da carga horária da disciplina. A Figura 15 mostra a alocação das estruturas citadas anteriormente que compõe a representação cromossômica proposta neste trabalho.

Figura 15: Estrutura geral do cromossomo



Fonte: elaborada pela autora.

Figura 17: Ilustração da representação geral do cromossomo



Fonte: elaborada pela autora.

3.2. Operadores Genéticos Utilizados

Como em todo problema a ser solucionado de forma computável por meio de um algoritmo, é necessário conhecer a dimensão (tamanho) do problema. Cooper e Kingston (1996) provaram que o problema de decisão associado à montagem de horários é do tipo NP-Completo e, por não existir um algoritmo específico que encontre uma solução ótima em tempo polinomial, são usadas técnicas de busca heurística que encontre uma solução aceitável para o problema.

Apesar de o presente trabalho levar em consideração unicamente o curso de BSI, de acordo com as características do escalonamento de horários é possível encontrar o tamanho do espaço de busca de acordo com a Equação 1.

$$S = \sum_{p=1}^n \frac{e!}{(e - b_p)!} \quad (1)$$

Onde:

S = espaço de busca;

p = período;

n = número total de períodos de determinado curso (no caso de BSI são 8 períodos);

e = tamanho da matriz de horários do curso (no caso de BSI a matriz é 3x5, desta forma a matriz contém 15 blocos a serem preenchidos);

b_p = quantidade de blocos que serão preenchidos na matriz de acordo com o período, o mesmo pode ser obtido pelo somatório da divisão entre a carga horária e os créditos de cada disciplina de determinado período.

Conforme foi explanada na Seção 2, que descreve os passos a ser seguidos para elaboração de uma algoritmo genético, o primeiro passo é a definição da representação cromossômica, sendo esta a mais simples possível, mas que permita a descrição exata das características do problema.

Após a definição da representação cromossômica, o passo seguinte é geração da população inicial. Para este fim, foi gerado um indivíduo aleatoriamente, como mostra na Figura 17, e a partir deste cromossomo foram gerados novos indivíduos por meio de permutação.

Em seguida, foram definidos os operadores genéticos que seriam utilizados durante a execução do algoritmo na geração de novas populações. Segundo a literatura, os operadores de cruzamento e mutação, assim como a representação cromossômica, devem estar de acordo com a complexidade e restrições associadas ao problema.

O operador de cruzamento utilizado foi o baseado em ordem, onde era gerada aleatoriamente uma matriz com tamanho correspondente ao da matriz do período. Para seleção dos pais foi utilizado o método da roleta, que consiste em escolher os pais mais aptos ou não, a fim de manter a diversidade da população. Os valores da matriz de seleção (máscara) variam entre 0 ou 1, como ilustra a Figura 18.

Figura 18: Matriz de seleção / máscara

2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	
1	1	0	0	1	12
0	1	1	0	0	34
1	0	0	1	0	56

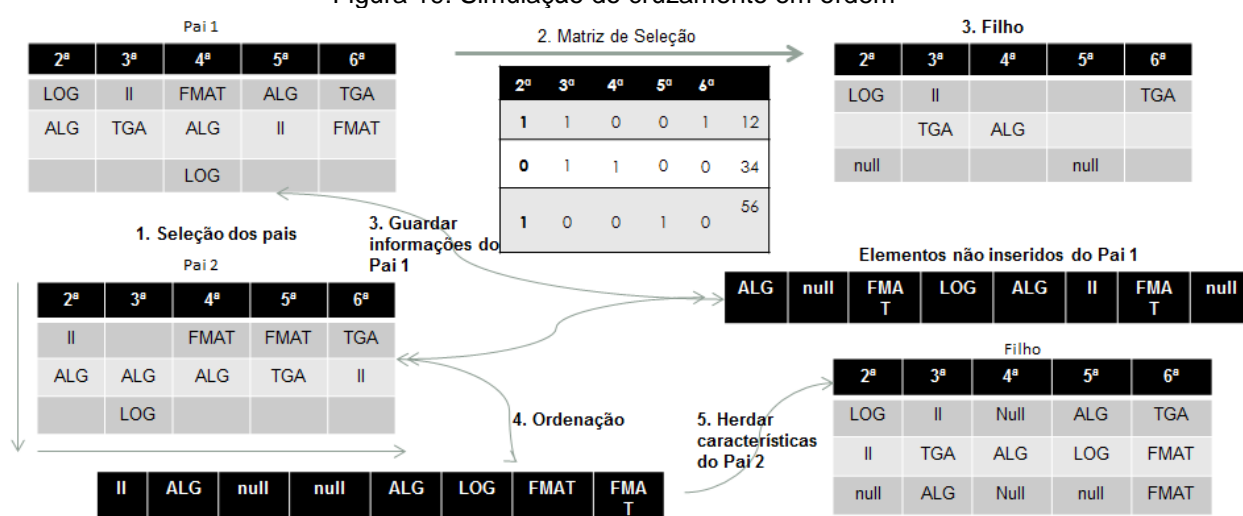
Fonte: elabora pela autora.

De acordo com o conceito do cruzamento em ordem exposto na Seção 2.2.6, quando o valor da posição i da matriz é igual a 1, indica que o filho herda o gene do pai1, quando 0, do pai2. Entretanto o uso da máscara não é suficiente para

gerar o filho de forma correta, devido ao fato de que características importantes dos pais podem ser perdidas ao longo das gerações, ou gerar anomalias.

Como solução, foi utilizado um vetor auxiliar, de tamanho dinâmico, onde são alocadas as informações do pai1 que não foram inseridas no filho. Em seguida, o vetor é ordenado de acordo com que os elementos do vetor aparecem no pai2, desta forma o vetor passa a ter as características do pai2. Após a ordenação do vetor, o filho herda as características do pai2. A Figura 19 ilustra como se dá o processo de cruzamento em ordem utilizado no algoritmo.

Figura 19: Simulação do cruzamento em ordem



Fonte: elaborada pela autora.

Após o processo de geração de um filho, o mesmo é submetido ao operador de mutação. O uso da mutação é justificado para inibir soluções proibidas conforme descrito na Seção 2.2.4.

A mutação utilizada foi à baseada em ordem por meio da permutação de genes do mesmo período. Além disto, este operador sofreu uma adaptação ao problema, devido a algumas restrições relacionadas ao mesmo.

Existem algumas restrições que devem ser respeitadas durante o processo de montagem de horário, sendo assim o operador de mutação ficou responsável em verificar se o indivíduo gerado não infringe nenhuma dessas restrições.

Como ocorre na biologia, onde uma mutação pode gerar uma ou mais anomalias ao indivíduo, no algoritmo proposto neste trabalho, o operador de

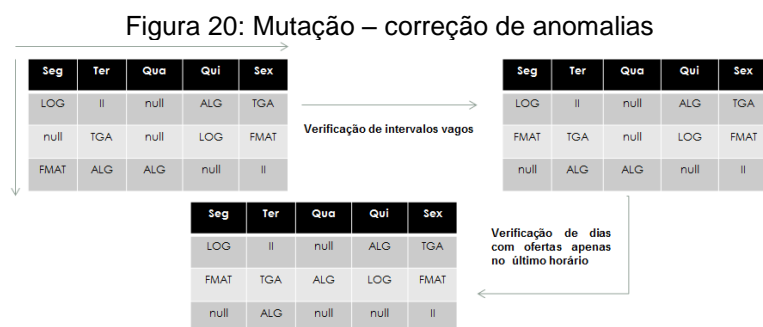
mutação atua na correção de anomalias que podem ter sido geradas durante o cruzamento dos indivíduos.

Sendo assim, a mutação tem como principal objetivo a correção de cromossomos mal formados. Porém, como o objetivo deste trabalho é gerar horários com uma blocagem igualitária, que possa favorecer as necessidades do corpo discente, também ficou agregada à mutação a responsabilidade de deixar os horários com uma distribuição de blocos mais uniforme, diminuindo o intervalo entre aulas.

A primeira verificação realizada pelo operador de mutação é detectar se existem mais de um bloco de aulas de uma turma no mesmo dia. Esta verificação é realizada em todos os dias e períodos do indivíduo. Em seguida, é feita correção de horários quanto a intervalos vagos entre aulas. Em caso da ocorrência desses intervalos, foi adotada a permutação entre os blocos, referentes ao mesmo dia, eliminando o intervalo vago.

Uma vez que o algoritmo proposto neste trabalho visa contribuir para melhorar o rendimento acadêmico do corpo discente, se fez necessário priorizar algumas características do corpo discente. O curso de BSI possui uma porcentagem significativa de alunos de outras cidades, os quais nem sempre podem permanecer até o fim do sexto horário. Esse fato motivou a eliminação da ocorrência de dias com disciplinas ofertadas apenas nos últimos horários. Assim como ocorre quando encontrado intervalos vagos, a mutação faz uma permutação entre os blocos referentes ao mesmo dia, alocando a turma em um bloco anterior.

Logo, o operador de mutação buscou priorizar a blocagem de horários, evitando intervalos vagos entre aulas e distribuindo igualmente a carga horária ao longo da semana, de forma a favorecer o rendimento acadêmico do discente, como é ilustrado na Figura 20.

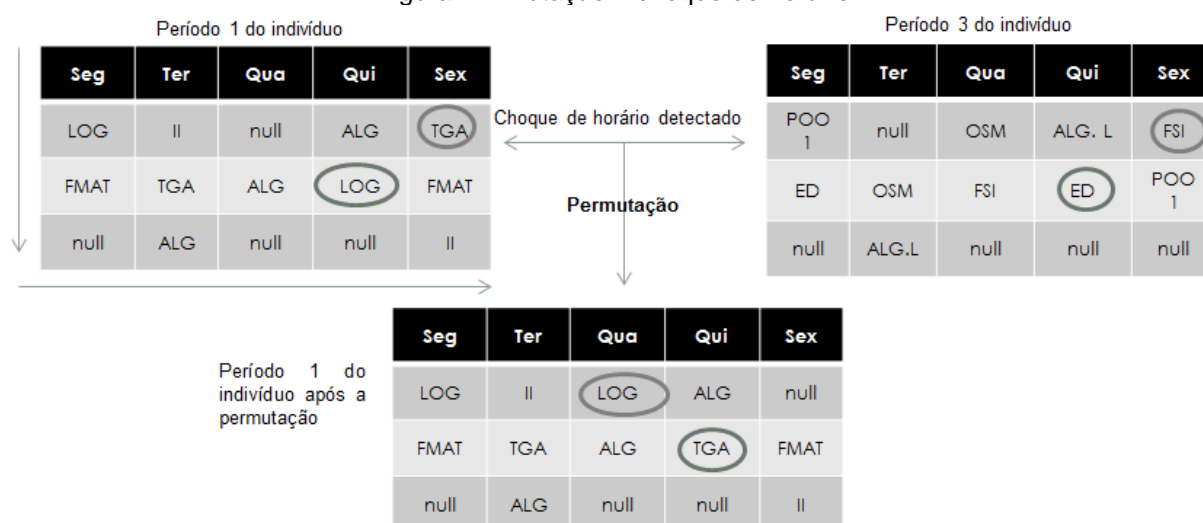


Fonte: elaborada pela autora.

Posteriormente, a mutação busca adequar os horários às preferências dos professores, quando possível, visto que o foco deste trabalho está na blocagem de horários quanto ao corpo discente e não ao corpo docente. Entretanto, para garantir que uma montagem de horários é bem elaborada, é necessário respeitar todas as restrições associadas ao problema. Por exemplo, o mesmo professor não pode ministrar duas disciplinas distintas simultaneamente. Consequentemente, o operador de mutação também é responsável pela verificação de choque de horários entre professores.

Por exemplo, supondo que as disciplinas lógica (LOG) e estrutura de dados (ED) são ministradas pelo mesmo professor. O mesmo acontece com as disciplinas teoria geral da administração (TGA) e fundamentos de sistemas de informação (FSI). Ao detectar que há choque de horário é feita uma permutação em um dos períodos, como mostra a Figura 21.

Figura 21: Mutação – choque de horário



Fonte: elaborada pela autora.

Como é mostrada na figura, a verificação de choque de horários é realizada da seguinte forma: um período é tomado como base e para cada posição desta matriz é comparada com as demais verificando se a turma não possui o mesmo professor. O mesmo procedimento é realizado para os demais períodos, respeitando-se a política adotada pelo curso quanto ao oferecimento das disciplinas. Ao detectar-se o choque de horário, são armazenadas em estruturas de dados auxiliares as informações referentes às turmas e as posições as quais foram

encontradas. Para correção deste problema, são realizadas permutações entre os blocos e comparados às informações nas estruturas auxiliares para que a permutação realizada não gere um novo choque de horário.

3.3. Função de Avaliação

Após a geração do filho, por meio da submissão deste indivíduo aos operadores genéticos, é necessário agregar ao mesmo um valor de aptidão, uma vez que cada indivíduo é visto como uma possível solução.

Posteriormente, a pesquisa determinou uma função de avaliação que agrega ao indivíduo sua aptidão. É importante ressaltar que, o valor de aptidão do indivíduo influencia na seleção dos indivíduos da população a serem submetidos ao operador genético de cruzamento a fim de gerar novos filhos.

A função de avaliação é dada pela soma de punições quando o indivíduo infringe alguma restrição. A punição se faz necessária, pois os indivíduos gerados podem infringir alguma restrição associada ao problema e não corresponder às preferências dos professores. Por se tratar de um problema de otimização, foi utilizada a minimização como tipo de otimização. Sendo assim, quanto menor o valor de aptidão, melhor é o indivíduo.

Os critérios adotados como punição a serem agregados ao valor de aptidão do indivíduo foram baseados nos objetivos do presente trabalho, focando na blocagem de horários a favor do corpo discente. Os critérios e a justificativa de seus usos são explanados a seguir:

- a. Último horário – como já foi descrito anteriormente, o corpo discente do curso de BSI possui alunos de outras cidades não residentes na cidade do campus, com isso eles dependem de transportes escolares que possuem horários definidos de chegada e saída. Sendo assim, se um dia oferecer apenas uma disciplina no último horário, pode resultar em um longo tempo ocioso por parte do aluno e/ou o mesmo poderá ter a necessidade de sair da sala de aula antes que a mesma tenha terminado, consequentemente o aluno perderá uma parte da explanação do conteúdo ministrado;
- b. Aulas vagas – uma vez que grande parte dos alunos não estão envolvidos em outras atividades acadêmicas que possam ocupar seu espaço de tempo integralmente, a ociosidade gera certo desinteresse em participar da aula

seguinte em caso de um grande intervalo de tempo (equivalente há duas horas-aulas ou mais) entre aula;

- c. Aulas de uma mesma disciplina no mesmo dia – o curso de BSI possui uma grade curricular bem diversificada quanto aos tipos de disciplinas. De acordo com estudos pedagógicos recentes (GARCIA *et al*, 2014), aulas de uma mesma disciplina no mesmo dia, além de cansativas, induz o aluno à desmotivação. É evidente que durante o curso existem disciplinas que requerem mais tempo e, por isso, torna-se necessário a ocorrência de mais aulas, o que pode até ser motivador, a depender do perfil do aluno. Entretanto estas disciplinas não podem ser consideradas como referências para as demais, além do fato de poderem ser divididas em vários blocos de duas aulas;
- d. Preferência dos professores – geralmente esse é um dos critérios mais importantes para a montagem de horários. Todavia, este requisito foi atendido na medida do possível, uma vez que os professores, a priori, possuem dedicação exclusiva. Assim, a questão da preferência dos professores foi tratada como uma punição, quando o algoritmo não puder atender a este requisito. Por fim, vale salientar que o foco deste trabalho é a criação de horários focados na sequência didática iterativa pedagógica e não às restrições dos professores;
- e. Choque de horário – o choque de horário é uma restrição primordial na montagem de horários, apesar da mutação tentar corrigir essa ocorrência, caso ainda persista, uma punição é associada ao horário.

As punições anteriormente citadas estão numa mesma escala numérica, e o valor determinado para cada restrição quebrada é acrescido o valor 1, todas as vezes que isso ocorrer. Sendo assim a função de avaliação de um indivíduo é obtida de acordo com a Equação 2.

$$F(i) = \sum_{p=1}^x (a_p + v_p + u_p + pf) + (k * ch) \quad (2)$$

Onde:

p = período

x = número total de períodos;

a = número de aulas vagas no primeiro horário, desde que possua aula no segundo horário;

v = número de aulas vagas entre aulas;

u = número de aulas que são ofertadas apenas no último horário em determinado(s) dia(s);

pf = número de infrações quanto a disponibilidade dos professores;

ch = número de choque de horários nos períodos;

k = peso atribuído ao choque de horários.

3.4. Implementação

Para implementação da proposta, a linguagem escolhida foi Java. Essa escolha seu deu em função da possibilidade do algoritmo proposto neste trabalho vir a ser utilizado em um sistema web, que possui um conjunto de ferramentas mais apropriadas para este tipo de problema.

A linguagem Java possui a disposição dos seus desenvolvedores um conjunto de bibliotecas que implementam soluções para problemas baseados em algoritmos genéticos. Entretanto, para implementação do algoritmo proposto neste trabalho não foi utilizada nenhuma das bibliotecas disponíveis, visto que para implementar o algoritmo se faria necessário realizar modificações em alguns métodos, a saber, os métodos relativos ao cruzamento e mutação.

Uma vez que os métodos a serem utilizados deveriam realizar operações específicas para o problema aqui descrito, a fim de gerar soluções que estivessem de acordo com o objetivo do presente trabalho, optou-se pela implementação completa da solução. Consequentemente, os métodos e classes utilizados no algoritmo foram criados de acordo com a necessidade de encontrar uma solução aceitável para o problema de escalonamento de horários, tendo em vista os objetivos deste trabalho.

O algoritmo segue a sequência lógica ilustrada na Figura 4. Após a geração da população inicial, foram aplicados os operadores de cruzamento e mutação para gerar novos indivíduos, que foram avaliados e acrescentados a uma nova população. Posteriormente a população corrente foi totalmente substituída pela nova população. Este processo se repetiu de acordo com o número de gerações resultando, portanto, no fim da busca por uma solução aceitável para o problema.

As informações necessárias para a execução do algoritmo foram: as turmas que seriam oferecidas de acordo com seus períodos e obedecendo a ordem

dos períodos quanto ao semestre corrente, de acordo com a política adotada pelo curso; e os parâmetros que definiam o tamanho da população e a quantidade de gerações. Quanto aos dados referentes às disciplinas e professores, foi utilizado o MySQL para armazenar estas informações e recupera-las quando necessário.

Para efeito de comparação do algoritmo proposto com a solução adotada anteriormente, foram tomados como base os horários dos semestres anteriores montado de forma manual. As informações obtidas desses horários referem-se às disciplinas ofertadas em cada semestre, e aos professores que as ministraram.

A cada iteração, os valores de aptidão do pior e melhor indivíduo da população eram mostrados como saída do algoritmo, e a média da população de tamanho igual a 100, como foi citado anteriormente, é obtida de acordo com a Equação 3.

$$M_p = \frac{\sum_{i=0}^{n-1} A_i}{n} \quad (3)$$

Onde:

M_p = aptidão média da população;

A = valor de aptidão do indivíduo;

n = tamanho da população.

Ademais, para fim de acompanhar a evolução do algoritmo, também é calculado a aptidão média das gerações a cada 100 iterações. Essa média é obtida de acordo com a Equação 4.

Os parâmetros genéticos utilizados foram: o tamanho da população, que foi escolhido a partir de sugestão da literatura, de valor igual a 100; e a quantidade de gerações, que foi definida com valor igual a 2000 para todas as simulações. É importante destacar que, embora seja possível que o algoritmo convirja (encontre a solução mais aceitável) antes de completar o número de geração predefinido, isto não afeta a solução do problema, pois o melhor indivíduo encontrado é armazenado e mostrado ao final da execução.

4. RESULTADOS E CONCLUSÕES

Os resultados foram obtidos através de simulações de horários geradas pelo algoritmo proposto no presente trabalho, que foram comparados aos horários montados anteriormente pelo coordenador do curso.

O mais relevante da realização deste trabalho é que, o algoritmo atendeu às diretrizes dos algoritmos genéticos, cumprindo com os objetivos apresentados anteriormente, além de respeitar as restrições associadas ao problema de escalonamento de horário. Consequentemente, o experimento provou ser possível encontrar diversas propostas de horários que podem ser utilizadas como soluções para o problema de escalonamento de horários tendo com foco principal a blocagem de horários, diferenciando-se das demais pesquisas realizadas na área.

As principais características do algoritmo que destacam a sua importância em diferenciar-se das demais soluções existentes na área, refere-se ao fato do algoritmo explorar o funcionamento dos operadores genéticos de cruzamento e, principalmente, o de mutação (APÊNDICE C) dando ênfase ao objetivo do trabalho. Em especial, a diferença se dá porque geralmente a mutação é realizada de acordo com uma taxa de mutação. Ao contrário disso, no algoritmo proposto, os indivíduos gerados após o cruzamento sempre são submetidos à operação de mutação, uma vez que a mutação age como corretor de anomalias e adaptação às preferências dos professores.

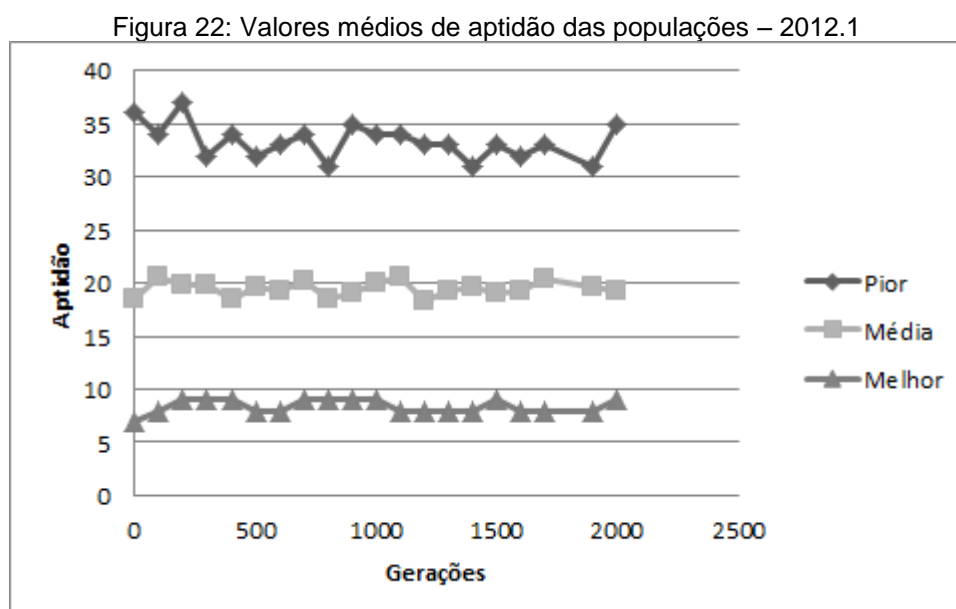
Quanto a função de avaliação, esta calcula as infrações encontradas numa possível solução, agregando o valor de aptidão ao indivíduo por meio de punições. Para fim de simulação e comparação, foram utilizados os horários anteriormente gerados de forma manual (ANEXO B), sendo assim foram simulados horários para os semestres de 2012.1 a 2015.2. Para todas as simulações o tamanho da população foi igual a 100 e a quantidade de gerações igual a 2000.

A partir de cada execução foi gerado gráficos que mostram o desempenho do algoritmo ao longo das execuções. Os dados extraídos, após a execução do algoritmo referente a cada semestre, correspondem aos valores dos piores e melhores indivíduos, e a média das populações a cada 100 gerações. Além desses dados, também foram extraída a médias das gerações a cada 100 iterações, como será detalhado adiante.

4.1 Geração de Horário para o Semestre 2012.1

A Figura 22 mostra a variação de aptidão dos indivíduos durante a evolução ao longo das gerações. Foram utilizados os valores de menor, maior e a média de aptidão dos indivíduos, a fim de mostrar os resultados obtidos.

É importante saliente que apesar dos piores valores sofrerem mais alterações que os de melhor aptidão, os melhores indivíduos têm valores bem aproximados, o que gera uma média grandes variações dos valores da média de aptidão ao longo das gerações.

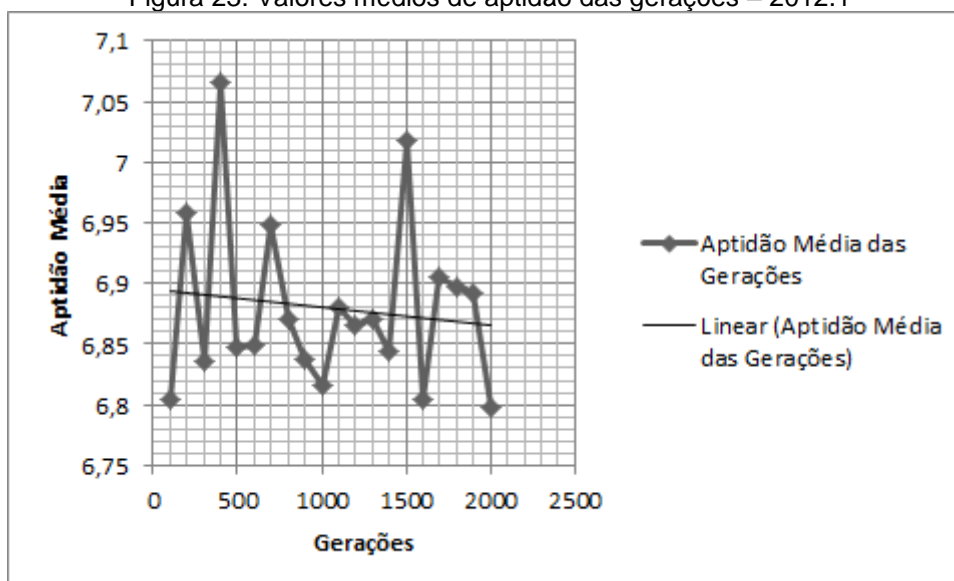


Fonte: elaborada pela autora.

Além de mostrar a variação dos valores de aptidão dos indivíduos, também deve ser levada em consideração a média de aptidão das gerações, a fim de mostrar o desempenho do algoritmo ao longo das iterações, como ilustra a Figura 23.

A variação das médias das avaliações mostra que ao longo das gerações, o espaço de busca é explorado amplamente, não ficando restrito a um mínimo ou máximo local.

Figura 23: Valores médios de aptidão das gerações – 2012.1



Fonte: elaborada pela autora.

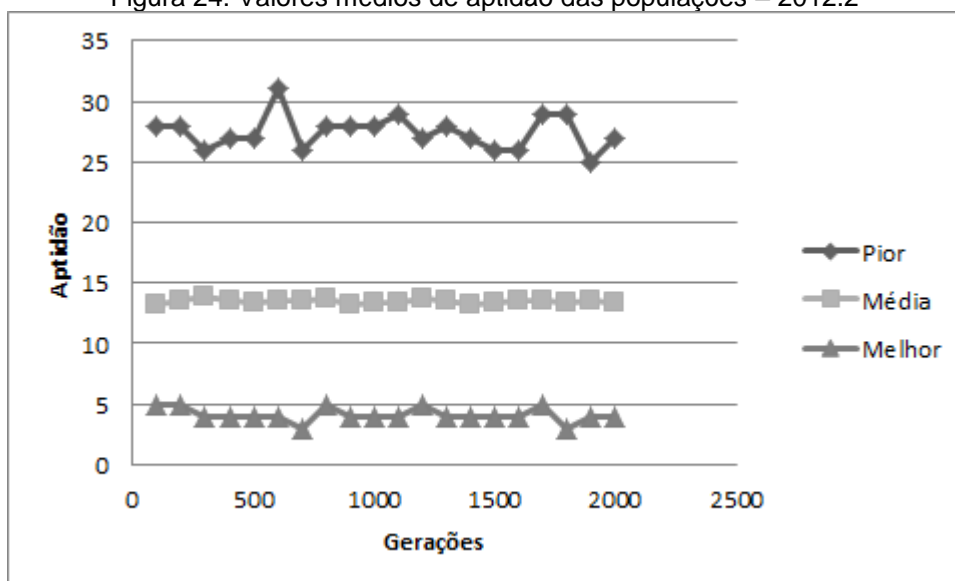
Evidentemente que, os operadores genéticos influenciam diretamente no valor de aptidão do indivíduo, visto que, quanto menor o seu valor de aptidão, melhor é o indivíduo. E que, apesar da variação dos valores dos piores quanto aos valores dos melhores indivíduos chega a ser o dobro ou mais, o valor médio de aptidão dos indivíduos na população não sofrem grandes alterações.

4.2 Geração de Horário para o Semestre 2012.2

A Figura 24 mostra os valores de aptidão obtidos na simulação do semestre 2012.2. A aptidão dos melhores indivíduos não sofrem grandes variações. Isto implica que a média das gerações não sofrem grandes variações ao longo das gerações, como ocorrem aos indivíduos de pior aptidão.

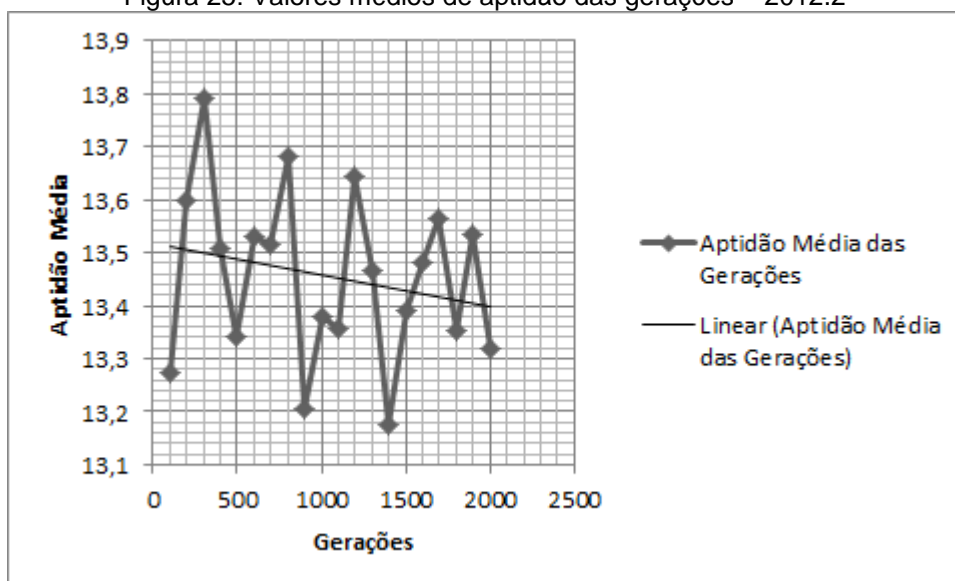
Ao longo das gerações, o valor médio de aptidão das gerações sofrem várias alterações, como ilustra a Figura 25. Assim como ocorre no desempenho dos algoritmos genéticos em várias outras aplicações, o gráfico mostra que, ao longo das gerações ocorre uma diminuição dos valores médios, isso acontece porque os algoritmos genéticos possuem como característica a exploração do espaço de busca total. Sendo assim, apesar das variações, é possível gerar indivíduos como uma possível solução.

Figura 24: Valores médios de aptidão das populações – 2012.2



Fonte: elaborada pela autora.

Figura 25: Valores médios de aptidão das gerações – 2012.2

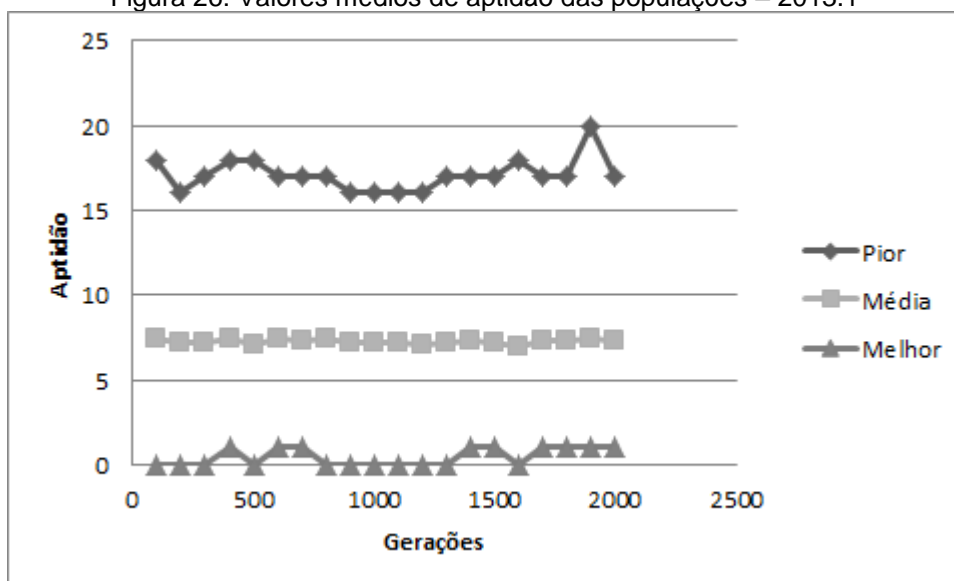


Fonte: elaborada pela autora.

4.3 Geração de Horário para o Semestre 2013.1

Nos dados obtidos na simulação referente ao semestre 2013.1, pode se observar que os primeiros indivíduos possuem valores bem aproximados, em outras palavras, os melhores indivíduos que são mais propícios a serem selecionados para a reprodução, possuem o valor de aptidão bem próximo como mostra a Figura 26.

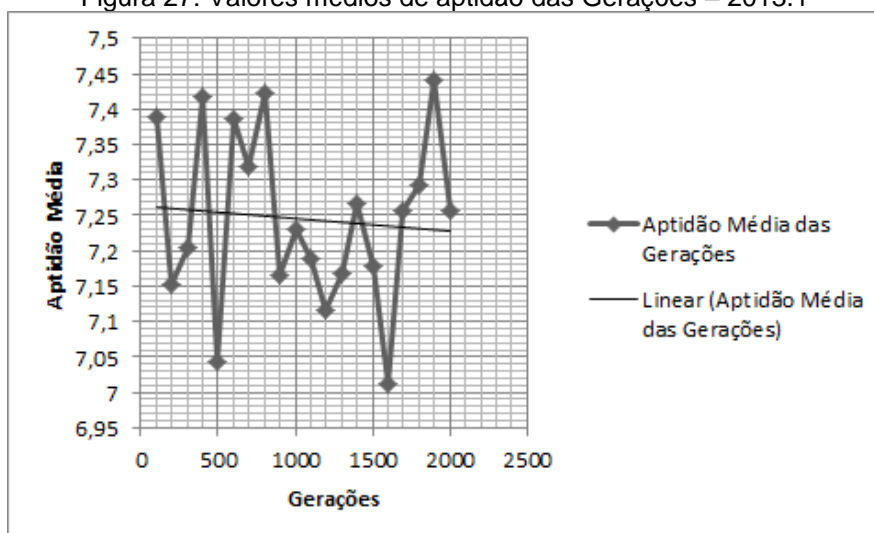
Figura 26: Valores médios de aptidão das populações – 2013.1



Fonte: elaborada pela autora.

Apesar dos indivíduos apresentarem aptidão semelhante, ao longo das 300 primeiras gerações observou-se que houve um crescimento significativo quanto à média dos indivíduos. Esse comportamento deve-se ao fato dos primeiros indivíduos, apesar de terem valores de aptidão baixos, os mesmos não estavam adequados ao problema, e ao longo das demais gerações, é possível identificar que o operador de mutação conseguiu adaptar os indivíduos se tornarem possíveis soluções mais aceitáveis. Este comportamento pode ser visto na Figura 27.

Figura 27: Valores médios de aptidão das Gerações – 2013.1

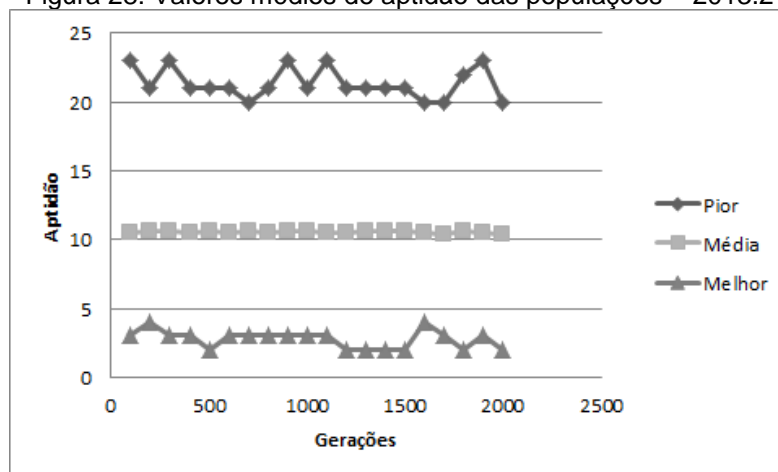


Fonte: elaborada pela autora.

4.4 Geração de Horário para o Semestre 2013.2

Como pode ser observado nas simulações anteriores aos valores de aptidão possuíam valores mais distribuídos, entretanto, na simulação do semestre 2013.2 observou-se que os valores obtidos estão mais aglomerados. Neste caso pode-se observar que os fatores e restrições associadas ao problema influencia diretamente no valor de aptidão dos indivíduos, como mostra a Figura 28.

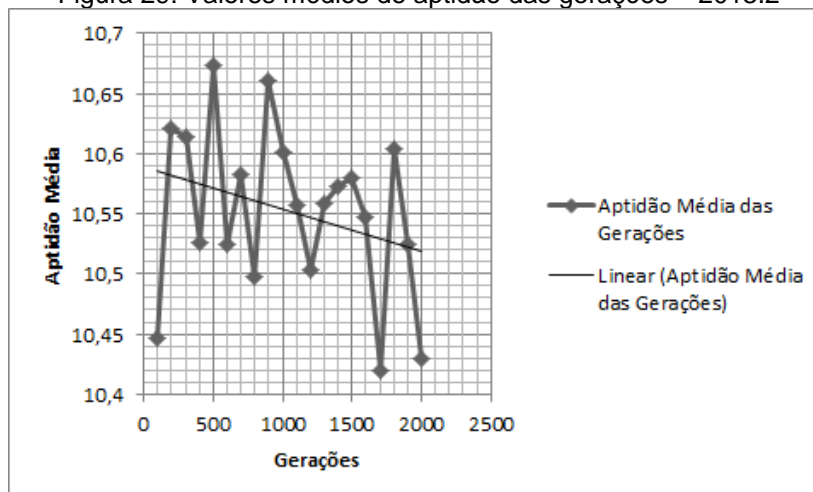
Figura 28: Valores médios de aptidão das populações – 2013.2



Fonte: elaborada pela autora.

Devido aos fatores descritos anteriormente, a Figura 29 ilustra a variação dos valores médios de aptidão das gerações ao longo das iterações realizadas pelo algoritmo.

Figura 29: Valores médios de aptidão das gerações – 2013.2

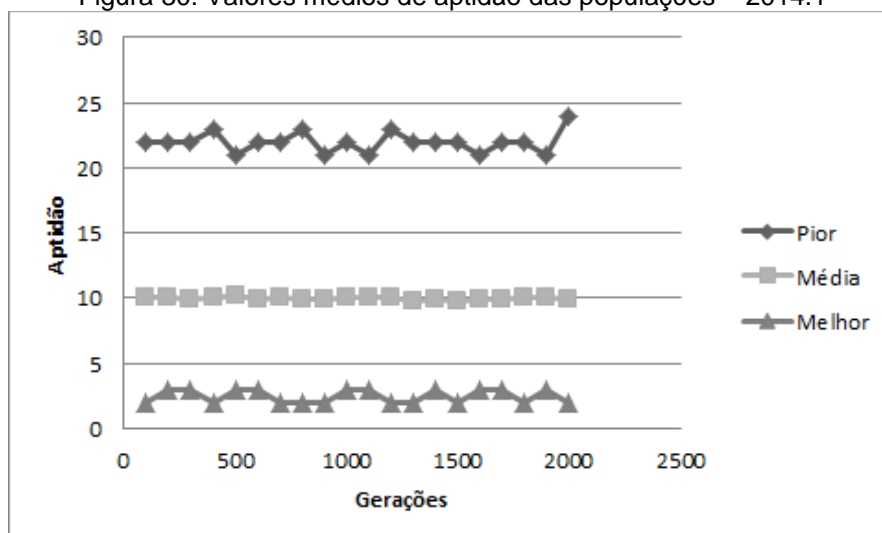


Fonte: elaborada pela autora.

4.5 Geração de Horário para o Semestre 2014.1

Os dados obtidos para simular uma proposta de solução aceitável para o semestre 2014.1 mostra uma variação dos valores de aptidão entre 7 e 34. Uma amostra dos dados foi utilizada para gerar um gráfico que é ilustrado na Figura 30.

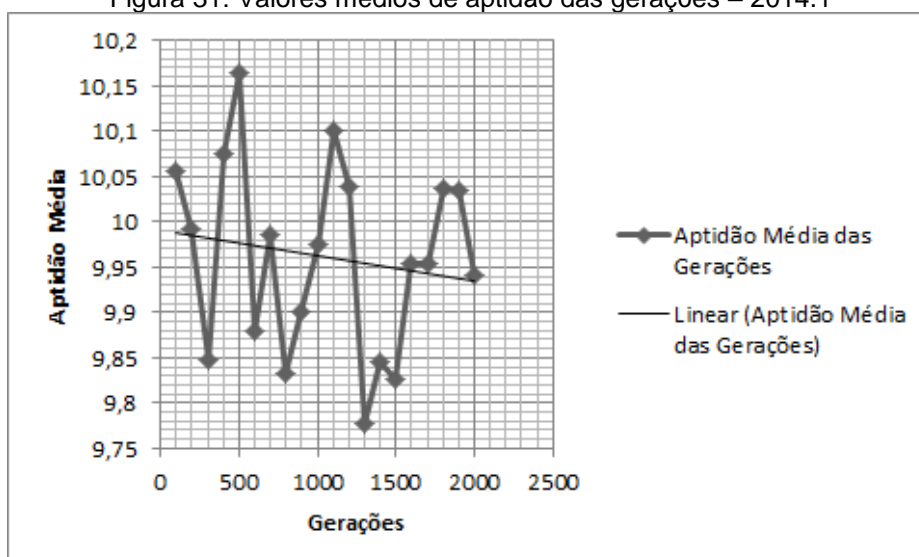
Figura 30: Valores médios de aptidão das populações – 2014.1



Fonte: elaborada pela autora.

Nesta simulação, os dados estavam em constante variação como mostra a Figura 31. Apesar dessas variações, o desempenho do algoritmo não se mostrou negativo quanto à convergência e a busca por uma solução aceitável.

Figura 31: Valores médios de aptidão das gerações – 2014.1

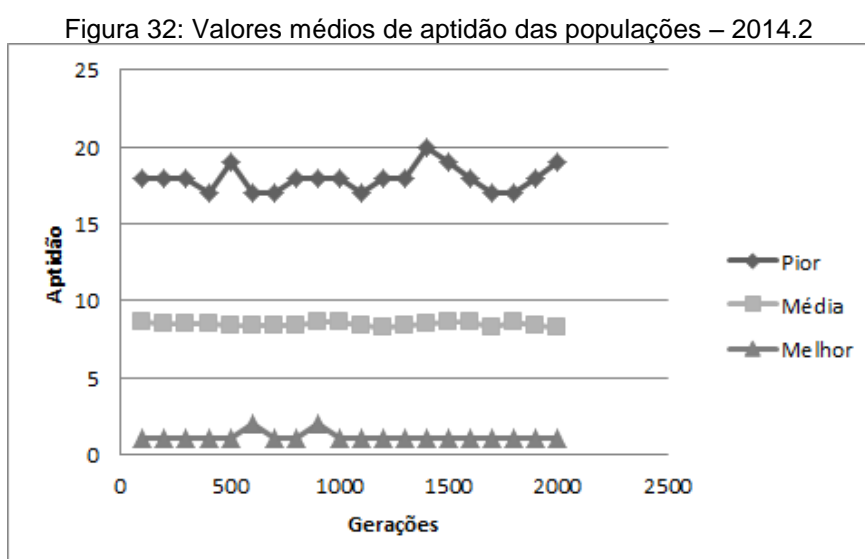


Fonte: elaborada pela autora.

Para esta simulação houve algumas adaptações de acordo com a política de oferta de disciplinas adotadas pelo curso de BSI. Essas adaptações referem-se a grande oferta de disciplinas optativas. É importante ressaltar que os semestres ímpares, por serem os semestres de entrada de novos alunos, a oferta de disciplinas se torna maior que em semestres pares.

4.6 Geração de Horário para o Semestre 2014.2

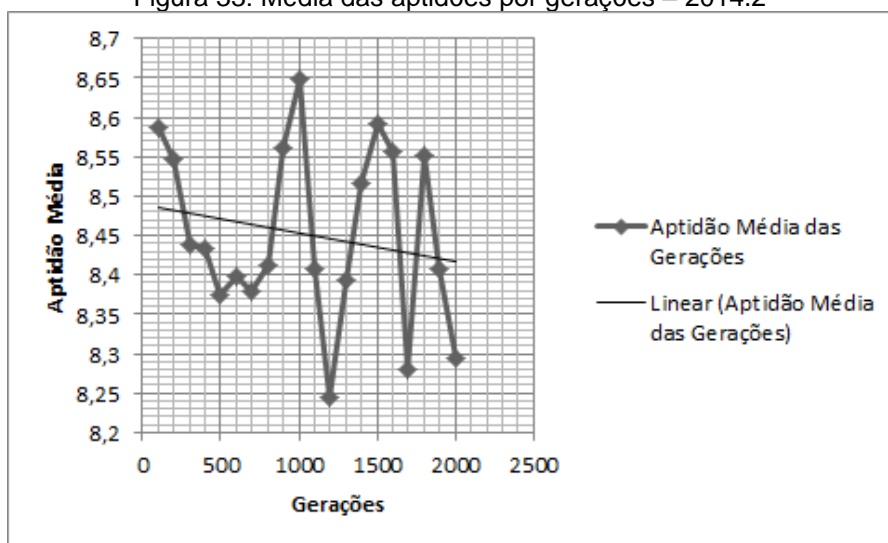
Outra simulação realizada pelo algoritmo foi quanto a encontrar uma solução aceitável, de acordo com os objetivos do presente trabalho, respeitando as informações obtidas pela solução feita de forma manual referente às disciplinas e aos professores que as ministraram. Uma amostragem dos valores de aptidão dos indivíduos obtidos nesta simulação está disposta na Figura 32.



Fonte: elaborada pela autora.

A Figura 33 ilustra a média dos valores de aptidão dos indivíduos ao longo das gerações. É importante salientar que o gráfico mostra máximos e mínimos locais, e apesar disto ocorrer, o algoritmo não ficou restrito a esse pontos, pelo contrário, o algoritmo continuou pela busca de uma solução aceitável até o fim da execução do algoritmo.

Figura 33: Média das aptidões por gerações – 2014.2



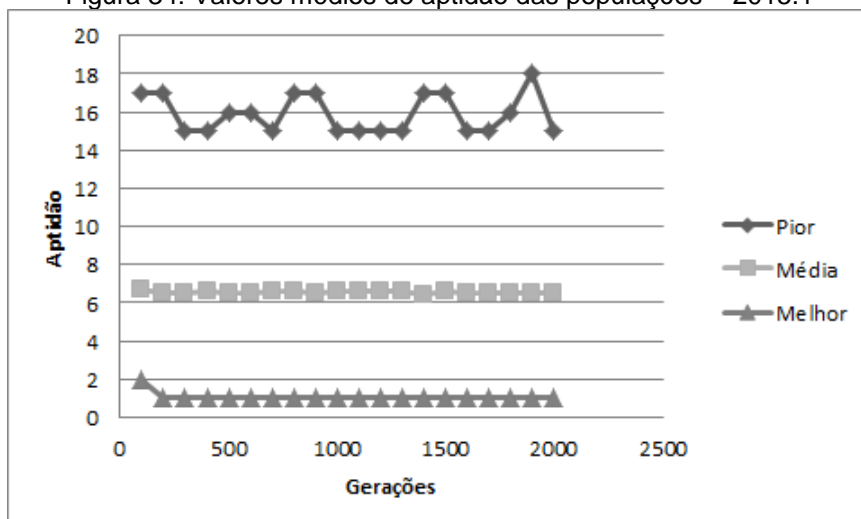
Fonte: elaborada pela autora.

Assim como nas simulações anteriores, as Figuras 32 e 33 levam em consideração o pior e o melhor indivíduo de uma população e a média de aptidão da população.

4.7 Geração de Horário para o Semestre 2015.1

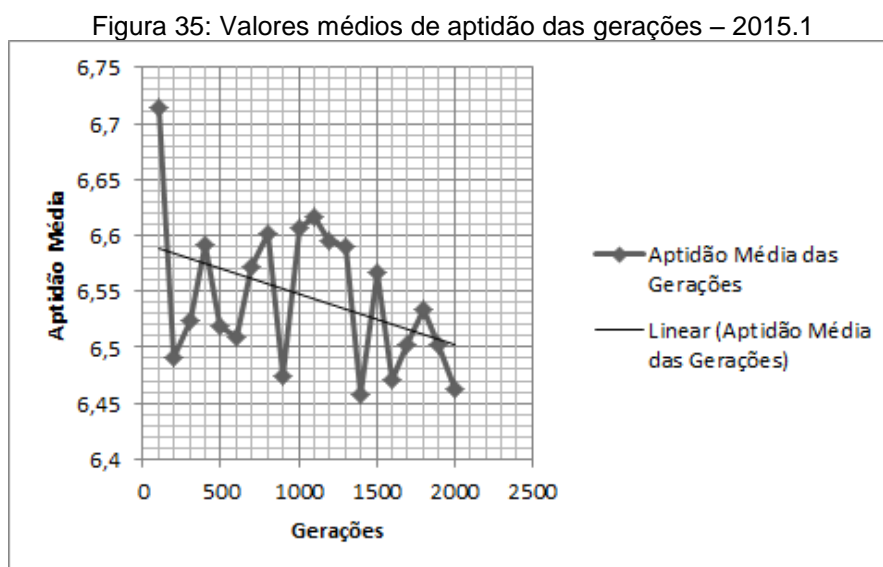
Outra simulação efetuada pelo algoritmo refere-se ao semestre 2015.1. Como as simulações anteriores, os primeiros indivíduos possuem seus valores de aptidão bem distribuídos ao longo das primeiras gerações como pode ser vista na Figura 34.

Figura 34: Valores médios de aptidão das populações – 2015.1



Fonte: elaborada pela autora.

É importante observar que ao longo das gerações é comum encontrar pontos mínimos e máximos, como mostra a Figura 35. Apesar da variação ao longo das gerações, há uma tendência de redução da aptidão, visto que trata-se de uma otimização por minimização.



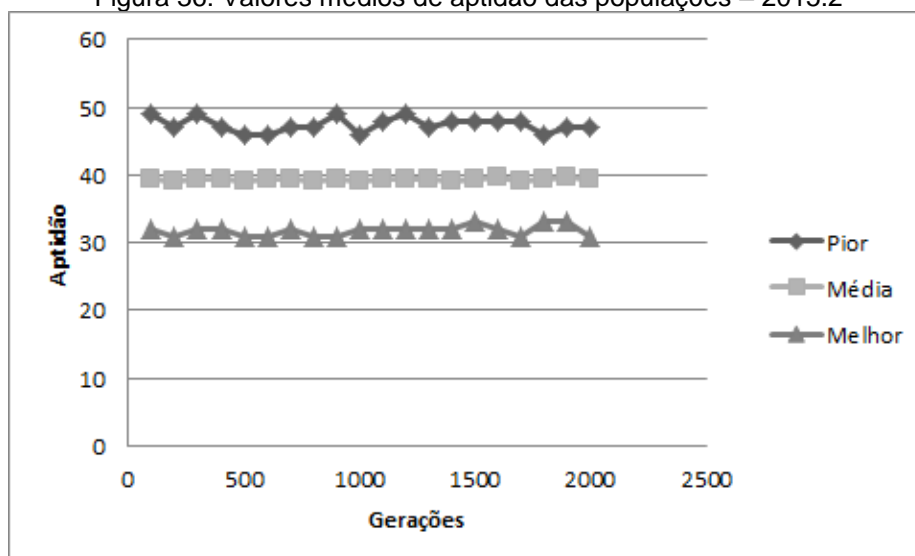
Fonte: elaborada pela autora.

4.8 Geração de Horário para o Semestre 2015.2

Para simular uma solução aceitável para o semestre 2015.2 existem algumas peculiaridades que não existiam das simulações anteriores. Na figura 36 mostra que os valores de aptidão são superiores em comparação as simulações anteriores. Isso se deve ao grande número de restrições associadas a este horário, em outras palavras, a todos os professores existiam preferências associadas, o que se fosse levado em consideração na população inicial dificultaria a execução do algoritmo, ou até mesmo tornando impossível encontrar uma solução qualquer.

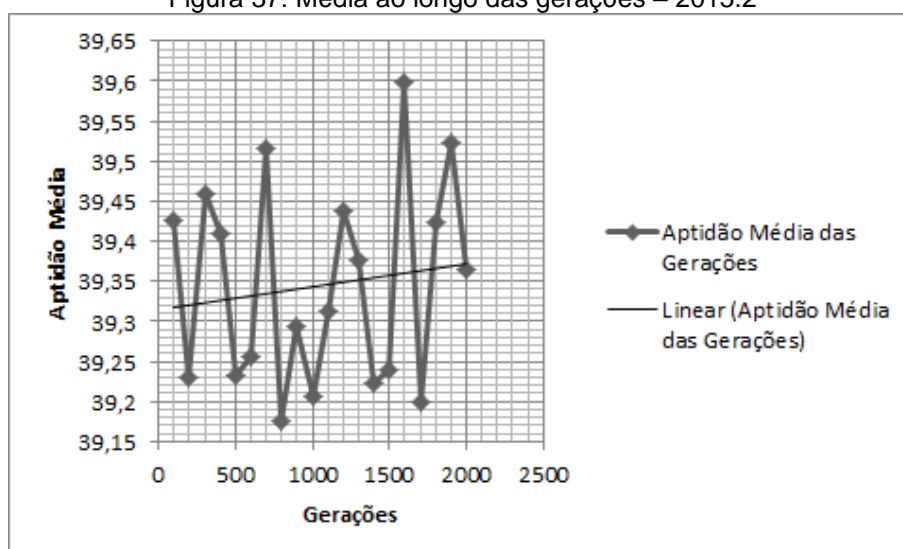
A Figura 37 apresenta os valores médios de aptidão dos indivíduos, obtidos a cada 100 gerações, como foi citado na Seção 3.4. De acordo com o gráfico, é possível perceber o valor médio de aptidão, de uma forma geral, apresenta uma tendência de queda ao longo das gerações, demonstrando que o algoritmo tende a obter indivíduos mais aptos com o passar das gerações. Entretanto, é possível também perceber oscilações nos valores de aptidão demonstrando a não restrição a um espaço de busca local.

Figura 36: Valores médios de aptidão das populações – 2015.2



Fonte: elaborada pela autora.

Figura 37: Média ao longo das gerações – 2015.2



Fonte: elaborada pela autora.

Para melhor detalhamento dos valores obtidos o APÊDICE D é composto pelos valores de aptidão utilizados nas gerações dos gráficos ilustrados acima.

4.9 Avaliação

A avaliação dos horários gerados pela proposta foi realizada a partir da comparação entre a solução obtida por meio do algoritmo proposto neste trabalho e com os horários produzidos de forma manual para o curso de BSI oferecido no

âmbito do Centro de Ensino Superior do Seridó, para os semestres de 2012.1 a 2015.2.

A fim de validação do algoritmo, foi realizada comparações entre os valores de aptidão das devidas soluções encontradas pelo algoritmo com os horários anteriores (ANEXO B) e com um *software* (ASC *timetable*) como mostra a Tabela 2.

Tabela 2: Comparação das aptidões das soluções

Semestre	Aptidão Manual	Aptidão Algoritmo	
2012.1	19	2	População Final
		0	Melhor das Gerações
2012.2	21	6	População Final
		3	Melhor das Gerações
2013.1	23	3	População Final
		0	Melhor das Gerações
2013.2	25	5	População Final
		2	Melhor das Gerações
2014.1	23	3	População Final
		2	Melhor das Gerações
2014.2	12	4	População Final
		1	Melhor das Gerações
2015.1	24	3	População Final
		1	Melhor das Gerações
2015.2	17	34	População Final
		17	Melhor das Gerações

Fonte: elaborada pela autora.

De acordo com os resultados apresentados, conclui-se que o algoritmo proposto consegue gerar soluções aceitáveis em um período de tempo bastante inferior ao gasto se feito de forma manual, o que manualmente gastaria dias para encontrar uma solução, o algoritmo encontra soluções entre 1 minuto e 30 segundos a 1 minuto e 42 segundos. Ademais, ele auxilia o coordenador no processo de montagem de horários, permitindo também que seja feita várias simulações que implica em alterações de uma proposta de horário gerada anteriormente.

Quanto ao *software*, especificamente, os horários gerados por ele não obedecem à política adotada pelo curso, mesmo que estas fossem inseridas nas restrições. Apesar de ter sido realizada uma simulação para o semestre 2012.1 (ANEXO C), o *software* não encontrou uma solução para o problema. Outro ponto que é importante ressaltar é que o *software* não mostra o horário de acordo com os períodos ou níveis (neste caso é tratado como grupo) que as disciplinas estão inseridas. Desta forma, foi impossibilitado de obter os valores de aptidão para soluções sugeridas pelo software. Ademais, é preciso levar em consideração que o

Asc Timetable é um *software* que tem como objetivo encontrar soluções para cursos anuais que condizem com o ensino fundamental e médio, além de gerar horários principalmente para um único turno.

Embora os resultados encontrados sejam melhores, não houve acesso às restrições as quais os semestres de 2012.1 a 2015.1 foram submetidos para geração do horário de forma manual. Apesar disso, foi possível atingir o foco diferenciado, priorizando a blocagem de horários, o que implica diretamente ou indiretamente no favorecimento do rendimento acadêmico do corpo discente (GARCIA *et al.* 2014). Desta forma, foi possível atingir os objetivos gerais e específicos do presente trabalho.

Outro ponto que é importante ressaltar é que a quantidade de restrições influencia na execução do algoritmo. A ocorrência de muitas restrições, além de gerar soluções com alto valor de aptidão, é possível que o algoritmo entre em um ciclo infinito por não conseguir encontrar uma possível solução. Para resolver este caso se faz necessário ignorar algumas restrições para que o algoritmo consiga executar normalmente.

Por fim, o Apêndice B mostra os horários gerados na simulação referente a cada semestre, as disciplinas que foram ofertadas e os professores responsáveis por ministrar cada disciplina. Em alguns casos, os professores de algumas disciplinas ainda não tinham sido definidos, como solução optou-se por usar três interrogações para simbolizar a indefinição do professor.

Uma das dificuldades encontradas durante a pesquisa foi a falta de pseudocódigos que explicassem como os operadores genéticos utilizados nos trabalhos relacionados. Por este motivo, considero a inclusão do APÊNDICE C como diferencial entre alguns trabalhos relacionados, principalmente aos que utilizam algoritmos genéticos como técnica para encontrar uma solução aceitável para o problema.

A realização deste trabalho não faz jus apenas a contribuição acadêmica quanto a este área de pesquisa, mas também a aplicação de conceitos que não são vistos obrigatoriamente durante o curso de BSI, isso implica na ampliação dos conhecimentos utilizados e uma visão amplificada a respeito das áreas de pesquisa no mundo da computação.

Para trabalhos futuros, sugere-se acrescentar ao algoritmo proposto fatores a serem considerados na montagem de horário. Dentre esses fatores,

incluem-se a disponibilidade das salas e a disponibilidade dos alunos aptos a cursar determinada disciplina e testes de outros tipos de operadores genéticos. Ademais, sugere-se ainda acrescentar à função de avaliação uma punição para dias letivos que não possuam aula, a salvo se a quantidade de disciplinas a serem ofertadas não for suficiente para serem alocadas todos os dias. A inclusão desses últimos fatores implica em proporcionar maior ênfase ao rendimento acadêmico do corpo discente.

REFERÊNCIAS

ABBASZADEH, Mortaza; SAEEDVAND, Saeed; MAYANI, Hamid Asbagi. *Solving university scheduling problem with a memetic algorithm. **International Journal of Artificial Intelligence***, v. 1, n. 2, p. 79-90, 2012.

ABRAMSON, D. *Constructing school timetables using simulated annealing: sequential and parallel algorithms. **Management Science***, v. 37, n. 1, p.98-113, 1991.

AHANDANI, Morteza Alinia; BAGHMISHEH, Mohammad Taghi Vakil. *Memetic algorithms for solving university course timetabling problem. In: 1st International eConference on Computer and Knowledge Engineering. **Anais...*** 2011.

BATESON, Winston. ***Materials for the study of variation***. Londres: Micmillan co. e Nova Iorque, 1894.

BIRBAS, T.; DASKALAKI, S.; HOUSOS, E. *School timetabling for quality student and teachers schedules. **Journal of Scheduling***, v. 12, p. 177-197, 2009.

BRAZ Jr., Osmar de Oliveira. **Otimização de horários em instituições de ensino superior através de algoritmos genéticos**. Florianópolis: UFSC, 2000. Originalmente apresentada como dissertação de mestrado na Universidade Federal de Santa Catarina, 2000.

CAMPOS, Ricardo Luiz B. L. ERM2C: uma metodologia para melhoria do ensino aprendizado de lógica de programação. In: Encontro Regional de Computação Bahia-Alagoas-Sergipe. **Anais...** Maceió: Cesmac, 2010.

CASTRO, Leandro Nunes de. **Computação natural: uma jornada ilustrada**. São Paulo: Editora Livraria da Física, 2010. p.1-70.

CHEN, Ruey-Maw; SHIH, Hsiao-Fang. *Solving university course timetabling problems using constriction particle swarm optimization with local search. **Algorithms***, p. 227-224. In: MDPI. **Anais...** 2013.

COLONI, Alberto; DORIGO, Marco; MANIEZZO, Vittorio. **A genetic algorithm to solve the timetable problem**. Milano: Politécnico de Milano, 1993.

COOPER, Tim B.; KINGSTON, Jeffrey H. ***The complexity of timetable construction problems***. Berlin: Springer Berlin Heidelberg, 1996.

COPPIN, Ben. **Inteligência Artificial**. Rio de Janeiro: LTC, 2010.

DARWIN, Charles Robert. ***On the origin of species***. Londres: John Murray, 1859.

DERAKHSHI, Mohammad-Reza Feiza; BABAEI, Hamed. *A survey of approaches for university course timetabling problem*. In: IMS. **Anais...** Adrasan: Sakarya University, 2012. p. 307-321.

FREDRICH, Ana Paula. **Um estudo sobre algoritmos meméticos e sua eficiência em relação aos algoritmos genéticos**. Cascavel: UNIOESTE, 2010.

GARCIA, Tânia Cristina Meira; SOBRINHO, Djanní Martinho dos Santos; GARCIA, Tulia Fernandes Meira. **Profissão Docente**. Natal: EDUFRRN, 2014. p.89 – 218.

GAREY, M. R.; JOHNSON, D. S. ***Computers and intractability: a guide to the theory of NP-Completeness***. Nova Iorque: WH Freeman & Company, 1990.

GHAEMI, Sehrane; VAKILI, Mohammad Taghi; AGHAGOLZADEH, Ali. *Using a genetic algorithm optimizer tool to solve University timetable scheduling problem*. In: Signal Processing and Its Applications, 2007. 9th International Symposium, Sharjah, United Arab Emirates. **Anais...** Sharjah: IEEE, 2007. p. 1-4.

GLOVER, Fred; TAILLARD, Eric; WERRA, Dominique de. *A user's guide to tabu search*. **Annals of Operations Research**, 1993, v. 41, p 1 – 28.

GOLDBARG, Marco Cesar; LUNA, Henrique Pacca L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: 2000. p. 13-14.

Holland, John Henry. ***Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence***. Ann Arbor: University of Michigan Press, 1975.

JAT, Sadaf Naseem; YANG, Shengxiang. *A guided search genetic algorithm for the university course timetabling problem*. In: MISTA, 2009. **Anais...** 2009. Dublin: MISTA, 2009.

KENNEDY, James; EBERHART, Russel. *Particle swarm optimization*. In: **Encyclopedia of Machine Learning**. Nova Iorque: Springer, 2010. p. 760-766

KIRKPATRICK, Scott; GELATT, Charles D. Jr.; VECCHI, Mario P. *Optimization by simulated annealing*. **Science**, v. 220, n. 4598, p. 671-680, 1983.

LARA, Carlos; FLORES, Juan J.; CALDERÓN, Félix. *Solving a school timetabling problem using a bee algorithm*. **MICAI 2008: advances in artificial intelligence**, v. 5317. Atizapán de Zaragoza: Springer, 2008. p. 664-674.

LINDEN, Ricardo. **Algoritmos genéticos**, 3 ed. Rio de Janeiro: Editora Ciência Moderna, 2012.

LOBO, Eduardo Luiz M. **Uma solução do problema de horário escolar via algoritmo genético paralelo**. Belo Horizonte: CEFET, 2005. Originalmente apresentada como dissertação de mestrado no Centro Federal de Educação Tecnológica de Minas Gerais, 2005.

LUKAS, Samuel; ARIBOWO, Arnold; MUCHRI, Milyandreana. *Genetic algorithm and heuristic search for solving timetable problem – case study: Universitas Pelita Harapan Timetable*. *Applications of Digital Information and Web Technologies*, 2009. *Second International Conference on the, Anais...* Tangerang: UPH, 2009, p.629-633.

MARTE, Michael. **Models and algorithms for school timetabling – a constraints-programming approach**. Munique: LMU, 2002. Originalmente apresentado como tese de doutorado na Universidade de Munique, 2002.

MICHELAN, Roberto; MAIA, Alexandra Carniel Perdigão. **Algoritmos Genéticos**. 2006.

MIRANDA, Marcio Nunes de. **Algoritmos genéticos: fundamentos e aplicações**, 2007.

MOURA, Arnaldo Vieira. *Getting a grip on school timetables*. **International Journal of Operational Research**. Nova Iorque: EurekaAlert, 2010. p. 152-170.

NGUYEN, Khang; LU, Tien; LE, Trung; TRAN, Nuon. *Memetic algorithm for a university course timetabling problem*. **Informatics in control, automation and robotics**. Berlim: Springer, 2011, p. 67-71.

NOGUEIRA, Fernando. **Programação inteira**. Juiz de Fora: UFJF, 2010. Disponível em: <http://www.ufjf.br/epd015/files/2010/06/ProgramacaoInteira.pdf>. Acesso em: 26/abr/2015.

PACHECO, Marco Aurélio Cavalcante. Algoritmos genéticos: princípios e aplicações. **ICA**. Rio de Janeiro: PUC, 1999. Disponível em: < <http://www.ica.ele.puc-rio.br/downloads/38/ce-apostila-comp-evol.pdf> >. Acesso em: 04/set/2014.

PENA, Ana Cerdeira; CARPENTE, Luisa; FARIÑA, Antonio; SECO, Diego. *New approaches for the school timetabling problem*. **MICAI 2008: advances in artificial intelligence**, v. 5317. Atizapán de Zaragoza: Springer, 2008. p. 261-267.

PILLAY, Nelshia. *A survey of school timetabling research*. **Ann Oper Res**, Nova lorque, n. 1, v. 218, p. 261-293, 2014.

POULSEN, Camilo José Bornia. **Desenvolvimento de um modelo para o school timetabling problem baseado na meta-heurística simulated annealing**. Porto Alegre: UFRGS, 2012. Originalmente apresentado como dissertação de mestrado na Universidade Federal do Rio Grande do Sul, 2012.

PRIETCH, Soraia Silva; PAZETO, Tatiana Annoni. Estudo sobre a evasão em um curso de licenciatura em informática e considerações de melhoria. In: ENCONTRO REGIONAL DE COMPUTAÇÃO BAHIA-ALAGOAS-SERGIPE. **Anais...** Maceió: Cesmac, 2010.

RAGHAVJEE, Rushil; PILLAY, Nelishia. *An application of genetic algorithms to the school timetabling problem*. In: *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*. **Anais...** Nova lorque: ACM Digital Library, p. 193-199, 2008.

RAGHAVJEE, Rushil; PILLAY, Nelishia. *Using genetic algorithm to solve the south African school timetabling problem*. In: *Second World Congress on Biologically Inspired Computing*, Kitakyushu, Japão, dec. 2010. **Anais...** Kitakyushu: 2010.

RAGHAVJEE, Rushil; PILLAY, Nelishia. *The effect of construction heuristic on the performace of a genetic algorithm for the school timetabling problem*. In: *Proceedings of the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment*. **Anais...** Nova lorque: ACM Digital Library, 2011 p. 187-194.

RECHENBERG, I. ***Cybernetic solution path of an experimental***. Hants: Royal Aircraft Establishment, 1965.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**, 3ª ed. Rio de Janeiro: Elsevier, 2013.

SIGL, Branimir; GOLUB, Marin; MORNAR, Vedran. *Solving timetable scheduling problem using genetic algorithms*. In: *25th Conference Information Technology Interfaces*. Cavtat, Croácia, p. 519-524, jun. 2003. **Anais...** Cavtat: 2003.

SILVA, Gilson Gomes da. Montagem de horários: depoiment. [22 de maio de 2014]. Caicó: Quais as diretrizes para montagem de horários? Entrevista concedida a Weslane Almeida.

TOLEDO, Maria Inez Melo de. DNA – Material da hereditariedade: processos de reprodução celular. **Centro de Referência Virtual do Professor**. 2008.

TURING, Alan M. Computing machinery and intelligence. **Mind**, v. 59, n. 239, p. 433-460, 1950.

VIANNA, Valdisio. **Meta-heurísticas e programação paralela em otimização combinatória**. Fortaleza: EUFC, 1998.

WINSTON, Patrick Henry. **Artificial Intelligence**. 3 ed. Boston: Addison-Wesley, 1992.

APÊNDICES

APÊNDICE A – Outras Técnicas de Busca

Outras Técnicas de Busca

Além dos algoritmos genéticos, as técnicas a seguir também são utilizadas para resolver problemas de busca e otimização, como no caso do problema de escalonamento de horários. Os conceitos abaixo servem para esclarecer o funcionamento destas técnicas, visto que o uso de cada técnica será visto na seção dos trabalhos relacionados (seção 2.4).

Busca Tabu (*Tabu Search*)

O método da busca tabu foi proposta por Fred Glover em 1986. Ela é caracterizada como uma meta-heurística e um procedimento adaptativo auxiliar que guia um algoritmo de busca local na exploração contínua dentro de um espaço de busca (LINDEN, 2009).

A base da busca tabu é otimizar a busca por uma solução que utilize conceitos de definição do espaço e estrutura da vizinhança, visto que no processo do algoritmo é utilizando uma memória vinculada a cada vizinho para que o algoritmo não fique preso aos máximos e mínimos locais (GLOVER, 1993).

Durante a execução do algoritmo, são realizados movimentos na vizinhança que alteram os valores dos vizinhos, e estes movimentos e valores são salvos na memória para que os movimentos que gerem os mesmos resultados não sejam repetidos. Após armazenar os valores e os movimentos feitos na vizinhança, o algoritmo poderá substituir parte ou toda a vizinhança.

Segundo Glover (1993), o algoritmo converge quando atinge a condição de parada, a qual está associada, basicamente, à combinação de algumas ideias lógicas: quando encontrado uma solução ótima, quando encontrado uma solução pré-definida, e/ou um número de iterações predefinido.

Têmpera Simulada (*Simulated Annealing*)

A técnica de têmpera simulada (SA, do inglês, *Simulated Annealing*) criada por Scott Kirkpatrick em 1983. O autor foi influenciado pelo algoritmo de Metropolis de aproximação numérica simulada, utilizando conceitos físicos da termodinâmica aplicados no tratamento térmico de materiais, buscando a otimização combinatorial na busca por soluções de problemas característicos de alta

complexidade (KIRKPATRICK *et al.*,1983). Nesse processo, o metal é aquecido de forma contínua até a certa temperatura, quando atinge a temperatura ideal é posteriormente resfriado. A capacidade do SA de ser um procedimento genérico de solução faz com que ele seja classificado como meta-heurística.

A ideia do SA é aceitar, de uma forma controlada, movimentos que pioram a solução corrente. Quanto pior for a solução na vizinhança da solução atual, maior a probabilidade de que um movimento para essa região não seja aceito. À medida que as iterações do método evoluem, a probabilidade de o mesmo movimento ser aceito diminui juntamente com um parâmetro chamado “temperatura” (POUSEN, 2012).

O propósito da técnica é melhorar a solução vigente. O algoritmo usa a perturbação da solução corrente (conhecido como estado da solução/energia) dentro de um laço (*loop*) que se mantém enquanto a temperatura corrente seja maior do que uma temperatura mínima. Dentro desse laço há outro que garante que os movimentos de perturbação sejam executados num determinado número de vezes. A cada iteração, o custo da nova solução é comparado ao da solução corrente.

A aceitação de um movimento é definida por meio de uma função de determinação de probabilidades que depende diretamente da temperatura, e é aceita quando a probabilidade for menor que um.

Após atingir a “temperatura” ideal, a solução corrente é comparada com a melhor obtida até o momento. Ao identificar que a melhor solução foi encontrada, a temperatura é diminuída a cada nova execução do algoritmo a uma taxa de resfriamento, a qual é previamente definida pelo programador, fazendo com que a solução encontrada seja mantida até uma condição de parada pré-definida.

Colônia de Abelhas (*Bee Algorithm*)

O algoritmo da colônia de abelhas é uma técnica do ramo da computação natural, ou Bioinspirada, que é baseada no comportamento de enxames de abelhas na procura pela melhor solução para um problema de otimização.

Cada solução candidata é considerada uma fonte de alimento (flor), e uma população (colônia) de n agentes (abelhas) é usada para pesquisar o espaço de solução. A cada iteração é avaliada a rentabilidade (*fitness*) da solução corrente,

selecionam-se as melhores m abelhas locais, busca-se na vizinhança dos m locais e adiciona-se uma série de novas abelhas escolhidas aleatoriamente.

As abelhas serão recrutadas considerando as que possuem melhor aptidão (*fitness*), em caso de não encontrar abelhas de melhor aptidão o tamanho da população pode ser reduzido. Se não houver melhora no condicionamento da população, são registradas as informações sobre a população corrente em um determinado ponto para um número predefinido de ciclos, o máximo local de aptidão é considerado encontrado, logo após é abandonada a solução corrente e gerada uma nova população aleatoriamente e o processo é reinicializado (LARA, 2008).

A convergência do algoritmo se dá quando é encontrada uma solução de adequação aceitável ou por um número de ciclos pré-determinado. Visto que o algoritmo da abelha engloba tanto a pesquisa global quanto a local.

Programação Inteira (Integer Programming)

A técnica de programação matemática é utilizada em problemas de otimização. Devido às diversas peculiaridades inerentes aos diversos contextos de programação, os métodos de solução sofreram especialização e particularizações e foram criadas diversas subáreas dentro da programação matemática, entre elas, a programação inteira (GOLDBARG e LUNA, 2000).

Problemas que podem ser estruturados e solucionados por meio de modelos quantitativos podem ser expressos matematicamente. Os modelos de pesquisa operacional são estruturados de forma lógica e amparados no ferramental matemático de representação.

A programação inteira é um modelo de otimização utilizado quando qualquer variável não pode assumir valores contínuos, ficando condicionadas a assumir valores inteiros. Os métodos de programação inteira utilizam metáforas como a da evolução genética ou do resfriamento de metais a fim de encontrar a melhor solução (GOLDBARG e LUNA, 2000).

A programação inteira é utilizada como técnica de otimização de busca para problemas que podem ser representados por valores inteiros, as restrições e a função objetivo são valores inteiros. Em caso do problema não encontrar o valor inteiro, é utilizada a abordagem de ramificar e limitar (*branch-bound*), onde este é dividido em outros problemas a fim de encontrar uma solução de valor inteiro.

A convergência se dá quando o algoritmo é executado uma quantidade de iterações pré-definida. É possível que a solução viável seja encontrada antes de completar o número de execuções previamente definido, mas só saberá que a solução encontrada é realmente a solução aceitável após a finalização das iterações. Isso se dar pelo fato de que se o problema tiver como solução ótima Z^t é realizado o teste se Z^{t+1} pode ser uma solução aceitável, caso não seja isto prova que Z^t é de fato a solução do problema (NOGUEIRA, 2010).

Algoritmos Meméticos

Os algoritmos meméticos podem ser definido como uma tentativa de imitar a evolução cultural. São também considerados um casamento entre uma busca global na população base e buscas locais heurísticas realizadas em cada um dos indivíduos.

Uma tendência relativamente nova no campo dos GAs é a incorporação de técnicas de busca local. O que os pesquisadores da área de algoritmos genéticos pensavam ter em mãos uma ferramenta que poderia resolver qualquer tipo de problema de otimização de forma simples, constataram que há alguns problemas que necessitam embutir conhecimentos específicos para que possa obter resultados de alta qualidade de forma consistente, o que não seria possível encontrar esses resultados através dos GAs (LINDEN, 2012).

Os algoritmos meméticos são aplicados em problemas que seja necessário embutir o máximo de conhecimento sobre o problema, casos em que não é possível obter um bom desempenho através dos GAs. Sendo assim, os algoritmos meméticos são uma evolução dos GAs, divergindo no fato de que os algoritmos genéticos visam reproduzir a evolução biológica, enquanto que os algoritmos meméticos visam reproduzir a evolução cultural.

Enquanto nos algoritmos genéticos os genes são submetidos ao processo de transmissão de informação onde os descendentes herdam habilidades e características dos seus progenitores, os memes (analogia ao gene, porém no contexto cultural) são adaptados pelo indivíduo que o recebe com base no seu conhecimento e para atender suas necessidades.

Segundo Nguyen *et al.* (2011), as combinações de buscas na execução dos algoritmos meméticos o favorecem em relação ao desempenho sobre os

algoritmos genéticos, visto que os algoritmos meméticos tendem a evitar a convergência prematura e possui menor custo de tempo.

Propagação de Enxames de Partículas

Há fenômenos na natureza que são produzidos por pequenos elementos básicos ou partículas. O método de propagação de enxame de partículas, criado por Kennedy e Eberhart (1995), também é do ramo da computação evolucionária e foi influenciada pelo comportamento humano. Segundo os autores, “o comportamento humano não só ajusta o movimento físico, mas cognitivo ou de variáveis experienciais também”.

Os sistemas baseados em partículas são compostos de representações através de pontos no espaço, com atributos que variam ao longo do tempo. Os atributos podem ser posição, tempo de vida, velocidade, forma, tamanho, cor e transparência.

No sistema baseado em partículas, essas podem ou não interagir umas com as outras. Cada uma atua de forma independente no espaço. Mesmo assim, as formas resultantes são representadas por nuvens de partículas que, conjuntamente, definirão o volume do objeto, e as partículas são individualmente dinâmicas, podendo nascer e morrer ao passar das iterações.

Sua implementação requer apenas operadores matemáticos, o enxame otimizador é semelhante ao cruzamento utilizado em algoritmos genéticos e utiliza os mesmos paradigmas da computação evolutiva. A simulação do algoritmo é como se as partículas voassem sobre o espaço de solução a fim de encontrar a melhor solução mais rapidamente.

APÊNDICE B – Horários Propostos pelo Algoritmo

Horários 2012.1

- Melhor indivíduo dentre todas as gerações (Aptidão = 0)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
FMAT (Luciano)	II (Luiz Paulo)	LOG (João Paulo)	ALG (Flavius)	ALG (Flavius)	1 2	T
II (Luiz Paulo)	LOG (João Paulo)	ALG (Flavius)	FMAT (Luciano)	TGA (Gilson)	3 4	
		TGA (Gilson)			5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
II (Luiz Paulo)	PROG (José Enéias)	LOG (João Paulo)	PROG (José enéias)	LOG (João Paulo)	1 2	M
PROG (José Enéias)	II (Luiz Paulo)				3 4	
					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	ALGL (Deilson)	FSI (Gilson)	OSM (Karlíane)	1 2	T
POO1 (Fabrício)	FSI (Gilson)	ED (João Paulo)	ALGL (Deilson)	ED (João Paulo)	3 4	
OSM (Karlíane)					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
IHC (Karliane)	ES2 (Taciano)	EMP (Gilson)	SO (Borges)	PWEB (Fabrício)	1 2	T
PABD (Taciano)	PWEB (Fabrício)	IHC (Karliane)	EMP (Gilson)	SO (Borges)	3 4	
ES2 (Taciano)	PABD (Taciano)				5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
SAD (Flavius)	SDIS (Borges)	DIR (Rogério)	CEC (Celso)	MATF (Luciano)	1 2	T
GPS (Karliane)	DIR (Rogério)	CEC (Celso)	SDIS (Borges)	GPS (Karliane)	3 4	
CEC (Celso)	SAD (Flavius)	MATF (Luciano)			5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

- Melhor indivíduo da última população (Aptidão = 2)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
FMAT (Luciano)	II (Luiz Paulo)	ALG (Flavius)	LOG (João Paulo)	ALG (Flavius)	1 2	T
II (Luiz Paulo)	LOG (João Paulo)	FMAT (Luciano)	ALG (Flavius)	TGA (Gilson)	3 4	
		TGA (Gilson)			5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
II (Luiz Paulo)	PROG (José Enéias)	LOG (João Paulo)	LOG (João Paulo)	II (Luiz Paulo)	1 2	M
PROG (José Enéias)	PROG (José Enéias)				3 4	
					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	ALGL (Deilson)	FSI (Gilson)	OSM (Karliane)	1 2	T
FSI (Gilson)	ALGL (Deilson)	POO1 (Fabrício)	OSM (Karliane)	ED (João Paulo)	3 4	
ED (João Paulo)					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
IHC (Karliane)	ES2 (Taciano)	PWEB (Fabrício)	S0 (Borges)	EMP (Gilson)	1 2	T
PWEB (Fabrício)	SO (Borges)	ES2 (Taciano)	EMP (Gilson)	IHC (Karliane)	3 4	
PABD (Taciano)	PABD (Taciano)				5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
SAD (Flavius)	SDIS (Borges)	MATF (Luciano)	GPS (Karliane)	CEC (Celso)	1 2	T
DIR (Rogério)	DIR (Rogério)	SAD (Flavius)	CEC (Celso)	MATF (Luciano)	3 4	
CEC (Celso)	GPS (Karliane)	SDIS (Borges)			5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

Horário 2012.2

- Melhor indivíduo dentre todas as gerações (Aptidão =3)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LOG (João Paulo)	ALG (José Enéias)	LOG (João Paulo)	ALG (José Enéias)	ALG (José Enéias)	1 2	M
					3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CAL (Désio)	MTC (Gilson)	LPT (Célia)	TGS (Gilson)	PROG (Flavius)	1 2	T
PROG (Flavius)	TGS (Gilson)	PROG (Flavius)	MTC (Gilson)	CAL (Désio)	3 4	
LPT (Célia)					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ARQ (Luiz Paulo)	ARQ (Luiz Paulo)				1 2	M
					3 4	
					5 6	

						4
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
ING (Leomarques)	BD (Taciano)	ARQ (Luiz Paulo)	ES1 (Fabrício)	POO2 (Fabrício)	1 2	T
ARQ (Luiz Paulo)	ING (Leomarques)	POO2 (Fabrício)	BD (Taciano)	ES1 (Fabrício)	3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TSI (José Enéias)	CEC (Ricardo)	FIL (Talitha)	GPS (Karlíane)	RED (Borges)	1 2	T
SEG (Borges)	SEG (Borges)	GPS (Karlíane)	CEC (Ricardo)	TSI (José Enéias)	3 4	
RED (Borges)	FIL (Talitha)	CEC (Ricardo)			5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TEES (Taciano)	TEES (Taciano)				1 2	M
					3 4	
					5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
SAD (Flavius)	AEX (João Paulo)	PV (Karliane e Luiz Paulo)	DIR (Vyrna)	ETIC (Carlos)	1 2	T
DIR (Vyrna)	SAD (Flavius)	ETIC (Carlos)	AEX (João Paulo)	PV (Karliane e Luiz Paulo)	3 4	
					5 6	

- Melhor indivíduo da última população: (Aptidão = 6)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LOG (João Paulo)	ALG (José Enéias)	LOG (João Paulo)	ALG (José Enéias)	ALG (José Enéias)	1 2	M
					3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CAL (Désio)	LPT (Célia)	TGS (Gilson)	MTC (Gilson)	PROG (Flavius)	1 2	T
PROG (Flavius)	CAL (Désio)	PROG (Flavius)	TGS (Gilson)	LPT (Célia)	3 4	
MTC (Gilson)					5 6	

						3
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
ARQ (Luiz Paulo)	ARQ (Luiz Paulo)				1 2	M
					3 4	
					5 6	

Seg	Ter	Qua	Qui	Sex	Aula	Turno
ING (Leomarques)	BD (Taciano)	ARQ (Luiz Paulo)	ES1 (Fabrício)	POO2 (Fabrício)	1 2	T
BD (Taciano)	POO2 (Fabrício)	ING (Leomarques)	ARQ (Luiz Paulo)	ES1 (Fabrício)	3 4	
					5 6	

Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

Seg	Ter	Qua	Qui	Sex	Aula	Turno
TSI (José Enéias)	CEC (Ricardo)	FIL (Talitha)	GPS (Karlíane)	RED (Borges)	1 2	T
CEC (Ricardo)	FIL (Talitha)	SEG (Borges)	CEC (Ricardo)	GPS (Karlíane)	3 4	
SEG (Borges)	RED (Borges)	TSI (José Enéias)			5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TEES (Taciano)	TEES (Taciano)				1 2	M
					3 4	
					5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
SAD (Flavius)	AEX (João Paulo)	PV (Karliane e Luiz Paulo)	DIR (Vyrna)	ETIC (Carlos)	1 2	T
AEX (João Paulo)	SAD (Flavius)	ETIC (Carlos)	PV (Karliane e Luiz Paulo)	DIR (Vyrna)	3 4	
					5 6	

Horário 2013.1

- Melhor indivíduo dentre todas as gerações (Aptidão = 0)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
FMAT (???)	ALG (Flavius)	II (Luiz Paulo)	LOG (João Paulo)	II (Luiz Paulo)	1 2	T
ALG (Flavius)	LOG (João Paulo)	ALG (Flavius)	FMAT (???)		3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CAL (???)	PROG (Walber)	ALG (Flavius)	II (Luiz Paulo)	FMAT (Barroca)	1 2	M
PROG (Walber)	II (Luiz Paulo)	CAL (???)	PROG (Walber)	ALG (Flavius)	3 4	
FMAT (Barroca)	ALG (Flavius)				5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	FSI (Gilson)	OSM (Karlíane)	ALGL (Désio)	1 2	T
ALGL (Désio)	FSI (Gilson)	ED (João Paulo)	ED (João Paulo)	POO1 (Fabrício)	3 4	
OSM (Karlíane)					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
POO1 (Flavius)	POO1 (Flavius)				1 2	M
					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
RED (Borges)	ES2 (Taciano)	PWEB (Fabrício)	SO (Borges)	PABD (Taciano)	1 2	T
ES2 (Taciano)	RED (Borges)	PABD (Taciano)	PWEB (Fabrício)	SO (Borges)	3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
PDM (Walber)	SDIS (Borges)	PV (Karlíane)	MATF (Manassés)	TEES (Taciano)	1 2	T
DIR (Vyrna)	MIC (Luiz Paulo)	TEES (Taciano)	SDIS (Borges)	PDM (Walber)	3 4	
MIC (Luiz Paulo)	DIR (Vyrna)	MATF (Manassés)	PV (Karlíane)		5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

- Melhor indivíduo da última população: (Aptidão = 3)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
FMAT (???)	ALG (Flavius)	II (Luiz Paulo)	LOG (João Paulo)	II (Luiz Paulo)	1 2	T
ALG (Flavius)	LOG (João Paulo)	FMAT (???)	ALG (Flavius)		3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CAL (???)	PROG (Walber)	ALG (Flavius)	II (Luiz Paulo)	FMAT (Barroca)	1 2	M
CAL (???)	PROG (Walber)	II (Luiz Paulo)	PROG (Walber)	ALG (Flavius)	3 4	
FMAT (Barroca)	ALG (Flavius)				5 6	

						3
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	ALGL (Désio)	OSM (Karliane)	FSI (Gilson)	1 2	T
POO1 (Fabrício)	OSM (Karliane)	ED (João Paulo)	ALGL (Désio)	ED (João Paulo)	3 4	
FSI (Gilson)					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
POO1 (Flavius)	POO1 (Flavius)				1 2	M
					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
RED (Borges)	ES2 (Taciano)	PWEB (Fabrício)	SO (Borges)	PABD (Taciano)	1 2	T
ES2 (Taciano)	PWEB (Fabrício)	PABD (Taciano)	RED (Borges)	SO (Borges)	3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
PDM (Walber)	DIR (Vyrna)	PV (Karliane)	MIC (Luiz Paulo)	TEES (Karliane)	1 2	T
SDIS (Borges)	PV (Karliane)	MIC (Luiz Paulo)	MATF (Manassés)	PDM (Walber)	3 4	
DIR (Vyrna)	MATF (Manassés)	TEES (Karliane)	SDIS (Borges)		5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

Horário 2013.2

- Melhor indivíduo dentre todas as gerações (Aptidão = 2)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
II (Luiz Paulo)	II (Luiz Paulo)				1 2	M
					3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LPT (???)	PROG (Flavius)	CAL (???)	TGS (Gilson)	MTC (Gilson)	1 2	T
TGS (Gilson)	CAL (???)	PROG (Flavius)	LPT (???)	PROG (Flavius)	3 4	
MTC (Gilson)					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	ED (João Paulo)	POO1 (Fabrício)	ED (João Paulo)	1 2	M
					3 4	
					5 6	

						4
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
PWEB (Fabrício)	ING (???)	EST (???)	BD (Taciano)	ES1 (Karliane)	1 2	T
ARQ (Luiz Paulo)	BD (Taciano)	ES1 (Karliane)	ARQ (Luiz Paulo)	ING (???)	3 4	
EST (???)	PWEB (Fabrício)				5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
EMP (Gilson)	RED (Borges)	GPS (Karliane)	FIL (???)	CEC (???)	1 2	T
CEC (???)	GPS (Karliane)	CEC (???)	EMP (Gilson)	RED (Borges)	3 4	
FIL (???)					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						8
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
SAD (Flavius)	TEES (Taciano)	ETIC (???)	SAD (Flavius)	TEES (Taciano)	1 2	T
ETIC (???)					3 4	
					5 6	

- Melhor indivíduo da última população (Aptidão = 5)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ALG (Flavius)	ALG (Flavius)	ALG (Flavius)			1 2	M
					3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LPT (???)	PROG (Flavius)	CAL (???)	TGS (Gilson)	MTC (Gilson)	1 2	T
MTC (Gilson)	TGS (Gilson)	LPT (???)	CAL (???)	PROG (Flavius)	3 4	
PROG (Flavius)					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	ED (João Paulo)	ED (João Paulo)	POO1 (Fabrício)	1 2	M
					3 4	
					5 6	

						4
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
PWEB (Fabrício)	ES1 (Karliane)	BD (Taciano)	ARQ (Luiz Paulo)	ING (???)	1 2	T
EST (???)	BD (Taciano)	PWEB (Fabrício)	ES1 (Karliane)	EST (???)	3 4	
ARQ (Luiz Paulo)	ING (???)				5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
EMP (Gilson)	RED (Borges)	GPS (Karliane)	CEC (???)	FIL (???)	1 2	T
GPS (Karliane)	FIL (???)	RED (Borges)	EMP (Gilson)	CEC (???)	3 4	
CEC (???)					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
					1 2	
					3 4	
					5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
SAD (Flavius)	TEES (Taciano)	ETIC (???)	SAD (Flavius)	ETIC (???)	1 2	T
TEES (Taciano)					3 4	
					5 6	

Horário 2014.1

- Melhor indivíduo dentre todas as gerações (Aptidão = 2)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TGA (Gilson)	ALG (Flavius)	II (Tiago)	LOG (João Paulo)	FMAT (Deilson)	1 2	T
FMAT (Deilson)	TGA (Gilson)	ALG (Flavius)	II (Tiago)	ALG (Flavius)	3 4	
LOG (João Paulo)					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
POO1 (Odilon)	ALG (Flavius)	CAL (Deilson)	CAL (Deilson)	ALG (Flavius)	1 2	M
ALG (Flavius)	POO1 (Odilon)				3 4	
					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	ALGL (Deilson)	OSM (Karlíane)	FSI (Gilson)	1 2	T
POO1 (Fabrício)	ED (João Paulo)	ED (João Paulo)	ALGL (Deilson)	OSM (Karlíane)	3 4	
FSI (Gilson)					5 6	

						4
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
PWEB (Fabrício)	BD (Taciano)	ARQ (Tiago)	PWEB (Fabrício)	BD (Taciano)	1 2	M
ARQ (Tiago)					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
APM (João Paulo)	PABD (Taciano)	RED (Borges)	ES2 (Taciano)	POO2 (Fabrício)	1 2	T
SO (Borges)	RED (Borges)	ES2 (Taciano)	POO2 (Fabrício)	SO (Borges)	3 4	
PABD (Taciano)	APM (João Paulo)				5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
GPS (Karliane)	GPS (Karliane)				1 2	M
					3 4	
					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
DIR (???)	PV (Tiago)	MATF (Guilherme)	DIR (???)	PV (Tiago)	1 2	T
MATF (Guilherme)					3 4	
					5 6	

						8
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
CLOUD (Gilson)	SDIS (Borges)	TES (Karliane)	MIN (Flavius)	PDM (Odilon)	1 2	M
PDM (Odilon)	MIN (Flavius)	CLOUD (Gilson)	SDIS (Borges)	TES (Karliane)	3 4	
					5 6	

- Melhor indivíduo da última população (Aptidão = 3)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TGA (Gilson)	ALG (Flavius)	II (Tiago)	LOG (João Paulo)	FMAT (Deilson)	1 2	T
LOG (João Paulo)	II (Tiago)	ALG (Flavius)	TGA (Gilson)	ALG (Flavius)	3 4	
FMAT (Deilson)					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
POO1 (Odilon)	ALG (Flavius)	CAL (Guilherme)	POO1 (Odilon)	CAL (Guilherme)	1 2	M
ALG (Flavius)	CAL (Guilherme)				3 4	
					5 6	

						3
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Fabrício)	ALGL (Deilson)	OSM (Karliane)	FSI (Gilson)	1 2	T
ALGL (Deilson)	ED (João Paulo)	OSM (Karliane)	ED (João Paulo)	POO1 (Fabrício)	3 4	
FSI (Gilson)					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
PWEB (Fabrício)	BD (Taciano)	ARQ (Tiago)	BD (Taciano)	PWEB (Fabrício)	1 2	M
ARQ (Tiago)					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
APM (João Paulo)	ES2 (Taciano)	POO2 (Fabrício)	SO (Borges)	PABD (Taciano)	1 2	T
RED (Borges)	POO2 (Fabrício)	SO (Borges)	RED (Borges)	APM (João Paulo)	3 4	
ES2 (Taciano)	PABD (Taciano)				5 6	

						6
SeMg	Ter	Qua	Qui	Sex	Aula	Turno
GPS (Karliane)	GPS (Karliane)				1 2	M / T
					3 4	
					5 6	

						7
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
DIR (???)	PV (Tiago)	MATF (Guilherme)	DIR (???)	PV (Tiago)	1 2	T
MATF (Guilherme)					3 4	
					5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CLOUD (Gilson)	SDIS (Borges)	TES (Karlíane)	MIN (Flavius)	PDM (Odilon)	1 2	M
PDM (Odilon)	MIN (Flavius)	SDIS (Borges)	TES (Karlíane)	CLOUD (Gilson)	3 4	
					5 6	

Horário 2014.2

- Melhor indivíduo dentre todas as gerações (Aptidão = 1)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ALG (Odilon)	ALG (Amarildo)	II (Tiago)	ALG (Odilon)	ALG (Amarildo)	1 2	M
ALG (Amarildo)	II (Tiago)	ALG (Odilon)			3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LPT (???)	CAL (Deilson)	MTC (Gilson)	PROG (João Paulo)	TGS (Adrianne)	1 2	T
MTC (Gilson)	LPT (???)	PROG (João Paulo)	CAL (Deilson)	PROG (João Paulo)	3 4	
TGS (Adrianne)					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ALGL (Deilson)	POO1 (Amarildo)	POO1 (Amarildo)	ALGL (Deilson)		1 2	M
					3 4	
					5 6	

						4
--	--	--	--	--	--	---

Seg	Ter	Qua	Qui	Sex	Aula	Turno
PWEB (Amarildo)	BD (Aislânia)	ING (???)	ARQ (Tiago)	EST (Deilson)	1 2	T
ES1 (Karliane)	ARQ (Tiago)	EST (Deilson)	ES1 (Karliane)	ING (???)	3 4	
BD (Aislânia)	PWEB (Amarildo)		ING		5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TAP (João Paulo)	PABD (Aislânia)	TAP (João Paulo)	PABD (Aislânia)		1 2	M
					3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
EMP (Adrienne)	RED (Rivaldo)	GPS (Karliane)	FIL (???)	CEC (???)	1 2	T
RED (Rivaldo)	EMP (Adrienne)	FIL (???)	CEC (???)	GPS (Karliane)	3 4	
CEC (???)					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CLOUD (Gilson)	ASR (Rivaldo)	TSI (Flavius)	TSI (Flavius)	CLOUD (Gilson)	1 2	M
ASR (Rivaldo)	TESBD (Aislânia)	TESBD (Aislânia)			3 4	
					5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
IHC (Tiago)	SAD (Flavius)	ETIC (Adrianne)	SAD (Flavius)	IHC (Tiago)	1 2	T
ETIC (Adrianne)					3 4	
					5 6	

- Melhor indivíduo da última população (Aptidão = 4)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ALG (Odilon)	ALG (Amarildo)	II (Tiago)	ALG (Odilon)	ALG (Amarildo)	1 2	M
ALG (Amarildo)	II (Tiago)	ALG (Odilon)			3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LPT (???)	PROG (João Paulo)	CAL (Deilson)	TGS (Adrianne)	MTC (Gilson)	1 2	T
LPT (???)	TGS (Adrianne)	MTC (Gilson)	PROG (João Paulo)	CAL (Deilson)	3 4	
PROG (João Paulo)					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ALGL (Deilson)	POO1 (Amarildo)	ALGL (Deilson)	POO1 (Amarildo)		1 2	M
					3 4	
					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
PWEB (Amarildo)	ES1 (Karliane)	ING (???)	EST (Deilson)	ARQ (Tiago)	1 2	T
BD (Aislânia)	BD (Aislânia)	EST (Deilson)	ARQ (Tiago)	ES1 (Karliane)	3 4	
ING (???)	PWEB (Amarildo)				5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TAP (João Paulo)	PABD (Aislânia)	TAP (João Paulo)	PABD (Aislânia)		1 2	M
					3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
EMP (Adrianne)	RED (Rivaldo)	GPS (Karliane)	FIL (???)	CEC (???)	1 2	T
FIL (???)	GPS (Karliane)	CEC (???)	RED (Rivaldo)	EMP (Adrianne)	3 4	
CEC (???)					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CLOUD (Gilson)	ASR (Rivaldo)	TSI (Flavius)	ASR (Rivaldo)	CLOUD (Gilson)	1 2	M
TSI (Flavius)	TESBD (Aislânia)	TESBD (Aislânia)			3 4	
					5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
IHC (Tiago)	SAD (Flavius)	ETIC (Adrianne)	IHC (Tiago)	SAD (Flavius)	1 2	T
ETIC (Adrianne)					3 4	
					5 6	

Horário 2015.1

- Melhor indivíduo dentre todas as gerações (Aptidão = 1)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TGA (Adrianne)	ALG (Flavius)	II (Tiago)	FMAT (Deilson)	LOG (João Paulo)	1 2	T
II (Tiago)	LOG (João Paulo)	ALG (Flavius)	TGA (Adrianne)	ALG (Flavius)	3 4	
FMAT (Deilson)					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CAL (Deilson)	PROG (Flavius)	CAL (Deilson)	PROG (Flavius)	PROG (Flavius)	1 2	M
AGL (Flavius)	AGL (Flavius)	AGL (Flavius)			3 4	
					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Amarildo)	ALGL (Deilson)	OSM (Karliane)	FSI (Gilson)	1 2	T
FSI (Gilson)	ALGL (Deilson)	ED (João Paulo)	ED (João Paulo)	POO1 (Amarildo)	3 4	
OSM (Karliane)					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
EST (Deilson)	BD (Aislânia)	BD (Aislânia)	EST (Deilson)		1 2	M
					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
	ES2 (Karlíane)	PABD (Aislânia)	SO (Tiago)	POO2 (Amarildo)	1 2	T
POO2 (Amarildo)	PABD (Aislânia)		ES2 (Karlíane)	SO (Tiago)	3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
RED (Rivaldo)	RED (Rivaldo)	TAP (Amarildo)	TAP (Amarildo)		1 2	M
					3 4	
					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TESBD (Aislânia)	PV (Tiago)	ASR (Rivaldo)	TSI (Adrianne)	DIR (???)	1 2	T
MATF (???)	DIR (???)	TSI (Adrianne)	PV (Tiago)	MATF (???)	3 4	
ASR (Rivaldo)	TESBD (Aislânia)				5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
PDM (???)	TES (Karliane)	SAD (Flavius)	AEX (João Paulo)	AEX (João Paulo)	1 2	M
TES (Karliane)	SAD (Flavius)	PDM (???)			3 4	
					5 6	

- Melhor indivíduo da última população (Aptidão = 3)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Tur no
TGA (Adrianne)	ALG (Flavius)	II (Tiago)	FMAT (Deilson)	LOG (João Paulo)	1 2	T
ALG (Flavius)	LOG (João Paulo)	FMAT (Deilson)	ALG (Flavius)	TGA (Adrianne)	3 4	
II (Tiago)					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CAL (Deilson)	PROG (???)	PROG (???)	CAL (Deilson)	PROG (???)	1 2	M
ALG (Flavius)	ALG (Flavius)	ALG (Flavius)			3 4	
					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ED (João Paulo)	POO1 (Amarildo)	FSI (Gilson)	OSM (Karliane)	ALGL (Deilson)	1 2	T
OSM (Karliane)	ED (João Paulo)	ED (João Paulo)	ALGL (Deilson)	POO1 (Amarildo)	3 4	
FSI					5	

(Gilson)					6	
						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
EST (Deilson)	BD (Aislânia)	EST (Deilson)	BD (Aislânia)		1 2	M
					3 4	
					5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
	ES2 (Karliane)	POO2 (Amarildo)	SO (Tiago)	PABD (Aislânia)	1 2	T
SO (Tiago)	POO2 (Amarildo)		PABD (Aislânia)	ES2 (Karliane)	3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
RED (Rivaldo)	RED (Rivaldo)	TAP (Amarildo)	TAP (Amarildo)		1 2	M
					3 4	
					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TESBD (Aislânia)	TSI (Adriane)	DIR (???)	MATF (???)	PV (Tiago)	1 2	T
ASR (Rivaldo)	ASR (Rivaldo)	TESBD (Aislânia)	PV (Tiago)	MATF (???)	3 4	
TSI	DIR				5	

(Adrienne)	(???)				6	
------------	-------	--	--	--	---	--

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turn o
PDM (???)	AEX (João Paulo)	TES (Karliane)	SAD (Flavius)	TES (Karliane)	1 2	M
SAD (Flavius)	AEX (João Paulo)	PDM (???)			3 4	
					5 6	

Horário 2015.2

- Melhor indivíduo entre todas gerações (Aptidão = 17)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LOG (Tiago)	ALG (Alexandre)		ALG (Alexandre)	FMAT (Deilson)	1 2	M
			LOG (Tiago)	ALG (Alexandre)	3 4	
		II (Tiago)	II (Tiago)	FMAT (Deilson)	5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
TGS (Gilson)	PROG (João Paulo)	PROG (João Paulo)		PROG (João Paulo)	1 2	T
LPT (???)		MTC (Adrianne)	CAL (Deilson)	MTC (Adrianne)	3 4	
CAL (Deilson)		LPT (???)		TGS (Gilson)	5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
	POO1 (Rodrigo)	POO1 (Rodrigo)			1 2	M
					3 4	
	ALGL (Deilson)	ALGL (Deilson)			5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
EST (Deilson)	ES1 (Aislânia)	PWEB (Rodrigo)	ARQ (Tiago)	BD (Aislânia)	1 2	T
ARQ (Tiago)	ING (???)	ES1 (Aislânia)	EST (Deilson)		3 4	
ING (???)		BD (Aislânia)		PWEB (Rodrigo)	5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
	FDLD (???)	PABD (Aislânia)			1 2	M
TAP (Rodrigo)		TAP (Rodrigo)	PABD (Aislânia)		3 4	
	FDLD (???)				5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CEC (???)	GPS (Karlíane)		FIL (???)	EMP (Adrianne)	1 2	T
	RED (Rivaldo)	FIL (???)	CEC (???)	RED (Rivaldo)	3 4	
	EMP (Adrianne)		CEC (???)	GPS (Karlíane)	5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
APM (Karlíane)	TSI (João Paulo)	APM (Karlíane)	PV (Alexandre)		1 2	M
MIN (Flavius)	ASR (Rivaldo)		TSI (João Paulo)	PV (Alexandre)	3 4	
			ASR (Rivaldo)	MIN (Flavius)	5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ETIC (Adrianne)		ETIC (Adrianne)			1 2	T
		SAD (Flavius)			3 4	
			SAD (Flavius)		5 6	

- Melhor indivíduo da última população (Aptidão = 34)

						1
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ALG (Alexandre)	ALG (Alexandre)	ALG (Alexandre)	LOG (Tiago)	II (Tiago)	1 2	M
LOG (Tiago)	II (Tiago)	FMAT (Deilson)	FMAT (Deilson)		3 4	
					5 6	

						2
Seg	Ter	Qua	Qui	Sex	Aula	Turno
LPT (???)	TGS (Gilson)	CAL (Deilson)	MTC (Adrianne)	LPT (???)	1 2	T
CAL (Deilson)	PROG (João Paulo)	MTC (Adrianne)	PROG (João Paulo)	TGS (Gilson)	3 4	
PROG (João Paulo)					5 6	

						3
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ALGL (Deilson)	POO1 (Rodrigo)	POO1 (Rodrigo)	ALGL (Deilson)		1 2	M
					3 4	
					5 6	

						4
Seg	Ter	Qua	Qui	Sex	Aula	Turno
ING (???)	PWEB (Rodrigo)	BD (Aislânia)	ARQ (Tiago)	PWEB (Rodrigo)	1 2	T
BD (Aislânia)	ARQ (Tiago)	ES1 (Aislânia)	ING (???)	ES1 (Aislânia)	3 4	
EST (Deilson)	EST (Deilson)				5 6	

						5
Seg	Ter	Qua	Qui	Sex	Aula	Turno
FDLD (???)	FDLD (???)	TAP (Rodrigo)	TAP (Rodrigo)	PABD (Aislânia)	1 2	M
PABD (Aislânia)					3 4	
					5 6	

						6
Seg	Ter	Qua	Qui	Sex	Aula	Turno
CEC (???)	EMP (Adrianne)	FIL (???)	EMP (Adrianne)	GPS (Karliane)	1 2	T
FIL (???)	CEC (???)	RED (Rivaldo)	RED (Rivaldo)	CEC (???)	3 4	
GPS (Karliane)					5 6	

						7
Seg	Ter	Qua	Qui	Sex	Aula	Turno
APM (Karliane)	MIN (Flavius)	APM (Karliane)	TSI (João Paulo)	ASR (Rivaldo)	1 2	M
MIN (Flavius)	TSI (João Paulo)	ASR (Rivaldo)	PV (Alexandre)	PV (Alexandre)	3 4	
					5 6	

						8
Seg	Ter	Qua	Qui	Sex	Aula	Turno
SAD (Flavius)	SAD (Flavius)	ETIC (Adrianne)	ETIC (Adrianne)		1 2	T
					3 4	
					5 6	

APÊNDICE C – Métodos para Geração de um Indivíduo e Operadores Genéticos
Utilizados

Geração de um Indivíduo

```
public void geracao(int periodo) throws CloneNotSupportedException {
    this.novaPopulacao = new ArrayList<>();
    Cromossomo pai1;
    Cromossomo pai2;
    Cromossomo filho = new Cromossomo();
    //Calculando nova geracao...

    calculaSomaAvaliacoes();

    for (int i = 0; i < this.getTamanho(); i++)
    {

        pai1 = this.populacao.get(this.roleta());

        filho.criandoPeriodos(filho, this.periodos);

        filho = pai1.cruzamentoEmOrdem(periodo);

        pai2 = this.populacao.get(this.roleta());

        pai2.setTurmasNaolInseridas(pai1.getTurmasNaolInseridas());

        filho.herdarDoSegundoCromossomo(pai2.ordenar(periodo), periodo);

        filho.mutacao2(listProf);

        filho.getRegras().avaliacao();

        this.novaPopulacao.add(filho);
    }
}
```

Cruzamento em ordem

```

public Cromossomo cruzamentoEmOrdem(int periodo)
{
    int index = 0;
    int matrizDeSelecao[][] = gerarMatriz();

    Cromossomo filho = new Cromossomo();
    criandoPeriodos(filho, periodos);
    turmasNaolInseridas = new Turma[tamanhoVetor(matrizDeSelecao)];

    for(int i = 0; i<3; i++)
    {
        for(int j = 0; j<5; j++)
        {
            if(matrizDeSelecao[i][j] == 1)
            {
                filho.periodos.get(periodo).getHorario()[i][j] = this.periodos.get(periodo).getHorario()[i][j];
            }
            else
            {
                Turma t = this.periodos.get(periodo).getHorario()[i][j];

                if(t != null)
                {
                    this.getTurmasNaolInseridas()[index] = t; // não inseridas
                    index++;
                }

                Turma w = new Turma();
                w.setNome("0");

                filho.getPeriodos().get(periodo).getHorario()[i][j] = w;
            }
        }
    }

    filho.regras.avaliacao();
    return filho;
}

```

```

public int[][] gerarMatriz()
{
    int matriz[][] = new int[3][5];

    for(int i = 0; i<3; i++)
    {
        for(int j = 0; j<5; j++)
        {
            matriz[i][j] = (int) (Math.random() * 2);
        }
    }

    return matriz;
}

```

```
private int tamanhoVetor(int matriz[][]){
    {
        int tam = 0;
        for(int i = 0; i < 3; i++)
        {
            for(int j = 0; j < 5; j++)
            {
                if(matriz[i][j]==0)
                    tam++;
            }
        }
        return tam;
    }
}
```

```
public Turma[] ordenar(int periodo)
{
    Turma aux[] = new Turma[turmasNaolInseridas.length];

    int index = 0;
    for(int linha = 0; linha < 3; linha++)
    {
        for(int coluna = 0; coluna<5; coluna++)
        {
            Turma t = this.periodos.get(periodo).getHorario()[linha][coluna];

            if(encontrarTurma(t, turmasNaolInseridas))
            {
                aux[index] = t;
                index++;
            }
        }
    }

    return aux;
}
```

```
public boolean encontrarTurma(Turma turma, Turma turmasNaolInseridas[])
{
    for(int i = 0; i < turmasNaolInseridas.length; i++)
    {
        if (turmasNaolInseridas[i] == turma)
        {
            //Turma t = turmasNaolInseridas[i];
            Turma aux = new Turma();
            aux.setNome("sub");
            turmasNaolInseridas[i] = aux;
            return true;
        }
    }
    return false;
}
```

```
public void herdarDoSegundoCromossomo(Turma turmas[], int periodo)
{
    int index = 0;
    for(int linha = 0; linha<3; linha++)
    {
        for(int coluna = 0; coluna<5; coluna++)
        {
            if(this.periodos.get(periodo).getHorario()[linha][coluna] != null)
            {
                if(this.periodos.get(periodo).getHorario()[linha][coluna].getNome().equals("0"))
                {
                    this.periodos.get(periodo).getHorario()[linha][coluna] = turmas[index];
                    index++;
                }
            }
        }
    }
}
```

Mutação

```

public void mutacao2(List<Professor> listProf)
{
    List<Integer> choqueLinha = new ArrayList<Integer>();

    List<Integer> choqueColuna = new ArrayList<Integer>();

    List<Professor> tChoque = new ArrayList<Professor>();

    for(int periodo = 0; periodo < this.periodos.size(); periodo++)
    {
        Período p = this.periodos.get(periodo);

        this.regras.aulaMesmoDia(p);

        this.regras.corrigirIntervaloVago(p.getHorario());

        this.regras.adaptarDisponibilidade();

        this.regras.choqueDeHorario(p, tChoque, choqueColuna, choqueLinha);
    }
}

```

```

public void aulaMesmoDia(Período p)
{
    for (Turma t: p.getTurmas())
    {
        int x = 0;
        for(int coluna = 0; coluna < 5; coluna++)
        {
            for (int linha = 1; linha < 3; linha++)
            {
                if(p.getHorario()[linha][coluna] == p.getHorario()[x][coluna])
                {
                    int a = (int) Math.random() % 5;
                    Turma aux = p.getHorario()[linha][coluna];
                    p.getHorario()[linha][coluna] = p.getHorario()[linha][a];
                    p.getHorario()[linha][a] = aux;
                }
            }
        }
    }
}

```

```

public void corrigirIntervaloVago(Turma horario[][])
{

```

```

for(int i = 0; i<5; i++)
{
    if ((horario[0][i] != null)
        && (horario[1][i] == null)
        && (horario[2][i] != null))
    {
        horario[1][i] = horario[2][i];
        horario[2][i] = null;
    }

    if ((horario[0][i] == null)
        && (horario[1][i] == null)
        && (horario[2][i] != null))
    {
        horario[1][i] = horario[2][i];
        horario[2][i] = null;
    }

    if ((horario[0][i] == null)
        && (horario[1][i] != null))
    {
        horario[0][i] = horario[1][i];
        horario[1][i] = horario[2][i];
        horario[2][i] = null;
    }
}
}

```

```

public void adaptarDisponibilidade()
{
    for(Periodo p : this.cromossomo.getPeriodos())
    {
        for(Turma t : p.getTurmas())
        {
            int disp[] = t.getProfessor().disponivel();

            for(int i = 0; i < 3; i++)
            {
                for(int j = 0; j<5; j++)
                {
                    if(p.getHorario()[i][j] == t)
                    {
                        if(t.getProfessor().verificarDisponibilidade(i*5+j))
                        {
                            int x = (int) (Math.random() % disp.length);
                            int linha = disp[x] %3;
                            int coluna = disp[x] % 5;
                            Turma aux = p.getHorario()[linha][coluna];
                            p.getHorario()[linha][coluna] = p.getHorario()[i][j];
                            p.getHorario()[i][j] = aux;
                        }
                    }
                }
            }
        }
    }
}

```

```

    public void choqueDeHorario(Periodo p, List<Professor> tChoque, List<Integer> choqueColuna,
    List<Integer> choqueLinha)
    {
        for(int linha = 0; linha<3; linha++)
        {
            for(int coluna = 0; coluna<5; coluna++)
            {
                for(int index = this.cromossomo.getPeriodos().indexOf(p)+2; index <
                this.cromossomo.getPeriodos().size(); index+=2)
                {
                    if((p.getHorario()[linha][coluna] != null)
                    && (this.cromossomo.getPeriodos().get(index).getHorario()[linha][coluna] != null))
                    {
                        if(p.getHorario()[linha][coluna].getProfessor() ==
                        this.cromossomo.getPeriodos().get(index).getHorario()[linha][coluna].getProfessor())
                        {
                            Turma t = p.getHorario()[linha][coluna];
                            int y = posicao(choqueColuna, tChoque, t.getProfessor(), 5);
                            int x = posicao(choqueLinha, tChoque, t.getProfessor(), 3);

                            tChoque.add(t.getProfessor());
                            choqueLinha.add(x);
                            choqueColuna.add(y);

                            p.getHorario()[linha][coluna] = p.getHorario()[x][y];
                            p.getHorario()[x][y] = t;
                        }
                    }
                }
            }
        }
    }

```

APÊNDICE D – Valores de Aptidão Obtidos pelo Algoritmo

Valores Obtidos Durante a Execução do Algoritmo

As tabelas mostram os valores médios de aptidão das populações e das gerações ao longo das iterações do algoritmo para cada semestre utilizado para simulação no presente trabalho.

Tabela 3: Valores médios de aptidão das populações – 2012.1

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão	Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
100	16	6,8043	1	1100	16	6,8797	0
200	16	6,9576	1	1200	16	6,8653	0
300	16	6,8357	0	1300	17	6,8698	1
400	15	7,0659	0	1400	15	6,8438	0
500	17	6,8476	1	1500	16	7,0185	0
600	16	6,8492	1	1600	15	6,8039	1
700	16	6,9475	1	1700	15	6,9051	1
800	16	6,8703	0	1800	17	6,8962	1
900	15	6,8372	1	1900	16	6,8923	1
1000	16	6,8157	1	2000	16	6,7975	0

Fonte: elaborada pela autora.

Tabela 4: Valores médios de aptidão das gerações – 2012.1

Gerações	Média das gerações	Gerações	Média das gerações
100	6,8043	1100	6,8797
200	6,9576	1200	6,8653
300	6,8357	1300	6,8698
400	7,0659	1400	6,8438
500	6,8476	1500	7,0185
600	6,8492	1600	6,8039
700	6,9475	1700	6,9051
800	6,8703	1800	6,8962
900	6,8372	1900	6,8923
1000	6,8157	2000	6,7975

Fonte: elaborada pela autora.

Tabela 5: Valores médios de aptidão das populações – 2012.2

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão	Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
100	28	13,2729	5	1100	29	13,3559	4
200	28	13,597	5	1200	27	13,6439	5
300	26	13,7914	4	1300	28	13,4655	4
400	27	13,5087	4	1400	27	13,1742	4
500	27	13,3408	4	1500	26	13,3895	4
600	31	13,5292	4	1600	26	13,4796	4
700	26	13,5145	3	1700	29	13,5633	5
800	28	13,6801	5	1800	29	13,3534	3
900	28	13,2044	4	1900	25	13,5339	4

1000	28	13,3774	4
------	----	---------	---

2000	27	13,3179	4
------	----	---------	---

Fonte: elaborada pela autora.

Tabela 6: Valores médios de aptidão das gerações – 2012.2

Gerações	Média das gerações	Gerações	Média das gerações
100	13,2729	1100	13,3559
200	13,5970	1200	13,6439
300	13,7914	1300	13,4655
400	13,5087	1400	13,1742
500	13,3408	1500	13,3895
600	13,5292	1600	13,4796
700	13,5145	1700	13,5633
800	13,6801	1800	13,3534
900	13,2044	1900	13,5339
1000	13,3774	2000	13,3179

Fonte: elaborada pela autora.

Tabela 7: Valores médios de aptidão das populações – 2013.1

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão	Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
100	18	7,3901	0	1100	16	7,1879	0
200	16	7,1535	0	1200	16	7,1171	0
300	17	7,2034	0	1300	17	7,1672	0
400	18	7,4166	1	1400	17	7,2679	1
500	18	7,0419	0	1500	17	7,1796	1
600	17	7,3865	1	1600	18	7,0109	0
700	17	7,3184	1	1700	17	7,2571	1
800	17	7,4228	0	1800	17	7,294	1
900	16	7,1651	0	1900	20	7,4419	1
1000	16	7,2295	0	2000	17	7,2569	1

Fonte: elaborada pela autora.

Tabela 8: Valores médios de aptidão das gerações – 2013.1

Gerações	Média das gerações	Gerações	Média das gerações
100	7,3901	1100	7,1879
200	7,1535	1200	7,1171
300	7,2034	1300	7,1672
400	7,4166	1400	7,2679
500	7,0419	1500	7,1796
600	7,3865	1600	7,0109
700	7,3184	1700	7,2571
800	7,4228	1800	7,2940
900	7,1651	1900	7,4419
1000	7,2295	2000	7,2569

Fonte: elaborada pela autora.

Tabela 9: Valores médios de aptidão das populações – 2013.2

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão	Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
----------	--------------	---------------	----------------	----------	--------------	---------------	----------------

100	23	10,4459	3
200	21	10,6203	4
300	23	10,6137	3
400	21	10,5256	3
500	21	10,6729	2
600	21	10,5247	3
700	20	10,5823	3
800	21	10,4977	3
900	23	10,6602	3
1000	21	10,6007	3

1100	23	10,5572	3
1200	21	10,503	2
1300	21	10,558	2
1400	21	10,5721	2
1500	21	10,58	2
1600	20	10,5474	4
1700	20	10,4195	3
1800	22	10,6043	2
1900	23	10,5252	3
2000	20	10,4301	2

Fonte: elaborada pela autora.

Tabela 10: Valores médios de aptidão das gerações – 2013.2

Gerações	Média das gerações
100	10,4459
200	10,6203
300	10,6137
400	10,5256
500	10,6729
600	10,5247
700	10,5823
800	10,4977
900	10,6602
1000	10,6007

Gerações	Média das gerações
1100	10,5572
1200	10,5030
1300	10,5580
1400	10,5721
1500	10,5800
1600	10,5474
1700	10,4195
1800	10,6043
1900	10,5252
2000	10,4301

Fonte: elaborada pela autora.

Tabela 11: Valores médios de aptidão das populações – 2014.1

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
100	22	10,0556	2
200	22	9,9912	3
300	22	9,8482	3
400	23	10,0748	2
500	21	10,1642	3
600	22	9,8793	3
700	22	9,9867	2
800	23	9,8336	2
900	21	9,9009	2
1000	22	9,9747	3

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
1100	21	10,1007	3
1200	23	10,04	2
1300	22	9,7763	2
1400	22	9,8448	3
1500	22	9,8264	2
1600	21	9,9546	3
1700	22	9,9546	3
1800	22	10,036	2
1900	21	10,0355	3
2000	24	9,9405	2

Fonte: elaborada pela autora.

Tabela 12: Valores médios de aptidão das gerações – 2014.1

Gerações	Média das gerações
100	10,0556
200	9,9912
300	9,8482
400	10,0748
500	10,1642

Gerações	Média das gerações
1100	10,1007
1200	10,0400
1300	9,7763
1400	9,8448
1500	9,8264

600	9,8793
700	9,9867
800	9,8336
900	9,9009
1000	9,9747

1600	9,9546
1700	9,9546
1800	10,0360
1900	10,0355
2000	9,9405

Fonte: elaborada pela autora.

Tabela 13: Valores médios de Aptidão das populações – 2014.2

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
100	18	8,5884	1
200	18	8,5481	1
300	18	8,4382	1
400	17	8,4336	1
500	19	8,3735	1
600	17	8,3993	2
700	17	8,38	1
800	18	8,4116	1
900	18	8,5607	2
1000	18	8,6497	1

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
1100	17	8,4079	1
1200	18	8,2445	1
1300	18	8,3935	1
1400	20	8,516	1
1500	19	8,5912	1
1600	18	8,5557	1
1700	17	8,2792	1
1800	17	8,5527	1
1900	18	8,407	1
2000	19	8,2941	1

Fonte: elaborada pela autora.

Tabela 14: Valores médios de aptidão das gerações – 2014.2

Gerações	Média das gerações
100	8,5884
200	8,5481
300	8,4382
400	8,4336
500	8,3735
600	8,3993
700	8,3799
800	8,4116
900	8,5607
1000	8,6497

Gerações	Média das gerações
1100	8,4079
1200	8,2445
1300	8,3935
1400	8,5159
1500	8,5912
1600	8,5557
1700	8,2792
1800	8,5526
1900	8,407
2000	8,2941

Fonte: elaborada pela autora.

Tabela 15: Valores médios de Aptidão das populações – 2015.1

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
100	17	6,7139	2
200	17	6,49	1
300	15	6,5245	1
400	15	6,5914	1
500	16	6,5192	1
600	16	6,509	1
700	15	6,5722	1
800	17	6,6021	1
900	17	6,4746	1
1000	15	6,6067	1

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
1100	15	6,6158	1
1200	15	6,5951	1
1300	15	6,5905	1
1400	17	6,458	1
1500	17	6,5666	1
1600	15	6,4705	1
1700	15	6,5025	1
1800	16	6,5343	1
1900	18	6,5028	1
2000	15	6,4621	1

Fonte: elaborada pela autora.

Tabela 16: Valores médios de aptidão das gerações – 2015.1

Gerações	Média das gerações	Gerações	Média das gerações
100	6,7139	1100	6,6158
200	6,4900	1200	6,5951
300	6,5245	1300	6,5905
400	6,5914	1400	6,4580
500	6,5192	1500	6,5666
600	6,5090	1600	6,4705
700	6,5722	1700	6,5025
800	6,6021	1800	6,5343
900	6,4746	1900	6,5028
1000	6,6067	2000	6,4621

Fonte: elaborada pela autora.

Tabela 17: Valores médios de Aptidão das populações – 2015.2

Gerações	Pior aptidão	Aptidão Média	Melhor aptidão	Gerações	Pior aptidão	Aptidão Média	Melhor aptidão
100	49,000	39,425	32,000	1100	48,000	39,313	32,000
200	47,000	39,230	31,000	1200	49,000	39,437	32,000
300	49,000	39,458	32,000	1300	47,000	39,376	32,000
400	47,000	39,410	32,000	1400	48,000	39,224	32,000
500	46,000	39,231	31,000	1500	48,000	39,239	33,000
600	46,000	39,257	31,000	1600	48,000	39,598	32,000
700	47,000	39,515	32,000	1700	48,000	39,200	31,000
800	47,000	39,176	31,000	1800	46,000	39,424	33,000
900	49,000	39,295	31,000	1900	47,000	39,522	33,000
1000	46,000	39,206	32,000	2000	47,000	39,364	31,000

Fonte: elaborada pela autora.

Tabela 18: Valores médios de aptidão das gerações – 2015.2

Gerações	Média das gerações	Gerações	Média das gerações
100	39,4254	1100	39,3133
200	39,2304	1200	39,4370
300	39,4584	1300	39,3760
400	39,4102	1400	39,2237
500	39,2314	1500	39,2385
600	39,2567	1600	39,5981
700	39,5149	1700	39,1996
800	39,1759	1800	39,4242
900	39,2945	1900	39,5222
1000	39,2063	2000	39,3640

Fonte: elaborada pela autora.

ANEXOS

ANEXO A - Estrutura Curricular do Curso de Bacharelado em Sistemas de
Informação

DADOS DA ESTRUTURA CURRICULAR

Código: 01A

Matriz Curricular: SISTEMAS DE INFORMAÇÃO - CAICÓ - Presencial - MT - BACHARELADO

Unidade de Vinculação: CENTRO DE ENSINO SUPERIOR DO SERIDÓ (18.00)

Município de funcionamento: CAICÓ - RN

Período Letivo de Entrada em Vigor: 2011 . 1

Carga Horária Mínima:	Obrigatória	Optativas	Complementar	Total
	2820h	180h	h	3000h

Créditos Obrigatórios: 148cr Total - (30cr Práticos) / (118cr Teóricos)

Carga Horária Obrigatória: 2490h Total - (720h Práticas) / (1770h Teóricas)

Carga Horária Obrigatória de Atividade Acadêmica Específica: 270 hrs

Carga Horária de Componentes Eletivos: Máxima (240 horas)

Carga Horária por Período Letivo: Mínima (120 horas)

Créditos por Período Letivo: Mínimo 8 Médio 24 Máximo 28

Prazos em Períodos Letivos: Mínimo 8 Médio 8 Máximo 12

Componentes Optativos

Componente Curricular	CH Detalhada	Tipo	Natureza
BS11111 PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS12103 SISTEMAS DISTRIBUÍDOS - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OPTATIVO
BS12203 MINERAÇÃO DE DADOS - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS12401 GESTÃO DA INFORMAÇÃO E DOS SISTEMAS DE INFORMAÇÃO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS12402 AUDITORIA E SEGURANÇA DE SISTEMAS DE INFORMAÇÃO - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OPTATIVO
BS12404 TRABALHO COOPERATIVO BASEADO EM COMPUTADOR - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS12501 COMPUTAÇÃO GRÁFICA - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OPTATIVO
BS12502 NOÇÕES DE ELETRÔNICA DIGITAL E PROGRAMAÇÃO DE MICROCONTROLADORES - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS12602 APRENDIZADO DE MÁQUINA - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS12701 INTERAÇÃO HUMANO-COMPUTADOR - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS14101 ADMINISTRAÇÃO DE SERVIDORES DE REDE - 60h (4cr)	15h (1cr) aula 45h (3cr) lab.	DISCIPLINA	OPTATIVO
BS14102 GERÊNCIA DE REDES - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OPTATIVO
BS14201 TÓPICOS ESPECIAIS ENGENHARIA DE SOFTWARE - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OPTATIVO
BS14202 TÓPICOS ESPECIAIS EM BANCOS DE DADOS - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OPTATIVO
BS14301 TÓPICOS EM SISTEMAS DE INFORMAÇÃO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OPTATIVO
BS14302 TÓPICOS ESPECIAIS EM SEGURANÇA DA INFORMAÇÃO - 60h (4cr) aula		DISCIPLINA	OPTATIVO

60h (4cr)		0h (0cr) lab.		
BSI4401	TÓPICOS AVANÇADOS EM PROGRAMAÇÃO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA OPTATIVO	
CEA0403	INTRODUÇÃO À ECONOMIA - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA OPTATIVO	
DAE0016	ENADE - INGRESSANTE - 0h (0cr)	0h (0cr) aula 0h (0cr) lab.	ATIVIDADE INTEGRADORA DE FORMAÇÃO OPTATIVO	
DAE0019	ENADE - CONCLUINTE - 0h (0cr)	0h (0cr) aula 0h (0cr) lab.	ATIVIDADE INTEGRADORA DE FORMAÇÃO OPTATIVO	
DED0327	SOCIOLOGIA GERAL - 90h (6cr)	90h (6cr) aula 0h (0cr) lab.	DISCIPLINA OPTATIVO	
DED0800	LÍNGUA BRASILEIRA DE SINAIS - LIBRAS - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA OPTATIVO	
DHC0702	HISTÓRIA E CULTURA INDÍGENA E AFRO-BRASILEIRA - 60h (4cr)	45h (3cr) aula 15h (1cr) lab.	DISCIPLINA OPTATIVO	
DHG0248	SEMINÁRIO DE MEIO AMBIENTE E DESENV. SUSTENT. NO SEMI-ARIDO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA OPTATIVO	
DIR0080	FUNDAMENTOS DO DIREITO E LEGISLAÇÃO DIGITAL - 30h (2cr)	30h (2cr) aula 0h (0cr) lab.	DISCIPLINA OPTATIVO	
CH Total: 1380hrs.				
Componentes Complementares				
Componente Curricular		CH Detalhada	Tipo	Natureza
CH Total: 0hrs.				
1º Nível				
Componente Curricular		CH Detalhada	Tipo	Natureza
BSI1101	ALGORITMOS E LÓGICA DE PROGRAMAÇÃO - 90h (6cr)	90h (6cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1102	INTRODUÇÃO À INFORMÁTICA - 60h (4cr)	45h (3cr) aula 15h (1cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1301	FUNDAMENTOS DE MATEMÁTICA - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1302	LÓGICA - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI3101	TEORIA GERAL DA ADMINISTRAÇÃO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
CH Total: 330hrs.				
2º Nível				
Componente Curricular		CH Detalhada	Tipo	Natureza
BSI1106	PROGRAMAÇÃO - 90h (6cr)	45h (3cr) aula 45h (3cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1201	TEORIA GERAL DOS SISTEMAS - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1303	CÁLCULO DIFERENCIAL E INTEGRAL - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
DED0005	METODOLOGIA DO TRABALHO CIENTIFICO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
DED0405	LEITURA E PRODUÇÃO DE TEXTO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
CH Total: 330hrs.				
3º Nível				
Componente Curricular		CH Detalhada	Tipo	Natureza
BSI1107	PROGRAMAÇÃO ORIENTADA A OBJETOS I - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1202	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1304	ÁLGEBRA LINEAR - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
		60h (4cr) aula		

BSI3102 ORGANIZAÇÃO, SISTEMAS E MÉTODOS - 60h (4cr)	0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
CEA0008 ESTRUTURA DE DADOS - 90h (6cr)	60h (4cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
CH Total: 330hrs.			
4º Nível			
Componente Curricular	CH Detalhada	Tipo	Natureza
BSI1105 ARQUITETURA DE COMPUTADORES - 60h (4cr)	45h (3cr) aula 15h (1cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1109 PROGRAMAÇÃO WEB - 60h (4cr)	15h (1cr) aula 45h (3cr) lab.	DISCIPLINA	OBRIGATORIO
BSI1305 PROBABILIDADE E ESTATÍSTICA - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI2201 BANCO DE DADOS - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI2301 ENGENHARIA DE SOFTWARE I - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
DED0006 INGLÊS TÉCNICO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
CH Total: 360hrs.			
5º Nível			
Componente Curricular	CH Detalhada	Tipo	Natureza
BSI1108 PROGRAMAÇÃO ORIENTADA A OBJETOS II - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI2101 SISTEMAS OPERACIONAIS - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI2202 PROJETO E ADMINISTRAÇÃO DE BANCO DE DADOS - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI2302 ENGENHARIA DE SOFTWARE II - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
CH Total: 240hrs.			
6º Nível			
Componente Curricular	CH Detalhada	Tipo	Natureza
BSI2102 REDES DE COMPUTADORES - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI2303 GESTÃO DE PROJETO DE SOFTWARE - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI3103 EMPREENDEDORISMO EM INFORMÁTICA - 60h (4cr)	30h (2cr) aula 30h (2cr) lab.	DISCIPLINA	OBRIGATORIO
BSI3202 CONTABILIDADE E CUSTOS - 90h (6cr)	90h (6cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
DED0328 FILOSOFIA I - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
CH Total: 330hrs.			
7º Nível			
Componente Curricular	CH Detalhada	Tipo	Natureza
BSI1110 PROGRAMAÇÃO VISUAL - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI5001 ESTÁGIO SUPERVISIONADO I - 120h (0cr)	0h (0cr) aula 0h (0cr) lab.	ESTÁGIO	OBRIGATORIO
BSI5003 TRABALHO CONCLUSÃO CURSO I - 90h (0cr)	0h (0cr) aula 90h (0cr) lab.	ATIVIDADE INTEGRADORA DE FORMAÇÃO	OBRIGATORIO
CEA0009 MATEMÁTICA FINANCEIRA - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
DIR0013 DIREITO E LEGISLAÇÃO SOCIAL - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
CH Total: 390hrs.			
8º Nível			
Componente Curricular	CH Detalhada	Tipo	Natureza

BSI0435	ÉTICA - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI2403	SISTEMAS DE APOIO À DECISÃO - 60h (4cr)	60h (4cr) aula 0h (0cr) lab.	DISCIPLINA	OBRIGATORIO
BSI5002	ESTÁGIO SUPERVISIONADO II - 120h (0cr)	0h (0cr) aula 0h (0cr) lab.	ESTÁGIO	OBRIGATORIO
BSI5004	TRABALHO CONCLUSÃO CURSO II - 90h (0cr)	0h (0cr) aula 0h (0cr) lab.	TRABALHO DE CONCLUSÃO DE CURSO	OBRIGATORIO
BSI5005	ATIVIDADES COMPLEMENTARES - 180h (0cr)	0h (0cr) aula 180h (0cr) lab.	ATIVIDADE INTEGRADORA DE FORMAÇÃO	OBRIGATORIO
CH Total: 510hrs.				

SIGAA | Superintendência de Informática - (84) 3215-3148 | Copyright © 2006-2015 - UFRN -
sistemas11b.sistemas11b1

ANEXO B - Montagens de Horários Realizadas nos Semestres Anteriores de Forma
Manual

Horários 2012.1

Nível	Local	Pré-req	Código	Turma	Disciplinas	CR/CH	Dia da Semana					Vagas	Professor
							2ª	3ª	4ª	5ª	6ª		
I	Lab. Turing		BSI1101	1	ALGORITMOS E LÓGICA DE PROGRAMAÇÃO	06/90	T345		T345			50	Flavius
	B-09		BSI1102	1	INTRODUÇÃO À INFORMÁTICA	04/60		T34		T34		50	Luiz Paulo
	B-09		BSI1102	2	INTRODUÇÃO À INFORMÁTICA	04/60		M45		M45		50	Luiz Paulo
	B-09		BSI1301	1	FUNDAMENTOS DE MATEMÁTICA	04/60	T12		T12			50	Luciano
	B-09		BSI1302	1	LÓGICA	04/60		T12		T12		50	João Paulo
	B-09		BSI1302	2	LÓGICA	04/60		M23		M23		50	João Paulo
	B-09		BSI3101	1	TEORIA GERAL DA ADMINISTRAÇÃO	04/60				T56	T34	50	Gilson
II	Lab. Turing		BSI1106	1	PROGRAMAÇÃO	06/90			M345		M345	40	Enéias
III	B-10	BSI1301	BSI1304	1	ÁLGEBRA LINEAR	04/60	T34		T34			50	Delson
	Lab. Turing	BSI1106	CEA0008	1	ESTRUTURA DE DADOS	06/90		T345		T345		50	João Paulo
	B-10	BSI1201	BSI1202	1	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO	04/60			T56		T12	50	Gilson
	B-10		BSI3102	1	ORGANIZAÇÃO, SISTEMAS E MÉTODOS	04/60		T12		T12		50	Karlíane
	Lab. Turing	BSI1106	BSI1107	1	PROGRAMAÇÃO ORIENTADA A OBJETOS I	04/60	T56		T12			50	Fabício
V	B-01		BSI3103	1	EMPREENDEDORISMO EM INFORMÁTICA	04/60			T34	T12		50	Gilson
	B-01/Lab. Turing	BSI2301	BSI2302	1	ENGENHARIA DE SOFTWARE II	04/60		T12		T56		50	Taciano
	Lab. Neumann	BSI1108	BSI1109	1	PROGRAMAÇÃO WEB	04/60	T12		T56			50	Fabício
	Lab. Neumann	BSI2201	BSI2202	1	PROJETO E ADMIN. DE BANCO DE DADOS	04/60			T12		T12	50	Taciano
	B-01	BSI1105	BSI2101	1	SISTEMAS OPERACIONAIS	04/60		T34		T34		50	João Borges
VII	B-02		DIR0013	2	DIREITO E LEGISLAÇÃO SOCIAL	04/60				T12	T12	40	Rogério/Oswaldo
	B-02	BSI1301	CEA0009	1	MATEMÁTICA FINANCEIRA	04/60		T12		T34		40	Luciano
	B-02	DED0405	BSI0435	1	ÉTICA	04/60				T56	T34	40	---
			BSI5001		ESTÁGIO SUPERVISIONADO I	120						---	---
			BSI5003		TRABALHO DE CONCLUSÃO DE CURSO I	90						---	---
	B-02	BSI1301	BSI3202	1	CONTABILIDADE E CUSTOS	06/90	T34	T34	T34			40	Celso
	B-02	BSI2302	BSI2303	1	GESTÃO DE PROJETO DE SOFTWARE	04/60	T56	T56				20	Karlíane
OP	B-02		BSI2403	1	SISTEMA DE APOIO A DECISÃO	04/60	T12		T12			40	Flavius
	B-02	BSI2101	BSI2103	1	SISTEMAS DISTRIBUÍDOS	04/60	T34		T34			20	João Borges
	B-02		BSI2701	1	INTERFACE HUMANO-COMPUTADOR	04/60			T56		T56	20	Karlíane

Horários BSI 2012.2

Horários BSI 2012.2													
Nível	Local	Pré-req	Código	Turma	Disciplinas	CR/CH	Dia da Semana					Vagas	Professor
							2ª	3ª	4ª	5ª	6ª		
I	Lab. Neumann		BSI1101	1	Algoritmos e Lógica de Programação	06/90				T345	M345	20	José Enéas
	B-02		BSI1302	1	Lógica	04/60		M34		M34		40	João Paulo
II	B-02		BSI1303	1	Cálculo Diferencial e Integral	04/60	T12		T12			50	Désio
	B-02		DED0405	1	Leitura e Produção de Texto	04/60	T34	T12				50	Célia
	Lab. Turing		BSI1106	1	Programação	06/90		T345		T345		50	Flavius
	B-02		DED0005	1	Metodologia do Trabalho Científico	04/60			T34		T12	50	Gilson
	B-02		BSI1201	1	Teoria Geral de Sistemas	04/60				T12	T34	50	Gilson
IV	B-04		BSI1305	1	Probabilidade e Estatística	04/60	T12		T12			50	Luciano
	Lab. Turing		BSI1108	1	Programação Orientada a Objetos II	04/60	T34		T34			50	Fabrizio
	Lab. Turing/B-04		BSI2301	1	Engenharia de Software I	04/60	T56	T56				50	Fabrizio
	Lab. Turing		BSI2201	1	Banco de Dados	04/60			T56		T12	50	Taciano
	B-04		BSI1105	1	Arquitetura de Computadores	04/60		T34		T34		50	Luiz Paulo
	B-04		BSI1105	2	Arquitetura de Computadores	04/60		M45		M45		50	Luiz Paulo
	B-04		DED0006	1	Inglês Técnico	04/60				T12	T34	50	Leomarques
VI	B-01		BSI2202	1	Contabilidade e Custos	06/90	T12	T12	T12			50	Ricardo
	Lab. Neumann		BSI2102	1	Redes de Computadores	04/60	T56		T56			50	João Borges
	B-01		BSI2303	1	Gestão de Projetos de Software	04/60		T34		T34		50	Karlane
	B-01		DED0328	1	Filosofia I	04/60					T3456	50	Talitha
	Lab. Neumann/B-01		BSI4302	1	Tópicos Especiais em Segurança da Informação	04/60		T56		T56		20	João Borges
	Lab. Neumann		BSI4301	1	Tópicos em Sistemas de Informação (Técnica de Algoritmos)	04/60				T12	T12	20	José Enéas
VIII	Lab. Ada Lovelace		BSI2403	1	Sistema de Apoio a Decisão	04/60		T12		T12		20	Flavius
	B-03		BSI1110	1	Programação Visual	04/60	T34		T34			50	Karlane e Luiz Paulo
	B-03		DIR0013	1	Direito e Legislação Social	04/60	T56	T56				50	Vyma
	B-03		BSI0435	1	Ética	04/60				T34	T12	50	Carlos
	B-03		BSI2603	1	Algoritmos Experimentais	04/60		T12		T12		20	João Paulo
	Lab. Neumann		BSI4201	1	Tópicos Especiais Engenharia de Software (Ambiente e Prática de Programação)	04/60		M34		M34		20	Taciano

Horários BSI 2013.1														
Nível	Local	Pré-req	Código	Turno	Disciplinas	CR/CH	De da Semana	2ª	3ª	4ª	5ª	6ª	Vagas	Professor
I	Lab. Turing		BSI1101	1	Algoritmos e Lógica de Programação	06/60	T345			T345			30	Flavius
	Lab. Turing		BSI1101	2	Algoritmos e Lógica de Programação	06/60	M123	M123					30	Flavius
	B-01		BSI1302	1	Lógica	04/60			T12		T12		30	João Paulo
	B-01		BSI1302	2	Lógica	04/60			M23		M23		30	João Paulo
	B-01		BSI1102	1	Introdução à Informática	04/60			T34		T34		30	Luiz Paulo
	B-01		BSI1102	2	Introdução à Informática	04/60			M45		M45		30	Luiz Paulo
	B-01		BSI1301	1	Fundamentos da Matemática	04/60		T12		T12			30	Substituto Matemática
	B-01		BSI1301	2	Fundamentos da Matemática	04/60		M12	M34				30	Barroca
B-01		BSI3101	1	Teoria Geral da Administração	04/60					T56	T34	50	Gilson	
II	B-09		BSI1303	1	Cálculo Diferencial e Integral	04/60					M34	M34	40	Concurso Eletivo Matemática
	Lab. Turing		BSI1106	1	Programação	06/60				M345		M345	20	Walber
III	B-02	BSI1301	BSI1304	1	Álgebra Linear	04/60					T12	T34	50	Desio
	Lab. Turing	BSI1106	BSI1107	1	Programação Orientada a Objetos I	04/60	T56			T12			50	Fabrizio
	Lab. Neumann	BSI1106	BSI1107	2	Programação Orientada a Objetos I	04/60	M45	M12					20	Flavius
	Lab. Turing		BSI3102	1	Organização, Sistemas e Métodos	04/60	T12	T12					50	Substituto Karliane
	Lab. Turing	BSI1106	CEAD008	1	Estrutura de Dados	06/60			T34	T34	T34		50	João Paulo
	B-02	BSI1201	BSI1202	1	Fundamentos de Sistemas de Informação	04/60				T56		T12	50	Gilson
V	B-03	BSI1105	BSI2102	1	Redes de Computadores	04/60	T56			T34			30	João Borges
	Lab. Neumann	BSI2301	BSI2302	1	Engenharia de Software II	04/60	T34					T34	50	Tadane
	Lab. Neumann	BSI1108	BSI1109	1	Programação Web	04/60	T12			T56			50	Fabrizio
	Lab. Neumann	BSI2301	BSI2302	1	Projeto e Adm. de Banco de Dados	04/60				T12		T12	50	Tadane
	B-03	BSI1105	BSI2101	1	Sistemas Operacionais	04/60			T34		T34		50	João Borges
VII			BSI5001	1	Estágio Supervisionado I	08/120								
			BSI5003	1	Trabalho de Conclusão de Curso I	06/60								
	Lab. Neumann	BSI1109	BSI1110	1	Programação Visual	04/60	T56			T34			50	Substituto Karliane
	B-04	BSI1301	CEAD009	1	Matemática Financeira	04/60				T12		T12	50	Manassés
	B-04		DIR0013	1	Direito e Legislação Social	04/60					T34	T34	50	Vyma
OP	Lab. Turing		BSI1111	1	Programação para Dispositivos Móveis	04/60			T56		T56		25	Walber
	Lab. Ada	BSI2102	BSI2103	1	Sistemas Distribuídos	04/60			T56		T56		20	João Borges
	Labepi		BSI2503	1	Microcontroladores	04/60			T56		T56		10	Luiz Paulo
	Lab. Neumann		BSI4201	1	Tópicos Especiais Engenharia de Software (Teste de Software)	04/60			T56		T56		20	Substituto Karliane

Horários BSI 2013.2

Nível	Local	Pré-req	Código	Turno	Disciplinas	CR/CH	Dia da Semana					Vagas	Professor
							2ª	3ª	4ª	5ª	6ª		
I	B-02		BSI1102	1	Introdução à Informática	04/60		M45		M45		50	Lutz Paulo
II	B-02		BSI1303	1	Cálculo Diferencial e Integral	04/60	T45		T12			50	
II	B-02		DED0405	1	Leitura e Produção de Texto	04/60		T45		T34		50	
II	Lab. Turing		BSI1106	1	Programação	06/90	T123	T123				50	Flavius
II	B-02		DED0005	1	Metodologia do Trabalho Científico	04/60			T34		T12	50	Gilson
II	B-02		BSI1201	1	Teoria Geral de Sistemas	04/60				T12	T34	50	Gilson
III	Lab. Turing		BSI1107	1	Programação Orientada a Objetos I	04/60	M45				M45	50	Fabrizio
III	Lab. Turing/B-03/B-08		GEA0008	1	Estrutura de Dados	06/90	T12	T12	T12			50	João Paulo
IV	B-04		BSI1305	1	Probabilidade e Estatística	04/60		T12	T12			50	
IV	Lab. Turing		BSI1108	1	Programação Orientada a Objetos II	04/60	T56				T34	50	Fabrizio
IV	Lab. Turing		BSI2301	1	Engenharia de Software I	04/60		T56		T56		50	Karlane
IV	Lab. Turing		BSI2301	1	Banco de Dados	04/60	T34				T12	50	Tadano
IV	B-04		BSI1105	1	Arquitetura de Computadores	04/60		T34		T34		50	Lutz Paulo
IV	B-04		DED0006	1	Inglês Técnico	04/60			T34	T12		50	
VII	B-01		BSI3202	1	Contabilidade e Custos	06/90	T12	T12	T12			50	
VII	Lab. Neumann		BSI2102	1	Redes de Computadores	04/60		T56		T56		50	João Borges
VII	B-01		BSI2303	1	Gestão de Projetos de Software	04/60		T34	T34			50	Karlane
VII	B-01		BSI3103	1	Empreendedorismo	04/60			T56	T34		50	Gilson
VII	B-01		DED0328	1	Filosofia I	04/60				T12	T12	50	
VIII	Lab. Ada Lovelace		BSI2403	1	Sistema de Apoio à Decisão	04/60	T56	T56				20	Flavius
VIII	B-03		BSI0435	1	Ética	04/60				T34	T34	50	Carlos
VIII	Lab. Neumann		BSI4201	1	Tópicos Especiais Engenharia de Software (Ambiente e Prática de Programação)	04/60	M45				M45	20	Tadano

TURMAS 2014.2 - QUADRO GERAL DE DISCIPLINAS POR PROFESSOR

PERÍODO	CÓDIGO	DISCIPLINA	CH (H)	DOCENTE	TURMA	LOCAL	HORÁRIO
2º	DED0405	LEITURA E PRODUÇÃO DE TEXTO	60	A DEFINIR	1		2T34 5T12
4º	DED0006	INGLÊS TÉCNICO	60	A DEFINIR	1		46T34
6º	BSI3202	CONTABILIDADE DE CUSTOS	90	A DEFINIR	1		3T34 46T56
6º	DED0328	FILOSOFIA	60	A DEFINIR	1		23T12
2º	BSI1201	TEORIA GERAL DOS SISTEMAS	60	ADRIANE	1		46T12
6º	BSI3103	EMPREENDEDORISMO EM INFORMÁTICA	60	ADRIANE	1		56T34
8º	BSI0435	ÉTICA	60	ADRIANE	1		4T34 5T12
4º	BSI2201	BANCO DE DADOS	60	AISLÂNIA	1		23T56
5º	BSI2202	PROJETO E ADMINISTRAÇÃO DE BANCO DE DADOS	60	AISLÂNIA	1		2M34 4M12
OPT	BSI4402	TÓPICOS ESPECIAIS EM BANCO DE DADOS	60	AISLÂNIA	1		24M56
1º	BSI1101	ALGORITMOS E LÓGICA DE PROGRAMAÇÃO	90	AMARILDO	1		456M12
3º	BSI1107	PROGRAMAÇÃO ORIENTADA A OBJETOS I	60	AMARILDO	1		45M34
4º	BSI1109	PROGRAMAÇÃO WEB	60	AMARILDO	1		56T12
1º	BSI1301	FUNDAMENTOS DE MATEMÁTICA	60	DEILSON	1		23M12
2º	BSI1303	CÁLCULO DIFERENCIAL E INTEGRAL	60	DEILSON	1		23T12
3º	BSI1304	ÁLGEBRA LINEAR	60	DEILSON	1		23M34
4º	BSI1305	PROBABILIDADE E ESTATÍSTICA	60	DEILSON	1		23T34
8º	BSI2403	SISTEMAS DE APOIO À DECISÃO	60	FLAVIUS	1		23M12
OPT	?	TÓPICOS EM INTELIGÊNCIA ARTIFICIAL	60	FLAVIUS	1		23M34
2º	DED0005	METODOLOGIA DO TRABALHO CIENTÍFICO	60	GILSON	1		45T56
OPT	BSI1203	CLOUD COMPUTING	60	GILSON	1		45M34
2º	BSI1106	PROGRAMAÇÃO	90	JOÃO PAULO	1		345T34
OPT	BSI4401	TÓPICOS AVANÇADOS EM PROGRAMAÇÃO	60	JOÃO PAULO	1		35T12
4º	BSI2301	ENGENHARIA DE SOFTWARE I	60	KARLIANE	1		24T12
6º	BSI2303	GESTÃO DE PROJETO DE SOFTWARE	60	KARLIANE	1		3T56 4T34
OPT	BSI1111	PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS	60	ODILON	1		26M12
OUTROS	LMC0013	ALGORITMO E LÓGICA DE PROGRAMAÇÃO	90	ODILON	-		345M12
6º	BSI2102	REDES DE COMPUTADORES	60	RIVALDO	1		5T56 6T12
OPT	BSI4101	ADMINISTRAÇÃO DE SERVIDORES DE REDE	60	RIVALDO	1		6M3456
1º	BSI1102	INTRODUÇÃO À INFORMÁTICA	60	TIAGO	1		35M34
4º	BSI1105	ARQUITETURA DE COMPUTADORES	60	TIAGO	1		3T12 5T34
OPT	BSI2701	INTERAÇÃO HUMANO-COMPUTADOR	60	TIAGO	1		3T34 5T56

TURMAS 2013.1 - QUADRO GERAL DE DISCIPLINAS POR PROFESSOR

Período	Local	Pré-requisito	Código	Turma	Disciplinas	CR/CH	Horários	Vagas	Professor
1º	LAB. ALAN TURING	-	BSI1101	1	ALGORITMOS E LÓGICA DE PROGRAMAÇÃO	06/90	23T34S	30	FLAVIUS
	LAB. ALAN TURING	-	BSI1101	2	ALGORITMOS E LÓGICA DE PROGRAMAÇÃO	06/90	23M123	30	FLAVIUS
	B1	-	BSI1102	1	INTRODUÇÃO À INFORMÁTICA	04/60	45T12	50	TIAGO
	B1	-	BSI1301	1	FUNDAMENTOS DE MATEMÁTICA	04/60	23T12	50	DEILSON
	B1	-	BSI1302	1	LÓGICA	04/60	45T34	50	JOÃO PAULO
	B1	-	BSI3101	1	TEORIA GERAL DA ADMINISTRAÇÃO	04/60	5T56 6T12	50	ADRIANNE
2º	B1	BSI1301	BSI1303	1	CÁLCULO DIFERENCIAL E INTEGRAL	04/60	23M12	50	DEILSON
	LAB. VON NEUMANN	BSI1101	BSI1106	1	PROGRAMAÇÃO	06/90	45M34S	20	A DEFINIR
3º	LAB. ALAN TURING	BSI1106	BSI1107	1	PROGRAMAÇÃO ORIENTADA A OBJETOS I	04/60	5T34 6T12	40	AMARILDO
	B2	BSI1301	BSI1202	1	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO	04/60	5T56 6T34	50	GILSON
	B2	BSI1301	BSI1304	1	ÁLGEBRA LINEAR	04/60	23T34	50	DEILSON
	LAB. VON NEUMANN	-	BSI3102	1	ORGANIZAÇÃO, SISTEMAS E MÉTODOS	04/60	2T12 4T34	20	KARLIANE
	B2	BSI1106	CEA0008	1	ESTRUTURA DE DADOS	06/90	34S12	50	JOÃO PAULO
4º	LAB. VON NEUMANN	BSI1104 OU BSI1106	BSI2201	1	BANCO DE DADOS	04/60	23M12	20	AISLÂNIA
	B1	BSI1301	BSI1305	1	PROBABILIDADE E ESTATÍSTICA	04/60	23M34	50	DEILSON
5º	LAB. ALAN TURING	BSI1107 E BSI1109	BSI1108	1	PROGRAMAÇÃO ORIENTADA A OBJETOS II	04/60	5T12 6T34	40	AMARILDO
	B3	BSI1105	BSI2101	1	SISTEMAS OPERACIONAIS	04/60	45T34	50	TIAGO
	LAB. ALAN TURING	BSI2201	BSI2202	1	PROJETO E ADMINISTRAÇÃO DE BANCO DE DADOS	04/60	23T12	40	AISLÂNIA
	LAB. VON NEUMANN	BSI3201 E BSI2201 E BSI1109	BSI2302	1	ENGENHARIA DE SOFTWARE II	04/60	2T34 4T12	20	KARLIANE
6º	LAB. VON NEUMANN	BSI1105	BSI2102	1	REDES DE COMPUTADORES	04/60	5T56 6T12	20	RIVALDO
7º	B4	BSI1108	BSI1110	1	PROGRAMAÇÃO VISUAL	04/60	45T56	50	TIAGO
	B4	BSI1301	CEA0009	1	MATEMÁTICA FINANCEIRA	04/60	45T12	50	A DEFINIR
	B4	-	DIRE0013	1	DIREITO E LEGISLAÇÃO SOCIAL	04/60	45T34	50	A DEFINIR
8º	LAB. VON NEUMANN	BSI2202	BSI2403	1	SISTEMAS DE APOIO À DECISÃO	04/60	23M45	20	FLAVIUS
OPT	LAB. IMD	BSI1107	BSI1111	1	PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS	04/60	2M12 4M34	20	A DEFINIR
OPT	LABEPI	CEA0008	BSI2603	1	ALGORITMOS EXPERIMENTAIS	04/60	35M34	10	JOÃO PAULO
OPT	LAB. VON NEUMANN	-	BSI4301	1	TÓPICOS EM SISTEMAS DE INFORMAÇÃO	04/60	56T34	20	ADRIANNE
OPT	LAB. VON NEUMANN	BSI1107	BSI4401	1	TÓPICOS AVANÇADOS EM PROGRAMAÇÃO	04/60	56M12	20	AMARILDO
OPT	LAB. VON NEUMANN	BSI2102	BSI4101	1	ADMINISTRAÇÃO DE SERVIDORES DE REDE	04/60	5T56 6T12	20	RIVALDO
OPT	LAB. VON NEUMANN	BSI2201	BSI4202	1	TÓPICOS ESPECIAIS EM BANCO DE DADOS	04/60	23T56	20	AISLÂNIA
OPT	LAB. IMD	-	BSI4201	1	TÓPICOS ESPECIAIS ENGENHARIA DE SOFTWARE	04/60	34M12	20	KARLIANE

TURMAS 2015.2 - QUADRO GERAL DE DISCIPLINAS POR PROFISSÃO

PERÍODO	CÓDIGO	DISCIPLINA	CR/CH	DOCENTE	TURMA	PRÉ-REQ*	UNIDADE	LOCAL	HORÁRIO
1ª	1801.001	ALGORITMOS E LÓGICA DE PROGRAMAÇÃO	04/80	ALEXANDRE	1	8	-02	LAB. ALAN TURING / B2	16H07-1712
1ª	1801.002	INTRODUÇÃO À INFORMÁTICA	04/80	TURAO	1	24	-02	B1	18H00
1ª	1801.003	FUNDAMENTOS DE MATEMÁTICA	04/80	OSILSON	1	81	-02	B1	21H07
1ª	1801.004	FÍSICA	04/80	TURAO	1	25	-02	B1	18H00
1ª	1801.006	PROGRAMAÇÃO	04/80	JOÃO PAULO	1	61	-02	LAB. ALAN TURING	14H14
1ª	1801.007	TEORIA GERAL DOS SISTEMAS	04/80	OSILSON	1	64	-02	B1	1712-1814
1ª	1801.008	CÁLCULO DIFERENCIAL E INTEGRAL	04/80	OSILSON	1	62	-02	B1	21H12
1ª	01.00008	METODOLOGIA DO TRABALHO CIENTÍFICO	04/80	ADRIANE	1	50	-02	B1	1700-1712
1ª	01.00009	LEITURA E PRODUÇÃO DE TEXTO	04/80	A. DEFFNER	1	67	-02	B1	21H1-1700
1ª	1801.007	PROGRAMAÇÃO ORIENTADA A OBJETOS I	04/80	RICARDO	1	29	-02	LAB. VON NEUMANN	18H00
1ª	1801.004	ALGORITMOS LINEARES	04/80	OSILSON	1	69	-02	B1	21H00
1ª	1801.009	ARQUITETURA DE COMPUTADORES	04/80	TURAO	1	80	-02	B2	16H00
1ª	1801.009	PROGRAMAÇÃO WEB	04/80	RICARDO	1	88	-02	LAB. ALAN TURING	16H12
1ª	1801.009	PROBABILIDADE E ESTATÍSTICA	04/80	OSILSON	1	68	-02	B2	21H00
1ª	1801.001	BANCO DE DADOS	04/80	APULVIR	1	80	-02	LAB. ALAN TURING	21H00
1ª	1801.001	LINGUAGEM DE SOFTWARE I	04/80	APULVIR	1	24	-02	LAB. ALAN TURING	14H12
1ª	01.00008	INTRODUÇÃO	04/80	A. DEFFNER	1	82	-02	LAB. ALAN TURING	1712-1704
1ª	1801.001	PROJETO E ADMINISTRAÇÃO DE BANCO DE DADOS	04/80	APULVIR	1	27	-02	LAB. ALAN TURING	21H00
1ª	1801.001	REDES DE COMPUTADORES	04/80	RIVELLO	1	25	-02	LAB. VON NEUMANN	1704-1712
1ª	1801.001	GESTÃO DE PROJETO DE SOFTWARE	04/80	SERLANY	1	80	-02	LAB. VON NEUMANN	21H00
1ª	1801.001	IMPLICAÇÕES COMPLEXAS EM INFORMÁTICA	04/80	ADRIANE	1	104	-02	B2	1704-1700
1ª	1801.001	CONTABILIDADE E CUSTOS	04/80	A. DEFFNER	1	86	-02	B2	21H12
1ª	01.00138	PSICOLOGIA I	04/80	A. DEFFNER	1	107	-02	B2	1712-1704
1ª	1801.110	PROGRAMAÇÃO VISUAL	04/80	ALEXANDRE	1		-02	LAB. VON NEUMANN	18H00
1ª	1801.001	ÉTICA	04/80	ADRIANE	1	67	-02	B2	1712-1704
1ª	1801.001	ATIVIDADES DE APOIO À DISCIPLINA	04/80	FLAVIUS	1	81	-02	LAB. VON NEUMANN	2112-1704
OPT	1801.001	MINERAÇÃO DE DADOS	04/80	FLAVIUS	1	87	-02	LAB. VON NEUMANN	21H07
OPT	1801.001	TÓPICOS EM MULTIMÍDIAS DE INFORMAÇÃO	04/80	JOÃO PAULO	1	149	-02	LAB. VON NEUMANN	18H00
OPT	1801.001	TÓPICOS AVANÇADOS EM PROGRAMAÇÃO	04/80	RICARDO	1	81	-02	LAB. VON NEUMANN	16H07
OPT	1801.001	APRENDIZADO DE MÁQUINA	04/80	SERLANY	1	88	-02	LAB. VON NEUMANN	21H01-18H12
OPT	1801.001	ADMINISTRAÇÃO DE SERVIÇOS DE NÚCLO	04/80	RIVELLO	1	62	-02	LAB. ALAN TURING	16H18-1700
OPT	01.00008	FUNDAMENTOS DO DIREITO E LEGISLAÇÃO DIGITAL	04/80	A. DEFFNER	1	20	-02	LAB. VON NEUMANN	18H00
OPT									

* ALINHOS PREVIÁVEIS E EFETOS PARA CURSOS O-CONJUNTO

ANEXO C – Horários Sugeridos pelo *Software Asc Timetable*

Horário 2012.1

aSc TimeTables 2014 - [document.roz]

Principal Arquivo Especificações Visualizar Horário Opções Ajuda Procurar: Personalizar

Novo Abrir Salvar Imprimir Visualizar impressão Total Disciplinas Turmas Salas Professores Seminários Relações Teste Criar novo Verificação Escola Perguntas? Comentários? Escreva-nos

Training_01.roz document.roz

	Segunda			Terça			Quarta			Quinta			Sexta		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
II															
ALG ₁	ALP	ALP													
II 2															
LOG					LOG	LOG									
LOG 2		LOG	LOG												
FMAT							FMAT	FMAT							
TGA													TGA	TGA	
PROG ₂							PROG								
ALGL ₁					ALGL										
ED ₂	ED														
FSI ₁															FSI
OSM ₁										OSM					
POOI ₁															POOI
EMP ₁								EMP							
ESII ₁	ESII														
DIR ₁			DIR												
MATF ₁							MATF								
ETIC ₁						ETIC									
CEC ₂												CEC			
GPS ₁	GPS														
SAD ₁					SAD										
SDIS ₁			SDIS												
IHC ₁												IHC			

ALP PROG POOI FSI ALGL OSM ED ESII GPS EMP
CEC MATF DIR ETIC SAD SDIS IHC

Ready NUM 70%