

FACULDADE INDEPENDENTE DO NORDESTE
CURSO DE ENGENHARIA DA COMPUTAÇÃO

RODRIGO BORGES DE OLIVEIRA

**FERRAMENTA COMPUTACIONAL PARA GERAÇÃO DE HORÁRIOS
ESCOLARES UTILIZANDO INTELIGÊNCIA ARTIFICIAL**

VITÓRIA DA CONQUISTA – BA

2016

RODRIGO BORGES DE OLIVEIRA

**FERRAMENTA COMPUTACIONAL PARA GERAÇÃO DE HORÁRIOS
ESCOLARES UTILIZANDO INTELIGÊNCIA ARTIFICIAL**

Monografia apresentada como requisito
parcial para obtenção do título de
Bacharel em Engenharia da Computação.

Orientador: MSc. Marcos Gomes Prado

VITÓRIA DA CONQUISTA – BA

2016

O48f

Oliveira, Rodrigo Borges de

Ferramenta computacional para geração de horários escolares utilizando inteligência artificial./ Rodrigo Borges de Oliveira .__ Vitória da Conquista, 2016.

111.f.il

Monografia (Graduação em Engenharia da computação)
Faculdade Independente do Nordeste - FAINOR

Orientador (a): Prof. Marcos Gomes Prado

1. Algoritmos genéticos. 2. Engenharia de software. 3. Ferramenta computacional. 4. Horários escolares. 5. Inteligência artificial. I. Título.

CDD: 006.3

RODRIGO BORGES DE OLIVEIRA

FERRAMENTA COMPUTACIONAL PARA GERAÇÃO DE HORÁRIOS ESCOLARES UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Monografia apresentada como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação pela Faculdade Independente do Nordeste.

Aprovado em 06/12/2016

BANCA EXAMINADORA / COMISSÃO AVALIADORA

Prof. MSc. Marcos Gomes Prado
Faculdade Independente do Nordeste – FAINOR

Prof. Esp. Joaquim Júnior Lacerda dos Santos
Faculdade Independente do Nordeste – FAINOR

Prof. MSc. Marcelo Barbosa de Almeida
Faculdade Independente do Nordeste – FAINOR

VITÓRIA DA CONQUISTA – BA

2016

Dedico este trabalho a todos aqueles que
me serviram como fonte de inspiração.

AGRADECIMENTOS

Aos meus pais e à minha irmã, por sempre me apoiarem e me incentivarem a seguir aquilo que gosto.

Aos meus familiares, em especial Tio Raymundo, por me ajudar a ter uma oportunidade melhor de educação.

Aos meus colegas, grandes amigos, pelo companheirismo e ajuda nos estudos em todos esses anos de faculdade.

A todos os professores que me fizeram buscar mais conhecimento.

Ao meu orientador, prof. MSc. Marcos Prado, por me ajudar a tornar esta monografia possível.

*“Tudo o que temos de decidir é o que fazer
com o tempo que nos é dado.”*

(J. R. R. Tolkien)

RESUMO

A geração de horários escolares, quando feita de forma manual, pode demandar muito tempo devido à sua característica complexa de possuir várias restrições em um único problema. O uso de sistemas computacionais aliados às técnicas de inteligência artificial, proporciona um auxílio e uma melhoria em diversas atividades realizadas cotidianamente em distintas áreas de atuação. Esta monografia descreve o projeto e desenvolvimento de um sistema para gerar horários escolares utilizando algoritmos genéticos. Seguiu-se os passos de modelagem do negócio, especificação de requisitos, projeto do software, codificação e testes no ciclo de desenvolvimento do software, onde foram utilizadas modelagens através de diagramas de caso de uso, de atividades, de classes e de entidade relacionamento para realizar o planejamento da ferramenta computacional proposta. Observou-se que o uso de algoritmos genéticos para resolução do problema de horários escolares proporciona uma grande melhoria na geração de grades onde uma solução não seria facilmente encontrada através de um trabalho manual.

Palavras-chave: Algoritmos Genéticos. Engenharia de Software. Ferramenta Computacional. Horários Escolares. Inteligência Artificial.

ABSTRACT

The generation of school schedules, when done manually, can take a long time because of its complex feature of having multiple constraints on a single problem. The use of computer systems combined with the techniques of artificial intelligence, provides an aid and an improvement in several activities carried out daily in different areas of operation. This monograph describes the design and development of a system to generate school schedules using genetic algorithms. The steps followed to accomplish it made use of business modeling, requirements specification, software design, coding and testing in the software development cycle, where modeling through use case, activity, class and entity relationship diagrams were used to plan the proposed computational tool. It was observed that the use of genetic algorithms to solve the problem of school schedules provides a great improvement in the generation of timetables where a solution would not be easily found through manual work.

Keywords: Artificial Intelligence. Computational Tool. Genetic Algorithms. School Timetables. Software Engineering.

LISTA DE FIGURAS

Figura 1 – Método da roleta.....	27
Figura 2 – Cruzamento em um ponto.....	28
Figura 3 – Cruzamento em dois pontos.....	28
Figura 4 – Cruzamento uniforme.....	29
Figura 5 – Mutação.....	30
Figura 6 – Ciclo de desenvolvimento do software.....	34
Figura 7 – Diagrama de caso de uso referente aos cadastros de dados.....	38
Figura 8 – Diagrama de caso de uso referente às funcionalidades.....	39
Figura 9 – Diagrama entidade relacionamento.....	42
Figura 10 – Diagrama de atividades do fluxo básico do funcionamento geral.....	43
Figura 11 – Diagrama de atividades do gerenciamento de professores.....	44
Figura 12 – Diagrama de atividades do gerenciamento de disciplinas.....	45
Figura 13 – Diagrama de atividades do gerenciamento de turmas.....	46
Figura 14 – Diagrama de atividades do gerenciamento de restrições.....	47
Figura 15 – Diagrama de atividades do gerenciamento de aulas conjuntas.....	48
Figura 16 – Diagrama de atividades do gerenciamento de horários.....	49
Figura 17 – Exemplo da população inicial.....	53
Figura 18 – Exemplo da primeira geração.....	54
Figura 19 – Exemplo da segunda geração.....	55
Figura 20 – Exemplo da terceira geração.....	56
Figura 21 – Diagrama de classes referente aos dados gerais.....	57
Figura 22 – Diagrama de classes referente ao algoritmo genético.....	58
Figura 23 – Relacionamento entre os dados gerais e o algoritmo genético.....	59
Figura 24 – Tela principal no Linux.....	61
Figura 25 – Barra de menus.....	61
Figura 26 – Barra de tarefas.....	61
Figura 27 – Tela principal no Windows.....	62
Figura 28 – Tela de geração de horário por turma.....	62
Figura 29 – Tela de exibição de horários.....	63

LISTA DE QUADROS

Quadro 1 – Descrição do caso de uso “Gerenciar professores”	77
Quadro 2 – Descrição do caso de uso “Cadastrar professor”	77
Quadro 3 – Descrição do caso de uso “Editar professor”	78
Quadro 4 – Descrição do caso de uso “Listar professores”	78
Quadro 5 – Descrição do caso de uso “Remover professor”	79
Quadro 6 – Descrição do caso de uso “Gerenciar disciplinas”	79
Quadro 7 – Descrição do caso de uso “Cadastrar disciplina”	80
Quadro 8 – Descrição do caso de uso “Editar disciplina”	81
Quadro 9 – Descrição do caso de uso “Listar disciplinas”	81
Quadro 10 – Descrição do caso de uso “Remover disciplina”	82
Quadro 11 – Descrição do caso de uso “Gerenciar turmas”	82
Quadro 12 – Descrição do caso de uso “Cadastrar turma”	83
Quadro 13 – Descrição do caso de uso “Editar turma”	84
Quadro 14 – Descrição do caso de uso “Listar turmas”	85
Quadro 15 – Descrição do caso de uso “Remover turma”	85
Quadro 16 – Descrição do caso de uso “Adicionar disciplina à turma”	86
Quadro 17 – Descrição do caso de uso “Remover disciplina da turma”	86
Quadro 18 – Descrição do caso de uso “Gerenciar restrições”	87
Quadro 19 – Descrição do caso de uso “Definir restrições”	87
Quadro 20 – Descrição do caso de uso “Listar restrições”	87
Quadro 21 – Descrição do caso de uso “Definir indisponibilidades do professor”	88
Quadro 22 – Descrição do caso de uso “Habilitar/desabilitar restrições de uma turma”	89
Quadro 23 – Descrição do caso de uso “Exibir indisponibilidades de um professor”	90
Quadro 24 – Descrição do caso de uso “Exibir restrições de uma turma”	90
Quadro 25 – Descrição do caso de uso “Gerenciar aulas conjuntas”	91
Quadro 26 – Descrição do caso de uso “Adicionar aula conjunta”	91
Quadro 27 – Descrição do caso de uso “Editar aula conjunta”	92
Quadro 28 – Descrição do caso de uso “Listar aulas conjuntas”	93
Quadro 29 – Descrição do caso de uso “Remover aulas conjuntas”	94
Quadro 30 – Descrição do caso de uso “Gerenciar horários”	94
Quadro 31 – Descrição do caso de uso “Gerar horário por curso”	95

Quadro 32 – Descrição do caso de uso “Gerar horário por turma”	96
Quadro 33 – Descrição do caso de uso “Exibir horário”	96
Quadro 34 – Descrição do caso de uso “Remover horário”	97
Quadro 35 – Especificação do requisito “Cadastrar professor”	98
Quadro 36 – Especificação do requisito “Editar professor”	99
Quadro 37 – Especificação do requisito “Listar professores”	99
Quadro 38 – Especificação do requisito “Remover professor”	99
Quadro 39 – Especificação do requisito “Cadastrar disciplina”	99
Quadro 40 – Especificação do requisito “Editar disciplina”	100
Quadro 41 – Especificação do requisito “Listar disciplinas”	100
Quadro 42 – Especificação do requisito “Remover disciplina”	100
Quadro 43 – Especificação do requisito “Cadastrar turma”	100
Quadro 44 – Especificação de requisito “Editar turma”	101
Quadro 45 – Especificação do requisito “Listar turmas”	101
Quadro 46 – Especificação do requisito “Remover turma”	101
Quadro 47 – Especificação do requisito “Adicionar disciplina à turma”	101
Quadro 48 – Especificação do requisito “Remover disciplina da turma”	102
Quadro 49 – Especificação do requisito “Definir indisponibilidades do professor” .	102
Quadro 50 – Especificação do requisito “Habilitar/desabilitar restrições de uma turma”	102
Quadro 51 – Especificação do requisito “Exibir indisponibilidades de um professor”	102
Quadro 52 – Especificação do requisito “Exibir restrições de uma turma”	103
Quadro 53 – Especificação do requisito “Adicionar aula conjunta”	103
Quadro 54 – Especificação do requisito “Editar aula conjunta”	103
Quadro 55 – Especificação do requisito “Listar aulas conjuntas”	103
Quadro 56 – Especificação do requisito “Remover aulas conjuntas”	104
Quadro 57 – Especificação do requisito “Gerar horário por curso”	104
Quadro 58 – Especificação do requisito “Gerar horário por turma”	104
Quadro 59 – Especificação do requisito “Exibir horário”	104
Quadro 60 – Especificação do requisito “Remover horário”	105
Quadro 61 – Especificação do requisito “Usabilidade”	105
Quadro 62 – Especificação do requisito “Confiabilidade”	105
Quadro 63 – Especificação do requisito “Portabilidade”	105

Quadro 64 – Especificação do requisito “Funcionalidade” 105

LISTA DE TABELAS

Tabela 1 – Testes com população de 5 indivíduos.....	65
Tabela 2 – Testes com população de 15 indivíduos.....	66
Tabela 3 – Testes com população de 25 indivíduos.....	67
Tabela 4 – Testes com população de 50 indivíduos.....	68
Tabela 5 – Testes com população de 100 indivíduos.....	69
Tabela 6 – Média de execução por tamanho da população.....	70
Tabela 7 – Média de execução por taxa de crossover.....	70
Tabela 8 – Média de execução por taxa de mutação.....	70
Tabela 9 – Tempo médio para gerar horários de um curso.....	71

LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas
CSP – Constraint Satisfaction Problems
DER – Diagrama Entidade Relacionamento
IBM – International Business Machines
IDE – Integrated Development Environment
IEC – International Electrotechnical Commission
IEEE – Institute of Electrical and Electronics Engineers
IHM – Interface Homem-Máquina
ISO – International Organization for Standardization
JVM – Java Virtual Machine
MDI – Multiple Document Interface
NBR – Norma Brasileira
PSR – Problema de Satisfação de Restrições
RUP – Rational Unified Process
SGBD – Sistema de Gerenciamento de Banco de Dados
SQL – Structured Query Language
UFLA – Universidade Federal de Lavras
UML – Unified Modeling Language
UNIPAC – Universidade Presidente Antônio Carlos
VRM – Valores Restantes Mínimos

SUMÁRIO

1 INTRODUÇÃO	16
1.1 CONTEXTUALIZAÇÃO DO PROBLEMA DE PESQUISA	17
1.2 QUESTÃO CENTRAL DA PESQUISA	17
1.3 HIPÓTESE CENTRAL	17
1.4 OBJETIVO GERAL	18
1.5 OBJETIVOS ESPECÍFICOS	18
1.6 JUSTIFICATIVA	18
1.7 ESTRUTURA DOS CAPÍTULOS DO PROJETO DE MONOGRAFIA	18
2 ESTADO DA ARTE	20
3 REFERENCIAL TEÓRICO	22
3.1 INTELIGÊNCIA ARTIFICIAL	22
3.2 PROBLEMA DE SATISFAÇÃO DE RESTRIÇÕES	24
3.3 ALGORITMOS GENÉTICOS	25
3.3.1 Seleção	26
3.3.2 Crossover	27
3.3.3 Mutação	29
3.3.4 Substituição	30
3.3.5 Critério de Parada	31
4 METODOLOGIA	32
4.1 TIPOS DE PESQUISA QUANTO AOS OBJETIVOS	32
4.2 TIPOS DE PESQUISA QUANTO À FORMA DE ANÁLISE	33
4.3 TIPOS DE PESQUISA QUANTO AOS MEIOS	33
4.4 INSTRUMENTOS DE PESQUISA	34
5 DESENVOLVIMENTO	37
5.1 MODELAGEM DO NEGÓCIO	37
5.2 ESPECIFICAÇÃO DE REQUISITOS	39
5.3 PROJETO DE SOFTWARE	41
5.3.1 Projeto Arquitetural	41
5.3.2 Projeto Procedimental	42
5.3.2.1 Exemplo de Funcionamento do Algoritmo	53
5.3.3 Projeto de Dados	56
5.3.4 Projeto de Interface	60
5.4 CODIFICAÇÃO	63
5.5 TESTES	64
6 CONCLUSÃO	72
6.1 INDICAÇÃO DE TRABALHOS FUTUROS	72
REFERÊNCIAS	74
APÊNDICE A – Descrições dos casos de uso	77
APÊNDICE B – Documento de requisitos	98
APÊNDICE C – Manual do usuário	106

1 INTRODUÇÃO

Construir e gerenciar os horários de aulas de uma instituição escolar é uma tarefa árdua se feita manualmente. A complexidade do estabelecimento e da administração da distribuição dos horários cresce drasticamente, devido ao fator inversamente proporcional existente entre o número de aulas a serem lecionadas e a quantidade de professores disponíveis responsáveis por cada disciplina. Dependendo da quantidade de combinações possíveis que podem ser realizadas, esta é uma tarefa que pode demandar alguns dias de trabalho para ser concluída, mesmo para aqueles que são especialistas na área.

Nas corporações produtivas, os computadores são utilizados diariamente para agilizar as atividades a serem desenvolvidas e, conseqüentemente, melhorar a sua produtividade. Segundo Valente et al. (1999), desde que os primeiros computadores começaram a ser comercializados, passaram a ser utilizados na área da educação. A construção e gestão dos horários escolares, segundo Even et al. (1976 apud SOUSA; MORETTI; PODESTÁ, 2008), consiste em um problema não polinomial completo e, portanto, requer muito tempo e esforço computacional. Não obstante, ainda assim é possível fazer uso da capacidade de processamento que os sistemas computacionais possuem para resolver este problema. A inteligência artificial tem sido uma alternativa para buscar soluções, exemplo do que ora é apresentado neste trabalho.

A inteligência artificial é descrita de formas distintas. Russell e Norvig (2013) citam definições propostas por John Haugeland, Richard Bellman, Raymond Kurzweil, Eugene Charniak, Drew McDermott e Patrick Winston, e dividem essas definições em quatro categorias gerais que caracterizam o conceito de inteligência artificial. São elas: Sistemas que pensam como seres humanos, Sistemas que pensam racionalmente, Sistemas que atuam como seres humanos e Sistemas que atuam racionalmente. Observa-se que a divisão foi feita com base em dois pensamentos, sendo um quanto a atuação e outro em relação ao pensamento dos sistemas.

Logo, uma maneira de resolver o problema da criação e gerenciamento dos horários escolares é mediante o uso de softwares baseados na inteligência artificial, que pode proporcionar uma maior efetividade no seu funcionamento e, portanto, fazer com que o uso dessa modalidade de software permita que o trabalho que

levaria uma semana dure apenas algumas horas ou até mesmo minutos sem exigir participação direta de muitas pessoas.

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA DE PESQUISA

O grande número de requisitos que envolvem o problema de geração de horários o torna complexo de solucionar. Este nível de complexidade é diretamente proporcional à quantidade de restrições que serão impostas ao horário e elas variam de acordo com as necessidades da instituição. Muitas variáveis influenciam na geração destes horários, incluindo quantidade de disciplinas, quantidade de professores, disponibilidade dos professores, carga horária de cada disciplina, quantidade de turmas, existência de laboratórios, quadras, etc. Assim, cada variável influencia na forma como o horário deve ser gerado.

Os sistemas computacionais são grandes aliados na resolução de problemas e facilitam bastante o cumprimento das tarefas diárias. Entretanto, uma quantidade elevada de restrições pode implicar em um nível maior de processamento computacional. Deve-se então, buscar soluções computacionais que utilizem algoritmos capazes de melhorar o funcionamento do sistema.

A implementação de técnicas de inteligência artificial em softwares pode utilizar diferentes tipos de abordagens, incluindo, como destacam Sousa, Moretti e Podestá (2008), heurísticas diretas, redução ao problema de coloração de um grafo, fluxo em redes, busca tabu, *Simulated Annealing* e algoritmos genéticos.

1.2 QUESTÃO CENTRAL DA PESQUISA

Como desenvolver uma solução computacional para o problema de confecção e gerenciamento de horários em um ambiente educacional?

1.3 HIPÓTESE CENTRAL

Sistemas computacionais em conjunto com técnicas de inteligência artificial são muito eficientes na resolução de problemas mais complexos, podendo ser utilizados para solucionar a geração de horários escolares.

1.4 OBJETIVO GERAL

Desenvolver um sistema computacional utilizando algoritmos de inteligência artificial que permita a geração automática de horários escolares respeitando as restrições impostas pelo usuário.

1.5 OBJETIVOS ESPECÍFICOS

- Modelar o sistema proposto.
- Especificar os requisitos funcionais e não funcionais do sistema, visando garantir a definição das restrições e a geração dos horários.
- Descrever a estrutura dos projetos de dados, de arquitetura, de procedimentos e de interface.
- Codificar o software utilizando a inteligência artificial aliada a algoritmos genéticos.
- Realizar testes do sistema computacional.

1.6 JUSTIFICATIVA

O uso de softwares para auxílio de tarefas executadas em instituições é de grande valia para a melhoria da produtividade de suas atividades. Aqueles baseados em inteligência artificial possibilitam uma gama de facilidades não encontradas em softwares desenvolvidos segundo os preceitos tradicionais da programação. Em termos de arquitetura de software, os sistemas inteligentes possuem agentes que são capazes de aprender com o ambiente e são compostos por regras que permitem o desenvolvimento de uma ferramenta computacional mais robusta.

Buscando uma solução para o problema de geração de horários escolares, este projeto de monografia procura estudar e desenvolver uma resolução viável para este problema utilizando abordagens e técnicas de inteligência artificial em sistemas computacionais, de forma a trazer a oportunidade de relacionar conhecimentos teóricos disponibilizados na literatura de sistemas inteligentes com o desenvolvimento proposto.

1.7 ESTRUTURA DOS CAPÍTULOS DO PROJETO DE MONOGRAFIA

Este projeto de monografia é composto por seis capítulos, sendo o primeiro referente à introdução, que por sua vez aborda e descreve o problema, formula uma questão de como resolvê-lo, oferece uma hipótese para resolução do mesmo, indica

os objetivos gerais e específicos, e justifica o seu desenvolvimento. O segundo capítulo aborda trabalhos relacionados e os resultados obtidos por seus autores. O terceiro capítulo de revisão de literatura faz uso de embasamentos teóricos extraídos de artigos, monografias, dissertações, teses e livros. No capítulo quatro é descrita a metodologia do trabalho. O quinto capítulo, por sua vez, contém a descrição do sistema proposto. Por fim, o capítulo seis apresenta as considerações finais relacionadas à monografia.

2 ESTADO DA ARTE

Neste capítulo serão descritos alguns trabalhos acadêmicos relacionados ao tema proposto. Inicialmente serão apresentadas teses, seguidas por dissertações, monografias e artigos.

No que diz respeito às teses, foi identificado um trabalho intitulado “AST Um modelo para automação de horários escolares” desenvolvido por Santos (2008), onde foi utilizada programação para solucionar o problema de geração de horários escolares com o uso de heurísticas com trocas locais através de um grafo híbrido, que proporcionou grandes vantagens para o algoritmo desenvolvido. Os resultados foram promissores e precisos, e tornou-se possível não ter a necessidade de ajustar os horários obtidos após a geração dos mesmos.

Em relação aos trabalhos de dissertação, dois trabalhos foram destacados. No primeiro, sob o título “Uma solução do problema de horário escolar via algoritmo genético paralelo”, Lobo (2005) abordou a implementação de uma heurística de otimização de horários escolares utilizando a linguagem Java e a arquitetura Corba, com princípios de programação paralela e a aplicou a um dos campus da Universidade Presidente Antônio Carlos (UNIPAC) em Minas Gerais. Os resultados obtidos por ele foram satisfatórios e mostraram-se eficientes, no entanto, não eliminou-se a possibilidade de um mesmo professor ser alocado para duas turmas simultaneamente.

A dissertação de mestrado de Hamawaki (2005) sob o título “Geração automática de grade horária usando algoritmos genéticos: O caso da faculdade de engenharia da UFU” apresentou características dos algoritmos genéticos na construção de horários voltados às instituições de ensino e demonstrou as dificuldades em implantar uma solução para este tipo de problema. Em seu trabalho, ela desenvolveu uma representação genética para resolver o problema, que trouxe resultados satisfatórios.

Tratando-se de monografias, a intitulada “Sistema automático de geração de horários para a UFLA utilizando algoritmos genéticos” de Ramos (2002), foi feita a adaptação de um software de geração de horários escolares previamente desenvolvido para que ele pudesse utilizar algoritmos genéticos e adaptá-los para uso na Universidade Federal de Lavras – UFLA. Contudo, apesar de ter apresentado um resultado satisfatório, adaptações devem ser feitas para que o sistema possa ser

implantado definitivamente na UFLA, incluindo adição de outras restrições não consideradas pelo software. Ramos observou uma dificuldade em implementar o algoritmo, devido ao fato de ter sido necessário alterar um software já existente ao invés de desenvolver um completamente desde o seu estágio inicial.

Em termos de artigos, foi encontrado um desenvolvido por Cordenonsi, Aramburu e Almaça (2003) com o título “Resolução do problema de quadro de horários através de um algoritmo de satisfação de restrições”, onde foi estudado o uso de um algoritmo de satisfação de restrições para resolver o problema de quadro de horários, porém com algumas limitações, como uma instituição com quatro períodos de aulas e sem empecilhos para deslocamento dos professores. Constatou-se uma dificuldade ao resolver problemas com uma quantidade maior de restrições, mas ainda assim concluiu-se que o modelo oferece bastante praticidade e economia de tempo.

3 REFERENCIAL TEÓRICO

Este capítulo contém informações referentes a bases teóricas obtidas através de pesquisas necessárias para compreensão de alguns assuntos que envolvem o tema desta monografia. Inicialmente serão apresentados conceitos referentes à inteligência artificial, assim como um breve histórico, os tipos de agentes e as propriedades do ambiente de atuação destes. Em seguida será abordado o problema de satisfação de restrições e a sua solução chamada de técnica de propagação de restrições. Como próximo item, será feita uma descrição geral acerca de algoritmos genéticos, através da apresentação de um breve histórico, conceituação, elementos e etapas que o compõem.

3.1 INTELIGÊNCIA ARTIFICIAL

Em relação às outras ciências existentes, a inteligência artificial é uma das mais novas, tendo pesquisas relacionadas iniciadas logo após a Segunda Guerra Mundial e o surgimento do seu nome em 1956. No entanto, mesmo sendo considerada recente, mais de meio século já foi dedicado ao estudo desta área e, ao longo destes anos, vários conceitos de diversos estudiosos surgiram acerca do assunto. Como visto na introdução, capítulo 1, Russel e Norvig (2013) utilizaram conceitos de diversos autores para definir a inteligência artificial. Entre eles, Poole et al. (1998 apud RUSSEL; NORVIG, 2013, p. 5) define que “A Inteligência Computacional é o estudo do projeto de agentes inteligentes.”

Segundo Silva (2005), agentes inteligentes são aqueles que, através do uso de sensores, analisam o ambiente em que estão inseridos e, por meio das informações obtidas, agem neste meio utilizando atuadores. Ele define, também, que um agente inteligente ideal é capaz de realizar decisões com base nas informações sensoriais captadas no ambiente em que se está inserido e de conhecimentos pré-programados nele de forma a interpretar novas informações adquiridas da melhor maneira possível. Descreve também, que os agentes inteligentes são divididos em quatro classes distintas segundo o nível de inteligência que eles possuem. Podem ser agentes estímulo-resposta, é o tipo mais simples de agente e possui um comportamento composto por regras previamente codificadas; agentes com memória, são agentes estímulo-resposta que possuem estados em relação ao ambiente em que está inserido e fazem tomadas de decisões que envolvem as regras codificadas e os estados do ambiente; agentes guiados por

objetivos, são aqueles que procuram atingir um estado específico; agentes guiados pela função de utilidade, são agentes que, através de um método, avaliam qual das possíveis ações é a melhor para que ele atinja o objetivo mais facilmente.

Esses agentes inteligentes podem atuar em ambientes com características distintas que influenciam diretamente na maneira como eles devem agir. Por esta razão, Silva (2005), Aguiar (2014) e Reis (2003) definem cinco propriedades que um ambiente de atuação de um agente pode possuir.

1. Acessibilidade: caso um agente possa obter totalmente o estado do ambiente em que está inserido, diz-se que este ambiente é acessível ao agente. Caso contrário, ele será inacessível ou parcialmente acessível. Há ainda a definição de que se as informações obtidas pelo agente forem suficientes para realizar uma tomada de decisão ótima, o ambiente será efetivamente acessível.
2. Determinístico: caso o estado seguinte dependa exclusivamente do estado corrente e das ações que o agente efetua nele. Não sendo determinístico, o ambiente será não determinístico.
3. Estático: caracteriza-se como um ambiente estático, quando ele não sofre alterações a medida que o agente processa qual será sua próxima ação. Se sofrer mudanças, passa a ser definido como ambiente dinâmico.
4. Episódico: dividindo-se as percepções e ações tomadas por um agente em episódios, considera-se o ambiente como episódico se o sucesso de cada ação depender apenas do episódio atual. Caso dependa de episódios anteriores, este ambiente será definido como não episódico.
5. Discreto: quando há um número limitado de percepções e ações. Se for ilimitado, é considerado um ambiente contínuo.

O problema de alocação de horários escolares possui um ambiente não determinístico, pois assim que o agente escolher um horário que o professor será alocado, não só o próximo estado será afetado, mas todos os outros posteriores a ele. Pela mesma razão, ele é considerado como não episódico. O ambiente é também contínuo e acessível, pois o agente é capaz de obter todas as informações do estado e tomar inúmeras decisões. Como este ambiente não sofre alterações entre processamento de informações e tomadas de decisões, é considerado um ambiente estático. Portanto, a complexidade deste tipo de problema é dada, principalmente, pela sua característica não determinística, não episódica e contínua.

3.2 PROBLEMA DE SATISFAÇÃO DE RESTRIÇÕES

Uma das formas de resolução do problema de geração de horários escolares é através do problema de satisfação de restrições (PSR) ou *Constraint Satisfaction Problems* (CSP). Russel e Norvig (2013) descrevem que o PSR é definido por um conjunto de variáveis, onde cada variável possui um domínio de pelo menos um valor possível, e por um conjunto de restrições, sendo cada restrição relacionada a um subconjunto das variáveis e determina quais combinações de valores que podem ser assumidas. Uma atribuição de valores é um estado e aquela que respeitar todas as restrições impostas é chamada de atribuição consistente ou válida. Após todas as variáveis serem atribuídas, dá-se o nome de atribuição completa. Caso esta satisfaça todas as restrições, atinge-se a solução do problema.

Há três abordagens heurísticas para seleção de qual valor ou variável será o(a) próximo(a) a ser escolhido(a), conforme é apresentado por Pires (2006). A heurística de variável mais restritiva (heurística de grau) escolhe a variável com o maior número de restrições e atribui o primeiro valor que ela possuir em seu domínio. A abordagem de variável mais restringida (valores restantes mínimos – VRM, variável mais restrita ou primeira falha) consiste em escolher a variável que possui o menor domínio, ou seja, aquela que pode assumir uma menor quantidade de valores. Em seguida, atribui-se o primeiro valor contido no domínio da variável escolhida. A técnica de valor menos restritivo faz a tomada de decisão de acordo com a ordem em que as variáveis estão dispostas, porém os valores escolhidos ocorrem conforme a análise de qual valor dentre aqueles presentes no domínio da variável afetará menos o domínio das outras variáveis, ou seja, avalia qual valor que quando escolhido restringe menos as outras variáveis, dando a elas um maior grau de liberdade.

Para melhoria de uso das restrições e otimização dos algoritmos de problema de satisfação de restrições, utiliza-se a verificação prévia (forward checking). Sempre que um valor for escolhido para uma variável, o algoritmo verificará se este valor selecionado está contido no domínio de outras variáveis que possuam restrições com aquela recentemente atribuída e removerá esta informação do domínio de cada uma delas. No entanto, a verificação prévia apenas detecta inconsistências imediatas (próximas) à variável recém-atribuída. Para verificações mais a frente, existe a técnica de propagação de restrições, onde além de remover

os valores presentes no domínio de variáveis que possuam restrições com a variável recém-atribuída, procura-se também inconsistências de valores entre essas outras variáveis, baseando-se nas restrições que elas possuem entre si.

3.3 ALGORITMOS GENÉTICOS

Lobo (2005) descreve um breve histórico e conceituação acerca de algoritmos genéticos. Ele afirma que o estudo de soluções computacionais baseadas na genética da biologia teve seu início na década de 1950, dando início aos sistemas computacionais evolucionários. O avanço no estudo desta área foi significativo após os desenvolvimentos de sistemas computacionais baseados em evolução genética do cientista americano John Holland e a partir da publicação de seu livro *Adaptation in Natural and Artificial Systems* no ano de 1975. Os algoritmos genéticos baseiam-se, segundo Ribeiro Filho (2000 apud RAMOS, 2002), nas leis de seleção e evolução natural de Charles Darwin apresentadas em 1858, em conjunto com o conhecimento sobre os princípios da genética desenvolvidos por Gregor Mendel em 1865.

Um algoritmo genético é conceituado por Lobo (2005) como qualquer modelo computacional em forma de população inspirado na genética da biologia e que faz uso de operadores de cruzamento e mutação de maneira a gerar novas amostras (indivíduos) para aperfeiçoamento da população. Portanto, assim como na biologia, estas populações sofrem alterações ao longo das gerações que surgem posteriormente. Por conta das características que possui, os algoritmos genéticos são, afirma Ramos (2002), muito utilizados quando um problema possui um espaço de busca extenso, pois estes algoritmos permitem descobrir sub-regiões de alta qualidade de uma maneira rápida.

Mitchell (1996 apud RAMOS, 2002), Lucas (2002) e Lobo (2005) destacam alguns termos importantes da genética que também são utilizados nos algoritmos genéticos. São eles, cromossomo (indivíduo), representa uma possível solução ao problema; gene, unidade básica que representa características de um cromossomo. Um ou mais genes estão contidos em um cromossomo; alelo, valor de um gene; locus, local ou posição de um gene em um cromossomo; genótipo, codificação da solução; fenótipo, decodificação ou significado do genótipo; população, conjunto de cromossomos ou indivíduos; geração, cada iteração que gera uma nova população pelo algoritmo genético; operações genéticas, operações feitas sobre cada

cromossomo; espaço de busca, região de soluções viáveis para otimizar o problema; função objetivo (aptidão ou fitness), avalia quão bom um indivíduo é em relação a uma solução ótima através do uso de uma função específica para o problema; aptidão bruta, é o valor definido pela função fitness para um indivíduo; aptidão máxima, o indivíduo com aptidão máxima é o melhor da população; crossover (cruzamento), é um operador genético que gera um indivíduo para a nova população a partir de dois outros da antiga geração.

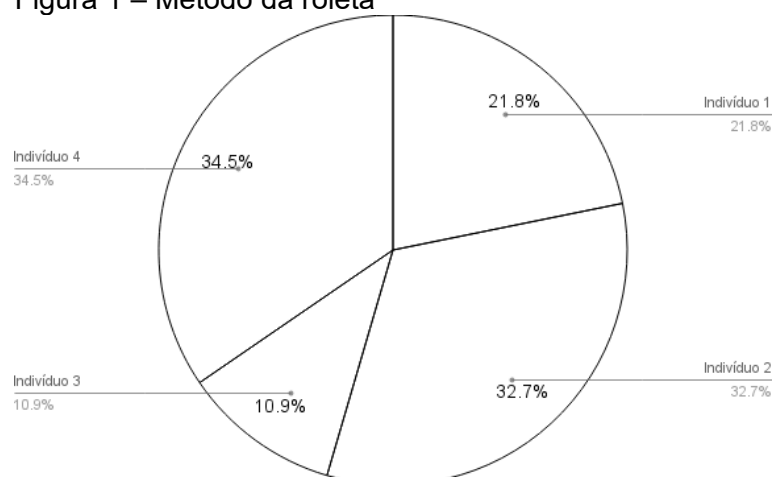
Segundo Vélez (2015), o fluxo de um algoritmo genético consiste em criar uma população inicial a ser evoluída, avaliá-la através da função fitness, selecionar indivíduos, efetuar o cruzamento entre eles, gerar possíveis mutações nos cromossomos, substituir indivíduos da antiga população por outros da nova geração e verificar se o objetivo foi alcançado (critério de parada). Atingindo este último, encerra-se a execução do algoritmo, caso contrário, retorna-se à etapa de avaliação e repete-se o fluxo a partir dela.

3.3.1 Seleção

Nesta etapa, alguns indivíduos da população são selecionados para que posteriormente ocorra o cruzamento entre eles e seja gerada uma nova população, chamada de temporária. Silva (2001) descreve dois métodos utilizados para o processo de seleção, o por roleta e o por torneio.

No método de seleção por roleta, é atribuído a cada indivíduo uma porção da roleta de acordo com a sua aptidão. Portanto, indivíduos com uma maior aptidão possuem uma maior fatia da roleta e, conseqüentemente, possuirão uma maior chance de serem selecionados pelo sorteio. Este processo é repetido várias vezes até preencher a quantidade de indivíduos necessários.

Figura 1 – Método da roleta



Fonte: Autoria própria, 2016.

Em relação ao método de seleção por torneio, escolhe-se dois ou mais indivíduos pertencentes a uma população de forma aleatória. Há duas formas de efetuar a seleção, uma onde define-se uma probabilidade que é posteriormente utilizada para selecionar um destes indivíduos e outra que analisa o cromossomo com melhor aptidão, que será o escolhido.

Há ainda dois outros métodos enfatizados por Ferreira, Braga e Ludermir (2003 apud LOBO, 2005). O método da amostragem universal estocástica ocorre de forma similar ao da roleta, porém realiza apenas um sorteio, selecionando todos os indivíduos necessários. Aqueles que possuírem uma maior porção, podem ser escolhidos mais de uma vez. Caso isto ocorra, cópias deles são criadas para cada vez que ele for selecionado. E o método de seleção elitista substitui os piores indivíduos da nova geração pelos melhores da antiga, de forma a garantir que os bons cromossomos não sejam eliminados no processo evolutivo. Ele é geralmente utilizado em conjunto com outras formas de seleção.

3.3.2 Crossover

Após a seleção dos indivíduos, forma-se pares entre eles para que ocorra a reprodução, ou seja, novos cromossomos são gerados para a nova geração. Os indivíduos filhos possuirão partes dos genes dos pais de forma mista, a fim de permitir a geração de cromossomos com novas características e com um maior potencial para uma solução ótima. Lobo (2005) e Silva (2001) descrevem três métodos de reprodução distintos, que determinam como serão feitas as misturas dos genes.

Uma forma de reprodução é o cruzamento em um ponto. Neste método, um ponto de corte dos cromossomos pais é escolhido aleatoriamente e dois filhos são gerados. Ao primeiro serão atribuídos dois pedaços, sendo cada um pertencente a um de seus genitores. As duas outras partes dos pais serão utilizadas para gerar o segundo filho. Antes de ocorrer o crossover, aplica-se uma taxa de probabilidade de cruzamento, que determinará se o ponto de corte será ou não aplicado. Caso não seja, cada filho será uma cópia idêntica de um dos pais. A figura 2 demonstra um exemplo deste método de crossover.

Figura 2 – Cruzamento em um ponto

Pai₁	1011	0111
Pai₂	1101	1111

Filho₁	1011	1111
Filho₂	1101	0111

Fonte: Adaptado de Silva, 2001.

Similar ao cruzamento em um ponto, há o método de cruzamento em dois pontos. O que o difere à outra maneira de reprodução é o fato de ao invés de selecionar apenas um ponto de corte, escolher-se dois distintos, também aleatórios. Um filho terá o primeiro e o último pedaços de um pai, e a parte central do outro genitor. O segundo novo indivíduo gerado possuirá as partes restantes. Na figura 3 é possível observar um exemplo de cruzamento em dois pontos.

Figura 3 – Cruzamento em dois pontos

Pai₁	011	11001	110
Pai₂	101	11000	100

Filho₁	011	11000	110
Filho₂	101	11001	100

Fonte: Adaptado de Silva, 2001.

No método de cruzamento uniforme, uma máscara com n bits é gerada aleatoriamente, sendo n a quantidade de genes dos pais. Posteriormente, os genes dos pais são atribuídos ao primeiro filho de acordo com a máscara gerada. Se o bit dela for 1, copia-se o respectivo gene do primeiro genitor para o filho, caso seja 0,

faz-se a cópia do gene do segundo pai. Todos os bits da máscara são percorridos e utilizados para efetuar a escolha de cada gene. Para o segundo filho, inverte-se os pais e realiza-se a seleção das características do indivíduo da mesma forma que o anterior. A figura 4 apresenta um exemplo de como é feito o cruzamento uniforme.

Figura 4 – Cruzamento uniforme

Máscara	11100100001100
Pai ₁	11011100101100
Filho ₁	11001110101101
Pai ₂	00101110101101

Fonte: Adaptado de Silva, 2001.

3.3.3 Mutação

Após a ocorrência do cruzamento, os novos indivíduos gerados podem sofrer mutações em alguns de seus genes. Silva (2001) destaca a importância de um operador de mutação ser aplicado a uma nova população, enfatizando que este processo permite diversificar a nova geração de indivíduos e introduz novos elementos na população. Entretanto, deve-se utilizar o operador de mutação com cautela, pois ao alterar os genes de um indivíduo, parte das características dos seus genitores são perdidas. Logo, deseja-se que a maior parte dos genes do cromossomo seja preservada.

Como forma de garantir de que apenas uma pequena fração dos genes serão alterados pelo operador de mutação, utiliza-se uma taxa de probabilidade de mutação para cada gene do indivíduo. Desta forma, cada unidade característica do cromossomo sofre uma pequena chance de ser alterada. Geralmente, esta taxa de mutação é aplicada, segundo Silva (2001), com uma probabilidade de 0,1% a 5% de chance de ocorrência.

Em um cromossomo onde cada bit corresponde a um gene, a operação de mutação consiste em substituir aquele bit pelo seu complemento. Portanto, se um gene cujo valor é 1 sofrer mutação, este dado passará a ser 0. Caso a informação inicial seja 0, será alterada para 1.

Figura 5 – Mutação

Antes da mutação	Filho ₁	11000011
	Filho ₂	00010111
Depois da mutação	Filho ₁	01100011
	Filho ₂	00010111

Fonte: Adaptado de Silva, 2001.

3.3.4 Substituição

Após o processo de ocorrência de mutação, há a substituição de indivíduos da antiga população por outros da nova geração. Vélez (2015) afirma que há duas classes de algoritmos genéticos que são classificadas de acordo com o método de substituição empregado. Na classe de algoritmos genéticos com substituição geracional, a população de pais é totalmente substituída pela de filhos. Na segunda classe, a de algoritmos genéticos com brecha geracional, alguns pais sobrevivem e são mantidos na nova população.

Para os algoritmos genéticos com brecha geracional, existem algumas formas distintas de efetuar a substituição dos indivíduos antigos pelos novos. A primeira maneira foi definida por De Jong e Sarma (1992 apud VÉLEZ, 2015) e consiste em substituir os indivíduos com menor aptidão pelos filhos gerados. Outra forma é fazer a substituição dos indivíduos pelos filhos de modo aleatório.

O método de torneio a morte é feito utilizando as características da substituição dos menos aptos e de forma aleatória. Portanto, consiste em selecionar um grupo de indivíduos aleatoriamente e substituir aquele com menor aptidão pelo novo cromossomo.

Há, também, uma forma de estratégia de substituição onde os indivíduos mais velhos da população são os escolhidos para serem substituídos. Para evitar a substituição de indivíduos velhos que possuam uma boa aptidão, pode-se combinar este método com o elitismo ou utilizar outra forma de substituição chamada seleção conservativa. Esta última foi definida por Smith (2007 apud VÉLEZ, 2015) e é um operador misto que utiliza a estratégia do indivíduo mais velho aliada ao torneio de seleção binária. Neste método, um indivíduo da população é escolhido aleatoriamente e é comparado ao mais velho. O que for menos apto é substituído

pelo novo cromossomo. O elitismo foi proposto por De Jong (1975 apud SILVA, 2001) e consiste em preservar o melhor indivíduo de uma geração para a outra. Assim, o melhor cromossomo da atual população será mantido na nova geração.

A forma de substituição dos progenitores considera ou não a substituição dos pais pelos filhos. Através de alguma estratégia de análise, incluindo avaliação do indivíduo que possui melhor aptidão, a substituição do pai pelo filho é ou não realizada. Portanto, o indivíduo com melhores características de uma solução ótima ou aquele que pode gerar indivíduos mais aptos são preservados.

3.3.5 Critério de Parada

Após a substituição dos indivíduos da população anterior pelos novos cromossomos gerados, faz-se uma avaliação para identificar se algum critério não já foi alcançado. Lobo (2005) afirma que as seguintes condições podem ser alcançadas: atingir o número máximo de evoluções estipulado, a aptidão do melhor indivíduo não melhorar mais nas novas gerações, os indivíduos de uma população atingirem uma aptidão similar entre eles, atingir o melhor valor possível para a função fitness ou haver diminuição da diversidade dos indivíduos de uma população.

Tendo atingido alguma condição do critério de parada, significa que o algoritmo deve encerrar as evoluções e armazenar aquele indivíduo com a melhor aptidão dentre aqueles de uma mesma população. Observa-se que o critério de parada permite que mesmo não obtendo um cromossomo solução para o problema, é possível que o algoritmo genético seja interrompido. Este tipo de implementação é importante para casos em que os novos indivíduos de uma população não consigam obter melhores resultados do que as gerações anteriores. Desta forma, evita-se que o algoritmo seja executado mais vezes sem a possibilidade de alcançar uma solução e garante-se que um bom cromossomo para o problema seja obtido como resultado final.

4 METODOLOGIA

Neste tópico são apresentadas as formas de pesquisa utilizadas para elaboração deste projeto de monografia, além dos instrumentos de pesquisa necessários para realização da mesma.

Para Fonseca (2002 apud GERHARDT; SILVEIRA, 2009), a metodologia é a análise do desenvolvimento e dos instrumentos utilizados para realizar uma pesquisa científica ou estudo.

4.1 TIPOS DE PESQUISA QUANTO AOS OBJETIVOS

Em relação aos fins, foram utilizados dois tipos distintos de pesquisa para o desenvolvimento desse projeto de monografia. Na fase inicial utilizou-se a pesquisa exploratória, que, segundo Selltitz et al. (1967 apud GIL, 2002), consiste em uma maneira de aproximar o pesquisador do problema em questão. Os meios que podem ser utilizados para a formulação dessa pesquisa são levantamento bibliográfico, entrevistas com pessoas que possuam experiência prática com o fenômeno em estudo e análise documental.

Para a segunda etapa, foi realizada uma pesquisa descritiva. Esta tem como objetivo, conforme afirmação de Gil (2002), descrever as características do objeto em estudo ou fornecer uma relação entre variáveis. Esta pesquisa pode ser desenvolvida através do uso de técnicas padronizadas de coleta de dados, incluindo o questionário e a observação sistemática, ou seja, aquela onde há planejamento e faz-se observações em ambientes controlados.

No que se trata da pesquisa exploratória, foram feitas consultas e análises de livros, artigos, monografias, dissertações e teses, de forma a obter um maior conhecimento acerca do tema e permitir uma maior compreensão para desenvolvimento e solução do problema de pesquisa. O foco da pesquisa exploratória foi obter um maior entendimento sobre os algoritmos genéticos e suas características. Os resultados da pesquisa exploratória foram expostos nos capítulos 2 e 3 desta monografia, que abordam respectivamente, o estado da arte e o referencial teórico.

Em relação à pesquisa descritiva, foram feitas descrições sobre o desenvolvimento do software de geração de horários escolares e informadas as características referentes a ele nos capítulos 1 e 5, referentes, respectivamente, à introdução e ao desenvolvimento da pesquisa abordada neste trabalho.

4.2 TIPOS DE PESQUISA QUANTO À FORMA DE ANÁLISE

Em relação ao formato de pesquisa utilizado quanto à abordagem, Silva e Menezes (2005) afirmam existir dois tipos distintos de elaborar a pesquisa. O primeiro deles é a maneira quantitativa, onde as informações e opiniões obtidas são feitas através de números. Estes são analisados e classificados conforme recursos e técnicas estatísticas. A segunda forma de classificação de uma pesquisa quanto à abordagem é a qualitativa. Neste tipo, a coleta de dados é feita no ambiente natural e são feitas interpretações acerca do fenômeno em estudo em relação ao processo e ao seu significado.

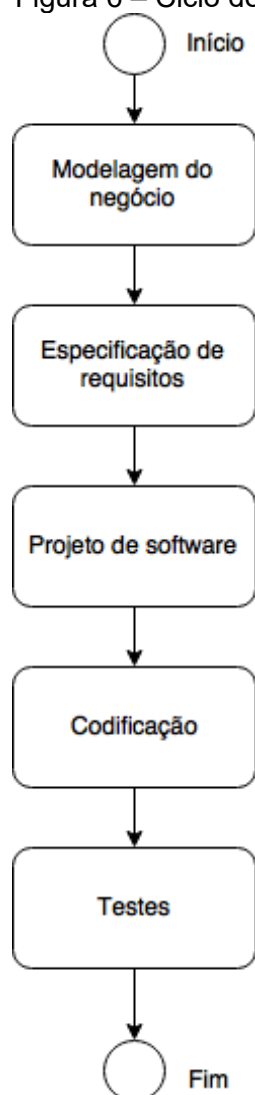
Para o desenvolvimento do software de geração de horários escolares, foi utilizada uma pesquisa do tipo qualitativa, por se tratar de uma análise do objeto de estudo e do processo de desenvolvimento do mesmo. Para a fase de testes do produto, uma pesquisa quantitativa foi feita, onde foram analisados o tamanho da população utilizada, o número de restrições definido e a quantidade necessária de evoluções para obter a solução do problema.

4.3 TIPOS DE PESQUISA QUANTO AOS MEIOS

Quanto aos procedimentos técnicos, foi realizado o desenvolvimento de um produto, neste caso um software, através do uso de métodos da engenharia de software baseados no IBM Rational Unified Process (RUP), seguindo as fases de modelagem do negócio, especificação de requisitos, projeto de software, codificação e testes, conforme figura 6. Portanto, a pesquisa quanto aos meios utilizada caracteriza-se como um estudo de caso único, diferenciando-se do múltiplo por se tratar de apenas um problema específico.

Segundo Pressman (1995), a engenharia de software faz uma abordagem conjunta de três elementos (métodos, ferramentas e procedimentos) de forma a possibilitar um melhor desenvolvimento do software e uma alta qualidade do produto final. Os métodos detalham o processo de construção do produto em questão, as ferramentas apoiam os métodos de forma automatizada ou semi-automatizada, e os procedimentos fazem a união entre os métodos e as ferramentas.

Figura 6 – Ciclo de desenvolvimento do software



Fonte: Autoria própria, 2016.

4.4 INSTRUMENTOS DE PESQUISA

No que se trata dos instrumentos utilizados nas etapas do ciclo de desenvolvimento do software, fez-se uso de diagramas do Unified Modeling Language (UML) e de entidade-relacionamento. Para a fase de modelagem do negócio, construiu-se o diagrama de caso de uso da UML, em conjunto com as suas descrições demonstrando as precondições, fluxo básico, fluxo alternativo e pós-condições.

Com base no diagrama de caso de uso, realizou-se a fase de elicitação dos requisitos funcionais do processo de engenharia de requisitos. Em relação à captura dos requisitos não-funcionais, um estudo geral do sistema e da norma NBR ISO/IEC 9126-1, que aborda o modelo de qualidade de produto da engenharia de software, foi feito de forma a analisar o que deveria ser oferecido aos usuários além de suas

funcionalidades. A segunda parte do processo de engenharia de requisitos se trata da análise e negociação de requisitos. Ela foi feita através de reuniões com o orientador da monografia, onde observou-se as exigências que a ferramenta computacional deveria ser capaz de cumprir. A próxima etapa foi a de documentação dos requisitos, tomando-se como base a especificação proposta pelo IEEE STD 830-1998. Ela foi elaborada através do uso de quadros informando a prioridade, o tipo e as dependências de cada requisito. A última fase consistiu em validar os requisitos elaborados no documento.

Na etapa de projeto de software do ciclo de desenvolvimento do software, construiu-se os diagramas de classes (compostos pelas entidades que o sistema deve possuir, bem como os seus atributos e métodos) e de atividades (onde foi informado o ciclo de funcionamento de cada caso de uso de gerenciamento) para auxiliar na codificação do sistema. Como forma de planejamento da composição do banco de dados, elaborou-se o diagrama entidade-relacionamento, descrevendo-se de forma visual a relação entre as entidades (tabelas), seus atributos e os tipos de dados. Esta fase de projeto de software foi dividida em projeto arquitetural, procedimental, de dados e de interface.

Em relação ao projeto arquitetural utilizou-se os estilos arquiteturais referentes às arquiteturas: orientada a objetos (chamada e retorno), baseada em agentes inteligentes (independente do processo) e repositório (banco de dados). Em relação aos agentes inteligentes utilizados, considerou-se duas possibilidades distintas de implementação de algoritmos que poderiam resolver o problema abordado, algoritmo de satisfação de restrições e algoritmos genéticos. No entanto, percebeu-se através de análises de alguns trabalhos relacionados como o de Lobo (2005), Hamawaki (2005), Ramos (2002) e, Cordenonsi, Aramburu e Almaça (2003), que os resultados obtidos ao utilizar algoritmos genéticos foram mais satisfatórios em relação ao outro algoritmo. Logo, escolheu-se desenvolver agentes inteligentes baseados na premissa de algoritmos genéticos.

No que se trata do projeto procedimental, descreveu-se, utilizando linguagem natural, como é o funcionamento geral do algoritmo desenvolvido para solucionar o problema de geração de horários escolares. Porém, um destaque foi dado para como são gerados os novos indivíduos e de que forma eles são avaliados pelo algoritmo para determinar seu grau de aptidão em relação à solução.

O projeto de dados, por sua vez, foi apresentado através do auxílio do diagrama de classes, onde demonstra-se os tipos de dados utilizados. Tratando-se de estrutura de dados, a principal utilizada foi o Array List, que consiste em uma lista dinâmica de dados altamente manipulável, ou seja, permite inserir um novo elemento de maneira simples, além de possibilitar a recuperação ou remoção de dados em qualquer posição da lista.

Na fase do projeto de interface, visou-se o desenvolvimento de uma interface de acordo com a usabilidade, também definida entre os requisitos não-funcionais, segundo a norma NBR ISO/IEC 9126-1. Segundo a Associação Brasileira de Normas Técnicas – ABNT (2003), esta característica significa a capacidade em que um software possui de ser compreendido pelos usuários, de forma que estes aprendam a operá-lo sem maiores dificuldades.

Para o desenvolvimento do software de geração de horários escolares tornou-se necessário o uso, como equipamento, de um computador para codificar o produto e como materiais, o ambiente de desenvolvimento integrado (IDE) NetBeans em sua versão 8.1 e, o MySQL Server 5.7 e o MySQL Workbench versão 6.3.6 para, respectivamente, criar e modelar o banco de dados utilizando a linguagem de pesquisa Structured Query Language (SQL).

Em termos de codificação, foi escolhida a linguagem de programação Java para desenvolvimento do software, pois se trata de uma linguagem que permite portabilidade e compatibilidade entre vários sistemas operacionais distintos, incluindo Mac OS, Windows e Linux, desde que a máquina virtual Java (JVM) esteja instalada no sistema desejado, além de tornar possível a manipulação de objetos, pois esta trabalha essencialmente com orientação a objetos.

Para definir a quantia mais adequada do tamanho da população e das taxas de crossover e de mutação, realizou-se diversos testes de geração de horários envolvendo diferentes valores para cada campo, de forma a obter uma execução mais eficiente do algoritmo genético, ou seja, atingir um melhor desempenho do software. A análise foi feita utilizando-se a quantidade de gerações e o tempo de execução total para encontrar a solução do problema, que foram obtidos através das experimentações descritas.

5 DESENVOLVIMENTO

Neste capítulo são descritas as etapas envolvidas no desenvolvimento da ferramenta computacional para geração de horários escolares. Conforme apresentado na figura 6, a construção do projeto seguiu os passos do ciclo de desenvolvimento do software baseado no IBM RUP.

5.1 MODELAGEM DO NEGÓCIO

O diagrama de caso de uso elaborado na fase de modelagem do negócio, figuras 7 e 8, possui um ator representando o funcionário de uma instituição educacional e 34 casos de uso relacionados às atividades que ele pode realizar no software. Este diagrama tem como objetivo apresentar de forma visual e simplificada quais são as atividades do sistema que o usuário possui uma interação direta, facilitando, então, a compreensão das funcionalidades da ferramenta.

Os casos de uso referentes ao gerenciamento de alguma atividade (Gerenciar professores, Gerenciar disciplinas, Gerenciar turmas, Gerenciar restrições, Gerenciar aulas conjuntas e Gerenciar horários) foram definidos como sendo menus do sistema. Enquanto que “Definir restrições” e “Listar restrições” foram considerados como submenus de “Gerenciar restrições”. Os demais casos de uso são atividades executadas a partir de algum menu/submenu caracterizados pelos relacionamentos *extend* representados no diagrama.

De forma a obter mais detalhes relacionados aos casos de uso e, assim, facilitar o entendimento dos procedimentos realizados em cada atividade, descreveu-se, através de quadros, os casos de uso projetados para o sistema, fornecendo informações como precondições, fluxo básico, fluxos alternativos e pós-condições que dizem respeito a cada caso de uso correspondente. Tais descrições foram dispostas no apêndice A.

A figura 7 a seguir descreve os casos de uso relacionados aos cadastros e manipulação de informações gerais sobre os professores, disciplinas e turmas, incluindo edição, listagem e remoção de cada um.

Figura 7 – Diagrama de caso de uso referente aos cadastros de dados



Fonte: Autoria própria, 2016.

Os casos de uso relacionados às funcionalidades gerais do sistema como gerenciamento de aulas conjuntas entre turmas, gerenciamento de restrições e gerenciamento de horários foram expostos na figura 8.

Figura 8 – Diagrama de caso de uso referente às funcionalidades



Fonte: Autoria própria, 2016.

5.2 ESPECIFICAÇÃO DE REQUISITOS

Os requisitos funcionais obtidos a partir das fases de elicitação, análise e negociação de requisitos dizem respeito aos casos de uso que, através da interação do usuário com o sistema, geram novas atividades. São eles Cadastrar professor, Editar professor, Listar professores, Remover professor, Cadastrar disciplina, Editar disciplina, Listar disciplinas, Remover disciplina, Cadastrar turma, Editar turma,

Listar turmas, Remover turma, Adicionar disciplina à turma, Remover disciplina da turma, Definir indisponibilidades do professor, Habilitar/desabilitar restrições de uma turma, Exibir indisponibilidades de um professor, Exibir restrições de uma turma, Adicionar aula conjunta, Editar aula conjunta, Listar aulas conjuntas, Remover aulas conjuntas, Gerar horário por curso, Gerar horário por turma, Exibir horário e Remover horário.

No que se trata dos requisitos não-funcionais capturados pela análise da norma NBR ISO/IEC 9126-1 e das funcionalidades gerais do sistema, obteve-se os requisitos de usabilidade, visando a apreensibilidade e operacionalidade, ou seja, permitir que o usuário consiga aprender a operar e controlar o sistema; confiabilidade, garantir que os dados sejam preservados corretamente, além de proporcionar a integridade e validade das informações inseridas pelo ator funcionário; portabilidade, refere-se à capacidade que o software possui em ser executado em diferentes sistemas operacionais; funcionalidade, atender às necessidades do usuário com acurácia dos resultados, ou seja, possibilitar que a solução encontrada pelo algoritmo seja útil e adequada.

A documentação elaborada na terceira fase do processo de engenharia de requisitos pode ser conferida no apêndice B. Uma informação disponibilizada sobre os elementos que a compõe é em relação à prioridade que cada requisito possui em relação ao seu desenvolvimento e funcionamento geral do sistema. Aqueles considerados essenciais são imprescindíveis para a operação correta do software proposto. Os demais foram descritos como não-essenciais.

A etapa de validação dos requisitos funcionais e do requisito não-funcional relacionado à funcionalidade do sistema foi feita através de testes de funcionamento das atividades. Quanto aos requisitos não-funcionais, no que se trata da usabilidade, apenas fez-se a avaliação se a experiência de usuário proporcionava uma forma fácil de operação e compreensão do sistema. Para promover uma confiabilidade do software, permitiu-se apenas a inserção de dados válidos. A escolha de informações já cadastradas foi fornecida através de menus de seleção ao invés de deixar o usuário livre para realizar a busca através de digitação de texto. Como o software foi desenvolvido através da linguagem de programação Java, o quesito de portabilidade pôde ser cumprido sem alterações no código. Para validação deste requisito, testes foram realizados nos sistemas operacionais Windows e Linux.

5.3 PROJETO DE SOFTWARE

Nesta fase fez-se a elaboração dos projetos utilizados como auxílio para desenvolvimento do sistema. As quatro etapas realizadas foram projeto arquitetural, projeto procedimental, projeto de dados e projeto de interface.

5.3.1 Projeto Arquitetural

Para o desenvolvimento do sistema utilizou-se a composição de sete objetos, sendo quatro referentes à estrutura do software (professor, disciplina, turma e aula) e os outros três relacionados ao algoritmo genético (indivíduo, população e algoritmo).

Para armazenamento e recuperação dos dados, utilizou-se a abordagem de repositórios onde o banco de dados utilizado é do tipo relacional e é composto por cinco tabelas. A figura 9 demonstra o Diagrama Entidade Relacionamento (DER) elaborado através da ferramenta Workbench do Sistema de Gerenciamento de Banco de Dados (SGBD) MySQL. Nele foram representados os atributos de cada tabela, bem como as relações que elas possuem entre si e a característica dos dados.

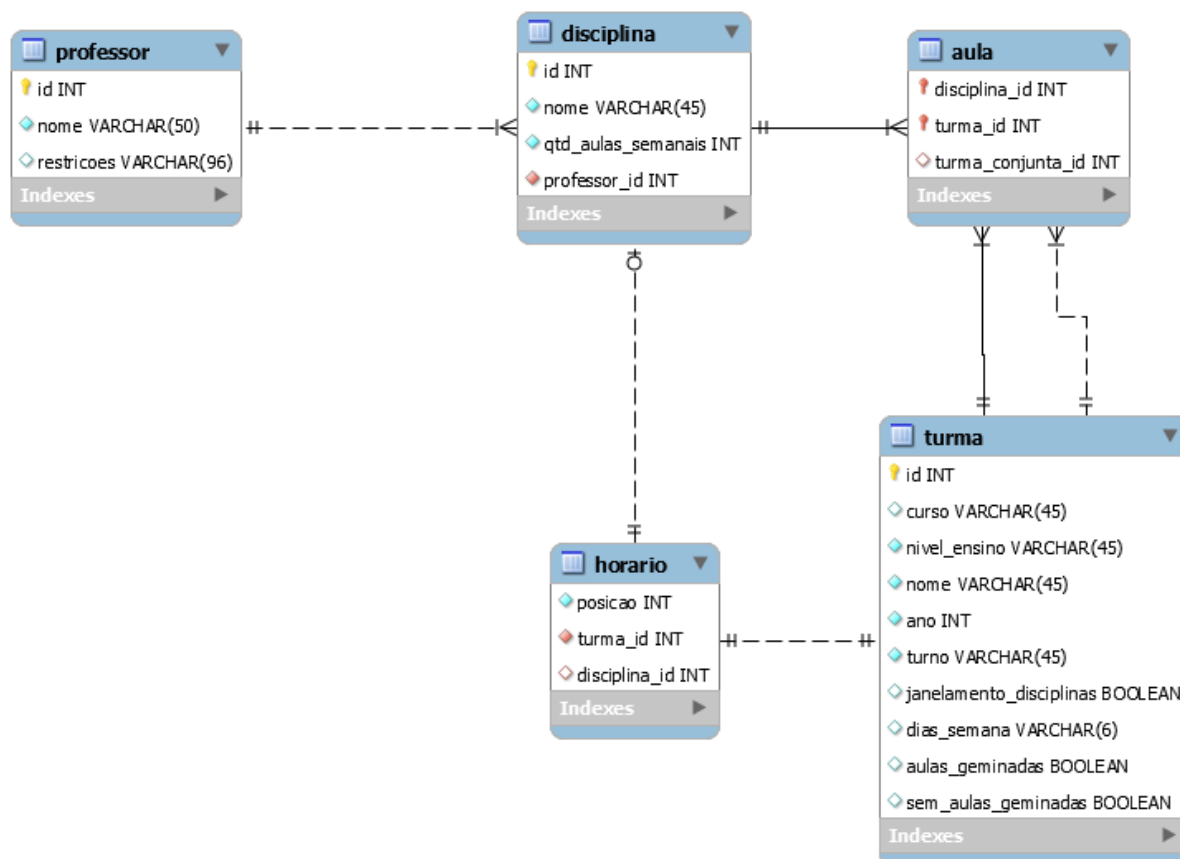
As três entidades principais, professor, disciplina e turma, possuem um atributo identificador único e não nulo definido como chave primária. Assim, cada elemento inserido na tabela possui uma característica privada e distinta dos demais itens. Uma referência ao professor foi criada na entidade disciplina em forma de chave estrangeira não nula, pois cada componente curricular é obrigatoriamente lecionado por um docente. Disciplinas idênticas, porém com professores distintos, devem ser cadastradas separadamente, permanecendo-se única no banco de dados.

A criação da tabela aula tornou-se necessária devido ao relacionamento de muitos para muitos existente entre as entidades disciplina e turma. Um atributo relacionado ao armazenamento da identificação da turma conjunta foi criado, podendo ser nulo caso a aula seja exclusiva.

De forma a possibilitar uma melhor consulta dos horários cadastrados no banco de dados, criou-se uma entidade específica para tal fim. Caso ela fosse considerada como um atributo da tabela turma, não seria possível pesquisar por uma posição específica do horário de uma determinada turma utilizando a linguagem

SQL. Se a disciplina for nula, significa que a aula é vaga para aquela posição do horário.

Figura 9 – Diagrama entidade relacionamento



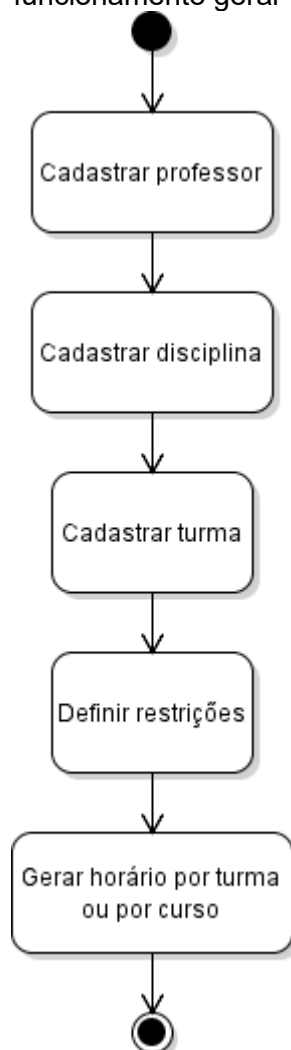
Fonte: Autoria própria, 2016.

5.3.2 Projeto Procedimental

Fez-se uso de diagramas de atividades para exemplificar o fluxo básico do funcionamento geral do sistema e a interação entre o usuário e o software, de maneira a possibilitar uma compreensão geral das tarefas do sistema e suas possíveis sequências de atividades.

O fluxo simples apresentado na figura 10 diz respeito à sequência de tarefas básicas que o usuário deve realizar para obter a geração de pelo menos um horário. Visto que para cadastrar uma disciplina é necessário informar um professor, torna-se necessário executar previamente a tarefa de cadastro de professores. Apesar de o cadastro de turmas dispensar a adição de componentes curriculares, para que o horário de uma turma seja gerado, deve-se possuir uma ou mais disciplinas armazenadas no banco de dados. Em seguida, caso haja restrições, é necessário que elas sejam cadastradas antes de gerar o horário.

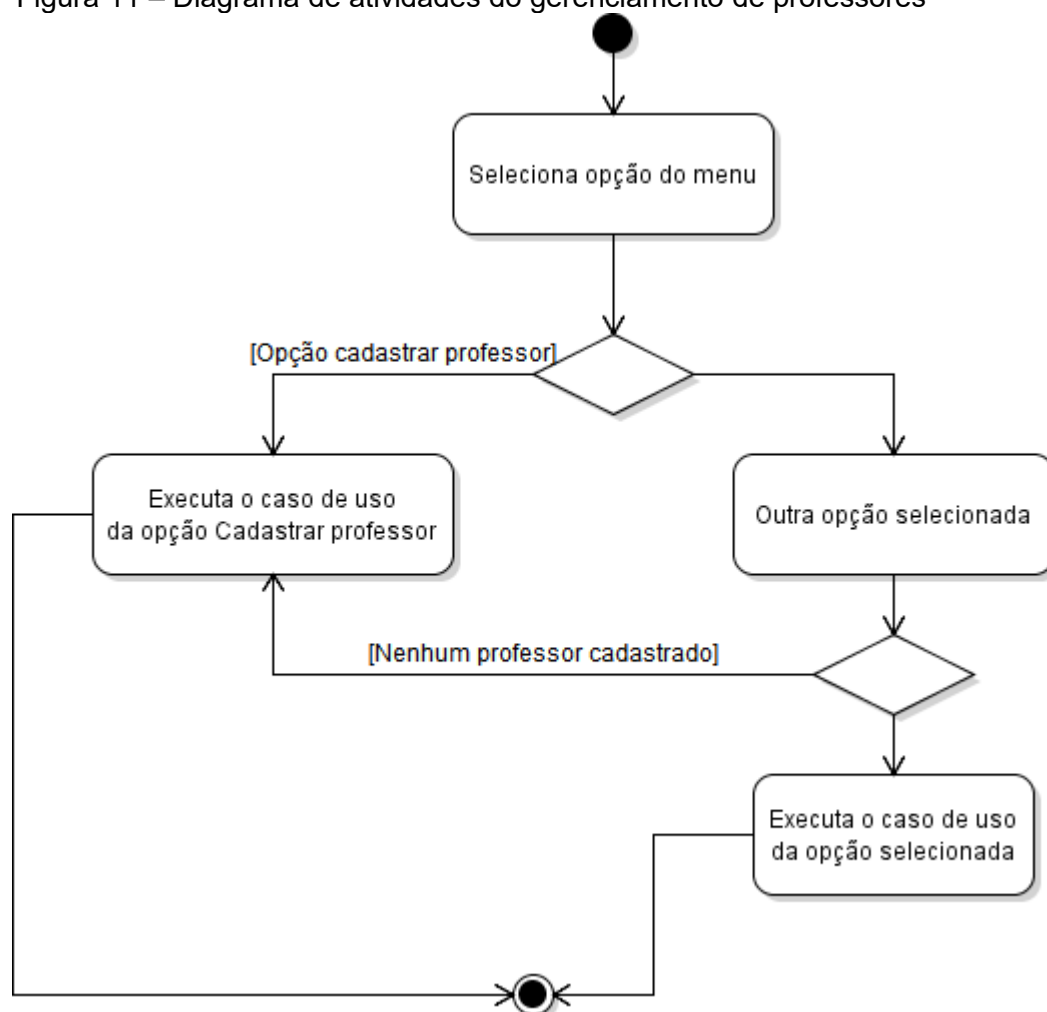
Figura 10 – Diagrama de atividades do fluxo básico do funcionamento geral



Fonte: Autoria própria, 2016.

De forma a descrever as possibilidades de interação entre o usuário e todas as tarefas disponíveis na ferramenta computacional, elas foram divididas em diagramas de atividades, figuras 11 a 16, onde cada um aborda o fluxo geral de um caso de uso de gerenciamento (Gerenciar professores, Gerenciar disciplinas, Gerenciar turmas, Gerenciar restrições, Gerenciar aulas conjuntas e Gerenciar horários).

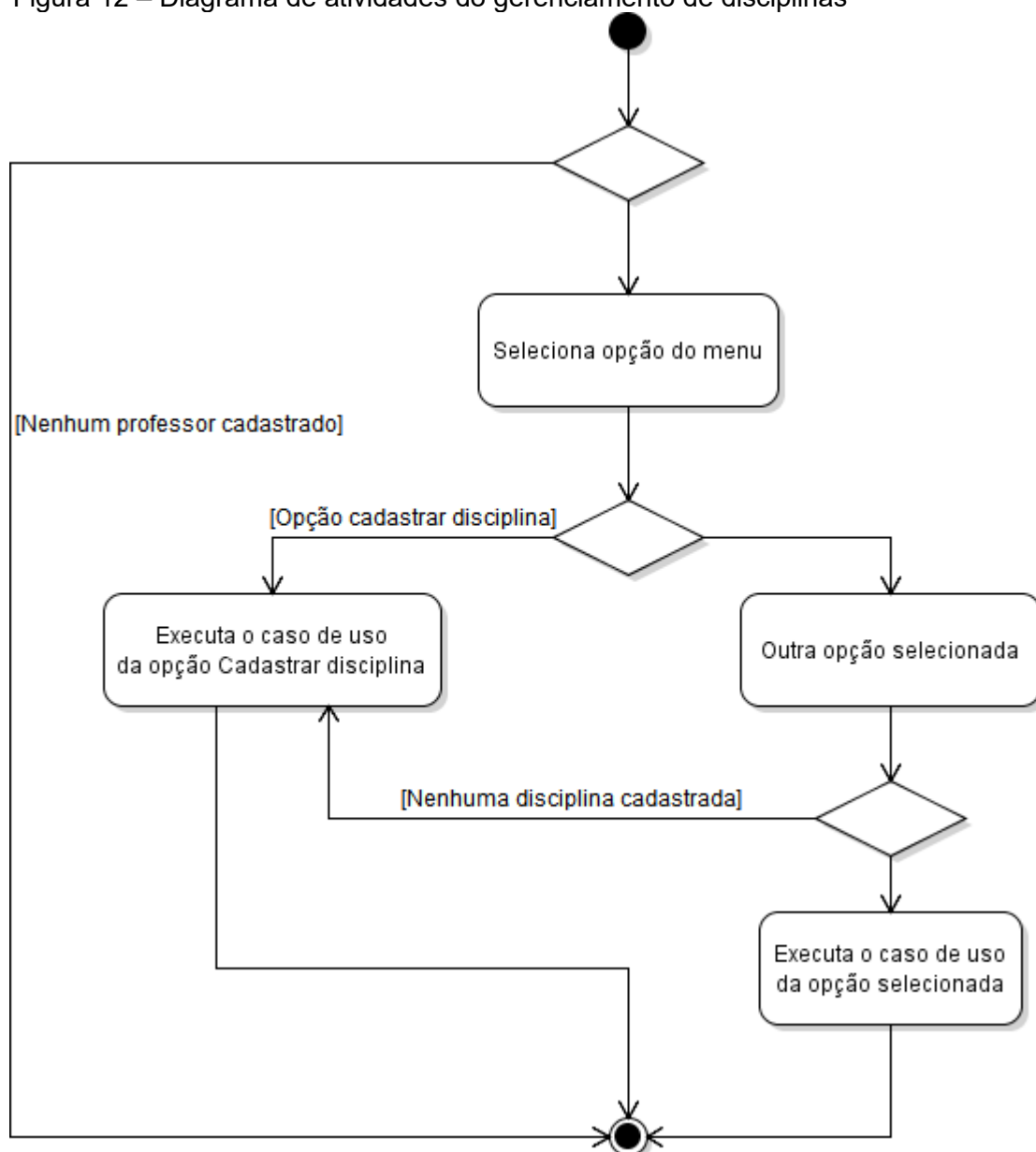
Figura 11 – Diagrama de atividades do gerenciamento de professores



Fonte: Autoria própria, 2016.

Observa-se no diagrama de atividades do gerenciamento de professores, que apenas a opção de cadastro de professores não possui precondições, porém tem as demais ações como suas dependentes diretas. Portanto, ela é a primeira opção que deve ser selecionada durante a execução inicial do software.

Figura 12 – Diagrama de atividades do gerenciamento de disciplinas

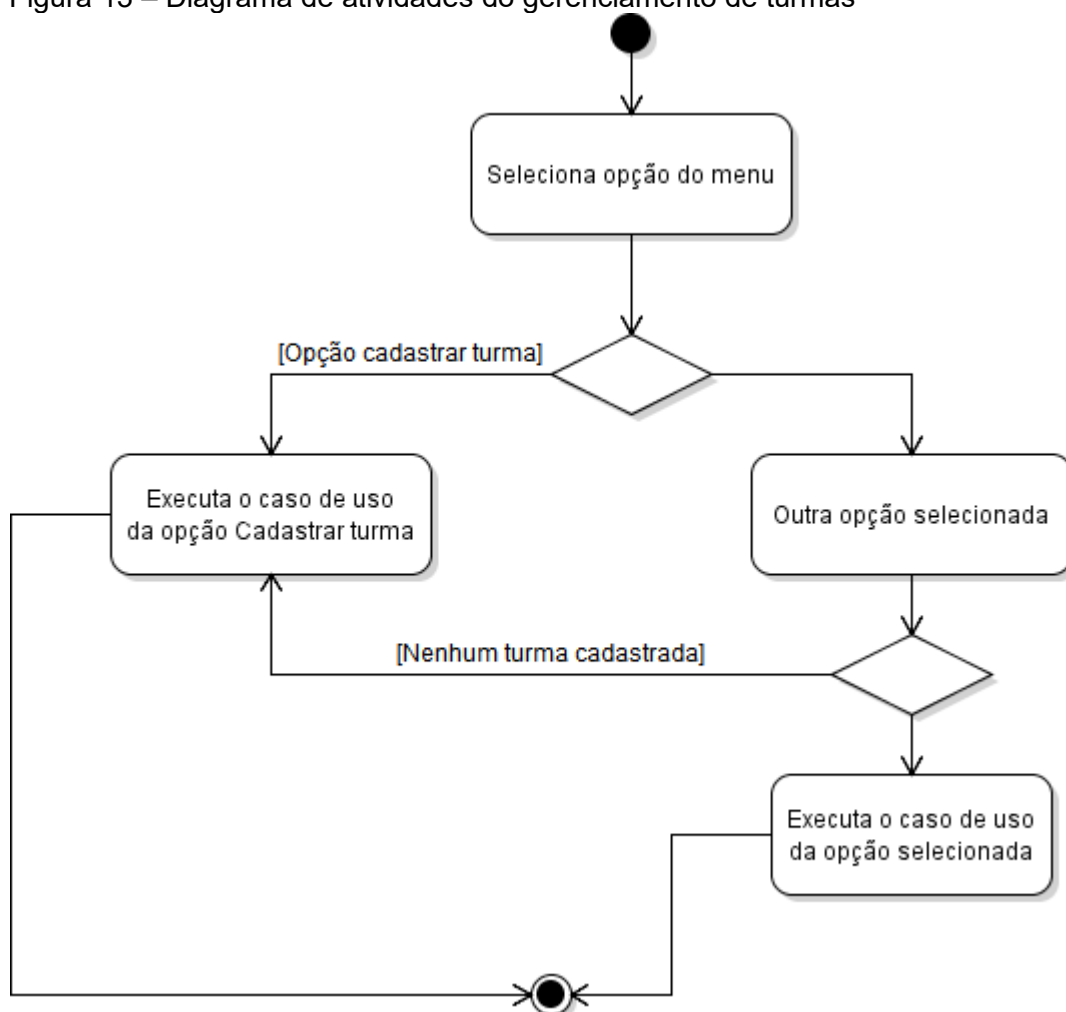


Fonte: Autoria própria, 2016.

As atividades associadas ao gerenciamento de disciplinas, por sua vez, são completamente dependentes do cadastro de professores, pois um dos campos pertencentes ao componente curricular é utilizado para informar o professor que o leciona.

A opção de cadastro de disciplinas possui as demais ações relacionadas à gerência de componentes curriculares como suas dependentes diretas, pois como elas são relacionadas à manipulação de disciplinas, torna-se necessário que ao menos uma tenha sido cadastrada anteriormente.

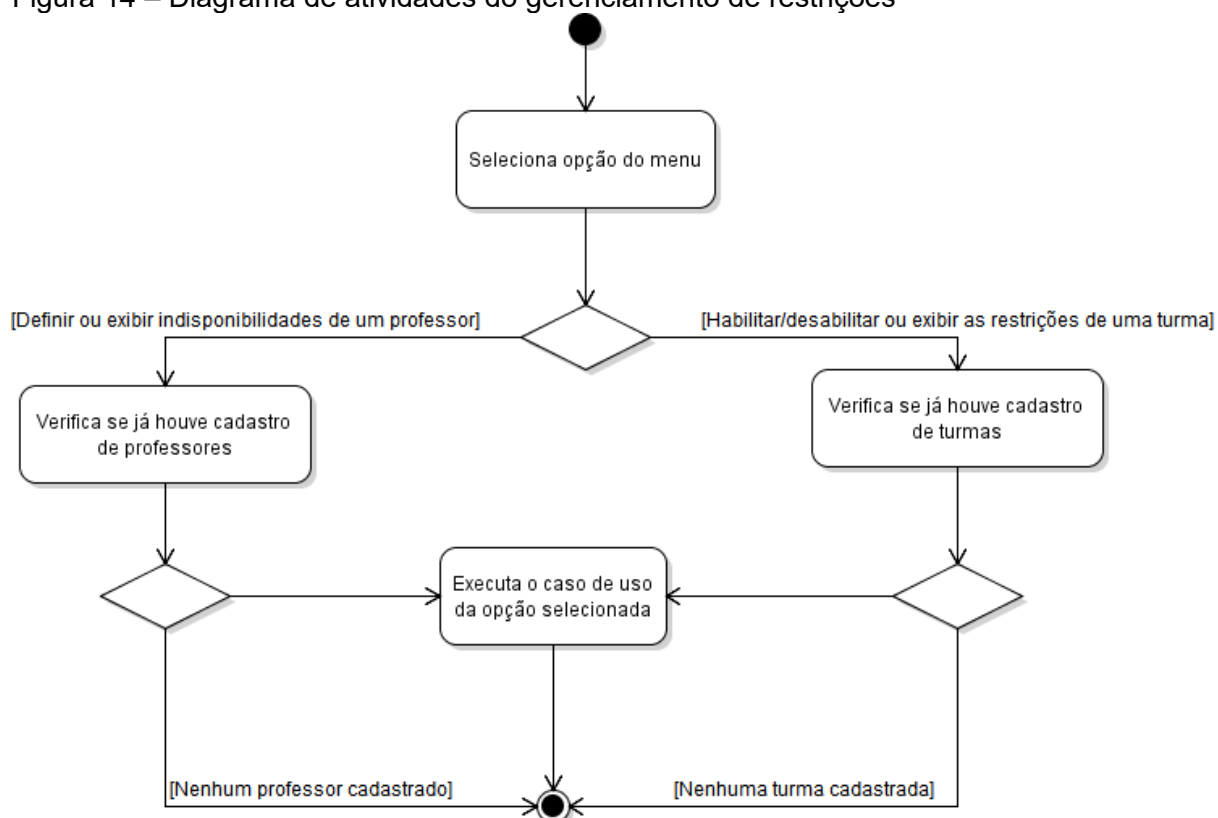
Figura 13 – Diagrama de atividades do gerenciamento de turmas



Fonte: Autoria própria, 2016.

As precondições presentes no gerenciamento de turmas são similares às daquelas da administração de professores. O cadastro não possui dependências, enquanto que as outras atividades têm como pré-requisito a condição de que, pelo menos, uma turma tenha sido armazenada. Como é possível adicionar disciplinas à turma em um momento posterior, não é obrigatório que elas sejam inseridas no ato de cadastro desta última.

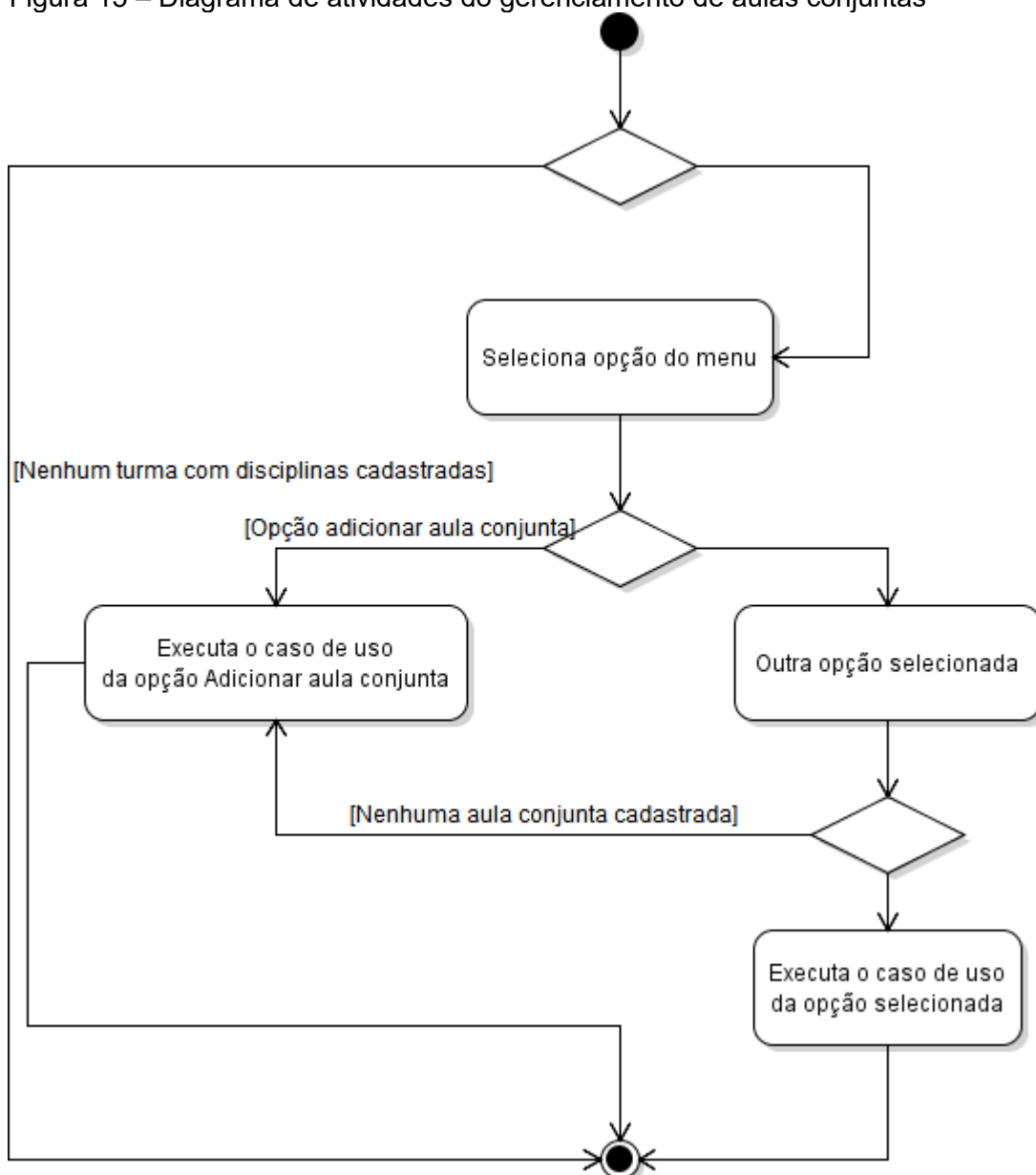
Figura 14 – Diagrama de atividades do gerenciamento de restrições



Fonte: Autoria própria, 2016.

A gerência de restrições possui atividades divididas em duas categorias, uma relacionada aos docentes e outra às turmas. Portanto, elas são dependentes, respectivamente, do cadastro de professores e do registro de turmas. Somente se a dependência correspondente for cumprida é que as ações relativas à categoria selecionada serão realizadas.

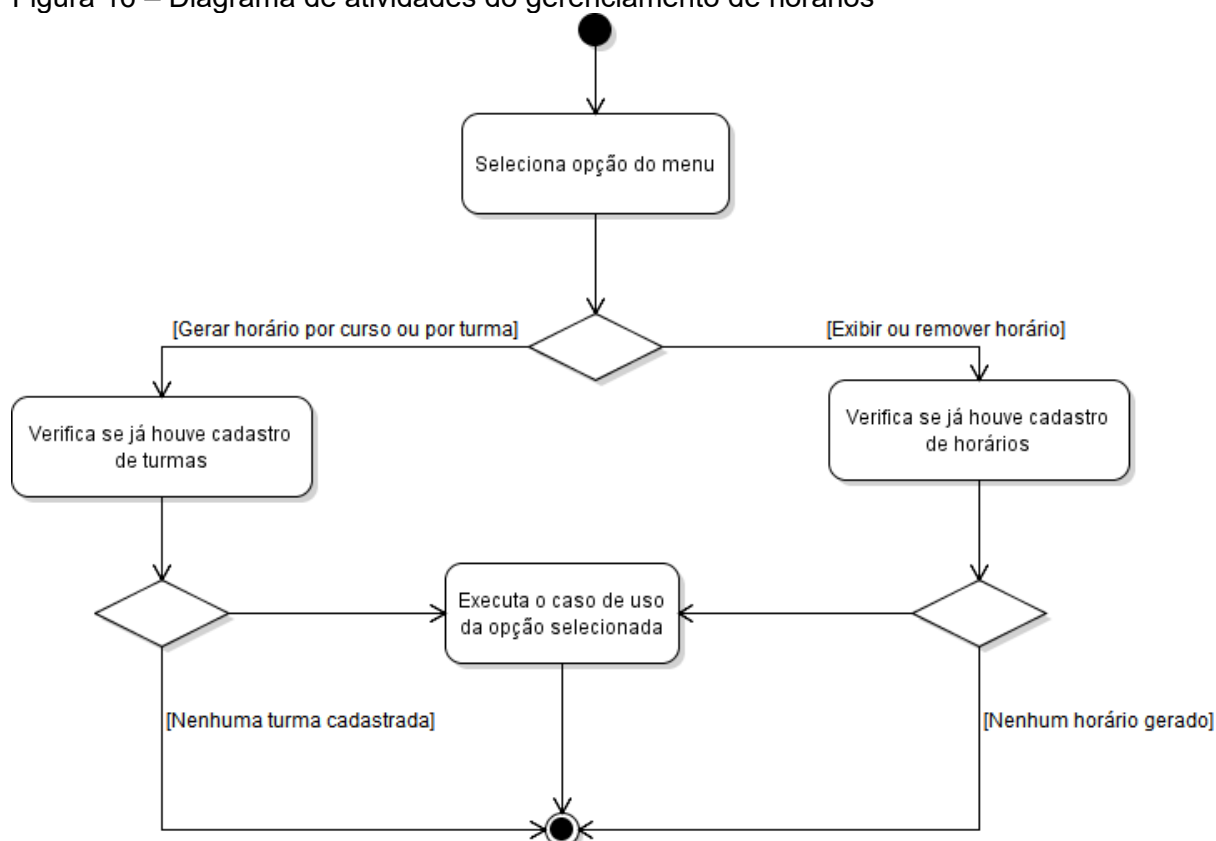
Figura 15 – Diagrama de atividades do gerenciamento de aulas conjuntas



Fonte: Autoria própria, 2016.

A execução das atividades relacionadas à aula conjunta depende do cadastro prévio de ao menos uma turma que possua no mínimo uma disciplina. Após cumprir este pré-requisito, é possível registrar uma aula conjunta entre duas ou mais turmas, e só então realizar as demais ações que compõem esse gerenciamento.

Figura 16 – Diagrama de atividades do gerenciamento de horários



Fonte: Autoria própria, 2016.

É possível notar no diagrama de atividades do gerenciamento de horários, o fato de que há duas restrições impostas às ações demonstradas. Uma é relacionada à geração do horário, que independente de ser por turma ou por curso, necessita do cadastro prévio de pelo menos uma turma, e a outra diz respeito à exibição e à remoção de horário, que depende da execução de ao menos uma das atividades de geração da grade escolar.

Para modelagem do algoritmo genético, fez-se, inicialmente, a definição da taxa de crossover como sendo 60% (0,6) e a de mutação foi configurada para 30% (0,3). Com o intuito de preservar o melhor indivíduo para as próximas gerações, ativou-se o elitismo. Também definiu-se o tamanho da população e o número máximo de gerações com valores, respectivamente, 15 e 10.000.

Após a configuração inicial das características do algoritmo, ele gera uma população inicial com indivíduos contendo genes totalmente aleatórios, que serão utilizados como base para as gerações seguintes. O critério de parada do algoritmo é ao encontrar uma solução para o problema proposto ou ao atingir a quantidade máxima de gerações definidas, finalizar a busca.

Enquanto não for alcançado o critério de parada, uma nova população representando uma geração mais recente é construída. Na criação da nova população, utiliza-se a anterior como base. Portanto, define-se inicialmente o tamanho que ela deve possuir e realiza-se a verificação se o elitismo está ativo. Caso positivo, uma cópia do melhor indivíduo é feita para que ele permaneça entre os cromossomos da nova geração.

Na construção da nova geração, enquanto o tamanho da população não for preenchido novos indivíduos são gerados seguindo uma sequência de criação. Primeiramente realiza-se a escolha de dois pais utilizando a seleção por torneio, onde três indivíduos aleatórios são sorteados dentre aqueles presentes na população base (geração anterior). Desses três cromossomos, os dois melhores são definidos como os pais.

Após a escolha dos pais, o algoritmo verifica, através da taxa de crossover, se ocorrerá ou não o cruzamento entre os indivíduos definidos na seleção por torneio. Isso é feito através de um sorteio de um número entre 0 e 1. Se ele for inferior ou igual à taxa, será realizado o crossover. Caso contrário, os pais escolhidos serão utilizados como indivíduos da nova geração.

O cruzamento entre os pais gerará dois novos indivíduos filhos que irão compor a nova população. Inicialmente faz-se o sorteio de duas posições aleatórias, definidas como primeiro e segundo pontos de corte. Os genes de cada pai são copiados para um filho distinto, de forma a gerar dois indivíduos diferentes, cada um possuindo genes idênticos a um dos pais. Em seguida, utiliza-se os pontos de corte sorteados previamente para realizar a troca das informações genéticas entre os filhos nas posições compreendidas entre esses pontos. Assim, ao final do cruzamento, dois cromossomos terão sido gerados com genes misturados dos pais. O primeiro possuirá os genes do pai localizados nas extremidades e o da mãe entre os pontos de corte. O segundo será exatamente o oposto.

Posterior à determinação dos genes dos dois filhos da nova população, verifica-se, separadamente, se estes cromossomos passarão por um processo de mutação ou não. Como há duas mutações distintas, os indivíduos podem sofrer uma, duas ou nenhuma alteração nos seus genes. Assim como ocorre com o cruzamento, essa possibilidade é feita gerando-se um número aleatório entre 0 e 1 e comparando-o à taxa de mutação.

A primeira forma de mutação faz a permutação entre dois alelos aleatórios dentre os genes do indivíduo. A segunda faz o sorteio de um alelo da população e realiza a mutação desse, de acordo com uma taxa de 50% (0,5). Então, é gerado um número aleatório entre 0 e 1, e se ele for menor ou exatamente igual a 0,5, o alelo escolhido é trocado por um novo sorteado entre aqueles que a turma pode possuir, ou seja, selecionado aleatoriamente entre as disciplinas que a turma possui. Senão, permuta-se aquela posição do gene por 0. Desta forma, o alelo sorteado no início do segundo método de mutação será definido como uma aula vaga.

Finalizada todas as operações com os genes do indivíduo, gera-se a aptidão dele através da função aptidão (fitness). Para tal, o algoritmo executa uma varredura de todos os genes do cromossomo para avaliar a sua qualidade de acordo com as restrições impostas à turma e aos professores das disciplinas. Para cada gene, obtém-se a disciplina relacionada a esta posição e realiza-se a verificação se há uma turma conjunta ou se a turma atual é conjunta de alguma outra para este componente curricular. Caso afirmativo, se a turma compartilhada já possuir um horário definido, o algoritmo penaliza o cromossomo com um ponto se a disciplina conjunta estiver em uma posição diferente entre as turmas. Se não possuir um horário ainda, analisa-se as restrições da turma conjunta.

Para a verificação de aulas geminadas, é feita uma observação se a turma conjunta está marcada para possuir apenas aulas geminadas ou não. Se possuir alguma dessas duas restrições, verifica-se de acordo com o desejado, ou seja, faz-se uma varredura para observar se a disciplina está duplicada em sequência ou não. Caso não esteja de acordo com a restrição imposta, penaliza-se o indivíduo somando um à aptidão.

Também analisa-se os dias da semana disponíveis na turma conjunta. O algoritmo verifica se a disciplina do gene atual está ou não em um dia que não foi marcado como indisponível na turma compartilhada. Caso não esteja de acordo com as restrições de dias semanais, penaliza-se o cromossomo com quatro pontos.

Se a turma conjunta possuir restrição quanto ao janelamento entre disciplinas, faz-se uma varredura dos genes que estão no mesmo dia em que o componente curricular sendo atualmente verificado. Caso a disciplina apareça duas vezes no mesmo dia, porém em horários não adjacentes, considera-se que pode ocorrer um janelamento. Portanto, penaliza-se o indivíduo adicionando um ponto a sua aptidão.

Ainda a respeito da análise da turma conjunta, quando ela não possuir um horário já definido, também realiza-se a verificação se ela possui uma turma que compartilhe com ela, o horário da disciplina atualmente sendo verificada. Caso possua, o algoritmo realiza a mesma análise que foi feita com a outra turma conjunta. Se já foi verificada antes, encerra-se a execução desta parte e procede-se à próxima etapa.

Tendo feito a verificação de turmas conjuntas, caso nenhuma delas possua um horário definido ainda, o algoritmo verifica se o professor responsável pela disciplina em análise possui alguma restrição no horário alocado. Caso afirmativo, é dada uma penalidade de cinco pontos à aptidão do indivíduo.

Há duas outras análises feitas pelo algoritmo, que não são relacionadas às restrições de professores ou da turma. A primeira diz respeito à quantidade de aulas em que a disciplina apareceu no horário gerado. Se for um valor diferente do esperado, penaliza-se com 10 pontos para cada aula a menos ou em excesso. A segunda faz uma análise parecida, não obstante relacionada aos horários vazios. Para cada aula vaga a menos ou em excesso, a penalidade atribuída à aptidão do cromossomo é de 15 pontos.

As três últimas análises de aptidão feitas pelo algoritmo são referentes às restrições de aulas geminadas, dias da semana disponíveis e permissão ou não de ter janelamento entre disciplinas. Estas verificações são feitas de maneiras similares às realizadas nas turmas conjuntas anteriormente e atribui uma penalidade idêntica àquela para cada um desses três tipos de restrições em casos de inconformidade.

Após finalizar a função fitness de cada novo filho gerado, eles são armazenados como indivíduos da nova geração e caso essa população não tenha sido preenchida ainda, retorna-se ao processo de seleção por torneio e faz-se todo o processo de criação dos dois filhos novamente. Caso a nova população já esteja completa, ordena-se os indivíduos presentes nela em ordem crescente em relação à aptidão dos cromossomos. Assim, será possível obter o melhor indivíduo simplesmente selecionando o primeiro da população.

Feito a criação da nova população, é realizada uma verificação referente ao melhor indivíduo dela. Se ele possuir uma aptidão igual a zero, significa que a solução foi encontrada e o algoritmo atingiu o critério de parada. Senão, faz-se uma análise se a geração máxima definida já foi alcançada. Caso afirmativo, o melhor

indivíduo possível foi obtido e o algoritmo interrompe a execução. Não tendo atingido a geração máxima ainda, retorna-se à fase de criação da nova geração.

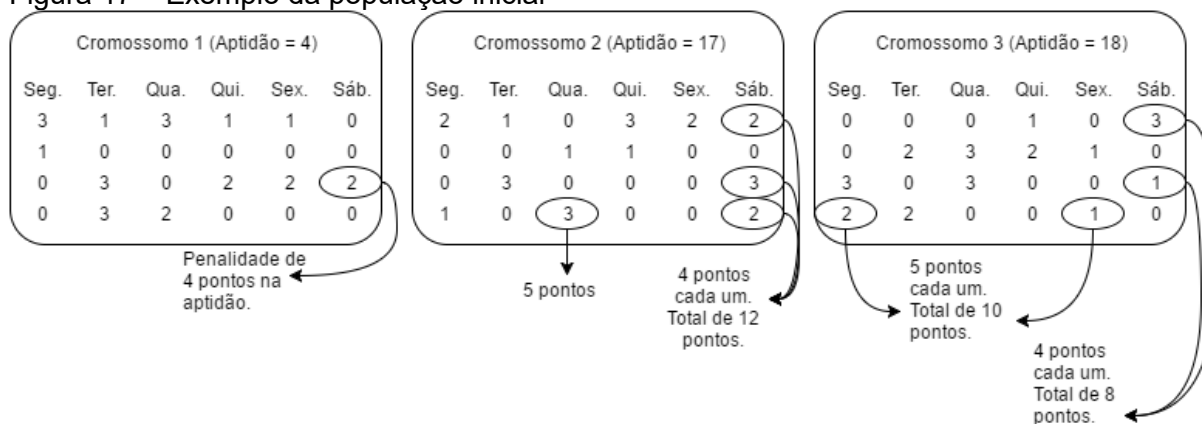
5.3.2.1 Exemplo de Funcionamento do Algoritmo

Considerando-se uma turma do turno noturno que possua as disciplinas: 1 – Sistemas Distribuídos, 2 – Algoritmos e Programação de Computadores e 3 – Inteligência Artificial, onde todos os componentes curriculares são lecionados quatro vezes por semana, foi gerado um exemplo do funcionamento do algoritmo genético.

Definiu-se, também, as seguintes restrições para os professores de cada disciplina: Sistemas Distribuídos (última aula da sexta-feira), Algoritmos e Programação de Computadores (última aula da segunda-feira) e Inteligência Artificial (última aula da quarta-feira). Em relação à turma, a única restrição imposta é que ela não deve possuir aulas nos sábados.

Na figura 17 foram descritos os cromossomos da população inicial que é utilizada como base para a criação das novas gerações. O tamanho de cada população foi definido como 3 indivíduos e o valor 0 indica uma aula vaga no horário.

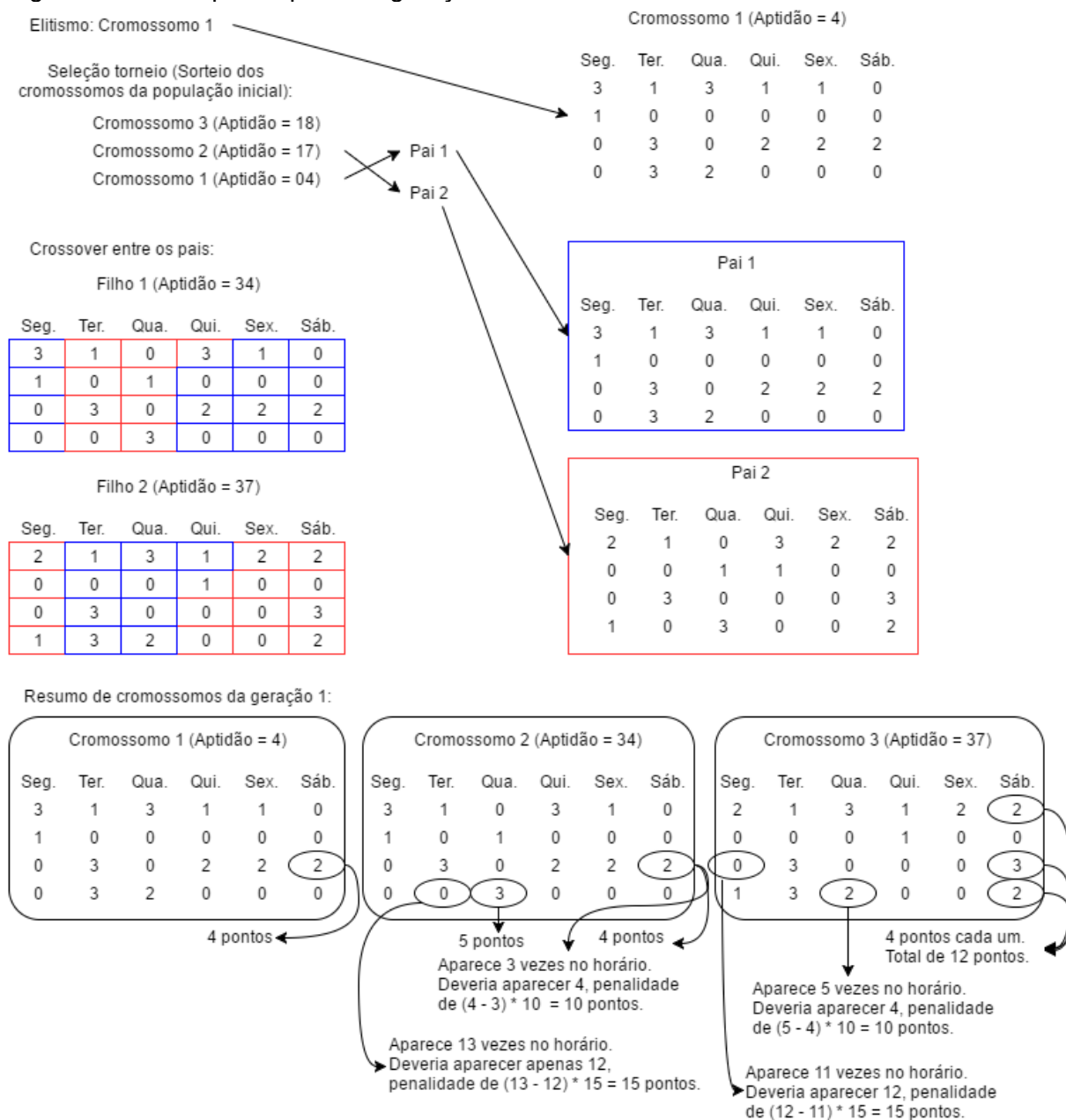
Figura 17 – Exemplo da população inicial



Fonte: Autoria própria, 2016.

A partir da população inicial, é criada a primeira geração, figura 18, que possuirá os primeiros cromossomos considerados como possíveis soluções para o problema.

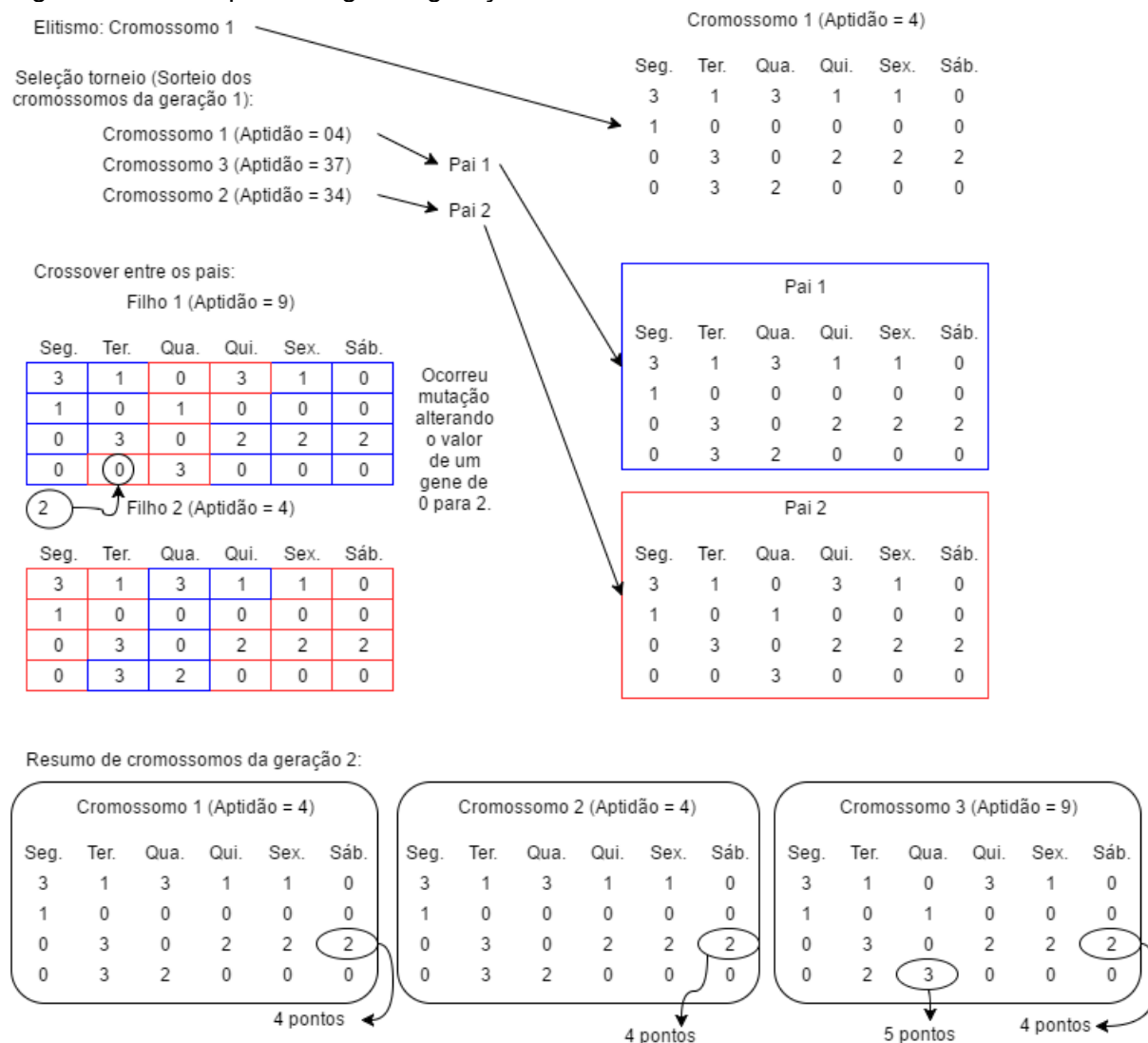
Figura 18 – Exemplo da primeira geração



Fonte: Autoria própria, 2016.

Como não foi possível encontrar uma solução a partir dos cromossomos gerados, a população da primeira geração é utilizada como base para criação de uma nova, figura 19.

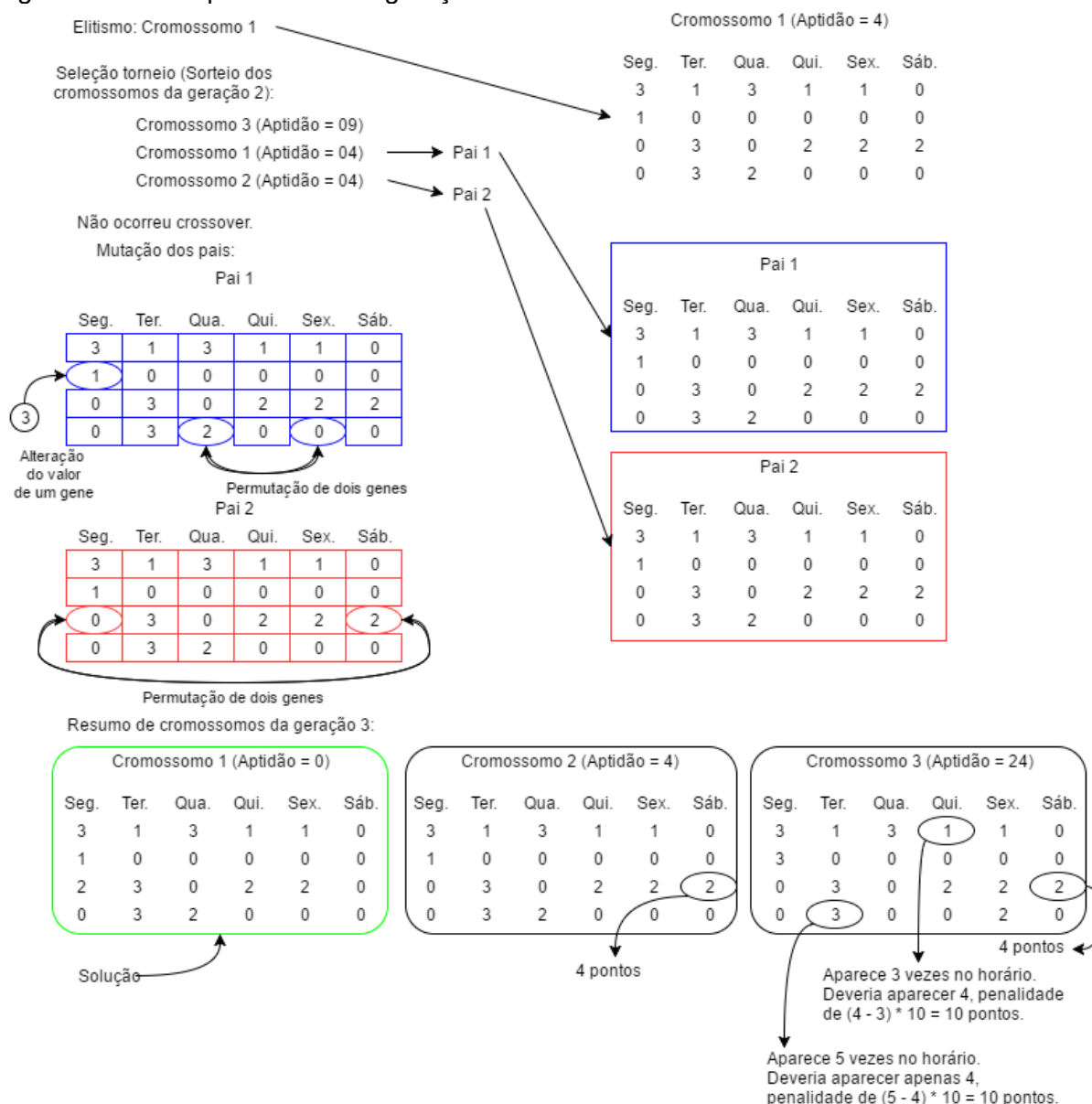
Figura 19 – Exemplo da segunda geração



Fonte: Autoria própria, 2016.

Observa-se que o melhor indivíduo (cromossomo) é preservado ao longo das gerações devido à característica da técnica de substituição de elitismo. A figura 20 demonstra a criação da terceira geração, onde os passos finais para que o algoritmo encontre a solução são descritos.

Figura 20 – Exemplo da terceira geração

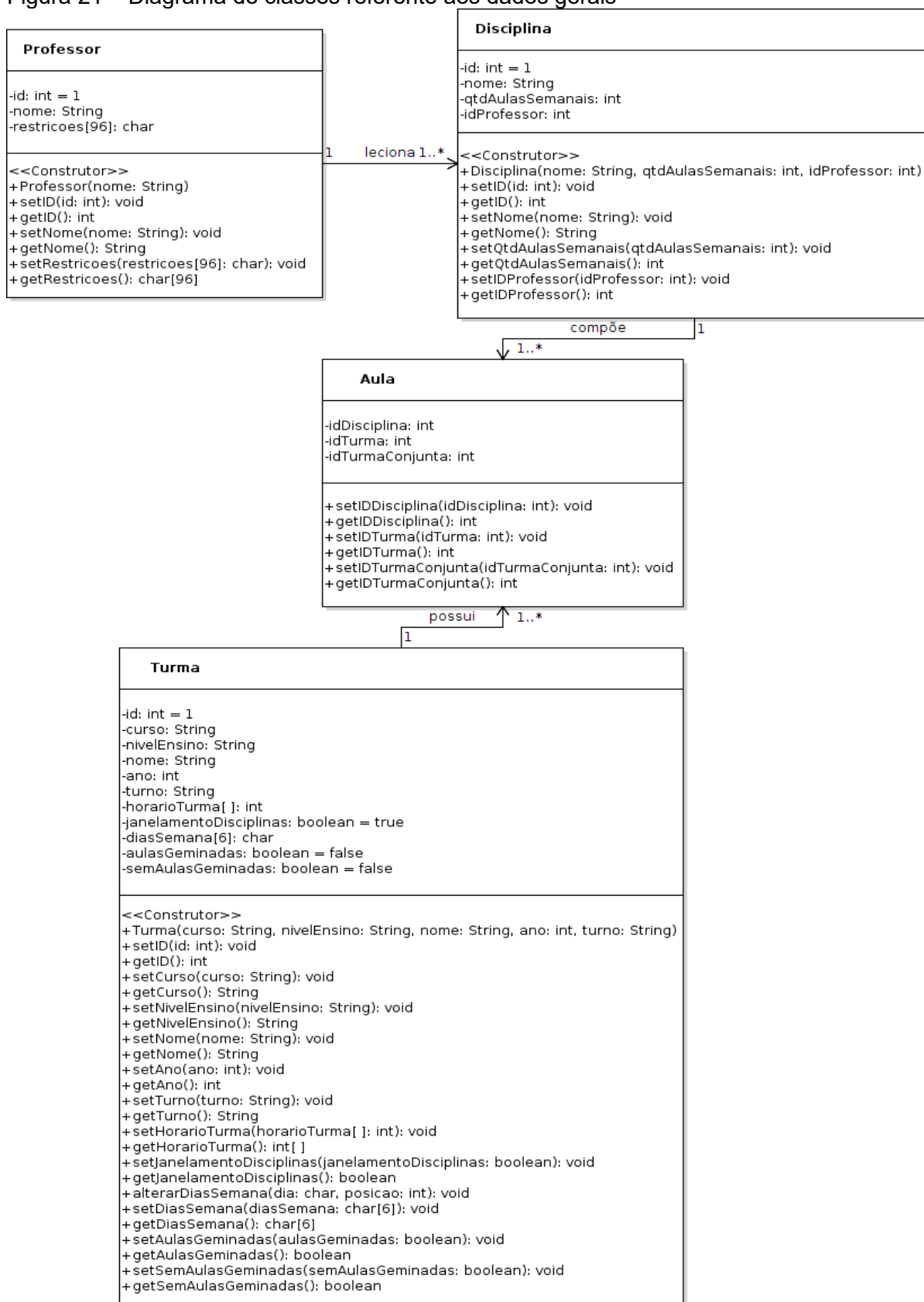


Fonte: Autoria própria, 2016.

5.3.3 Projeto de Dados

Os objetos necessários para o funcionamento do sistema, bem como os atributos e os métodos referentes a eles, foram planejados através do auxílio do diagrama de classes, figuras 21 e 22. Nele é possível observar também, os tipos de todos os dados relacionados à classe, inclusive os valores de retorno e parâmetros dos métodos. As entidades relacionadas aos dados de professores, disciplinas, aulas e turmas podem ser observadas na figura 21.

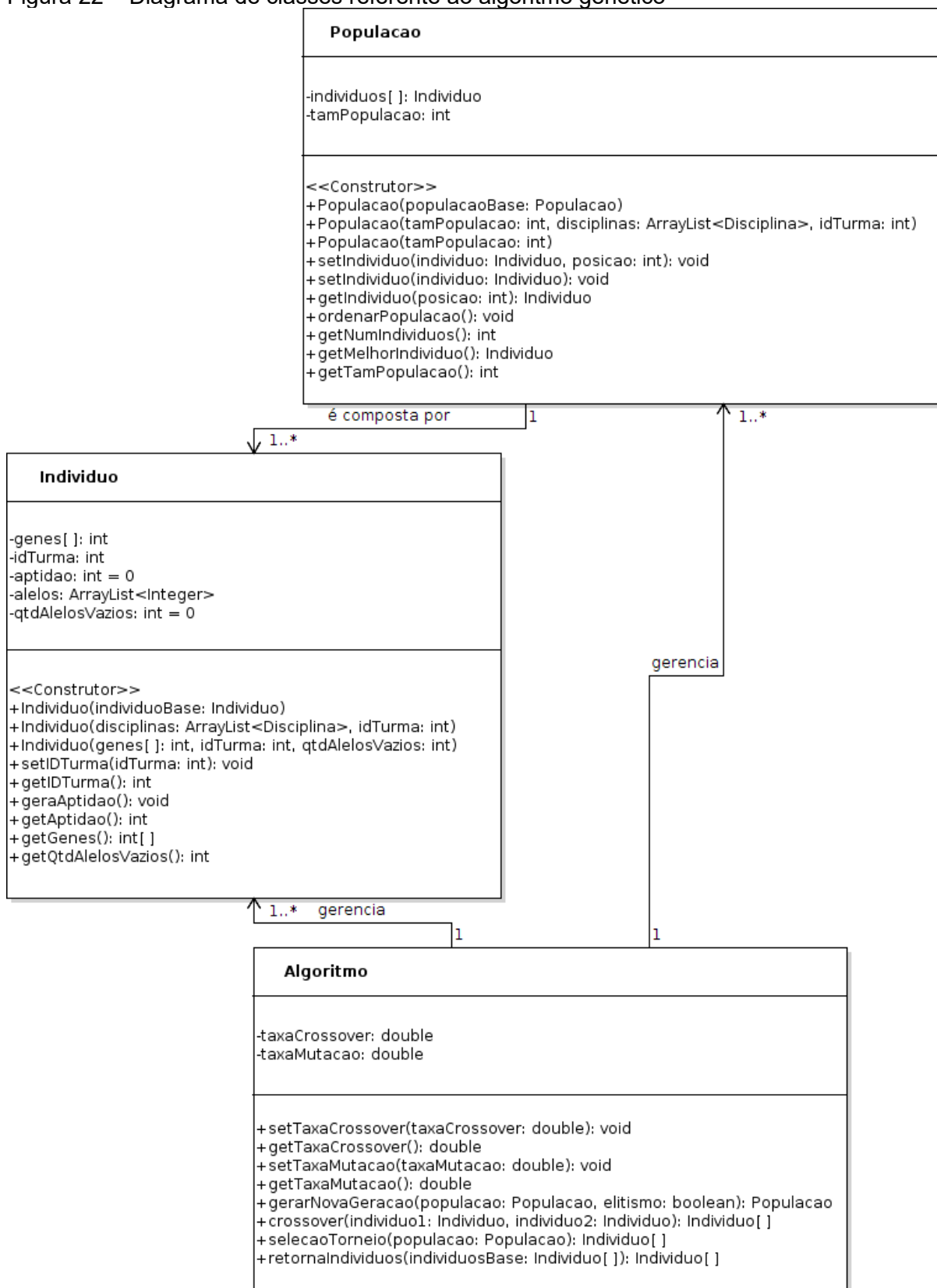
Figura 21 – Diagrama de classes referente aos dados gerais



Fonte: Autoria própria, 2016.

A figura 22 demonstra as entidades que compõem o algoritmo genético, sendo a classe algoritmo a responsável por toda a geração da nova população.

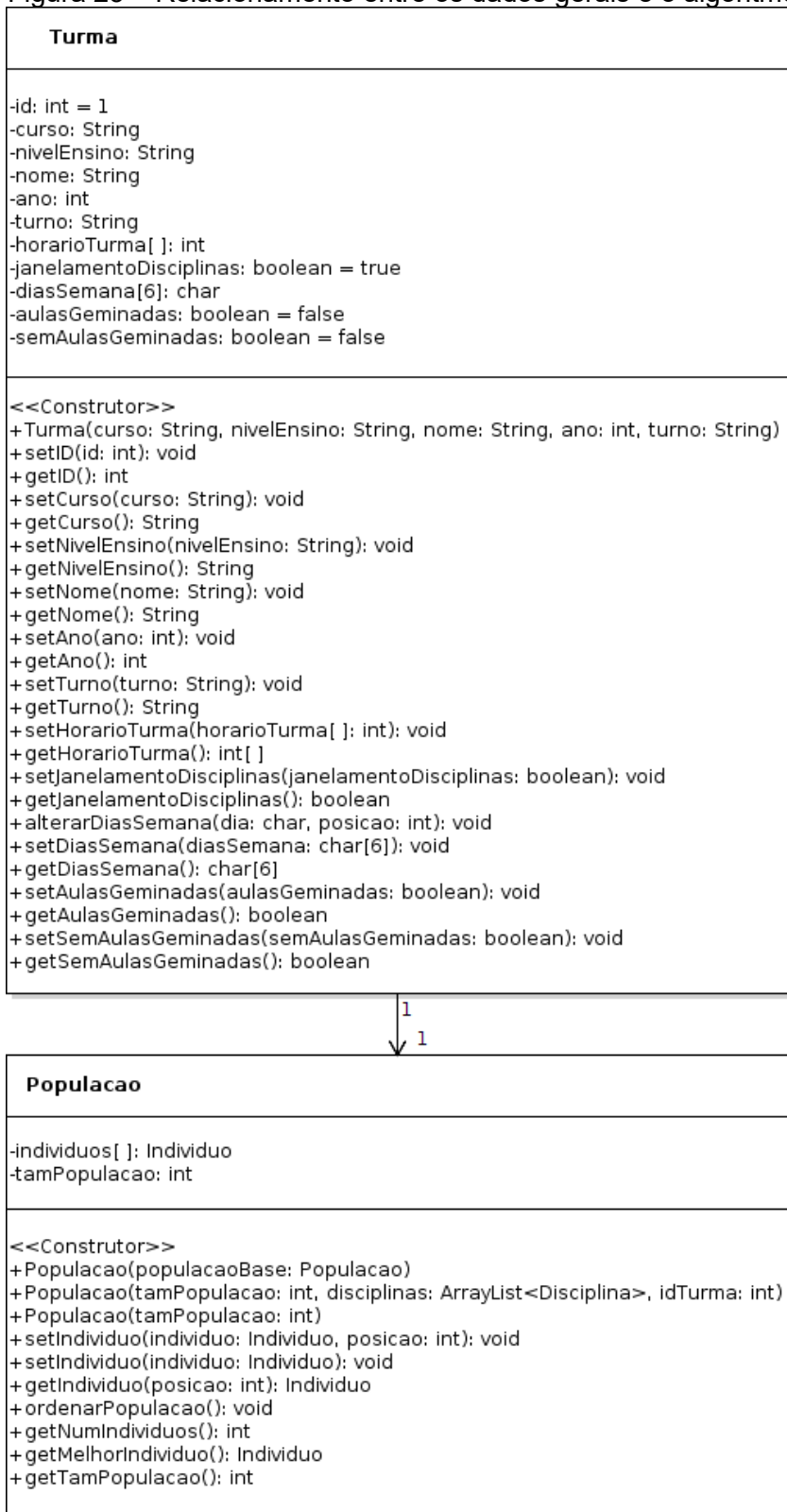
Figura 22 – Diagrama de classes referente ao algoritmo genético



Fonte: Autoria própria, 2016.

Na figura 23 pode-se observar o relacionamento de um para um existente entre as classes turma e população, fazendo a junção entre os diagramas expostos nas figuras 21 e 22.

Figura 23 – Relacionamento entre os dados gerais e o algoritmo genético



Fonte: Autoria própria, 2016.

O usuário possui interação direta com as classes professor, disciplina, aula e turma, sendo estas responsáveis pelo cadastro de dados para uso na geração dos horários escolares. Os demais objetos são utilizados para a composição do algoritmo genético, que tem como tarefa buscar uma solução adequada para as necessidades do usuário.

5.3.4 Projeto de Interface

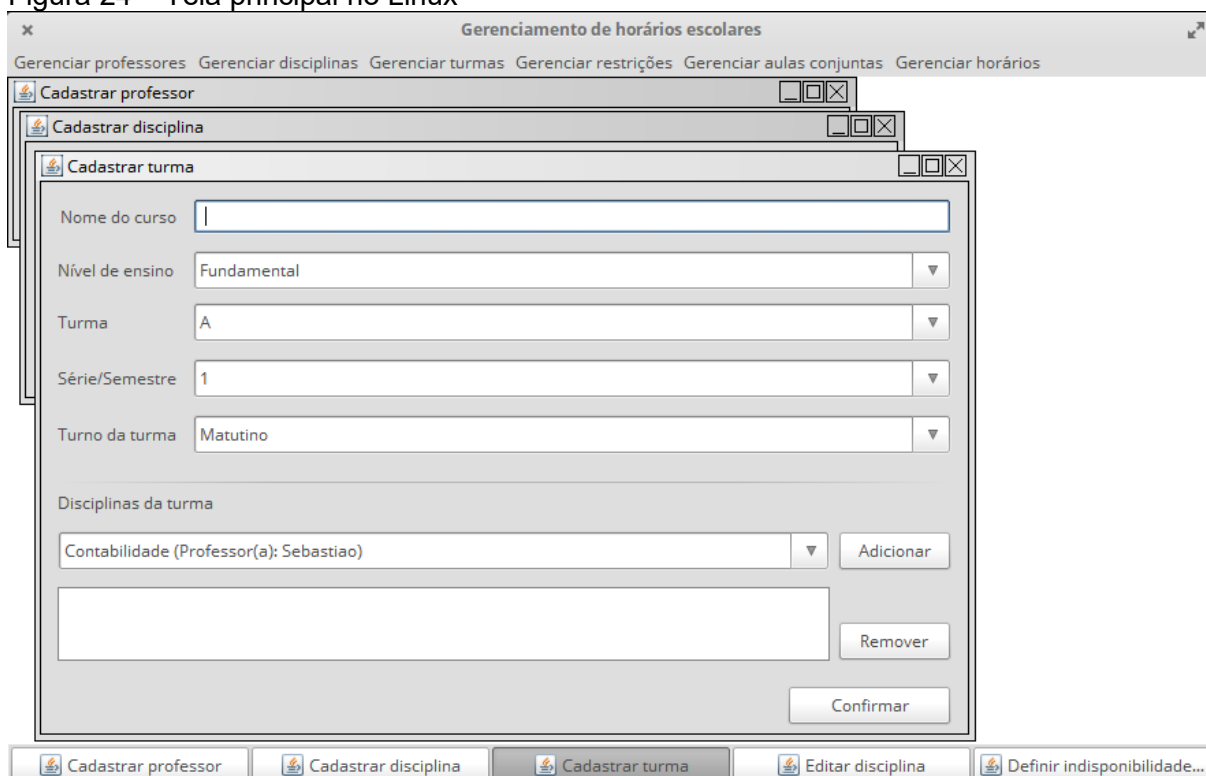
Para o projeto de interface, fez-se algumas modelagens visando obter uma melhor usabilidade para o usuário, conforme o requisito não-funcional RNF001. É importante que o software seja fácil de compreender como executar e operar as tarefas.

Para a tela inicial, utilizou-se uma abordagem da área de estudo Interface Homem-Máquina (IHM) chamada Multiple Document Interface (MDI), que consiste em apresentar novas atividades do sistema como janelas internas à principal. Assim, a tela principal possui algo similar a uma área de trabalho.

A figura 24 apresenta um exemplo da interface projetada no sistema operacional Linux. Nela é possível observar que no topo da tela, figura 25, estão localizados os menus contendo as funcionalidades que o funcionário pode executar. As janelas internas são abertas no espaço em branco da tela principal e podem ser arrastadas de acordo com as necessidades do usuário, bem como minimizadas, maximizadas ou fechadas. Assim como ocorre em alguns sistemas operacionais de computador, as janelas abertas no ambiente Linux são indicadas na parte inferior da tela, figura 26, de forma que permita que o usuário possua um fácil acesso às janelas minimizadas ou em segundo plano.

As telas internas possuem uma aparência um pouco distinta das janelas do sistema, de modo a diferenciá-las das demais e evitar que o usuário confunda as telas pertencentes à ferramenta com as de outros softwares. Essas janelas internas do sistema, possuem campos não editáveis em forma de caixa de seleção, para que o funcionário não possua dúvidas do que é permitido inserir e para que ele não forneça dados inválidos. Essa forma de abordagem também proporciona uma interação mais ágil entre o usuário e o software.

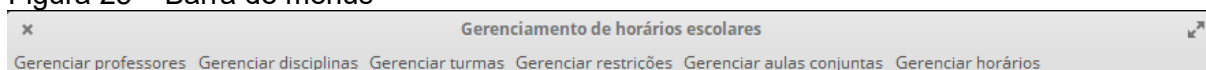
Figura 24 – Tela principal no Linux



Fonte: Autoria própria, 2016.

A figura 25 a seguir demonstra a barra de menus no Linux. É possível observar que todos os gerenciamentos são diretamente expostos nela, garantindo um fácil acesso ao usuário.

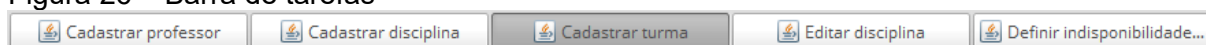
Figura 25 – Barra de menus



Fonte: Autoria própria, 2016.

É possível notar na figura 26, que a demonstração das janelas abertas proporciona uma melhor interação do usuário com a ferramenta computacional. Observa-se também, que a janela em foco é apresentada de forma destacada das demais na barra de tarefas.

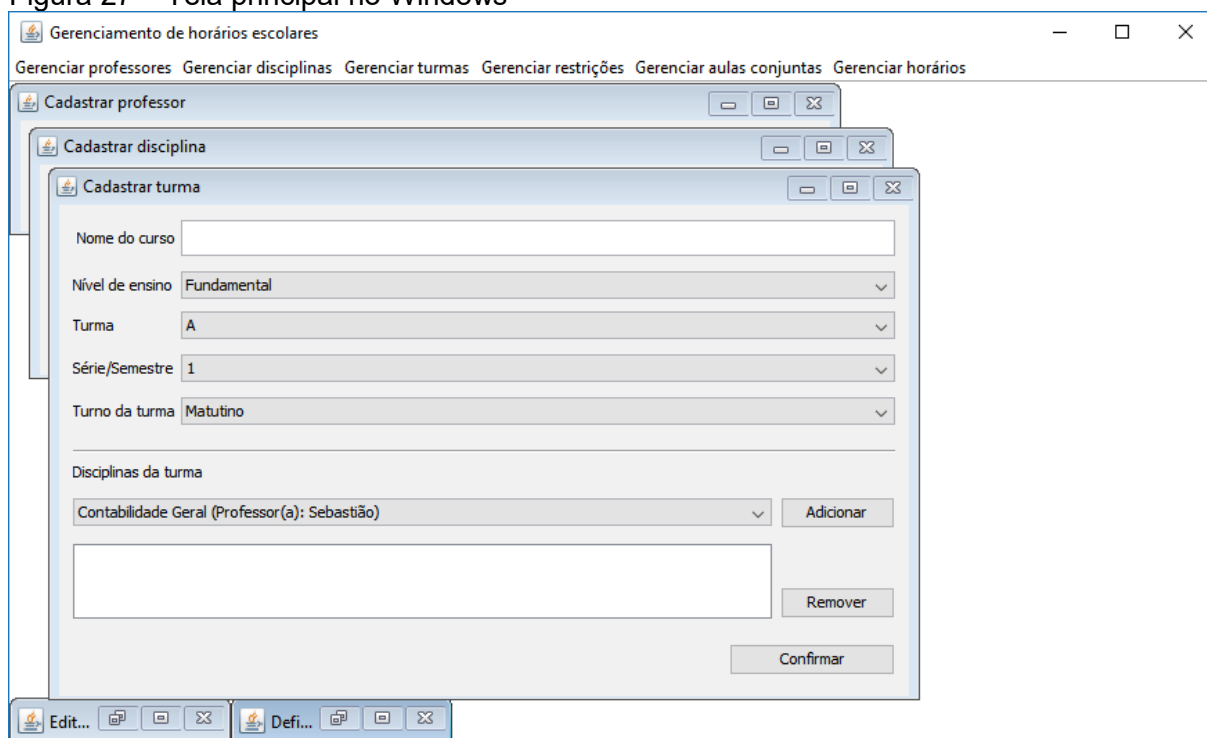
Figura 26 – Barra de tarefas



Fonte: Autoria própria, 2016.

No sistema operacional Windows, a interface é similar à apresentada no Linux, apenas com a diferença de que somente são exibidas na barra de tarefas, as janelas que foram minimizadas. A figura 27 demonstra a tela principal no Windows.

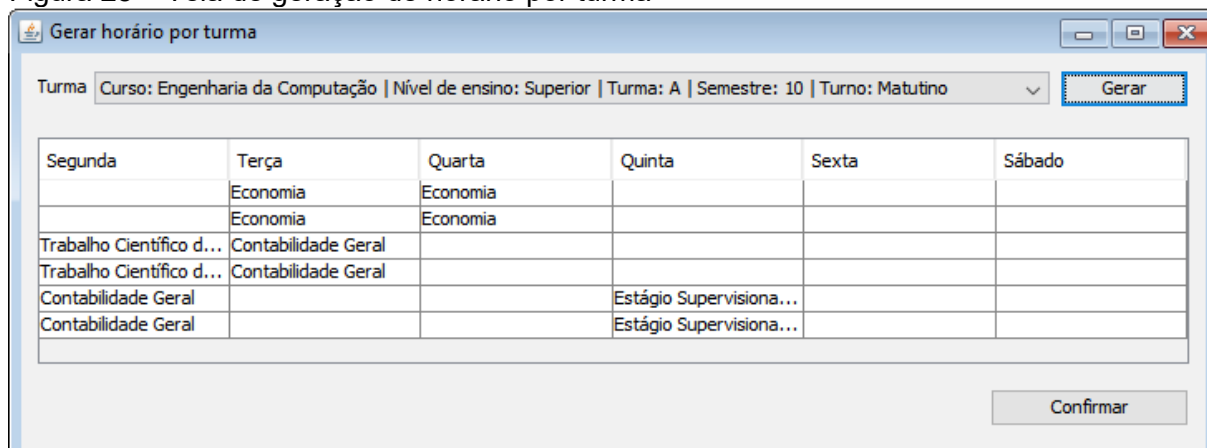
Figura 27 – Tela principal no Windows



Fonte: Autoria própria, 2016.

Após o algoritmo encontrar uma possível solução para o horário, ele é exibido em uma tabela na tela. Assim, o funcionário pode confirmar a grade gerada ou, caso precise de um horário diferente do encontrado pelo algoritmo, pedir para que um novo seja gerado. A tela de geração é exibida na figura 28.

Figura 28 – Tela de geração de horário por turma



Fonte: Autoria própria, 2016.

A interface de exibição de horários gerados é similar ao de criação, porém com um painel na parte inferior, que informa ao usuário qual é o professor que leciona cada disciplina da turma. A tela de exibição de horários pode ser conferida na figura 29.

Figura 29 – Tela de exibição de horários

Turma: Curso: Engenharia da Computação | Nível de ensino: Superior | Turma: A | Série/Semestre: 10 | ... Selecionar

Segunda	Terça	Quarta	Quinta	Sexta	Sábado
	Economia	Economia			
	Economia	Economia			
Trabalho Científico...	Contabilidade Geral				
Trabalho Científico...	Contabilidade Geral				
Contabilidade Geral			Estágio Supervisio...		
Contabilidade Geral			Estágio Supervisio...		

Disciplina: Trabalho Científico de Formatura II | Professor(a): Francisco
 Disciplina: Estágio Supervisionado II | Professor(a): Igor
 Disciplina: Contabilidade Geral | Professor(a): Sebastião
 Disciplina: Economia | Professor(a): Wivaldo

Fonte: Autoria própria, 2016.

5.4 CODIFICAÇÃO

Após a finalização das fases relacionadas à modelagem e elaboração do projeto, deu-se início à construção da ferramenta computacional para geração de horários escolares.

O planejamento realizado durante as etapas anteriores foi de extrema importância para que o processo de codificação do software fosse facilitado e se tornasse mais rápido. Através das especificações definidas no diagrama de classes, pôde-se construí-las apenas seguindo a modelagem proposta, evitando, assim, atrasos no avanço das atividades dessa fase.

Os diagramas de caso de uso e de atividades foram utilizados para definir as implementações necessárias para alcançar as interações que o usuário possui com o sistema e garantir que todas as funcionalidades desejadas fossem codificadas.

A descrição do algoritmo, por sua vez, foi utilizada como forma de implementar a lógica adequada para o funcionamento correto da inteligência artificial a ser utilizada para resolução do problema proposto. Os passos descritos no projeto procedimental, no que se diz respeito ao algoritmo genético, foram essenciais para obter um desenvolvimento mais preciso em relação à quantidade inicial de correções a serem feitas.

O projeto de interface foi decisivo para que as janelas finais do software pudessem corresponder adequadamente aos requisitos propostos, proporcionando uma boa usabilidade e experiência para o usuário, seguindo as diretrizes descritas pelo requisito não-funcional de código RNF001.

As primeiras funcionalidades implementadas foram aquelas relacionadas aos requisitos funcionais de prioridade essencial. A sequência de desenvolvimento dessas tarefas prioritárias foi: cadastrar professor, cadastrar disciplina, cadastrar turma, adicionar disciplina à turma, gerar horário por turma, exibir horário, definir indisponibilidades do professor e habilitar/desabilitar restrições de uma turma.

Por utilizar a linguagem de programação Java, o requisito não-funcional relacionado à portabilidade já estava sendo cumprido no momento de codificação. As tarefas foram implementadas visando atingir de imediato os requisitos referentes à confiabilidade (RNF002) e à funcionalidade (RNF004).

5.5 TESTES

A realização dos testes sobre uma turma utilizou um total de 134 restrições envolvendo diferentes valores para o tamanho da população e para as taxas de crossover e de mutação. Para cada quantidade avaliada, fez-se cinco execuções e, a partir dos resultados, tabulou-se a média de gerações e do tempo necessários para obtenção da solução.

As tabelas de 1 a 5 demonstram os resultados obtidos através de testes envolvendo tamanhos da população variados entre 5 e 100 indivíduos. As avaliações foram feitas utilizando taxas de crossover entre 50% e 80%. Esses valores foram definidos de tal forma, pois é necessário que as novas gerações possam obter uma certa diversidade, não obstante, deve-se permitir que os pais sejam, em alguns casos aleatórios, totalmente preservados.

Em relação à taxa de mutação, definiu-se valores de teste entre 5% e 45%. Quantidades acima desta última não foram avaliadas, por tratar-se de uma unidade muito próxima ao valor mínimo considerado para as taxas de crossover. Tal característica é indesejada, pois dá ao algoritmo a possibilidade de criar novas populações muito distintas das gerações anteriores.

Tabela 1 – Testes com população de 5 indivíduos

Quantidade de restrições	Tamanho da população	Taxa de crossover	Taxa de mutação	Quantidade de gerações	Tempo de execução (segundos)
134	5	50%	5%	5902	106,4
134	5	50%	15%	2941,8	53,8
134	5	50%	25%	1464,6	26,8
134	5	50%	35%	1805	33
134	5	50%	45%	1472,8	27
134	5	60%	5%	6752,6	120,4
134	5	60%	15%	2964,4	53,4
134	5	60%	25%	1964,6	35,6
134	5	60%	35%	2143,8	40
134	5	60%	45%	2041,4	38,8
134	5	70%	5%	6724	122,6
134	5	70%	15%	3426,4	62,8
134	5	70%	25%	2652,6	48,4
134	5	70%	35%	1113	20,4
134	5	70%	45%	1562,2	28,8
134	5	80%	5%	4815,2	86,2
134	5	80%	15%	3273	61
134	5	80%	25%	2195,2	40
134	5	80%	35%	2318,4	42,4
134	5	80%	45%	1710,4	31,6

Fonte: Autoria própria, 2016.

Através da análise dos testes realizados com uma população de cinco indivíduos, pode-se observar que o melhor resultado encontrado foi para as taxas de crossover e de mutação cujos valores são, respectivamente, 70% e 35%.

Tabela 2 – Testes com população de 15 indivíduos

Quantidade de restrições	Tamanho da população	Taxa de crossover	Taxa de mutação	Quantidade de gerações	Tempo de execução (segundos)
134	15	50%	5%	2418,4	151,8
134	15	50%	15%	467,4	30,4
134	15	50%	25%	622,6	41,2
134	15	50%	35%	647,8	41,8
134	15	50%	45%	718,4	45,6
134	15	60%	5%	2179,8	139,4
134	15	60%	15%	1087,8	68,8
134	15	60%	25%	1072,2	71
134	15	60%	35%	625,8	40
134	15	60%	45%	605,4	40,2
134	15	70%	5%	2032,6	129,6
134	15	70%	15%	752,6	47,8
134	15	70%	25%	509,6	32
134	15	70%	35%	726,6	46,2
134	15	70%	45%	514,8	32,8
134	15	80%	5%	2290	144,2
134	15	80%	15%	1147,2	74,6
134	15	80%	25%	469	31
134	15	80%	35%	415	26,4
134	15	80%	45%	616,4	39,8

Fonte: Autoria própria, 2016.

A população contendo 15 indivíduos, por sua vez, apresentou valores de 80% e 35% como os mais satisfatórios para as taxas de crossover e mutação, respectivamente. Observa-se que as porcentagens do melhor resultado foram próximas para os testes realizados com o tamanho da população de 5 e 15 cromossomos, porém as médias de gerações e de tempo de execução foram distintas. Na população maior, é possível obter a solução em uma quantidade menor de iterações, pois por possuir mais indivíduos, permite que o algoritmo tente chegar ao resultado mais vezes a cada geração. No entanto, isso provoca um aumento no tempo de criação de cada nova população, tornando a população de tamanho 5 mais rápida do que a de 15 cromossomos. Esse padrão irá se repetir ao longo dos testes, devido à característica inversamente proporcional envolvendo o tamanho da

população e a quantidade de gerações necessárias para que a solução seja encontrada.

Tabela 3 – Testes com população de 25 indivíduos

Quantidade de restrições	Tamanho da população	Taxa de crossover	Taxa de mutação	Quantidade de gerações	Tempo de execução (segundos)
134	25	50%	5%	1119,8	122
134	25	50%	15%	857,2	94,8
134	25	50%	25%	368,2	40,6
134	25	50%	35%	446,2	49,2
134	25	50%	45%	348,4	39
134	25	60%	5%	1363,6	149
134	25	60%	15%	499,2	54,2
134	25	60%	25%	418	46
134	25	60%	35%	396,4	44,2
134	25	60%	45%	470,2	52,8
134	25	70%	5%	1285,2	140,4
134	25	70%	15%	703,6	77,2
134	25	70%	25%	414,4	45,6
134	25	70%	35%	339,8	37,8
134	25	70%	45%	322,8	36,2
134	25	80%	5%	1755,4	190
134	25	80%	15%	487,8	54,6
134	25	80%	25%	389,2	43,2
134	25	80%	35%	250	27,8
134	25	80%	45%	309,8	34,4

Fonte: Autoria própria, 2016.

Os dados obtidos pelos testes de população com tamanho de 25 indivíduos, demonstraram que o melhor resultado foi novamente para as taxas de crossover e de mutação correspondentes a 80% e 35%, respectivamente.

Observa-se que a característica inversamente proporcional entre o tamanho da população e a quantidade de gerações para encontrar a solução provocou uma diminuição de quase 50% quando comparado o melhor resultado entre os testes realizados com 15 e 25 cromossomos. Ainda assim, a média obtida pelo tempo de execução foi muito próxima.

Tabela 4 – Testes com população de 50 indivíduos

Quantidade de restrições	Tamanho da população	Taxa de crossover	Taxa de mutação	Quantidade de gerações	Tempo de execução (segundos)
134	50	50%	5%	622,2	146,6
134	50	50%	15%	273	65,2
134	50	50%	25%	277,8	66
134	50	50%	35%	194,8	46,6
134	50	50%	45%	269	63,8
134	50	60%	5%	575,2	134,4
134	50	60%	15%	393,4	93
134	50	60%	25%	216,4	52,4
134	50	60%	35%	220,8	52,2
134	50	60%	45%	326,6	78,2
134	50	70%	5%	806,2	186,4
134	50	70%	15%	279	67
134	50	70%	25%	191,4	45,6
134	50	70%	35%	336,8	79,2
134	50	70%	45%	264,4	63,6
134	50	80%	5%	659	153,4
134	50	80%	15%	223,4	53
134	50	80%	25%	218	52
134	50	80%	35%	231,2	55
134	50	80%	45%	238	56,8

Fonte: Autoria própria, 2016.

Observando-se os dados obtidos pelos testes envolvendo uma população com 50 indivíduos, nota-se que houve uma pequena mudança em relação ao melhor resultado. Os valores mais satisfatórios encontrados foram aqueles resultantes das taxas de crossover e de mutação correspondentes a, respectivamente, 70% e 25%.

Tabela 5 – Testes com população de 100 indivíduos

Quantidade de restrições	Tamanho da população	Taxa de crossover	Taxa de mutação	Quantidade de gerações	Tempo de execução (segundos)
134	100	50%	5%	416,8	200,4
134	100	50%	15%	143,8	71,4
134	100	50%	25%	122,2	60,2
134	100	50%	35%	153,4	75
134	100	50%	45%	278,6	135
134	100	60%	5%	414,4	201
134	100	60%	15%	157,2	76,4
134	100	60%	25%	130,8	64,2
134	100	60%	35%	160,6	79,2
134	100	60%	45%	179,8	89,2
134	100	70%	5%	329,4	157,8
134	100	70%	15%	201,8	99,2
134	100	70%	25%	160,8	78,2
134	100	70%	35%	144,8	71
134	100	70%	45%	191,8	94,4
134	100	80%	5%	529	255
134	100	80%	15%	156	77
134	100	80%	25%	140	67
134	100	80%	35%	180,4	89
134	100	80%	45%	211,6	104,2

Fonte: Autoria própria, 2016.

Entre os diferentes tamanhos de população avaliados, o que mostrou-se mais distinto foi o de 100 indivíduos, pois enquanto os melhores resultados para os demais haviam sido para uma taxa de crossover de 70% e 80%, a de 100 cromossomos foi igual a 50%. Entretanto, a mutação mais adequada foi a que corresponde a uma taxa de 25%, permanecendo-se similar às análises realizadas para outros tamanhos de população.

A partir dos dados descritos nas tabelas 1 a 5, organizou-se os resultados separando-os com base no tamanho da população, na taxa de crossover e na taxa de mutação, onde cada divisão foi representada nas tabelas 6 a 8 dispostas a seguir.

Tabela 6 – Média de execução por tamanho da população

Tamanho da população	Quantidade média de gerações	Tempo médio (segundos)
5	2962,17	53,97
15	995,97	63,73
25	627,26	68,95
50	340,83	80,52
100	220,16	107,24

Fonte: Autoria própria, 2016.

Conforme as médias demonstradas segundo a categoria de tamanho da população, a quantidade ideal de indivíduos para o algoritmo desenvolvido é de 5 cromossomos, pois mesmo precisando de mais gerações para encontrar a solução, o tempo médio para obter o resultado foi consideravelmente menor.

Tabela 7 – Média de execução por taxa de crossover

Taxa de crossover	Quantidade média de gerações	Tempo médio (segundos)
50%	974,088	61,5047619
60%	1094,416	66,85714286
70%	1039,464	64,3047619
80%	1009,144	61,78095238

Fonte: Autoria própria, 2016.

Ao observar os valores filtrados segundo a taxa de crossover, nota-se que o tempo médio para as porcentagens 50% e 80% do cruzamento são bem próximas. Entretanto, ao comparar a quantidade de gerações relacionadas a essas duas taxas, pode-se concluir que o uso de 50% para o crossover é mais adequado para construir um algoritmo com uma melhor qualidade.

Tabela 8 – Média de execução por taxa de mutação

Taxa de mutação	Quantidade média de gerações	Tempo médio (segundos)
5%	2149,54	151,85
15%	1021,8	66,78
25%	699,88	49,35
35%	642,53	49,82
45%	632,64	56,61

Fonte: Autoria própria, 2016.

Tratando-se da taxa de mutação, o valor de 45% foi o melhor em relação à quantidade média de gerações, porém 25% foi mais satisfatório segundo a análise do tempo médio. Considerando-se que os valores obtidos pela taxa de 35% são próximos à quantidade de gerações dos 45% e ao tempo médio dos 25%, pode-se

concluir que a taxa de mutação mais adequada para uma melhor performance do algoritmo é a de 35%.

Logo, com base nos testes realizados, obteve-se que os valores mais adequados para a construção do algoritmo são de 5 indivíduos para o tamanho da população e, 50% e 35% para as taxas de crossover e de mutação, respectivamente.

Ao analisar os resultados obtidos com o uso da população cujo tamanho foi de cinco indivíduos, observou-se que a quantidade de gerações para obter a solução chegou, em alguns casos, a valores superiores a seis mil. Portanto, decidiu-se definir o número máximo de gerações para o critério de parada como 30 mil. Assim, evita-se que o algoritmo seja interrompido muito cedo, ao mesmo tempo que previne-se que ele fique buscando uma solução desnecessariamente.

Com base nos valores mais adequados obtidos nos testes de geração de horário por turma, tamanho da população de 5 indivíduos, taxas de crossover e de mutação de 50% e 35%, respectivamente, e quantidade máxima de 30 mil gerações, realizou-se testes com 10 turmas representando os 10 semestres de engenharia da computação. Ao todo foram definidas apenas 18 restrições iniciais, pois levou-se em consideração que o próprio algoritmo gera mais limitações de alocação a medida que ele vai construindo os horários.

Para todas as turmas, a restrição para que sejam geradas somente aulas geminadas foi definida como habilitada e as demais limitações foram relacionadas aos dias disponíveis para aula. No que diz respeito às aulas conjuntas, foi definido que apenas duas turmas possuem uma disciplina em comum entre si. Foram realizadas cinco execuções da geração de horários por curso e os resultados, acompanhados da média final do tempo de execução, foram expostos na tabela 9.

Tabela 9 – Tempo médio para gerar horários de um curso

Quantidade de turmas	Número da execução	Tempo para gerar turmas (minutos)
10	1	11:43
10	2	15:37
10	3	10:34
10	4	13:05
10	5	10:53
Média		12:22

Fonte: Autoria própria, 2016.

6 CONCLUSÃO

O uso de técnicas baseadas na inteligência artificial para resolução do problema de geração de horários escolares provou-se ser uma boa solução para a dificuldade apontada, pois foi possível obter resultados em pouco tempo se comparado à tarefa manual, mesmo em casos de um alto grau de restrições. Portanto, pôde-se confirmar a hipótese e responder a questão central da pesquisa, ambas apresentadas no capítulo 1.

No que diz respeito aos objetivos, tanto o geral, quanto os específicos, foi possível alcançá-los de forma satisfatória. Como a ferramenta computacional desenvolvida utilizou algoritmos genéticos que permitiram a geração automática de horários escolares respeitando as restrições impostas pelo usuário, o objetivo geral foi atingido. Em relação aos específicos, eles foram cumpridos a medida que o ciclo de desenvolvimento de software foi realizado, visto que cada um deles corresponde a uma fase do fluxo mencionado.

Ao longo do desenvolvimento do software ora proposto, foi possível relacionar conteúdos de diversos componentes curriculares que compõem a grade curricular do curso de Engenharia da Computação. Foram eles: Algoritmo e Programação de Computadores I e II, Banco de Dados, Engenharia de Software, Estrutura de Dados, Inteligência Artificial e Interface Homem-Máquina. Tratando-se da disciplina Inteligência Artificial, a confecção desta monografia proporcionou a oportunidade de aprofundar o conhecimento sobre algoritmos que compõem essa área de estudo.

Apesar de ter constatado resultados satisfatórios, alguns problemas relacionados ao desempenho do software desenvolvido foram encontrados. Em situações onde há um número elevado de restrições, muitos conflitos são gerados devido à característica natural do problema de geração de horários escolares. Portanto, foi possível observar que em certas situações onde a solução seria facilmente encontrada através de uma tarefa manual, o algoritmo simplesmente leva mais tempo para gerar o horário. Logo, nestas ocasiões, seria mais eficiente confeccionar a grade da turma manualmente.

6.1 INDICAÇÃO DE TRABALHOS FUTUROS

Os trabalhos a seguir são indicados como melhoria e desenvolvimento futuro da ferramenta implementada nesta monografia:

- Desenvolvimento de uma versão web do software, permitindo que os usuários tenham acesso a ele a partir de diferentes dispositivos, incluindo aparelhos móveis.
- Melhoria de acessibilidade do produto, implementando ferramentas e práticas para tal fim.
- Realizar testes envolvendo outras técnicas de seleção, de crossover e de mutação.
- Desenvolver uma solução utilizando as técnicas do Problema de Satisfação de Restrições e comparar com os resultados obtidos pelo uso de algoritmos genéticos.

REFERÊNCIAS

- AGUIAR, Matheus França. **Utilização de técnicas de busca seletiva no auxílio de processos de tomada de decisão autônoma**. 2014. 63 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Universidade de Brasília, Brasília, 2014. Disponível em: <http://bdm.unb.br/bitstream/10483/15163/1/2014_MatheusFrancaAguiar.pdf>. Acesso em: 19 nov. 2016.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1**: Modelo de qualidade. Rio de Janeiro: ABNT, 2003. 21 p. Disponível em: <http://luizcamargo.com.br/arquivos/NBR_ISO_IEC_9126-1.pdf>. Acesso em: 06 out. 2016.
- CORDENONSI, A. Z.; ARAMBURU, L. G. C.; ALMAÇA, L. **Resolução do problema de quadro de horários através de um algoritmo de satisfação de restrições**. In: VIII SIMPÓSIO DE INFORMÁTICA E III MOSTRA DE SOFTWARE ACADÊMICO, 2003, Uruguaiana - RS, 2003.
- GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo (Org.). **Métodos de pesquisa**. Porto Alegre: UFRGS, 2009. 120 p. (Educação a distância). Disponível em: <<http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>>. Acesso em: 25 maio 2016.
- GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 147 p.
- HAMAWAKI, Cristiane Divina Lemes. **Geração automática de grade horária usando algoritmos genéticos: o caso da Faculdade de Engenharia Elétrica da UFU**. 104 f. Dissertação (Mestrado) - Universidade Federal de Uberlândia, Uberlândia, 2005.
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE STD 830-1998**: IEEE recommended practice for software requirements specifications. Nova Iorque: IEEE, 1998. 37 p. Disponível em: <<http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>>. Acesso em: 06 out. 2016.
- INTERNATIONAL BUSINESS MACHINES. **RATIONAL UNIFIED PROCESS**: Best practices for software development teams. Cupertino: Rational Software, 2011. 21 p. Disponível em: <https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP02>. Acesso em: 07 out. 2016.
- LOBO, Eduardo Luiz Miranda. **Uma solução do problema de horário escolar via algoritmo genético paralelo**. 2005. 95 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2005.

LUCAS, Diogo C.. **Algoritmos genéticos: uma introdução**. 2002. Disponível em: <<http://www.inf.ufrgs.br/~alvares/INF01048IA/ApostilaAlgoritmosGeneticos.pdf>>. Acesso em: 19 nov. 2016.

PIRES, Alexsandro Santos. **Implementação de um algoritmo heurístico para problemas de restrições**. 2006. 59 f. TCC (Graduação) - Curso de Ciências da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2006. Disponível em: <<http://campeche.inf.furb.br/tccs/2006-II/2006-2alexsandrospiresvf.pdf>>. Acesso em: 19 nov. 2016.

PRESSMAN, Roger S.. **Engenharia de software**. 3. ed. São Paulo: Makron Books, 1995. 530 p.

RAMOS, Patrícia de Siqueira. **Sistema automático de geração de horários para a UFLA utilizando algoritmos genéticos**. 87 f. TCC (Graduação) - Curso de Ciência da Computação, Departamento de Ciência da Computação, Universidade Federal de Lavras, Lavras, 2002.

REIS, Luís Paulo. **Coordenação em sistemas multi-agente: aplicações na gestão universitária e futebol robótico**. 2003. 451 f. Tese (Doutorado) - Curso de Engenharia, Faculdade de Engenharia da Universidade do Porto, Porto, 2003. Disponível em: <<https://web.fe.up.pt/~lpreis/Research.htm#Tese de Doutoramento>>. Acesso em: 19 nov. 2016.

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

SANTOS, J. R.. **AST Um modelo para automação de horários escolares**. 218 f. Tese (Doutorado) - Curso de Matemática Computacional, Universidade Federal de Pernambuco, Recife, 2008.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. Florianópolis: UFSC, 2005. 139 p.

SILVA, Elivaldo Elenildo da. **Otimização de estruturas de concreto armado utilizando algoritmos genéticos**. 194 f. Dissertação (Mestrado) - Curso de Engenharia de Estruturas, Universidade de São Paulo, São Paulo, 2001. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3144/tde-21022002-112505/pt-br.php>>. Acesso em: 21 abr. 2016.

SILVA, Filipe Correa Lima da. **Agentes inteligentes em jogos de computador**. 23 f. Monografia (Especialização) - Curso de Ciência da Computação, Instituto de Matemática e Estatística, São Paulo, 2005. Disponível em: <<https://www.ime.usp.br/~leliane/MAC5701/2005-1oSem/PlanosMonografias/Filipe.pdf>>. Acesso em: 26 mar. 2016.

SOUSA, Vânia Nobre de; MORETTI, Antônio Carlos; PODESTÁ, Valéria Abrão de. Pesquisa Operacional. **Programação da grade de horário em escolas de ensino fundamental e médio**, Rio de Janeiro, v. 28, n. 3, p.399-421, set. 2008. Anual.

VALENTE, José Armando (Org.). **O computador na sociedade do conhecimento**. São Paulo: Unicamp/nied, 1999.

VÉLEZ, Digna Isabel Arteaga. **Otimização de estruturas reticuladas utilizando algoritmos genéticos**. 115 f. Dissertação (Mestrado) - Curso de Estruturas e Construção Civil, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, 2015. Disponível em: <<http://www.pecc.unb.br/wp-content/uploads/dissertacoes/M15-6B-Digna-Vélez.pdf>>. Acesso em: 21 abr. 2016.

APÊNDICE A – Descrições dos casos de uso

Os quadros de 1 a 34 informam as características de cada caso de uso abordado no correspondente diagrama, figura 7. Os dados descritos nos quadros são referentes ao nome e ao código de cada caso de uso, bem como suas dependências de execução, funcionamento básico, fluxos alternativos ao principal e resultado final após término do caso de uso.

Quadro 1 – Descrição do caso de uso “Gerenciar professores”

NOME	Gerenciar professores (CDU001).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Gerenciar professores”. 2. O funcionário escolhe uma das opções apresentadas: Cadastrar professor (CDU002), Editar professor (CDU003), Listar professores (CDU004) ou Remover professor (CDU005). 3. Uma janela referente à opção escolhida é aberta.
FLUXOS ALTERNATIVOS	(A1) Alternativa ao passo 2 – Nenhuma opção selecionada 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe a janela referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 2 – Descrição do caso de uso “Cadastrar professor”

NOME	Cadastrar professor (CDU002).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Cadastrar professor”. 2. O funcionário preenche o campo com o nome do professor. 3. O funcionário confirma os dados inseridos. 4. O sistema armazena os dados no banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	(A1) Alternativa ao passo 4 – Erro ao armazenar os dados no banco de dados 1.a O sistema informa uma mensagem de erro. 1.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram cadastrados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 3 – Descrição do caso de uso “Editar professor”

NOME	Editar professor (CDU003).
PRECONDIÇÕES	Ter executado o caso de uso “CDU002 – Cadastrar professor”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Editar professor”. 2. O funcionário seleciona um professor dentre os apresentados. 3. O funcionário edita os campos referentes ao professor. 4. O funcionário confirma os dados inseridos. 5. O sistema atualiza os dados no banco de dados. 6. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum professor cadastrado</p> <p>1.a O sistema exibe a mensagem “Nenhum professor cadastrado” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU002 – Cadastrar professor” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU003 – Editar professor” é realizado.</p> <p>(A2) Alternativa ao passo 5 – Erro ao armazenar os dados no banco de dados</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário retorna ao passo 4 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram atualizados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 4 – Descrição do caso de uso “Listar professores”

NOME	Listar professores (CDU004).
PRECONDIÇÕES	Ter executado o caso de uso “CDU002 – Cadastrar professor”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Listar professores”. 2. O sistema exibe os professores cadastrados no banco de dados. 3. O funcionário encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum professor cadastrado</p> <p>1.a O sistema exibe a mensagem “Nenhum professor cadastrado” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU002 – Cadastrar professor” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU004 – Listar professores” é realizado.</p> <p>(A2) Alternativa ao passo 2 – Erro de comunicação com o banco de dados.</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Nenhuma.

Fonte: Autoria própria, 2016.

Quadro 5 – Descrição do caso de uso “Remover professor”

NOME	Remover professor (CDU005).
PRECONDIÇÕES	Ter executado o caso de uso “CDU002 – Cadastrar professor”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Remover professor”. 2. O funcionário seleciona um professor dentre os apresentados. 3. O funcionário confirma o professor selecionado para que ele seja removido. 4. O sistema remove o professor do banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum professor cadastrado</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhum professor cadastrado” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU002 – Cadastrar professor” é realizado. 1.c O fluxo básico do caso de uso “CDU005 – Remover professor” é realizado. <p>(A2) Alternativa ao passo 4 – Erro ao remover os dados do banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram removidos do banco de dados.

Fonte: Autoria própria, 2016.

Quadro 6 – Descrição do caso de uso “Gerenciar disciplinas”

NOME	Gerenciar disciplinas (CDU006).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Gerenciar disciplinas”. 2. O funcionário escolhe uma das opções apresentadas: Cadastrar disciplina (CDU007), Editar disciplina (CDU008), Listar disciplinas (CDU009) ou Remover disciplina (CDU010). 3. Uma janela referente à opção escolhida é aberta.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma opção selecionada</p> <ol style="list-style-type: none"> 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe a janela referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 7 – Descrição do caso de uso “Cadastrar disciplina”

NOME	Cadastrar disciplina (CDU007).
PRECONDIÇÕES	Ter executado o caso de uso “CDU002 – Cadastrar professor”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Cadastrar disciplina”. 2. O funcionário preenche os campos com os dados da disciplina. 3. O funcionário confirma os dados inseridos. 4. O sistema armazena os dados no banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum professor cadastrado</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhum professor cadastrado” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU002 – Cadastrar professor” é realizado. 1.c O fluxo básico do caso de uso “CDU007 – Cadastrar disciplina” é realizado. <p>(A2) Alternativa ao passo 4 – Erro ao armazenar os dados no banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram cadastrados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 8 – Descrição do caso de uso “Editar disciplina”

NOME	Editar disciplina (CDU008).
PRECONDIÇÕES	Ter executado o caso de uso “CDU007 – Cadastrar disciplina”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Editar disciplina”. 2. O funcionário seleciona uma disciplina dentre as apresentadas. 3. O funcionário edita os campos referentes à disciplina. 4. O funcionário confirma os dados inseridos. 5. O sistema atualiza os dados no banco de dados. 6. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma disciplina cadastrada</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhuma disciplina cadastrada” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU007 – Cadastrar disciplina” é realizado. 1.c O fluxo básico do caso de uso “CDU008 – Editar disciplina” é realizado. <p>(A2) Alternativa ao passo 5 – Erro ao armazenar os dados no banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 4 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram atualizados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 9 – Descrição do caso de uso “Listar disciplinas”

NOME	Listar disciplinas (CDU009).
PRECONDIÇÕES	Ter executado o caso de uso “CDU007 – Cadastrar disciplina”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Listar disciplinas”. 2. O sistema exibe as disciplinas cadastradas no banco de dados. 3. O funcionário encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma disciplina cadastrada</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhuma disciplina cadastrada” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU007 – Cadastrar disciplina” é realizado. 1.c O fluxo básico do caso de uso “CDU009 – Listar disciplinas” é realizado. <p>(A2) Alternativa ao passo 2 - Erro de comunicação com o banco de dados.</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário encerra o caso de uso.
PÓS-CONDIÇÕES	Nenhuma.

Fonte: Autoria própria, 2016.

Quadro 10 – Descrição do caso de uso “Remover disciplina”

NOME	Remover disciplina (CDU010).
PRECONDIÇÕES	Ter executado o caso de uso “CDU007 – Cadastrar disciplina”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Remover disciplina”. 2. O funcionário seleciona uma disciplina dentre as apresentadas. 3. O funcionário confirma a disciplina selecionada para que ela seja removida. 4. O sistema remove a disciplina do banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma disciplina cadastrada</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhuma disciplina cadastrada” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU007 – Cadastrar disciplina” é realizado. 1.c O fluxo básico do caso de uso “CDU010 – Remover disciplina” é realizado. <p>(A2) Alternativa ao passo 4 – Erro ao remover os dados do banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram removidos do banco de dados.

Fonte: Autoria própria, 2016.

Quadro 11 – Descrição do caso de uso “Gerenciar turmas”

NOME	Gerenciar turmas (CDU011).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Gerenciar turmas”. 2. O funcionário escolhe uma das opções apresentadas: Cadastrar turma (CDU012), Editar turma (CDU013), Listar turmas (CDU014) ou Remover turma (CDU015). 3. Uma janela referente à opção escolhida é aberta.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma opção selecionada</p> <ol style="list-style-type: none"> 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe a janela referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 12 – Descrição do caso de uso “Cadastrar turma”

NOME	Cadastrar turma (CDU012).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Cadastrar turma”. 2. O funcionário preenche os campos com os dados da turma. 3. O funcionário confirma os dados inseridos. 4. O sistema armazena os dados no banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 3 – Funcionário adiciona disciplinas à turma</p> <ol style="list-style-type: none"> 1.a O fluxo básico do caso de uso “CDU016 – Adicionar disciplina à turma” é realizado. 1.b O funcionário retorna ao passo 3. <p>(A2) Alternativa ao passo 4 – Erro ao armazenar os dados no banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram cadastrados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 13 – Descrição do caso de uso “Editar turma”

NOME	Editar turma (CDU013).
PRECONDIÇÕES	Ter executado o caso de uso “CDU012 – Cadastrar turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Editar turma”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O funcionário edita os campos referentes à turma. 4. O funcionário confirma os dados inseridos. 5. O sistema atualiza os dados no banco de dados. 6. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma turma cadastrada</p> <p>1.a O sistema exibe a mensagem “Nenhuma turma cadastrada” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU012 – Cadastrar turma” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU013 – Editar turma” é realizado.</p> <p>(A2) Alternativa ao passo 4 – Funcionário adiciona disciplinas à turma</p> <p>2.a O fluxo básico do caso de uso “CDU016 – Adicionar disciplina à turma” é realizado.</p> <p>2.b O funcionário retorna ao passo 4.</p> <p>(A3) Alternativa ao passo 5 – Erro ao armazenar os dados no banco de dados</p> <p>3.a O sistema informa uma mensagem de erro.</p> <p>3.b O funcionário retorna ao passo 4 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram atualizados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 14 – Descrição do caso de uso “Listar turmas”

NOME	Listar turmas (CDU014).
PRECONDIÇÕES	Ter executado o caso de uso “CDU012 – Cadastrar turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Listar turmas”. 2. O sistema exibe as turmas cadastradas no banco de dados. 3. O funcionário encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma turma cadastrada</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhuma turma cadastrada” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU012 – Cadastrar turma” é realizado. 1.c O fluxo básico do caso de uso “CDU014 – Listar turmas” é realizado. <p>(A2) Alternativa ao passo 2 – Erro de comunicação com o banco de dados.</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário encerra o caso de uso.
PÓS-CONDIÇÕES	Nenhuma.

Fonte: Autoria própria, 2016.

Quadro 15 – Descrição do caso de uso “Remover turma”

NOME	Remover turma (CDU015).
PRECONDIÇÕES	Ter executado o caso de uso “CDU012 – Cadastrar turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Remover turma”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O funcionário confirma a turma selecionada para que ela seja removida. 4. O sistema remove a turma do banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma turma cadastrada</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhuma turma cadastrada” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU012 – Cadastrar turma” é realizado. 1.c O fluxo básico do caso de uso “CDU015 – Remover turma” é realizado. <p>(A2) Alternativa ao passo 4 – Erro ao remover os dados do banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram removidos do banco de dados.

Fonte: Autoria própria, 2016.

Quadro 16 – Descrição do caso de uso “Adicionar disciplina à turma”

NOME	Adicionar disciplina à turma (CDU016).
PRECONDIÇÕES	Ter executado o caso de uso “CDU007 – Cadastrar disciplina”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona uma disciplina dentre as apresentadas. 2. O funcionário pressiona o botão adicionar. 3. O funcionário retorna ao passo 1 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 1 – Nenhuma disciplina cadastrada</p> <p>1.a O caso de uso é encerrado.</p> <p>(A2) Alternativa ao passo 2 – Funcionário remove disciplinas da turma</p> <p>2.a O fluxo básico do caso de uso “CDU017 – Remover disciplina da turma” é realizado.</p> <p>2.b O funcionário retorna ao passo 1 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Acrescenta a disciplina à lista de disciplinas da turma.

Fonte: Autoria própria, 2016.

Quadro 17 – Descrição do caso de uso “Remover disciplina da turma”

NOME	Remover disciplina da turma (CDU017).
PRECONDIÇÕES	Ter executado o caso de uso “CDU016 – Adicionar disciplina à turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona uma disciplina dentre as adicionadas. 2. O funcionário pressiona o botão remover. 3. O funcionário retorna ao passo 1 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 1 – Nenhuma disciplina adicionada</p> <p>1.a O caso de uso é encerrado.</p>
PÓS-CONDIÇÕES	Remove a disciplina da lista de disciplinas da turma.

Fonte: Autoria própria, 2016.

Quadro 18 – Descrição do caso de uso “Gerenciar restrições”

NOME	Gerenciar restrições (CDU018).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Gerenciar restrições”. 2. O funcionário escolhe uma das opções apresentadas: Definir restrições (CDU019) ou Listar restrições (CDU020). 3. Um submenu referente à opção escolhida é aberto.
FLUXOS ALTERNATIVOS	(A1) Alternativa ao passo 2 – Nenhuma opção selecionada 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe o submenu referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 19 – Descrição do caso de uso “Definir restrições”

NOME	Definir restrições (CDU019).
PRECONDIÇÕES	Ter realizado o fluxo básico do caso de uso “CDU018 – Gerenciar restrições”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Definir restrições”. 2. O funcionário escolhe uma das opções apresentadas: Definir indisponibilidades do professor (CDU021) ou Habilitar/desabilitar restrições de uma turma (CDU022). 3. Uma janela referente à opção escolhida é aberta.
FLUXOS ALTERNATIVOS	(A1) Alternativa ao passo 2 – Nenhuma opção selecionada 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe a janela referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 20 – Descrição do caso de uso “Listar restrições”

NOME	Listar restrições (CDU020).
PRECONDIÇÕES	Ter realizado o fluxo básico do caso de uso “CDU018 – Gerenciar restrições”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Listar restrições”. 2. O funcionário escolhe uma das opções apresentadas: Exibir indisponibilidades de um professor (CDU023) ou Exibir restrições de uma turma (CDU024). 3. Uma janela referente à opção escolhida é aberta.
FLUXOS ALTERNATIVOS	(A1) Alternativa ao passo 2 – Nenhuma opção selecionada 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe a janela referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 21 – Descrição do caso de uso “Definir indisponibilidades do professor”

NOME	Definir indisponibilidades do professor (CDU021).
PRECONDIÇÕES	Ter executado o caso de uso “CDU002 – Cadastrar professor”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Definir indisponibilidades do professor”. 2. O funcionário seleciona um professor dentre os apresentados. 3. O funcionário marca as indisponibilidades do professor nas tabelas de horários apresentadas. 4. O funcionário confirma os dados inseridos. 5. O sistema atualiza os dados no banco de dados. 6. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum professor cadastrado</p> <p>1.a O sistema exibe a mensagem “Nenhum professor cadastrado” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU002 – Cadastrar professor” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU021 – Definir indisponibilidades do professor” é realizado.</p> <p>(A2) Alternativa ao passo 5 – Erro ao armazenar os dados no banco de dados</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário retorna ao passo 4 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram atualizados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 22 – Descrição do caso de uso “Habilitar/desabilitar restrições de uma turma”

NOME	Habilitar/desabilitar restrições de uma turma (CDU022).
PRECONDIÇÕES	Ter executado o caso de uso “CDU012 – Cadastrar turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Habilitar/desabilitar restrições de uma turma”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O funcionário marca as restrições da turma nos campos apresentados. 4. O funcionário confirma os dados inseridos. 5. O sistema atualiza os dados no banco de dados. 6. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma turma cadastrada</p> <p>1.a O sistema exibe a mensagem “Nenhuma turma cadastrada” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU012 – Cadastrar turma” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU022 – Habilitar/desabilitar restrições de uma turma” é realizado.</p> <p>(A2) Alternativa ao passo 5 – Erro ao armazenar os dados no banco de dados</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário retorna ao passo 4 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram atualizados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 23 – Descrição do caso de uso “Exibir indisponibilidades de um professor”

NOME	Exibir indisponibilidades de um professor (CDU023).
PRECONDIÇÕES	Ter executado o caso de uso “CDU002 – Cadastrar professor”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Exibir indisponibilidades do professor”. 2. O funcionário seleciona um professor dentre os apresentados. 3. O sistema exibe as restrições do professor selecionado. 4. O funcionário encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum professor cadastrado</p> <p>1.a O sistema exibe a mensagem “Nenhum professor cadastrado” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU002 – Cadastrar professor” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU023 – Exibir indisponibilidades de um professor” é realizado.</p> <p>(A2) Alternativa ao passo 2 – Erro de comunicação com o banco de dados.</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Nenhuma.

Fonte: Autoria própria, 2016.

Quadro 24 – Descrição do caso de uso “Exibir restrições de uma turma”

NOME	Exibir restrições de uma turma (CDU024).
PRECONDIÇÕES	Ter executado o caso de uso “CDU012 – Cadastrar turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Exibir restrições de uma turma”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O sistema exibe as restrições da turma selecionada. 4. O funcionário encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma turma cadastrada</p> <p>1.a O sistema exibe a mensagem “Nenhuma turma cadastrada” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU012 – Cadastrar turma” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU024 – Exibir restrições de uma turma” é realizado.</p> <p>(A2) Alternativa ao passo 2 – Erro de comunicação com o banco de dados.</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Nenhuma.

Fonte: Autoria própria, 2016.

Quadro 25 – Descrição do caso de uso “Gerenciar aulas conjuntas”

NOME	Gerenciar aulas conjuntas (CDU025).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Gerenciar aulas conjuntas”. 2. O funcionário escolhe uma das opções apresentadas: Adicionar aula conjunta (CDU026), Editar aula conjunta (CDU027), Listar aulas conjuntas (CDU028) ou Remover aulas conjuntas (CDU029). 3. Uma janela referente à opção escolhida é aberta.
FLUXOS ALTERNATIVOS	(A1) Alternativa ao passo 2 – Nenhuma opção selecionada <ol style="list-style-type: none"> 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe a janela referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 26 – Descrição do caso de uso “Adicionar aula conjunta”

NOME	Adicionar aula conjunta (CDU026).
PRECONDIÇÕES	Ter feito o fluxo básico do caso de uso “CDU016 – Adicionar disciplina à turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Adicionar aula conjunta”. 2. O funcionário seleciona uma turma. 3. O funcionário seleciona uma disciplina da turma selecionada. 4. O funcionário seleciona uma turma que compartilhará o mesmo horário da disciplina selecionada. 5. O funcionário confirma os dados inseridos. 6. O sistema armazena os dados no banco de dados. 7. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	(A1) Alternativa ao passo 2 – Nenhuma turma com disciplinas cadastradas <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhuma turma possui uma disciplina cadastrada” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU016 – Adicionar disciplina à turma” é realizado. 1.c O fluxo básico do caso de uso “CDU026 – Adicionar aula conjunta” é realizado. (A2) Alternativa ao passo 6 – Erro ao armazenar os dados no banco de dados <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 5 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram atualizados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 27 – Descrição do caso de uso “Editar aula conjunta”

NOME	Editar aula conjunta (CDU027).
PRECONDIÇÕES	Ter executado o caso de uso “CDU026 – Adicionar aula conjunta”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Editar aula conjunta”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O funcionário seleciona uma disciplina que possua uma aula conjunta cadastrada. 4. O funcionário seleciona uma nova turma que possuirá uma aula conjunta com essa disciplina. 5. O funcionário confirma os dados inseridos. 6. O sistema atualiza os dados no banco de dados. 7. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma aula conjunta cadastrada</p> <p>1.a O sistema exibe a mensagem “Nenhuma aula conjunta cadastrada” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU026 – Adicionar aula conjunta” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU027 – Editar aula conjunta” é realizado.</p> <p>(A2) Alternativa ao passo 6 – Erro ao armazenar os dados no banco de dados</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário retorna ao passo 5 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram atualizados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 28 – Descrição do caso de uso “Listar aulas conjuntas”

NOME	Listar aulas conjuntas (CDU028).
PRECONDIÇÕES	Ter executado o caso de uso “CDU026 – Adicionar aula conjunta”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Listar aulas conjuntas”. 2. O sistema exibe as aulas conjuntas cadastradas no banco de dados. 3. O funcionário encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma aula conjunta cadastrada</p> <p>1.a O sistema exibe a mensagem “Nenhuma aula conjunta cadastrada” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU026 – Adicionar aula conjunta” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU028 – Listar aulas conjuntas” é realizado.</p> <p>(A2) Alternativa ao passo 2 – Erro de comunicação com o banco de dados.</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Nenhuma.

Fonte: Autoria própria, 2016.

Quadro 29 – Descrição do caso de uso “Remover aulas conjuntas”

NOME	Remover aulas conjuntas (CDU029).
PRECONDIÇÕES	Ter executado o caso de uso “CDU026 – Adicionar aula conjunta”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Remover aulas conjuntas”. 2. O funcionário seleciona uma aula conjunta dentre as apresentadas. 3. O funcionário confirma a aula conjunta selecionada para que ela seja removida. 4. O sistema remove a aula conjunta do banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma aula conjunta cadastrada</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhuma aula conjunta cadastrada” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU026 – Adicionar aula conjunta” é realizado. 1.c O fluxo básico do caso de uso “CDU029 – Remover aulas conjuntas” é realizado. <p>(A2) Alternativa ao passo 4 – Erro ao remover os dados do banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram removidos do banco de dados.

Fonte: Autoria própria, 2016.

Quadro 30 – Descrição do caso de uso “Gerenciar horários”

NOME	Gerenciar horários (CDU030).
PRECONDIÇÕES	Nenhuma.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção de menu “Gerenciar horários”. 2. O funcionário escolhe uma das opções apresentadas: Gerar horário por curso (CDU031), Gerar horário por turma (CDU032), Exibir horário (CDU033) ou Remover horário (CDU034). 3. Uma janela referente à opção escolhida é aberta.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma opção selecionada</p> <ol style="list-style-type: none"> 1.a O usuário não seleciona uma opção. 1.b Encerra o caso de uso.
PÓS-CONDIÇÕES	Exibe a janela referente à opção escolhida.

Fonte: Autoria própria, 2016.

Quadro 31 – Descrição do caso de uso “Gerar horário por curso”

NOME	Gerar horário por curso (CDU031).
PRECONDIÇÕES	Ter executado o caso de uso “CDU012 – Cadastrar turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Gerar horário por curso”. 2. O funcionário seleciona um nível de ensino dentre os apresentados. 3. O funcionário seleciona o curso dentre os apresentados. 4. O funcionário seleciona as turmas dentre as apresentadas. 5. O funcionário confirma os dados inseridos. 6. O sistema cadastra os dados no banco de dados. 7. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma turma cadastrada</p> <p>1.a O sistema exibe a mensagem “Nenhuma turma cadastrada” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU012 – Cadastrar turma” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU031 – Gerar horário por curso” é realizado.</p> <p>(A2) Alternativa ao passo 6 – Erro ao armazenar os dados no banco de dados</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário retorna ao passo 5 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram cadastrados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 32 – Descrição do caso de uso “Gerar horário por turma”

NOME	Gerar horário por turma (CDU032).
PRECONDIÇÕES	Ter executado o caso de uso “CDU012 – Cadastrar turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Gerar horário por turma”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O funcionário seleciona a opção de gerar para construir o horário da turma selecionada. 4. O funcionário confirma o horário gerado. 5. O sistema cadastra os dados no banco de dados. 6. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhuma turma cadastrada</p> <p>1.a O sistema exibe a mensagem “Nenhuma turma cadastrada” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU012 – Cadastrar turma” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU032 – Gerar horário por turma” é realizado.</p> <p>(A2) Alternativa ao passo 5 – Erro ao armazenar os dados no banco de dados</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário retorna ao passo 4 ou encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram cadastrados no banco de dados.

Fonte: Autoria própria, 2016.

Quadro 33 – Descrição do caso de uso “Exibir horário”

NOME	Exibir horário (CDU033).
PRECONDIÇÕES	Ter executado o caso de uso “CDU031 – Gerar horário por curso” ou o caso de uso “CDU032 – Gerar horário por turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Exibir horário”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O sistema exibe o horário da turma selecionada. 4. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum horário cadastrado</p> <p>1.a O sistema exibe a mensagem “Nenhum horário cadastrado” e encerra o caso de uso.</p> <p>1.b O fluxo básico do caso de uso “CDU031 – Gerar horário por curso” ou do caso de uso “CDU032 – Gerar horário por turma” é realizado.</p> <p>1.c O fluxo básico do caso de uso “CDU033 – Exibir horário” é realizado.</p> <p>(A2) Alternativa ao passo 2 – Erro de comunicação com o banco de dados.</p> <p>2.a O sistema informa uma mensagem de erro.</p> <p>2.b O funcionário encerra o caso de uso.</p>
PÓS-CONDIÇÕES	Nenhuma.

Fonte: Autoria própria, 2016.

Quadro 34 – Descrição do caso de uso “Remover horário”

NOME	Remover horário (CDU034).
PRECONDIÇÕES	Ter executado o caso de uso “CDU031 – Gerar horário por curso” ou o caso de uso “CDU032 – Gerar horário por turma”.
FLUXO BÁSICO	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção “Remover horário”. 2. O funcionário seleciona uma turma dentre as apresentadas. 3. O funcionário confirma a turma selecionada para que seu horário seja removido. 4. O sistema remove o horário da turma do banco de dados. 5. O funcionário retorna ao passo 2 ou encerra o caso de uso.
FLUXOS ALTERNATIVOS	<p>(A1) Alternativa ao passo 2 – Nenhum horário cadastrado</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “Nenhum horário cadastrado” e encerra o caso de uso. 1.b O fluxo básico do caso de uso “CDU031 – Gerar horário por curso” ou do caso de uso “CDU032 – Gerar horário por turma” é realizado. 1.c O fluxo básico do caso de uso “CDU034 – Remover horário” é realizado. <p>(A2) Alternativa ao passo 4 – Erro ao remover os dados do banco de dados</p> <ol style="list-style-type: none"> 2.a O sistema informa uma mensagem de erro. 2.b O funcionário retorna ao passo 3 ou encerra o caso de uso.
PÓS-CONDIÇÕES	Uma mensagem é exibida informando que os dados foram removidos do banco de dados.

Fonte: Autoria própria, 2016.

APÊNDICE B – Documento de requisitos

1 INTRODUÇÃO

Este documento apresenta os requisitos necessários para desenvolvimento da ferramenta computacional de geração de horários escolares obtidos nas fases iniciais do processo de engenharia de requisitos.

Os requisitos foram separados em funcionais, aqueles relacionados diretamente às funcionalidades do sistema, e não-funcionais, referentes à qualidade do software.

O objetivo deste documento é descrever os requisitos e suas características como prioridade, tipo e dependências. A prioridade se refere à essencialidade do requisito em relação ao funcionamento principal da ferramenta computacional, o tipo informa se o requisito é ou não funcional e o campo de dependências descreve se há ou não uma pré-condição para realização do requisito em questão.

2 DESCRIÇÃO GERAL DAS FUNCIONALIDADES DO SISTEMA

O sistema proposto tem como funcionalidades permitir cadastrar, editar, listar e remover professores, disciplinas, turmas, aulas conjuntas e restrições relacionadas aos professores e às turmas. A principal característica da ferramenta é, a partir das turmas cadastradas, gerar, exibir e remover horários escolares utilizando algoritmos genéticos, de forma que as restrições impostas sejam respeitadas.

3 ESPECIFICAÇÃO DE REQUISITOS

As especificações dos requisitos funcionais da ferramenta computacional estão dispostos nos quadros 35 a 60. Os não-funcionais foram descritos nos quadros 61 a 64.

Quadro 35 – Especificação do requisito “Cadastrar professor”

Nome do requisito	Cadastrar professor
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF001
Dependências	Nenhuma

Fonte: Autoria própria, 2016.

Quadro 36 – Especificação do requisito “Editar professor”

Nome do requisito	Editar professor
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF002
Dependências	RF001

Fonte: Autoria própria, 2016.

Quadro 37 – Especificação do requisito “Listar professores”

Nome do requisito	Listar professores
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF003
Dependências	RF001

Fonte: Autoria própria, 2016.

Quadro 38 – Especificação do requisito “Remover professor”

Nome do requisito	Remover professor
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF004
Dependências	RF001

Fonte: Autoria própria, 2016.

Quadro 39 – Especificação do requisito “Cadastrar disciplina”

Nome do requisito	Cadastrar disciplina
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF005
Dependências	RF001

Fonte: Autoria própria, 2016.

Quadro 40 – Especificação do requisito “Editar disciplina”

Nome do requisito	Editar disciplina
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF006
Dependências	RF005

Fonte: Autoria própria, 2016.

Quadro 41 – Especificação do requisito “Listar disciplinas”

Nome do requisito	Listar disciplinas
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF007
Dependências	RF005

Fonte: Autoria própria, 2016.

Quadro 42 – Especificação do requisito “Remover disciplina”

Nome do requisito	Remover disciplina
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF008
Dependências	RF005

Fonte: Autoria própria, 2016.

Quadro 43 – Especificação do requisito “Cadastrar turma”

Nome do requisito	Cadastrar turma
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF009
Dependências	Nenhuma

Fonte: Autoria própria, 2016.

Quadro 44 – Especificação de requisito “Editar turma”

Nome do requisito	Editar turma
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF010
Dependências	RF009

Fonte: Autoria própria, 2016.

Quadro 45 – Especificação do requisito “Listar turmas”

Nome do requisito	Listar turmas
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF011
Dependências	RF009

Fonte: Autoria própria, 2016.

Quadro 46 – Especificação do requisito “Remover turma”

Nome do requisito	Remover turma
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF012
Dependências	RF009

Fonte: Autoria própria, 2016.

Quadro 47 – Especificação do requisito “Adicionar disciplina à turma”

Nome do requisito	Adicionar disciplina à turma
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF013
Dependências	RF005

Fonte: Autoria própria, 2016.

Quadro 48 – Especificação do requisito “Remover disciplina da turma”

Nome do requisito	Remover disciplina da turma
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF014
Dependências	RF013

Fonte: Autoria própria, 2016.

Quadro 49 – Especificação do requisito “Definir indisponibilidades do professor”

Nome do requisito	Definir indisponibilidades do professor
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF015
Dependências	RF001

Fonte: Autoria própria, 2016.

Quadro 50 – Especificação do requisito “Habilitar/desabilitar restrições de uma turma”

Nome do requisito	Habilitar/desabilitar restrições de uma turma
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF016
Dependências	RF009

Fonte: Autoria própria, 2016.

Quadro 51 – Especificação do requisito “Exibir indisponibilidades de um professor”

Nome do requisito	Exibir indisponibilidades de um professor
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF017
Dependências	RF001

Fonte: Autoria própria, 2016.

Quadro 52 – Especificação do requisito “Exibir restrições de uma turma”

Nome do requisito	Exibir restrições de uma turma
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF018
Dependências	RF009

Fonte: Autoria própria, 2016.

Quadro 53 – Especificação do requisito “Adicionar aula conjunta”

Nome do requisito	Adicionar aula conjunta
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF019
Dependências	RF013

Fonte: Autoria própria, 2016.

Quadro 54 – Especificação do requisito “Editar aula conjunta”

Nome do requisito	Editar aula conjunta
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF020
Dependências	RF019

Fonte: Autoria própria, 2016.

Quadro 55 – Especificação do requisito “Listar aulas conjuntas”

Nome do requisito	Listar aulas conjuntas
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF021
Dependências	RF019

Fonte: Autoria própria, 2016.

Quadro 56 – Especificação do requisito “Remover aulas conjuntas”

Nome do requisito	Remover aulas conjuntas
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF022
Dependências	RF019

Fonte: Autoria própria, 2016.

Quadro 57 – Especificação do requisito “Gerar horário por curso”

Nome do requisito	Gerar horário por curso
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF023
Dependências	RF009

Fonte: Autoria própria, 2016.

Quadro 58 – Especificação do requisito “Gerar horário por turma”

Nome do requisito	Gerar horário por turma
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF024
Dependências	RF009

Fonte: Autoria própria, 2016.

Quadro 59 – Especificação do requisito “Exibir horário”

Nome do requisito	Exibir horário
Prioridade	Essencial
Tipo	Requisito funcional
Código	RF025
Dependências	RF023 ou RF024

Fonte: Autoria própria, 2016.

Quadro 60 – Especificação do requisito “Remover horário”

Nome do requisito	Remover horário
Prioridade	Não essencial
Tipo	Requisito funcional
Código	RF026
Dependências	RF023 ou RF024

Fonte: Autoria própria, 2016.

Quadro 61 – Especificação do requisito “Usabilidade”

Nome do requisito	Usabilidade
Prioridade	Não essencial
Tipo	Requisito não-funcional
Código	RNF001

Fonte: Autoria própria, 2016.

Quadro 62 – Especificação do requisito “Confiabilidade”

Nome do requisito	Confiabilidade
Prioridade	Essencial
Tipo	Requisito não-funcional
Código	RNF002

Fonte: Autoria própria, 2016.

Quadro 63 – Especificação do requisito “Portabilidade”

Nome do requisito	Portabilidade
Prioridade	Essencial
Tipo	Requisito não-funcional
Código	RNF003

Fonte: Autoria própria, 2016.

Quadro 64 – Especificação do requisito “Funcionalidade”

Nome do requisito	Funcionalidade
Prioridade	Essencial
Tipo	Requisito não-funcional
Código	RNF004

Fonte: Autoria própria, 2016.

APÊNDICE C – Manual do usuário

1 INTRODUÇÃO

Este documento detalha as funcionalidades do software de geração de horários escolares e descreve de que forma é possível operá-lo, visando auxiliar o usuário no momento de uso do produto.

2 INSTALAÇÃO DO SOFTWARE

Não é necessário instalação do software, pois é possível executá-lo sem tal necessidade. No entanto, há dois pré-requisitos para que ele funcione adequadamente. O primeiro deles é a instalação da máquina virtual java (JVM), que pode ser encontrada no link https://www.java.com/pt_BR/. Ela é necessária para executar o software.

O segundo item necessário para operação adequada do sistema, é o MySQL Server. Ele é o Sistema de Gerenciamento de Banco de Dados (SGBD) e é necessário para armazenar e recuperar todos os dados cadastrados na ferramenta de geração de horários escolares. O MySQL Server pode ser encontrado no link <http://dev.mysql.com/downloads/mysql/>. No momento de instalação, deve-se prestar bastante atenção no login e senha informados, pois eles serão necessários para uso do software.

Após o download e instalação do JVM e do MySQL Server, pode-se executar o software normalmente, abrindo-o com o Java(™) Platform SE Binary que foi instalado com a JVM. Para que o sistema tenha acesso ao SGBD MySQL Server, será necessário informar o login e a senha que foram informados no momento de instalação do banco de dados.

3 EXECUÇÃO DO SOFTWARE

Na primeira execução do software, as únicas opções disponíveis serão a de cadastro de professores localizada sob o menu gerenciar professores e a de cadastro de turmas sob o menu gerenciar turmas. Apesar de ser possível criar uma turma, não será possível adicionar disciplinas a ela ainda, visto que nenhum componente curricular foi cadastrado ainda.

3.1 GERENCIAR PROFESSORES

Ao selecionar a opção “Cadastrar professor”, uma janela interna à principal será exibida contendo o campo “Nome completo do professor”. Após preencher o campo com o nome do professor, basta pressionar o botão confirmar para finalizar o cadastro.

Após o cadastro de um professor, será permitido editar, listar e remover professores, além de definir e exibir suas indisponibilidades localizadas sob o menu “Gerenciar restrições”. Na tela de edição, deve-se selecionar o professor que deseja-se editar na caixa de seleção e em seguida alterar as informações necessárias e confirmar os dados inseridos para salvar as alterações.

A opção “Listar professores” exibirá todos os professores cadastrados no sistema em ordem alfabética e não é possível realizar nenhuma interação com essa tela. Ela serve apenas como exibição de informações.

Na tela referente à opção “Remover professor” é apresentada apenas uma caixa de seleção onde pode-se selecionar qual professor deseja-se remover. Após realizar a escolha, basta pressionar o botão confirmar para que o professor seja removido. Atenção: a ação de remoção do professor não pode ser desfeita e irá remover também as disciplinas que ele leciona. Ao remover as disciplinas, elas serão apagadas das turmas e dos horários já gerados.

3.2 GERENCIAR DISCIPLINAS

Após cadastrar pelo menos um professor, será possível selecionar a opção “Cadastrar disciplina”. Nela será necessário que os dados nome da disciplina e quantidade de aulas semanais que a disciplina possui sejam informados, além de ser preciso que o professor que leciona a disciplina seja selecionado. Em seguida, deve-se clicar no botão confirmar para finalizar o cadastro da disciplina.

Após o cadastro de uma disciplina, será permitido editar, listar e remover disciplinas. Na tela de edição, deve-se selecionar a disciplina que deseja-se editar na caixa de seleção e em seguida alterar as informações necessárias e confirmar os dados inseridos para salvar as alterações.

A opção “Listar disciplinas” exibirá todas as disciplinas cadastradas no sistema em ordem alfabética e não é possível realizar nenhuma interação com essa tela. Ela serve apenas como exibição de informações.

Na tela referente à opção “Remover disciplina” é apresentada apenas uma caixa de seleção onde pode-se selecionar qual disciplina deseja-se remover. Após realizar a escolha, basta pressionar o botão confirmar para que a disciplina seja removida. Atenção: a ação de remoção da disciplina não pode ser desfeita e ao removê-la, ela será apagada das turmas e dos horários já gerados.

3.3 GERENCIAR TURMAS

Ao selecionar a opção “Cadastrar turma”, uma janela interna à principal será exibida contendo os campos “Nome do curso”, “Nível de ensino”, “Turma”, “Série/Semestre” e “Turno da turma”. Após preencher os campos, basta pressionar o botão confirmar para finalizar o cadastro. O campo “Turma” serve para diferenciar turmas que contenham valores idênticos para os demais campos. Em caso de a turma ser única, pode-se selecionar qualquer valor. Recomenda-se, no entanto, que o valor “A” seja selecionado, pois se novas turmas surgirem, basta adicioná-las com os demais valores subsequentes.

Na tela de cadastro de turmas, também é possível adicionar disciplinas à turma, caso alguma já tenha sido cadastrada anteriormente. Essa ação é opcional e pode ser acessada novamente pelo menu de edição de turmas. Se desejar remover uma disciplina que foi adicionada, deve-se selecioná-la e clicar no botão remover para apagá-la da lista.

Após o cadastro de uma turma, será permitido editar, listar e remover turmas, além de definir e exibir suas restrições. Na tela de edição, deve-se selecionar a turma que deseja-se editar na caixa de seleção e em seguida alterar as informações necessárias e confirmar os dados inseridos para salvar as alterações.

A opção “Listar turmas” exibirá todas as disciplinas cadastradas no sistema e não é possível realizar nenhuma interação com essa tela. Ela serve apenas como exibição de informações.

Na tela referente à opção “Remover turma” é apresentada apenas uma caixa de seleção onde pode-se selecionar qual turma deseja-se remover. Após realizar a escolha, basta pressionar o botão confirmar para que a turma seja removida. Atenção: a ação de remoção da turma não pode ser desfeita e irá remover também as aulas conjuntas e o horário dela.

3.4 GERENCIAR RESTRIÇÕES

No menu “Gerenciar restrições” há dois submenus relacionados à definição e à listagem de restrições, que serão liberadas apenas após o cadastro de um professor e de uma turma. Em “Definir restrições”, há duas opções: uma é a “Definir indisponibilidades do professor”, e a outra é “Habilitar/desabilitar restrições de uma turma”. Já em “Listar restrições”, pode-se “Exibir indisponibilidades de um professor” ou “Exibir restrições de uma turma”.

Na janela da opção “Definir indisponibilidades do professor”, deve-se escolher um professor dentre os apresentados na caixa de seleção e marcar nos quadros representando os horários, somente as indisponibilidades do professor. Após terminar de definir as restrições dos professores, é necessário clicar no botão confirmar para armazenar as alterações realizadas.

Na tela “Habilitar/desabilitar restrições de uma turma”, deve-se escolher uma turma dentre as apresentadas na caixa de seleção e informar as suas restrições. Para o janelamento entre disciplinas, há duas opções disponíveis, uma permitir e outra não permitir, e uma delas deve ser selecionada. Nos dias da semana, deve-se marcar somente os dias disponíveis para aula. E entre as aulas geminadas, é possível habilitar (todas as aulas deverão ser duplas), desabilitar (todas as aulas deverão ser simples) ou desconsiderar (não dar relevância para o arranjo de aulas, elas podem ser tanto duplas, quanto simples, podendo ocorrer os dois casos em um mesmo horário). Após definição dos valores, deve-se confirmar os dados marcados para que eles sejam armazenados.

Ao selecionar a opção de “Exibir indisponibilidades de um professor”, deve-se escolher um professor na caixa de seleção e clicar em confirmar para que suas indisponibilidades sejam exibidas nos quadros de horários.

A última opção possível do gerenciamento de restrições é a de “Exibir restrições de uma turma”. Nela é necessário selecionar uma turma dentre as exibidas na caixa de seleção e clicar em selecionar. As restrições da turma serão exibidas de forma similar ao da janela de “Habilitar/desabilitar restrições de uma turma”.

3.5 GERENCIAR AULAS CONJUNTAS

Após cadastrar pelo menos duas turmas, será possível adicionar aulas conjuntas entre duas ou mais delas, selecionando a opção “Adicionar aula conjunta”. Nela, será necessário selecionar uma das turmas. Ao escolher, as disciplinas dessa turma serão exibidas na caixa de seleção “Disciplina conjunta”. Deve-se então selecionar a disciplina desejada e as possíveis turmas conjuntas serão exibidas na caixa de seleção “Turma conjunta”. Ao selecionar a turma conjunta desejada, pode-se confirmar os dados informados pressionando o botão confirmar.

Depois de cadastrar uma aula conjunta, as opções editar, listar e remover aulas conjuntas ficarão disponíveis. Em “Editar aula conjunta”, a mesma sequência de ações de “Adicionar aula conjunta” deve ser feita. No entanto, ao selecionar a turma, aquelas disciplinas da turma selecionada que foram cadastradas em “Adicionar aula conjunta” serão exibidas na caixa de seleção “Disciplina conjunta”. Deve-se então selecionar a disciplina desejada e as possíveis turmas conjuntas serão exibidas na caixa de seleção “Turma conjunta”. Ao selecionar a turma conjunta desejada, pode-se confirmar os dados informados pressionando o botão confirmar.

A opção “Listar aulas conjuntas” exibirá todas as aulas conjuntas entre turmas cadastradas no sistema e não é possível realizar nenhuma interação com essa tela. Ela serve apenas como exibição de informações.

Na tela referente à opção “Remover aula conjunta” é apresentada uma caixa de seleção onde pode-se selecionar de qual turma deseja-se remover alguma disciplina conjunta. Após realizar a escolha e clicar em selecionar, serão exibidas na caixa de seleção “Disciplina”, as disciplinas que essa turma possui em aulas conjuntas. Basta escolher a que deseja-se separar das demais turmas e pressionar o botão confirmar para que a disciplina deixe de ser conjunta. Atenção: a ação de remoção da disciplina conjunta não pode ser desfeita.

3.6 GERENCIAR HORÁRIOS

Após cadastrar pelo menos uma turma, será possível gerar o horário através de duas opções: “Gerar horário por curso” e “Gerar horário por turma”. Em “Gerar horário por curso”, será necessário selecionar o nível de ensino a partir da caixa de seleção. Se o nível for fundamental ou médio, o próximo passo será pressionar o botão gerar para que o horário de todas as turmas do nível selecionado seja gerado.

Caso seja nível superior, os cursos disponíveis serão exibidos na caixa de seleção “Curso”. Ao selecioná-lo, deve-se adicionar à lista, quais turmas da caixa de seleção localizada em “Turmas do curso” deseja-se gerar o horário. Caso deseje remover uma turma da lista, deve-se clicar nela e pressionar o botão remover. Depois de ter adicionado todas as turmas desejadas à lista, deve-se clicar em gerar para concluir. Atenção: ao gerar horários utilizando essa opção, todos os horários das turmas selecionadas que foram cadastrados anteriormente serão removidos.

Na opção “Gerar horário por turma”, deve-se apenas escolher qual turma dentre as apresentadas na caixa de seleção “Turma” deseja-se obter um horário. Em seguida, deve-se pressionar o botão gerar para que um horário seja gerado respeitando as restrições definidas anteriormente. Caso a turma já possua um horário definido, uma mensagem de aviso será exibida informando que deve-se remover o horário da turma primeiro para que um novo seja gerado corretamente. Será perguntado se deseja ou não continuar. Selecionando a opção Sim, será possível gerar o horário, porém será evitado que as disciplinas sejam posicionadas nos mesmos pontos que no horário anterior. Se desejar eliminar essa característica, é necessário selecionar a opção “Remover horário” localizado sob o menu “Gerenciar horários” e remover o horário da turma antes de gerar um novo. Caso selecione a opção Não, o programa será interrompido e nada ocorrerá. Após o software encontrar um horário adequado, ele será exibido no quadro de horário. Se estiver satisfeito com ele, pode-se cadastrá-lo pressionando o botão confirmar. Senão, pode-se clicar no botão gerar para tentar encontrar um horário mais adequado.

Após gerar pelo menos um horário, será possível utilizar as funções “Exibir horário” e “Remover horário”. Em “Exibir horário”, deve-se escolher de qual turma deseja que o horário seja exibido e clicar em selecionar. O horário será exibido no quadro e abaixo dele será apresentada uma lista com as disciplinas da turma e os professores que as lecionam.

Em “Remover horário”, deve-se selecionar de qual turma deseja que o horário seja removido e pressionar o botão confirmar. Atenção: essa ação não pode ser desfeita.