

UNIVERSIDADE SAGRADO CORAÇÃO

GABRIEL CAMPOS FERRAREZI

**DESENVOLVIMENTO DE UM SISTEMA
INTELIGENTE DE GERAÇÃO DE AVALIAÇÕES
UTILIZANDO ALGORITMOS GENÉTICOS**

BAURU
2015

GABRIEL CAMPOS FERRAREZI

**DESENVOLVIMENTO DE UM SISTEMA
INTELIGENTE DE GERAÇÃO DE AVALIAÇÕES
UTILIZANDO ALGORITMOS GENÉTICOS**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para a obtenção do título de bacharel em Ciência da Computação sob a orientação do Prof. Me. Patrick Pedreira Silva.

BAURU
2015

F374d	<p>Ferrarezi, Gabriel Campos</p> <p>Desenvolvimento de um sistema inteligente de geração de avaliações utilizando algoritmos genéticos / Gabriel Campos Ferrarezi. -- 2015. 60f. : il.</p> <p>Orientador: Prof. Me. Patrick Pedreira Silva.</p> <p>Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.</p> <p>1. Educação. 2. Inteligência artificial. 3. Algoritmos genéticos. I. Silva, Patrick Pedreira. II. Título.</p>
-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GABRIEL CAMPOS FERRAREZI

**DESENVOLVIMENTO DE UM SISTEMA INTELIGENTE DE
GERAÇÃO DE AVALIAÇÕES UTILIZANDO ALGORITMOS
GENÉTICOS**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação sob a orientação do Prof. Me. Patrick Pedreira Silva.

Banca examinadora:

Prof. Me. Patrick Pedreira Silva
Universidade Sagrado Coração

Prof. Me. Henrique Pachioni Martins
Universidade Sagrado Coração

Prof. Dr. Elvio Gilberto da Silva
Universidade Sagrado Coração

Bauru, 16 de novembro de 2015.

Dedico este trabalho a meus pais, meus amigos, meu irmão e minha namorada.

AGRADECIMENTOS

Agradeço aos meus pais, por me darem esta oportunidade de estudo, a meus amigos que me ajudaram a me manter focado no projeto e principalmente a minha namorada, por todo seu apoio e cooperação durante todo o desenvolvimento deste projeto.

RESUMO

A educação é uma área de grande importância para o desenvolvimento econômico do país, devido a grande demanda de novas formas de ensino, práticas e funcionais, viu-se necessária a introdução de outras áreas para facilitar o ensino, sendo uma destas áreas a da Tecnologia de Informação (TI), pois possui um grande potencial e aplicação abrangente. Dentro da TI, existe uma área conhecida por Inteligência Artificial (IA), que consiste em técnicas que simulam uma inteligência. Para testar o potencial da IA como forma de aprendizado, será desenvolvido um sistema de avaliação inteligente, onde, utilizando técnicas de Algoritmos Genéticos (AG), que simulam um sistema que se baseia na teoria da seleção natural, onde a partir de uma população é possível criar novos indivíduos que possuem melhor aptidão em resolver o problema, ou seja, para criar a melhor avaliação para o usuário, conforme informações passadas por provas diagnósticas. Este sistema permite criar avaliações, calcando-se numa base de perguntas da área a ser melhorada. Obtendo provas equilibradas, tanto nos pontos fortes como nos fracos, de caráter formativo com o perfil do usuário, sendo este atualizado ao finalizar cada avaliação para verificação dos conhecimentos.

Palavras-chave: Educação. Inteligência Artificial. Algoritmos Genéticos.

ABSTRACT

Education is a very important area for economic development of a country, due to high demand for new forms of teaching, practical and functional, it was necessary to introduce other areas to facilitate the teaching, one of these areas is Information Technology (IT), due its great potential and broad application. Within the IT there is an area known for Artificial Intelligence (AI), which consists of techniques that simulate intelligence. To test the potential of AI as a way of learning, an intelligent testing system will be developed, in which, using genetic algorithm techniques (AG), will simulate a system that is based on the theory of natural selection, where from a population it can create new individuals with better fitness in solving the problem, for instance, to create the best test for the user using information passed by diagnostic tests. This system allows creating tests, treading on a base of questions in the area to be improved. Getting balanced tests, with the strengths and the weaknesses, of pedagogical aspect within the user profile, which is updated at the end of each test as a check to the knowledge.

Keywords: Education. Artificial Intelligence. Genetic Algorithms.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo em cascata.	21
Figura 2 – Desenvolvimento incremental.	22
Figura 3 – Estrutura básica de um Algoritmo Genético.	28
Figura 4 – Algoritmo Genético simples.	29
Figura 5 – Exemplos de Genótipos e Fenótipos e seus respectivos problemas.	31
Figura 6 – Operador de cruzamento de um ponto.	35
Figura 7 – Operador de cruzamento multiponto.	35
Figura 8 – Operador de cruzamento uniforme.	36
Figura 9 – Exemplo do uso do operador de mutação.	37
Figura 10 – Exemplo da avaliação gerada pelo sistema a ser desenvolvido.	39
Figura 11 – Banco de dados com suas tabelas e funções.	40
Figura 12 – Modelagem do banco de dados.	41
Figura 13 – Exemplo de tabela junto de sua <i>sequence</i>	41
Figura 14 – Projeto VBP.	42
Figura 15 – Referências.	42
Figura 16 – Módulo de conexão.	43
Figura 17 – Tela de login.	43
Figura 18 – Função de criação de perfis.	44
Figura 19 – Tela da prova Início.	45
Figura 20 – Tela da prova Fim.	45
Figura 21 - Diagrama do funcionamento do sistema.	46

SUMÁRIO

1 INTRODUÇÃO.....	10
2 OBJETIVOS	12
2.1 OBJETIVO GERAL	12
2.2 OBJETIVOS ESPECÍFICOS	12
3 AVALIAÇÃO	13
3.1 AVALIAÇÃO COMO MEIO DE ENSINO	13
3.2 TIPOS DE AVALIAÇÃO	14
3.2.1 Avaliação Diagnóstica	14
3.2.2 Avaliação Formativa	15
3.2.3 Avaliação Somativa	15
3.2.4 Avaliação Interna	16
3.2.5 Avaliação Externa	17
3.3 ELABORAÇÃO DA AVALIAÇÃO	18
4 ENGENHARIA DE SOFTWARE	20
4.1 MODELOS DE PROCESSO DE SOFTWARE	21
4.1.1 Modelo Cascata	21
4.1.2 Desenvolvimento Incremental	22
5 INTELIGÊNCIA ARTIFICIAL	24
5.1 AGIR HUMANAMENTE	24
5.2 PENSAR HUMANAMENTE	24
5.3 PENSAR RACIONALMENTE	25
5.4 AGIR RACIONALMENTE	25
6 SELEÇÃO NATURAL E ALGORITMOS GENÉTICOS	26
6.1 SELEÇÃO NATURAL	26
6.2 ALGORITMOS GENÉTICOS	27
6.2.1 Indivíduos	30
6.2.2 População	31
6.2.3 Inicialização e Função de Avaliação	32
6.2.4 Seleção	32
6.2.5 Reprodução	34
6.2.6 Mutação	36
6.2.7 Atualização	37
6.2.8 Finalização	37
7 METODOLOGIA	38

8 DESENVOLVIMENTO	40
9 RESULTADOS	47
10 CONSIDERAÇÕES FINAIS	53
REFERÊNCIAS.....	54

1 INTRODUÇÃO

Atualmente, com a globalização e as novas tecnologias da informação e comunicação envolvendo a internet, o mundo vive em um processo onde as fronteiras desaparecem e as barreiras físicas deixam de ser um problema. Todos os segmentos e nichos são afetados pelo processo de integração. Aquilo que parecia distante tornou-se rotina na vida das pessoas, principalmente para as empresas. Não existe mais a necessidade da presença física para comercializar produtos ou serviços, nem mais a disponibilidade de viajar ao exterior para fechar negócios. A sofisticação tecnológica aproximou o cliente internacional, e com isso também mudou o perfil dos profissionais. (PILATTI; SANTOS, 2008).

Este novo contexto exige que o profissional tenha a capacidade de comunicação em nível internacional, se mantenha sempre atualizado e tenha competência para lidar com os novos desafios que surgem na busca pela conquista de espaço no mercado de trabalho. É necessário que ele tenha percepção e atente para aquilo que alavancará o sucesso. A importância que o mercado de trabalho dá à contratação de pessoas que tenham fluência em mais de um idioma, principalmente o inglês, para se comunicar com esses novos clientes de forma eficiente se amplia significativamente. (PILATTI; SANTOS, 2008).

Com essa eminente necessidade de aprendizado de outros idiomas, surge também o uso de novas formas e ferramentas de testar e avaliar o conhecimento que o indivíduo tem dos mesmos, porém é necessário que tal avaliação seja feita de forma automática e inteligente, poupando tempo e mantendo um controle dinâmico de qualidade. Portanto, é preciso desenvolver uma ferramenta que consiga atingir tais objetivos. Neste contexto, a Computação e, mais especificamente, a área de Inteligência Artificial (IA) pode dar grandes contribuições.

Segundo Russel e Norvig (2004), a Inteligência Artificial pode ser utilizada em inúmeros campos, indo de algo geral, como aprendizado e percepção, até algo mais específico, como xadrez, realizar contas matemáticas, escrever poesia ou até mesmo dirigir um automóvel tendo uma rota calculada, a IA pode ser utilizada em qualquer área intelectual. Dentro da IA, uma subárea que tem ganhado destaque na solução de problemas de aprendizado é a dos Algoritmos Genéticos.

De acordo com Luger (2004), os algoritmos genéticos implementam uma forma poderosa de subida de encosta que mantém múltiplas soluções, eliminando por sua vez as soluções menos promissoras e melhorando as demais, dessa forma é possível chegar na melhor proposta após algumas gerações e por sua vez no resultado esperado.

Considerando esse potencial dos algoritmos genéticos na solução de problemas que necessitem de alguma forma de otimização, esta investigação propõe utilizá-los como mecanismo de otimização no contexto de um sistema inteligente de elaboração automática de avaliações, especificamente avaliações que visam testar os conhecimentos do usuário na língua inglesa.

Com a utilização de algoritmos genéticos, é possível, a partir de uma base de perguntas pré-determinadas, chegar à melhor avaliação (prova) que atenda adequadamente as necessidades do usuário, como identificar e trabalhar conscientemente suas fraquezas e também mensurar e definir suas potencialidades, no que se refere aos conhecimentos de língua inglesa.

2 OBJETIVOS

Tendo em vista as necessidades na área da educação, este trabalho visa demonstrar o potencial dos algoritmos genéticos em resolver o problema descrito.

2.1 OBJETIVO GERAL

Desenvolver um sistema de avaliação inteligente utilizando algoritmos genéticos para criar avaliações (provas) para o perfil do usuário.

2.2 OBJETIVOS ESPECÍFICOS

- a) Realizar um levantamento bibliográfico sobre teoria dos algoritmos genéticos;
- b) Desenvolver um sistema utilizando a linguagem de programação Visual Basic 6, por abranger maior domínio na linguagem;
- c) Montar uma base de perguntas sobre língua inglesa;
- d) Utilizar o banco de dados PostgreSQL para armazenamento de perguntas e resultados;
- e) Definir a melhor forma de escolher perguntas para a geração de avaliações considerando a teoria dos algoritmos genéticos;
- f) Selecionar uma base de perguntas pré-determinadas do devido assunto a ser utilizada na geração das provas;
- g) Testar diferentes configurações para os algoritmos genéticos, definindo a melhor configuração;
- h) Armazenar os resultados das avaliações para verificação do comportamento dos algoritmos genéticos no processo de geração de provas;
- i) Testar o sistema para verificação da eficiência no processo de geração das avaliações.

3 AVALIAÇÃO

Segundo Ishimoto (2006) “é através da avaliação diagnóstica e formativa que o aluno conhece seus avanços e necessidades, encontrando estímulo para um estudo sistemático”.

[...] O processo avaliativo vem sendo discutido e repensado por teóricos preocupados com a proposição de instrumentos e de estratégias que possibilitem compreender melhor as experiências de aprendizagem vivenciadas pelo aluno. Estes estudos são importantes no referente às mudanças que a escola precisa sofrer para poder acompanhar as transformações que vêm ocorrendo na sociedade atual. (MORAES, 2011, p. 235).

As provas, dentro da abordagem avaliação, são ferramentas que surgiram da necessidade de dar à avaliação um caráter mais científico e objetivo, como maneira mais adequada de medir os resultados da aprendizagem, pois a mesma possibilita que os resultados sejam quantificados e assim analisados. (QUINQUER, 2003 apud MORAES, 2011, p. 235).

Sendo assim, uma avaliação, indiferente do tema, tem como função validar os conhecimentos do usuário, assim possibilitando que suas forças e fraquezas sejam armazenadas em dados, a partir dos quais é possível verificar em quais conteúdos o aluno deve melhorar.

3.1 AVALIAÇÃO COMO MEIO DE ENSINO

Dentro das muitas aplicações das avaliações, temos o meio de ensino a partir de ferramentas como as provas – que são realizadas com o único intuito de testar os conhecimentos do usuário, para então descobrir onde o mesmo deve melhorar – porém,

[...] Muitos docentes veem o ensino como mera transmissão e memorização de informações prontas, e o aluno [...] como um ser receptivo. Em consequência, a avaliação do ensino superior normalmente se restringe a medir a quantidade de informações retidas, e não a qualidade. O importante é estabelecer um diagnóstico correto para cada aluno e identificar as possíveis causas de seu fracasso e/ou dificuldades visando uma maior qualificação na sua aprendizagem. (ISHIMOTO, 2006).

Segundo Moraes (2011), uma avaliação pode e deve assumir a função de colaborar com o trabalho pedagógico, propiciando informações claras e precisas sobre o conhecimento e desenvolvimento do aluno. Com base nestas informações, é possível saber os pontos fortes e fracos de cada indivíduo em um devido assunto, informação crucial para a consecução de uma segunda avaliação, com caráter mais formativo, ou seja, “[...] as informações serão recolhidas

através de provas aplicadas nas unidades de ensino, e as atividades de recuperação serão dadas com base nos resultados, alcance dos objetivos apresentados.” (DEPRESBITERIS, 1991 apud MORAES, 2011, p. 239).

3.2 TIPOS DE AVALIAÇÃO

Segundo o Centro de Políticas Públicas e Avaliação da Educação (CAEd) existem vários tipos de avaliação, normalmente divididos em cinco grupos dispostos a seguir.

3.2.1 Avaliação Diagnóstica

Apesar de não ter uma definição clara pelos especialistas, a Avaliação Diagnóstica apresenta condição e função bem reconhecidas quanto à sua aplicabilidade. Trata-se de uma ação avaliativa realizada no início do processo de aprendizagem, buscando os conhecimentos, informações e competências já adquiridos pelos estudantes, como base para a organização das ações de ensino e aprendizagem futuras, de acordo com os resultados adquiridos. (AVALIAÇÃO..., c2014a).

Especificamente, o objetivo deste tipo de avaliação é a identificação das características de aprendizagem do estudante, buscando a forma de trabalho mais adequada a essas características. Tanto pontos fortes quanto fracos de cada aluno são colocados em evidência, gerando um “ponto de entrada” preciso para dar sequência à aprendizagem com o modo de ensino mais indicado para a situação. É um tipo interessante de avaliação, pois previne a detecção tardia de possíveis dificuldades de aprendizagem dos alunos, assim como também busca identificar as aptidões, capacidades, competências e interesses dos mesmos como pré-requisitos para ações de ensino consecutivas. (AVALIAÇÃO..., c2014a).

Logo, a Avaliação Diagnóstica é de ordem preventiva, pois conhecer as dificuldades dos alunos no início do processo educativo permite prever e atender suas reais necessidades. Tem também como característica a possibilidade de determinar as causas de dificuldades persistentes em alguns alunos. (AVALIAÇÃO..., c2014a).

Todos os dados recolhidos neste tipo de avaliação servem para auxiliar as redes de ensino a planejar estratégias iniciais, explorar, identificar e adaptar de acordo com as competências dos alunos, sugerindo procedimentos que ajudem os mesmos a atingirem novos níveis de conhecimento. (AVALIAÇÃO..., c2014a).

3.2.2 Avaliação Formativa

Similar à Avaliação Diagnóstica, a Avaliação Formativa funciona como uma ferramenta do processo ensino-aprendizagem. Não apresenta finalidade probatória e está incorporada no ato de ensinar e formar. Apesar de alguns autores considerarem a Avaliação Formativa como uma integração de outras modalidades de avaliação, uma vez que ela é aplicada durante o processo educacional, seu caráter é claro e especificamente pedagógico. (AVALIAÇÃO..., c2014b).

Dentre seus objetivos, pode-se citar sua intenção em melhorar o processo de ensino-aprendizagem através das informações coletadas durante a ação avaliativa. Ela pretende detectar as possíveis dificuldades que surgem no processo de aprendizagem e em seguida corrigi-las com agilidade. No entanto, seu maior foco reside na coleta de dados, informações e desenvolvimento do aluno e fornecimento dos mesmos ao professor, permitindo que a prática docente se ajuste às necessidades dos estudantes durante o processo. (AVALIAÇÃO..., c2014b).

Como importante característica da Avaliação Formativa, temos a capacidade de evidenciar com agilidade informações úteis sobre as etapas superadas e obstáculos encontrados pelos alunos durante o processo. Consiste em um rápido e contínuo *feedback* para o professor. Este tipo de avaliação permite meios para encontrar soluções de problemas que tenham surgido ao longo do trabalho feito com o aluno. Trata-se de um aspecto claro da Avaliação Formativa aproximar, possibilitar o conhecimento mútuo e o diálogo entre aluno e professor. (AVALIAÇÃO..., c2014b).

Todos os resultados encontrados por meio da Avaliação Formativa servem como base para a identificação da forma que o processo de aprendizagem acontece. Através das informações reveladas, é possível planejar, ajustar e redirecionar práticas pedagógicas buscando a melhoria da absorção de conteúdo dos estudantes. Ainda, os resultados desta avaliação funcionam para apoiar, reforçar, compreender, facilitar, e equilibrar as competências e aprendizagens dos alunos. (AVALIAÇÃO..., c2014b).

3.2.3 Avaliação Somativa

A Avaliação Somativa consiste numa modalidade avaliativa pontual que é aplicada no final do período letivo. Busca a determinação do grau de conteúdo dominado e retido pelos estudantes, considerando objetivos pré-estabelecidos para finalmente realizar um balanço

somatório de um ou mais trabalhos de formação. Também é conhecida como Avaliação das Aprendizagens. (AVALIAÇÃO..., c2014c).

Pode se afirmar que o principal objetivo da Avaliação Somativa se encontra nos resultados das aprendizagens para a visualização do balanço somatório de um ou mais trabalhos realizados com os alunos. As aprendizagens dos alunos são assim sintetizadas tendo por base critérios gerais. (AVALIAÇÃO..., c2014c).

A principal característica dessa modalidade de avaliação reside na capacidade de informar, classificar e situar o avaliado, evidenciando a conclusão, considerando que ocorre no final de um processo educacional. (AVALIAÇÃO..., c2014c).

As informações sintetizadas da Avaliação Somativa são destinadas ao registro e à publicação daquilo que a princípio demonstra ter sido assimilado pelos estudantes. Estes resultados funcionam para verificar, situar, classificar, informar e por fim, certificar. (AVALIAÇÃO..., c2014c).

3.2.4 Avaliação Interna

A Avaliação Interna é realizada pelo professor em sala de aula para verificar a aprendizagem dos alunos. Os resultados do processo de ensino e aprendizagem tornam-se explícitos por meio de uma avaliação intencional e sistemática onde o professor tem a liberdade de recorrer a variados instrumentos avaliativos. (AVALIAÇÃO..., c2014d).

Através da Avaliação Interna é possível para o professor verificar a forma que o processo de ensino e aprendizagem tem acontecido durante as aulas. Ela fornece informações específicas que refletem a realidade dos alunos e a eficácia do trabalho do professor. (AVALIAÇÃO..., c2014d).

A principal característica da Avaliação Interna consiste no universo interno da sala de aula. Quando aplicada, esta avaliação demonstra o esforço do professor em buscar resultados do seu próprio trabalho congruentes ao desempenho dos seus alunos. Suas formas de aplicação são múltiplas e variadas podendo ocorrer por meio de provas abertas ou objetivas, observação e registro, auto avaliação, dentre outras. (AVALIAÇÃO..., c2014d).

Os resultados demonstrados pela Avaliação Interna fornecem dados cruciais para os professores avançarem em seus métodos e práticas pedagógicas, ou retrocederem à determinada etapa que possa ter criado dificuldades para os estudantes, e assim, sanarem o problema. A Avaliação Interna também possibilita a discussão e planejamento de ações específicas para cada caso envolvendo o desempenho dos alunos. (AVALIAÇÃO..., c2014d).

3.2.5 Avaliação Externa

A Avaliação Externa também é conhecida como a Avaliação de Larga Escala, e consiste num instrumento importante para a elaboração de políticas públicas dos sistemas de ensino, assim como para o redirecionamento de metas para unidades escolares. Busca avaliar o desempenho de escolas, gerando resultados que funcionam como medidas de proficiência, permitindo aos gestores programarem políticas públicas, e também como reflexos do trabalho das unidades escolares. Como curiosidade, “a primeira iniciativa brasileira de avaliação em larga escala foi o Sistema Nacional de Avaliação da Educação Básica (Saeb) que se desenvolveu a partir de 1990 e foi aplicado inicialmente em 1995”. (AVALIAÇÃO..., c2014e).

Hoje, devido à necessidade de conhecimento das diretrizes específicas acerca das suas próprias realidades, cada Estado busca o desenvolvimento de sistemas adaptados e adequados de avaliação. (AVALIAÇÃO..., c2014e).

De forma geral, a Avaliação Externa funciona como uma forma de garantir e assegurar a qualidade na Educação. Isso reflete na cobrança de qualidade nas unidades escolares. As informações e resultados adquiridos revelam o panorama de desempenho educacional, além de retratar a realidade geral de ensino. (AVALIAÇÃO..., c2014e).

Como tipos de Avaliação Externa classificam-se as censitárias ou amostrais. Dentre suas características está a avaliação das redes e sistemas de ensino fora da sala de aula. Como requisitos na sua elaboração destacam-se metodologia e instrumentos específicos de análise, desta forma permitindo a manutenção da comparabilidade e credibilidade dos resultados adquiridos. A fim de atingir estes objetivos, os testes são construídos padronizados e seus resultados classificados numa escala de proficiência variando de zero a 500 em intervalos de 25 pontos. Estes intervalos são índices de consolidação de competências e habilidades durante o ensino e aprendizagem. (AVALIAÇÃO..., c2014e).

A importância dos resultados das Avaliações em Larga Escala está no fornecimento de subsídios para decisões e ações que busquem aprimorar o ensino nas escolas. Através destes resultados também é possível monitorar a evolução e desenvolvimento das redes e sistemas de ensino por meio de comparações de diferentes edições de testes do tipo. Assim, torna-se possível criar indicadores nacionais como o IDEB (Índice de Desenvolvimento da Educação Básica), além de conhecer o percentual de alunos em cada nível da escala de proficiência. (AVALIAÇÃO..., c2014e).

3.3 ELABORAÇÃO DA AVALIAÇÃO

Tendo em vista que a avaliação tem papel fundamental no aprendizado, a sua elaboração é de extrema importância, pois é necessário que as perguntas sejam passadas de forma objetiva, explicando o que você quer do usuário, evitando perguntas cujo sentido pode ser duplo ou fora de contexto.

A partir dos estudos feitos por Moraes (2011, p.244), ao se iniciar a construção de uma prova devem ser feitos vários questionamentos, o que quero com esta questão? Qual é a aprendizagem que o aluno deve demonstrar? Quanto essa questão possibilita perceber o que o aluno aprendeu e as dificuldades que ele apresenta que o impedem de chegar à resposta correta? São esses questionamentos que ajudam o avaliador a saber o que se deseja alcançar, ou seja, seu objetivo, assim evitando questões sem relevância para o momento e desperdiçando a oportunidade de obter melhores informações sobre o aprendizado do aluno.

Ainda seguindo os estudos de Moraes (2011, p.245):

[...] A elaboração das questões torna-se um fator de grande importância, pois é a partir de boas perguntas que os alunos terão condições de dar boas respostas. Pensar em perguntas contextualizadas e bem escritas são fatores imprescindíveis se o objetivo for uma prova formativa. (MORAES, 2011, p. 245).

De acordo com Haydt (1997) e Moretto (2003) apud Moraes (2011), existem alguns passos que podem ser seguidos para ajudar a na formação das questões:

1. Todas as questões devem ter um objetivo claro.
2. As questões devem ser elaboradas com perguntas de relevância, evitando equívocos oriundos de interpretações dúbias. Devem ainda induzir operações mentais, que sejam adequadas ao formato de ensino adotado, assim como ao nível dos alunos. As questões devem ter um equilíbrio de dificuldade, evitando efeitos tanto de desânimo como de ansiedade.
3. A questão deve ser contextualizada, dentro de um âmbito que compreensão, evitando exigir apenas memorização mecânica.
4. O uso da linguagem deve estar adaptado, tornando-a clara, precisa e contextualizada, desta forma o aluno saberá especificamente o que está sendo solicitado.
5. Todas as atividades devem ser similares àquelas realizadas em aula, com o uso das mesmas estratégias metodológicas anteriormente planejadas.
6. A prova deve ser previamente planejada.

7. Definir o estilo da questão segundo a série e conteúdo.
8. Todas as instruções devem estar claras e dúvidas esclarecidas.
9. Vocabulário simples e acessível.
10. Palavras que devem ser evitadas: sempre, invariavelmente, jamais, nunca.
11. Negativas devem ser evitadas.
12. Somente informar na questão dados estritamente necessários para a solução.

4 ENGENHARIA DE SOFTWARE

De acordo com Pressman (2011), a engenharia de software abrange um processo, um conjunto de métodos e de ferramentas que possibilitam aos profissionais desenvolverem desenvolverem um software de alta qualidade e é dividida em sete categorias:

1. Software de sistema: conjunto de programas feito para atender a outros programas.
2. Software de aplicação: programas sob medida que solucionam uma necessidade específica.
3. Software científico/de engenharia: programas cujo finalidade é o processamento numérico pesado.
4. Software embutido: sistema utilizado para implementar e controlar funções e características para o usuário final.
5. Software para linha de produtos: sistemas cuja finalidade é prover uma capacidade específica de utilização por muitos usuário diferentes.
6. Aplicações para web: programas centralizados no uso da rede.
7. Software de inteligência artificial: utiliza de algoritmos não numéricos para solucionar problemas complexos.

Pressman (2011) ainda diz que a essência da pratica de engenharia de software está em quatro pontos, compreender o problema, planejar uma solução, executar o plano e examinar o resultado para ter precisão.

De acordo com Hooker (1996) apud Pressman (2011) existem sete princípios que se concentram na pratica da engenharia de software como um todo, eles são:

1. A razão de existir.
2. Faça de forma simples.
3. Mantenha a visão.
4. O que um produz outros consomem.
5. Esteja aberto para o futuro.
6. Planeje com antecedência, visando reutilização.
7. Pense!

4.1 MODELOS DE PROCESSO DE SOFTWARE

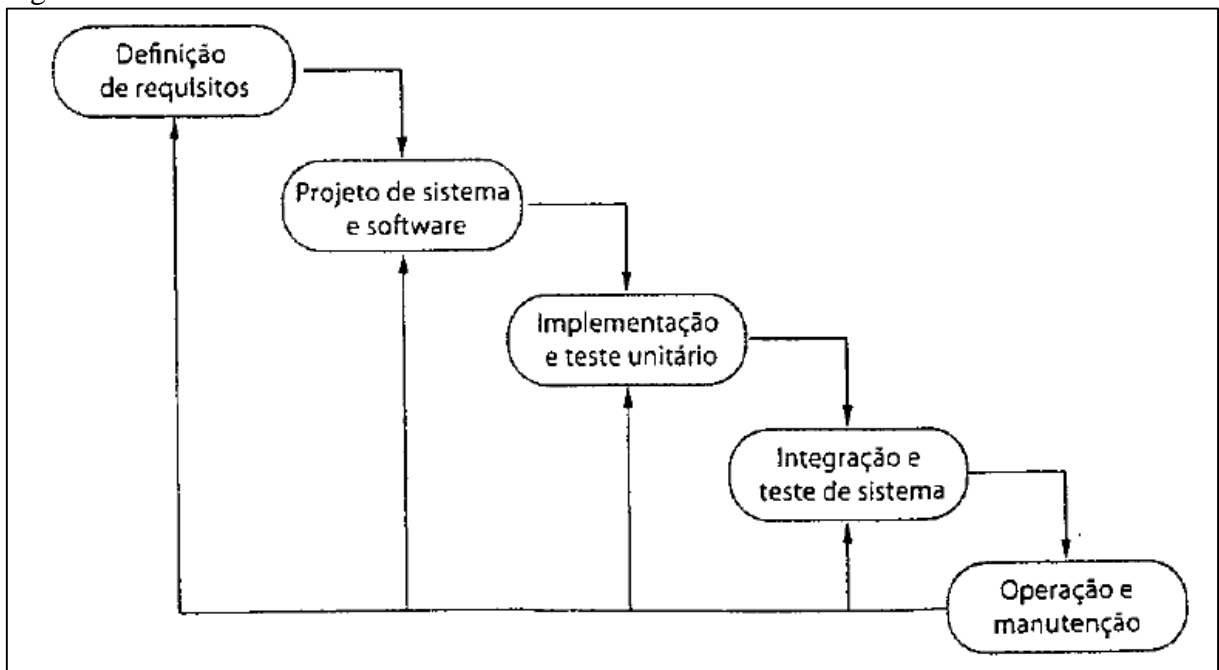
Sommerville (2012) aborda alguns modelos genéricos de processamento de software, entre eles o modelo cascata e o desenvolvimento incremental.

4.1.1 Modelo Cascata

Segundo Sommerville (2012), esse modelo considera as atividades fundamentais do processo de especificação, desenvolvimento, validação e evolução, e representa cada uma delas distintamente, como: especificação de requisitos, projeto de software, implementação, teste e assim por diante.

Este modelo, ilustrado na Figura 1, tem os principais estágios refletindo diretamente nas atividades fundamentais do desenvolvimento e tem como princípio planejar e programar todas as atividades do processo antes de inicia-las.

Figura 1 – Modelo em cascata.



Fonte: Sommerville (2012, p. 20).

- Análise e definição de requisitos: Serviços, restrições e metas do sistema devem ser estabelecidas por meio de consultas aos usuários;
- Projeto de sistema e software: O projeto de sistema aloca os requisitos, tanto para sistemas de hardware como para sistemas de software por meio de definição de uma arquitetura geral do sistema. O projeto de software envolve

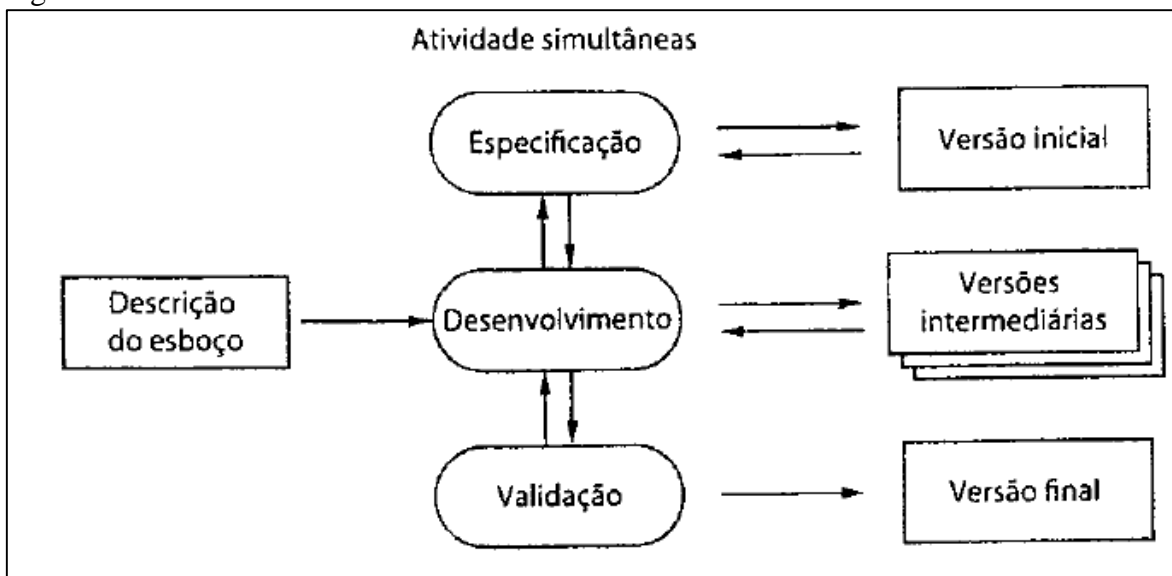
identificação e descrição das abstrações fundamentais do sistema de software e seus relacionamentos;

- Implementação e teste unitário: Ocorre a implementação do software como um conjunto de programas ou unidade de programa e é realizado o teste unitário verificando se cada unidade atende a sua especificação;
- Integração e teste de sistema: As unidades individuais são testadas como um todo para assegurar que os requisitos do software tenham sido atendidos, após isso o sistema é entregue ao cliente;
- Operação e manutenção: O sistema é instalado e colocado em uso. Aonde a manutenção envolve a correção de erros que não foram encontrados em estágios iniciais do ciclo, com melhora das unidades e ampliação de seus serviços em resposta da descoberta de novos requisitos.

4.1.2 Desenvolvimento Incremental

De acordo com Sommerville (2012), o desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expondo-a aos comentários dos usuários e continuando por meio da criação de várias versões até que um sistema adequado seja desenvolvido, sendo que as atividades de desenvolvimento, especificação e validação são intercalados, segundo a Figura 2.

Figura 2 – Desenvolvimento incremental.



Fonte: Sommerville (2012, p. 22).

O desenvolvimento incremental é uma abordagem mais ágil, melhor para sistemas de negócio e sistemas pessoais pois reflete a maneira que resolvemos problemas, movendo passo a passo em direção a uma solução e recuando quando percebemos que cometemos algum erro.

5 INTELIGÊNCIA ARTIFICIAL

De acordo com Luger (2004), um dos principais desencadeadores que proporcionou o estudo da Inteligência Artificial foi devido aos seres humanos pararem de interpretar as coisas como elas parecem ser, e sim como elas realmente são.

De acordo com Russel e Norvig (2004), existem quatro propostas dentro da IA que vem sendo seguidas: as preocupadas com processos de pensamento e raciocínio; as que endereçam a comportamento; as que medem o sucesso em termos de fidelidade para o desempenho humano; e as que medem contra um conceito ideal de inteligência. Ou seja, as propostas estão sempre centradas ao redor de humanos ou da racionalidade, agir ou pensar, onde uma proposta centrada em humanos deve ser uma ciência empírica, envolvendo hipóteses e confirmações experimentais e uma proposta racionalista envolve uma combinação de matemática e engenharia.

5.1 AGIR HUMANAMENTE

A proposta do Teste de Turing, por Alan Turing (1950), trata de prover uma definição satisfatória de inteligência, colocando um computador a prova, junto de um humano interrogador e que após algumas questões o humano não saiba dizer se as respostas são de uma pessoa ou não, tal proposta atinge o fator de agir humanamente. Tal proposta pode ser utilizada no problema em questão, pois com ela, o usuário passa a ter a sensação que outra pessoa está fazendo sua avaliação, e não uma máquina. (RUSSEL; NORVIG, 2004).

5.2 PENSAR HUMANAMENTE

Dentro da proposta de pensar humanamente temos a modelagem cognitiva. Para dizer que um programa pensa como um humano, precisamos de alguma forma para determinar como humanos pensam, ou seja, precisamos entender o funcionamento da mente humana. Existem duas formas de fazer isto: através de introspecção, tentando pegar nossos próprios pensamentos na medida em que eles surgem, ou através de experimentos psicológicos. (RUSSEL; NORVIG, 2004).

5.3 PENSAR RACIONALMENTE

O filósofo grego Aristóteles foi um dos primeiros a tentar codificar o “pensamento correto”, ou seja, pensar racionalmente. Utilizando o conceito de silogismo que provê padrões para estruturas de argumentos que sempre produzem conclusões corretas quando dadas corretas premissas, por exemplo, “Sócrates é um homem; Todos os homens são mortais; Logo, Sócrates é mortal”. Estas leis de pensamento eram supostas por governar a operação da mente; seu estudo iniciou o campo chamado de lógica. (RUSSEL; NORVIG, 2004).

5.4 AGIR RACIONALMENTE

Para que um programa atenda a esta proposta, um agente racional pode ser utilizado. Os agentes devem ter atributos que distinguem eles de meros programas, tal como operando sobre controle autônomo, percebendo seu ambiente, persistindo sobre um prolongado período de tempo, se adaptando às mudanças e sendo capaz de pensar em outros objetivos. Um agente racional é um que age de forma a alcançar o melhor resultado ou, quando em incerteza, o melhor resultado esperado. (RUSSEL; NORVIG, 2004).

Junto com a proposta de agir humanamente existe outra que poderia ser utilizada no problema em questão, sendo que a partir do pensamento lógico que deve ser feito com base nas regras propostas na elaboração da avaliação e nas informações coletadas do usuário, consiga produzir os melhores resultados. (RUSSEL; NORVIG, 2004).

6 SELEÇÃO NATURAL E ALGORITMOS GENÉTICOS

Segundo Lucas (2002, p. 5), os algoritmos genéticos utilizam conceitos do princípio da seleção natural, propostos por Darwin em 1858, sendo robustos, genéricos, e adaptáveis, eles são utilizados para uma ampla gama de problemas e em inúmeras áreas.

Linden (2008) define algoritmos genéticos como um ramo dos algoritmos evolucionários, que são modelos computacionais dos processos naturais de evolução, portanto podem ser definidos como uma técnica de busca que faz referência ao processo natural da evolução.

Buscando esclarecer um pouco mais a denotação de algoritmo genético, é possível dizer que se trata de uma variante de busca em feixe estocástica, ou seja, nestes estados que sucedem são gerados através da combinação de um par de estados pais, e não pela modificação de um estado específico. (RUSSEL; NORVIG, 2004).

6.1 SELEÇÃO NATURAL

De acordo com Lucas (2002), o neodarwinismo possui preceitos básicos com relação ao processo evolutivo das espécies sendo eles listados abaixo:

1. Indivíduos tendem a disputar continuamente pelos recursos limitados presente no meio ambiente.
2. Dentre todos os indivíduos presentes em um determinado meio, alguns possuem características especiais que permitem uma maior probabilidade de sobrevivência, tais indivíduos são ditos como mais adaptados ao ambiente.
3. Indivíduos mais adaptados possuem melhor chance de sobreviver, e conseqüentemente de reprodução.
4. Durante o processo de reprodução, um grande número de características é repassado à próxima geração, sendo assim, indivíduos que se reproduzem mais, tendem a repassar suas características com mais frequência às próximas gerações.
5. Características mais desejáveis tendem a se propagar na espécie ao longo do processo evolutivo, aumentando a adaptabilidade da espécie como um todo.
6. O processo reprodutivo possui falhas, portanto durante a replicação e transmissão dos genes aos novos indivíduos o fenômeno conhecido como mutação pode ocorrer. Tal fenômeno, normalmente, é prejudicial ao indivíduo,

mas em alguns casos pode incorporar a ele uma característica desejável que não foi herdada do conjunto de genes de seus pais, fazendo com que a natureza adquira a capacidade de explorar um número infinito de combinações possíveis.

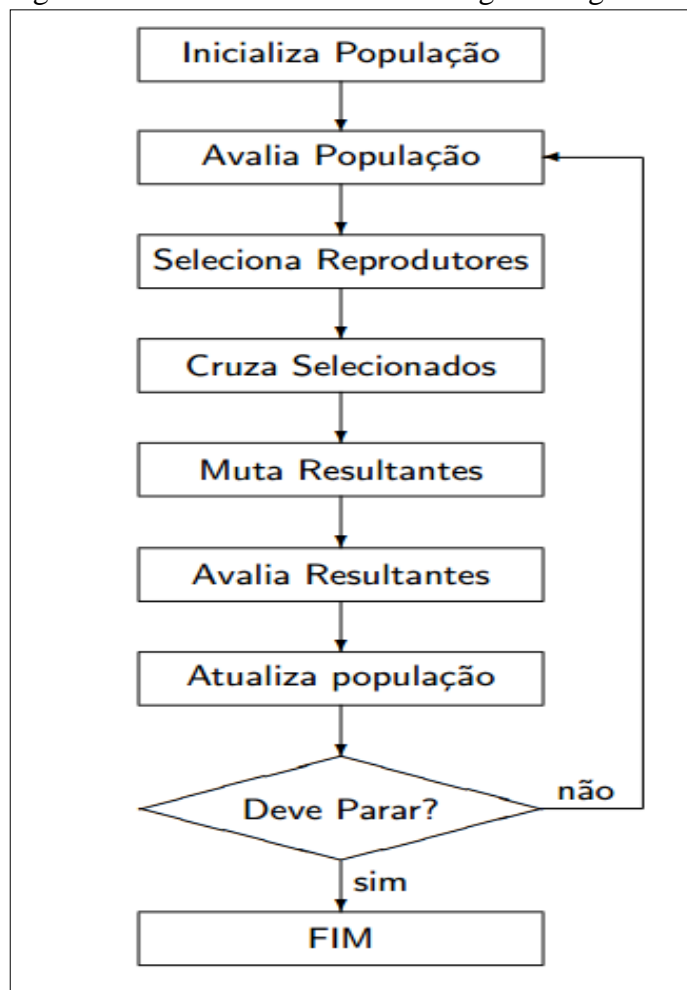
6.2 ALGORITMOS GENÉTICOS

Segundo Lucas (2002), a função dos Algoritmo Genéticos (AGs) é criar uma população de possíveis respostas para o problema a ser tratado para depois inseri-la ao processo evolutivo.

Holland (1975) apud Lucas (2002) decompôs o funcionamento de um algoritmo genético em algumas etapas, conforme Figura 3, sendo elas inicialização, avaliação, seleção, cruzamento, mutação, atualização e finalização.

Inicialmente deve-se pegar uma população aleatória, após, é feita uma avaliação para verificar a aptidão dos indivíduos para o problema em questão. Feita esta validação, serão selecionados os indivíduos para reprodução e ocorre o cruzamento, onde as características dos indivíduos são recombinadas, gerando novos indivíduos onde alguns podem vir a receber uma mutação para aumentar a variedade, e então a nova geração é adicionada à população para em seguida ser feita outra validação. Se esta validação tenha resultado negativo, o processo recomeça, caso contrário, é finalizado.

Figura 3 – Estrutura básica de um Algoritmo genético.



Fonte: Lucas (2002, p. 6).

Segundo Soares (1997) um algoritmo genético simples trabalha com uma população fixa, cujas cadeias de caracteres estão binariamente codificados. Um estudo deve ser feito sobre o problema a ser otimizado, e logo em seguida, definir a quantidade de indivíduos que formará a população, sua formação cromossômica e as probabilidades de aplicação dos operadores genéticos. Após as definições, o desempenho de cada indivíduo é avaliado, e a partir do resultado haverá a seleção que determina quem poderá reproduzir. Ocorrendo o cruzamento e a mutação, a nova geração substitui a anterior, sendo um processo cíclico que irá se repetir até que o critério de convergência seja atingido, conforme demonstrado no algoritmo da Figura 4.

Figura 4 – Algoritmo genético simples.

```

Algoritmo Genético Simples {
  Definindo {
    função desempenho
    formação do indivíduo e tamanho da população
    probabilidade dos operadores
  }
  Inicializar população aleatória
  Enquanto não alcançar critério de convergência faça {
    avaliar os indivíduos da população
    executar seleção
    executar cruzamento e mutação
  }
}

```

Fonte: Soares (1997, p. 11).

Devido à maneira que os AGs agem, algumas características se destacam:

- a) Busca codificada: para que seja possível a utilização de um AG na solução de um problema, é necessário que exista um conjunto de soluções viáveis que possam ser codificadas de alguma forma em uma população de indivíduos;
- b) Generalidade: os AG simulam um dos mais fortes fatores da natureza, a adaptabilidade, sendo assim, o conhecimento da área do problema é apenas necessário para a representação do mesmo e de suas possíveis soluções, portanto um programador que utiliza AG não deve se preocupar com a forma de chegar à solução e sim com o que ela deve se parecer;
- c) Paralelismo explícito: existindo cada indivíduo de uma população como um ente isolado e avaliado de forma independente, torna-se é possível dividir uma população em grupos ou ilhas, e então evoluí-la paralelamente;
- d) Busca estocástica: ao contrário dos outros métodos de busca de valores ótimos, os algoritmos genéticos são possuem comportamento determinístico, porém sua busca não é feita de forma aleatória, uma vez que as probabilidades de aplicação dos operadores genéticos faz com que seja possível, estatisticamente, prever o resultado – porém sem exatidão absoluta;
- e) Busca cega: devido ao fato de um AG ignorar o significado das estruturas que ele manipula e as melhores formas de trabalhar com elas faz com que o mesmo não precise de um domínio específico do problema, trazendo a generalidade por um lado, mas perdendo eficiência por outro;
- f) Eficiência mediana: devido a ser um método de busca cega, os AGs tendem a apresentar um desempenho menor que outras formas de buscas heurísticas

orientadas ao problema. Para sanar tal desvantagem, considera-se a hibridização, onde heurísticas provenientes de outras buscas são utilizadas;

- g) Paralelismo implícito: durante uma busca por uma população, a evolução do algoritmo genético tende a favorecer algumas características e os indivíduos que as possuem, avaliando implicitamente determinadas combinações ou esquemas como mais ou menos desejáveis, sendo chamada de busca por hiperplanos;
- h) Facilidade no uso de restrições: os AGs facilitam a codificação de problemas com diversos tipos de restrições e graus de importância, assim, junto de suas outras características, possibilitam uma avaliação mais fiel das características desejadas para solução.

6.2.1 Indivíduos

Segundo Lucas (2002), um indivíduo é a peça de maior importância para um algoritmo genético, pois nele que são codificadas as possíveis soluções do problema a ser tratado, e através de sua manipulação, a partir do processo evolutivo, é possível chegar numa resposta.

Tratando-se da parte mais importante de todo o desenvolvimento de um algoritmo genético, pois consistem nos principais responsáveis pelo desempenho do programa, os indivíduos, também chamados de *genoma* ou *cromossoma*, possuem, assim como seus outros nomes, um conjunto de genes, ou *genótipo*, e um problema, sendo que a representação do algoritmo tem como base o genótipo, e a avaliação o fenótipo, conforme alguns exemplos listados na Figura 5.

Figura 5 – Exemplos de Genótipos e Fenótipos e seus respectivos problemas.

Genótipo	Fenótipo	Problema
0010101001110101	10869	Otimização Numérica.
CGDEHABF	Comece pela cidade C, depois passe pelas cidades G, D, E, H, A, B e termine em F.	Caixeiro Viajante.
C1R4C2R5C4R1	Se condição 1 (C1) execute regra 4 (R4), se (C2) execute (R6), se (C4) execute (R1).	Regras de aprendizado para agentes.

Fonte Lucas (2002, p. 8).

Lucas (2002) diz que as características mais importantes para um indivíduo são:

- Genótipo: é a informação que consiste a estrutura de dados que engloba os genes;
- Fenótipo: é o resultado da decodificação do genoma;
- Grau de adaptação: representa a qualidade da resposta representada pelo indivíduo na solução do problema, calculado a partir de uma função chamada objetivo, denotada por $f_o(x)$;
- Grau de aptidão: representa o nível de adaptação em relação à população à qual ele pertence. Se o grau de adaptação de um indivíduo x é $f_o(x)$, então seu grau de aptidão será

$$f_A(x) = \frac{f_o(x)}{\sum_{i=1}^n f_o(i)} \quad (1)$$

onde n é o tamanho da população

6.2.2 População

Segundo Lucas (2002), uma população, como próprio nome diz, é um conjunto de indivíduos, desta forma, um AG não trabalha apenas com uma solução e sim com vários candidatos ao resultado final, onde, através do processo evolutivo, haverá a seleção, cruzamento e mutação de tais indivíduos até que finalmente algum requisito seja satisfeito.

Igual aos indivíduos, uma população possui características próprias, envolvendo todo o grupo, sendo elas as seguintes:

- Geração: é o número de vezes que uma população passou pelo ciclo evolutivo;
- Grau de convergência: é a representação de quão próximo à média de adaptação da atual geração está de suas anteriores. Os AGs têm como objetivo fazer a população convergir a um valor ótimo de adaptação;
- Diversidade: é o grau de variação entre os genótipos presentes na população, uma ampla diversidade é essencial para a amplitude da busca;
- Elite: são os indivíduos mais adaptados da população.

6.2.3 Inicialização e Função de Avaliação

A inicialização de um AG tem como base a aleatoriedade, utilizando funções para gerar indivíduos com genes aleatórios para fornecer uma maior diversidade para as próximas gerações. (LUCAS, 2002).

Após termos uma população é necessário passa-la as funções de avaliação, onde cada problema possui uma função de avaliação diferente, sendo as funções de avaliação para um problema de resolução de melhor caminho diferentes de uma função para o problema de escalonamento. Assim os indivíduos serão testados para lidar com o problema em questão, onde são analisados os graus de aptidão e adaptação, pois a seleção depende de tais testes para saber como o indivíduo se comportará perante o problema e qual foi seu desempenho. (GOLDBERG, 1989 apud Lucas (2002)).

Lucas (2002) afirma que calcular com exatidão completa o grau de adaptação dos indivíduos é uma tarefa complexa e, devido a tal operação ser repetida inúmeras vezes durante o processo evolutivo, seu custo pode ser consideravelmente alto, portanto em tais situações o uso de funções não determinísticas é comum, pois não avaliam a totalidade das características do indivíduo e sim a amostragem delas.

6.2.4 Seleção

Após a avaliação, os indivíduos são selecionados conforme sua aptidão para a reprodução, gerando assim novos indivíduos que serão avaliados novamente pela sua aptidão e assim sucessivamente.

Segundo Dawkin (1996) apud Lucas (2002) a seleção é um processo:

1. Dirigido, a seleção ocorre de forma determinística, e não aleatória como a mutação, sendo assim um indivíduo só conseguirá sobreviver em um ambiente e se reproduzir se e somente for capaz, graças a suas características e habilidades de reagir adequadamente a todas as ocorrências do seu meio.
2. Cumulativo, devido aos benefícios adquiridos durante a seleção passarem de uma geração a próxima, possibilita, junto a não aleatoriedade da seleção, o surgimento de organismos complexos igual às formas de vida que hoje conhecemos.

De acordo com Goldberg (1989), Wall (2000) e Geyer-Schultz (1997) apud Lucas (2002), o uso de um método de escala sobre o valor de adaptação de cada indivíduo normalmente é útil por reduzir a probabilidade de convergência prematura. Diversas funções podem ser aplicadas sobre a adaptação.

No scaling (Não escalar): o valor de retorno da função objetivo é usado sem nenhum tipo de alteração.

Linear scanling (Escala linear): o valor da função objetivo $f_o(x)$ sofre a seguinte alteração:

$$f_i(f_o(x)) = f_o(x) \times a + b, \quad (2)$$

Onde,

$$a = (c - 1) \times \frac{med}{\Delta} \quad (3)$$

e

$$b = med \times \frac{max - c \times med}{\Delta}, \quad (4)$$

se $\Delta \neq 0$, ou $a = 1$ e $b = 0$ em caso contrário, *max* representa o maior valor de adaptação encontrado, *med* a média de adaptação da população e *c* é uma constante pré-definida pertencente ao conjunto dos reais e maior que 1. Valores negativos de adaptação não são aceitos por este método.

Sigma truncation scaling (Escala por truncamento sigma): um múltiplo do desvio médio (denotado por d_m) é subtraído dos valores de adaptação. Todos os valores negativos são alterados para zero. A formula para seu calculo é a seguinte:

$$f_{st}(f_o(x)) = f_o(x) - med - c \times d_m, \quad (5)$$

Onde *c* é uma constante pré-definida pertencente ao conjunto dos reais e maior que 0.

Power Law scaling (Escala pela lei da potencia): o valor de adaptação do indivíduo é elevado a uma constante pré-determinada *k*:

$$f_{pl}(f_o(x)) = f_o(x)^k \quad (6)$$

Este método não aceita valores negativos de adaptação.

6.2.5 Reprodução

Após selecionados os indivíduos em pares, ocorre o processo chamado de cruzamento (*crossover*), onde parte dos genes dos pais são combinados para geração dos filhos. (LUCAS, 2002).

Segundo Geyer-Schultz (1997) apud Lucas (2002), o pareamento de indivíduos pode ocorrer de varias formas, sendo as principais as seguintes:

- Escolha aleatória: os pares são selecionados aleatoriamente;
- *Inbreeding* (Endogamia): os indivíduos pais são combinados para gerar a nova população;
- *Line breeding* (Reprodução em linha): um indivíduo de alto desempenho é cruzado com uma subpopulação e os filhos são escolhidos como pais;
- *Outbreeding* (Exogamia): indivíduos que codificam fenótipos diferentes são combinados;
- *Self-fertilization* (Autofertilização): o indivíduo é combinado consigo mesmo;
- *Positive assortive mating* (Acasalamento combinado positivo): indivíduos semelhantes são combinados;
- *Negative assortive mating* (Acasalamento combinado negativo): indivíduos diferentes são combinados;

Segundo Lucas (2002), é possível representar o funcionamento dos operados de cruzamento como uma seleção por máscara, onde esta seria representada por um vetor cujos elementos possam assumir valores binários e que possuam um comprimento igual ao dos cromossomos a serem combinados.

Os operadores tradicionais de cruzamento mais conhecidos para genomas de comprimento fixos, segundo Goldberg (1989), são:

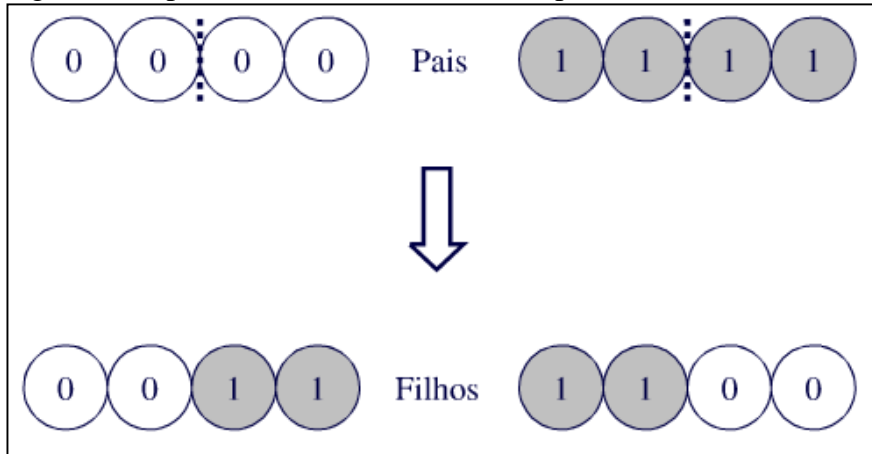
Cruzamento de um ponto (1PX): dados dois indivíduos x e y de comprimento l_g , é sorteado um número qualquer p sendo que $0 < p < l_g$, sendo que o primeiro filho f_0 receberá todos os genes de x de 1 até p e todos os genes de y de $p + 1$ até l_y e o segundo filho o inverso como visto na Figura 6.

Dentre os operadores de cruzamento tradicionais, o de ponto único é o que apresenta o pior desempenho.

Uma máscara de cruzamento seria uma série de zeros (0s), sucedidas de uma série de uns (1s) pertencente ao conjunto

$$\{0_n 1_m \mid n \geq 0, m \geq 0, n + m = l_g\}. \quad (7)$$

Figura 6 – Operador de cruzamento de um ponto.

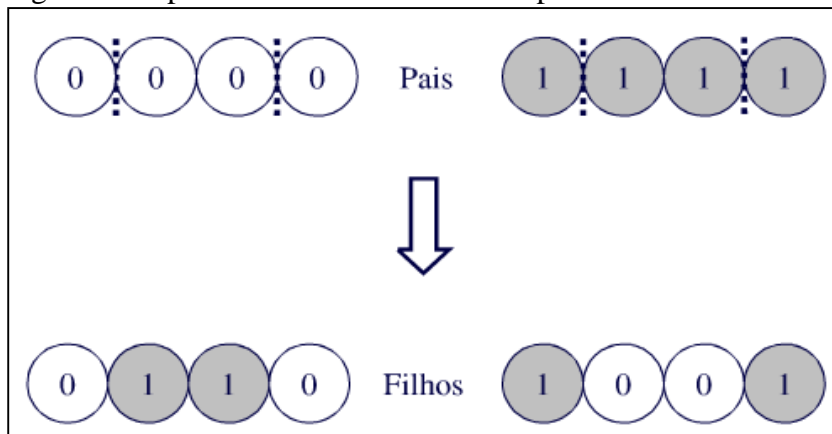


Fonte: Machado (2011).

Cruzamento multiponto (MPX): o cruzamento multiponto possui grande semelhança ao de ponto único, porém como o próprio nome diz, ele possui um número fixo n de pontos de corte. Uma operação de cruzamento com n pontos pode ser vista como um caso especial de uma $m > n$ pontos em que um ou mais $(m - n)$ pontos de corte se localizam no final do cromossomo, sendo o ponto de índice $i = l_g$.

Uma máscara para um cruzamento multiponto com n pontos apresentaria uma máscara com n mudanças em sua sequência de zeros e uns, conforme a Figura 7.

Figura 7 – Operador de cruzamento multiponto.

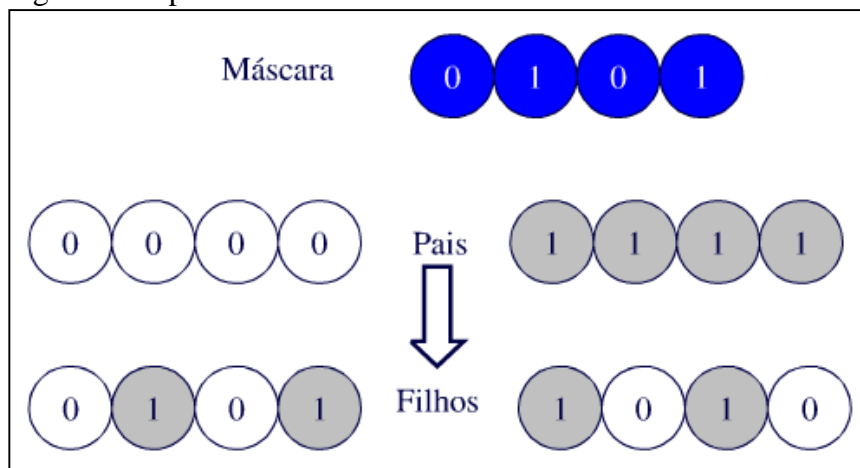


Fonte: Machado, 2011

Cruzamento segmentado (SX): semelhante ao cruzamento de multipontos, com a exceção de que o número de pontos de corte é sorteado cada vez que é executado.

Cruzamento uniforme (UX): para cada gene a ser preenchido nos cromossomos filhos, o operador sorteia de qual dos pais deve ser gerado, conforme Figura 8, sendo a máscara de tal operador numa sequência qualquer de zeros e uns.

Figura 8 – Operador de cruzamento uniforme.



Fonte: Machado, 2011

Cruzamento por combinação parcial (PMX): escolhe dois pontos de corte e faz com que os indivíduos filhos recebam por completo os genes do pai ou da mãe situados entre os pontos de corte, sendo que os da mãe vão para um filho e do pai para o outro, e então preenchendo os genes restantes com os valores considerados mais adequados para cada filho.

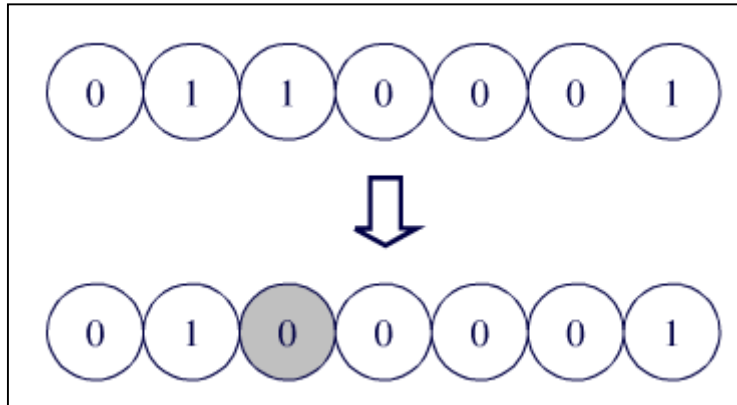
6.2.6 Mutação

Após o cruzamento, os indivíduos resultantes são submetidos ao processo de mutação, sendo que a mutação opera com uma probabilidade pré-determinada e efetuam algum tipo de alteração em sua estrutura. Tal operador tem importância devido ao fato de que uma vez bem sucedido em seu modo de atuar, garante que diversas alternativas serão exploradas, mantendo assim um nível mínimo de abrangência de busca, conforme exemplo na Figura 9, sendo os mais comuns:

- Mutação *flip*: cada gene a sofrer mutação recebe um valor sorteado do alfabeto válido;

- Muta o por troca (*swap mutation*): s o sorteados n pares de genes, e os elementos do par trocam de valor entre si;
- Muta o *creep*: um valor aleat rio   somado ou subtra do do valor do gene.

Figura 9 – Exemplo do uso do operador de muta o.



Fonte: Machado, 2011

6.2.7 Atualiza o

Depois de realizados o cruzamento e a muta o, os indiv duos resultantes s o inseridos na popula o segundo a pol tica adotada pelo algoritmo gen tico, sendo que normalmente a popula o mant m um tamanho fixo e os indiv duos s o criados em mesmo n mero que seus antecessores e os substituem por completo. Existem, por m, outras abordagens, o n mero de indiv duos gerados pode ser menor, o tamanho da popula o pode sofrer varia es e o crit rio de inser o pode ser alterado, como por exemplo, nos casos onde os filhos substituem os pais, ou que estes s o inseridos se possuem maior aptid o que o indiv duo a ser substituído, ou apenas o conjunto dos n melhores indiv duos s o mantidos. (LUCAS, 2002, p.17)

6.2.8 Finaliza o

Lucas (2002) afirma que a finaliza o n o envolve o uso de nenhum operador gen tico, ela simplesmente   composta de um teste que d  fim ao processo de evolu o caso o algoritmo gen tico tenha chegado a um ponto pr -determinado de parada, sendo estes variados, desde o n mero de gera es j  criadas at  o grau de converg ncia da popula o atual.

7 METODOLOGIA

As atividades deste trabalho de pesquisa podem ser divididas em duas etapas: fundamentação teórica e desenvolvimento de software. Na parte de fundamentação teórica (capítulos 3, 4, 5 e 6), foram abordadas teorias, formas de avaliação, princípios de engenharia de software e algoritmos necessários para o desenvolvimento do software proposto. A metodologia adotada nesta etapa consiste, basicamente, no estudo dos conceitos, teorias, métodos de avaliação e algoritmos relacionados ao tema desta pesquisa: desenvolvimento de uma ferramenta de autoavaliação utilizando algoritmos genéticos para o desenvolvimento das provas.

O desenvolvimento do projeto foi feito em Visual Basic 6 utilizando o banco de dados PostgreSQL para o armazenamento das questões, respostas, provas já realizadas, usuários, e qualquer outra informação de relevância para o desenvolvimento do algoritmo genético.

Cada questão, da língua inglesa, foi criada pertencendo a uma de dez categorias, sendo divididas em 5 de gramática e 5 de vocabulário e tem peso único para facilitar na contagem dos pontos e na criação de novas provas, possibilitando assim uma maior variedade de possíveis avaliações.

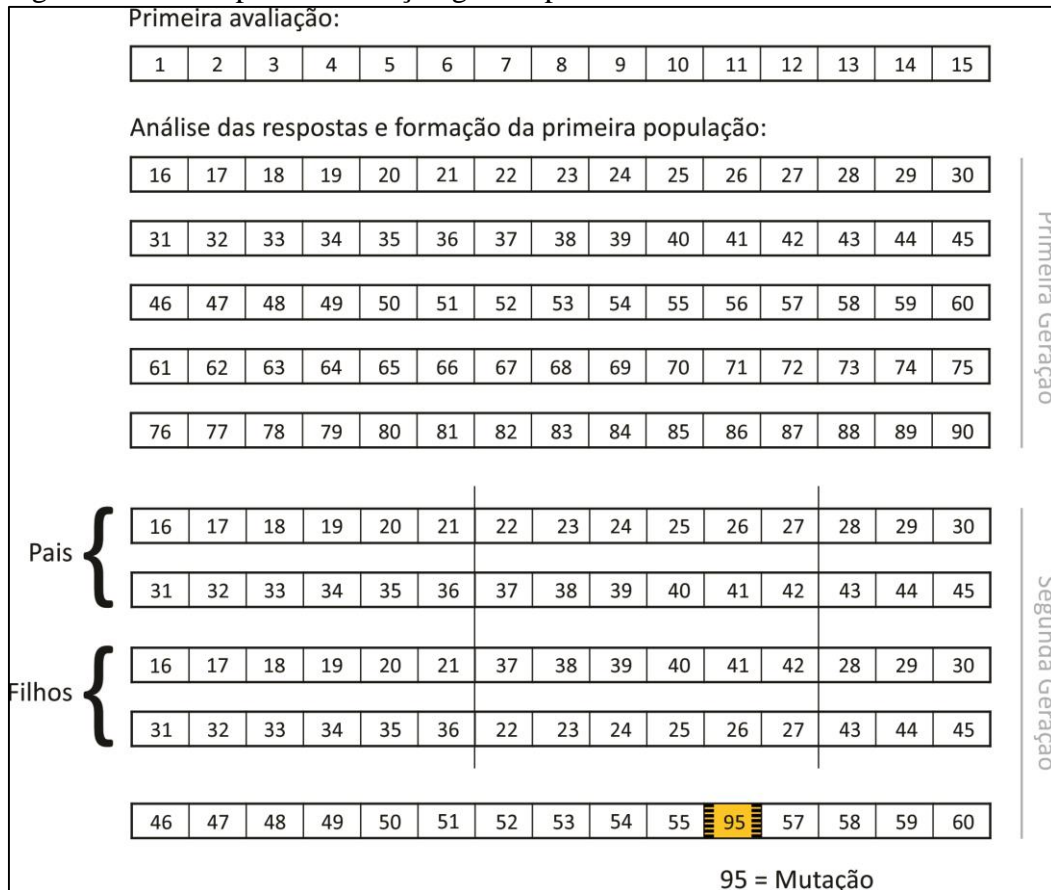
As primeiras cinco provas, número definido empiricamente a partir de testes e observações na melhor quantidade de provas para traçar o perfil de um usuário, tiveram caráter diagnóstico, com 10 perguntas dissertativas aleatórias da base de questões ainda não respondidas por este usuário. Seu peso total foi 10, facilitando o cálculo da pontuação do usuário. Tendo suas informações selecionadas, o usuário iniciou a realização de cada prova e ao finalizá-la os resultados foram armazenados no banco de dados e o perfil do usuário atualizado.

Após a realização das cinco primeiras provas e seus dados armazenados, o AG, utilizando estes dados montou cinco novas avaliações (indivíduos, do ponto de vista dos AG's), cada uma com 10 perguntas aleatórias da base de questões ainda não respondidas, únicas entre si para aumentar a diversidade e evitar inconsistências ao realizar o cruzamento e evitar que perguntas repetidas apareçam na mesma avaliação. Desta maneira a população original é formada. Utilizando ainda os dados coletados, os três indivíduos com melhor aptidão (de acordo com o perfil do usuário, tendo como base suas respostas iniciais) foram escolhidos, e os dois melhores cruzados para gerar dois novos, utilizando cruzamento multiponto – pontos na posição 4 e 7 da prova e o terceiro melhor indivíduo da geração anterior sofreu mutação. Assim, cinco indivíduos, dois pais, dois filhos e uma mutação irão

compor a nova geração. Esta prática foi repetida cinco vezes, número definido a partir de testes empíricos para o melhor resultado final, até que a avaliação com maior aptidão seja selecionada e apresentada ao usuário como nova prova a ser resolvida. Desta forma tem-se uma prova de caráter formativo, onde as perguntas irão focar em seus pontos fortes (categorias com mais acertos), porém sem negligenciar os fracos (categoria com menos acertos), para que com o tempo o resultado se altere e as novas gerações possuam avaliações diversificadas de acordo com as novas necessidades do usuário.

Um diagrama, representado na Figura 10, mostra um exemplo de como o sistema funciona, utilizando técnicas de AG para geração de novas avaliações.

Figura 10 – Exemplo da avaliação gerada pelo sistema a ser desenvolvido.



Fonte: Elaborado pelo autor.

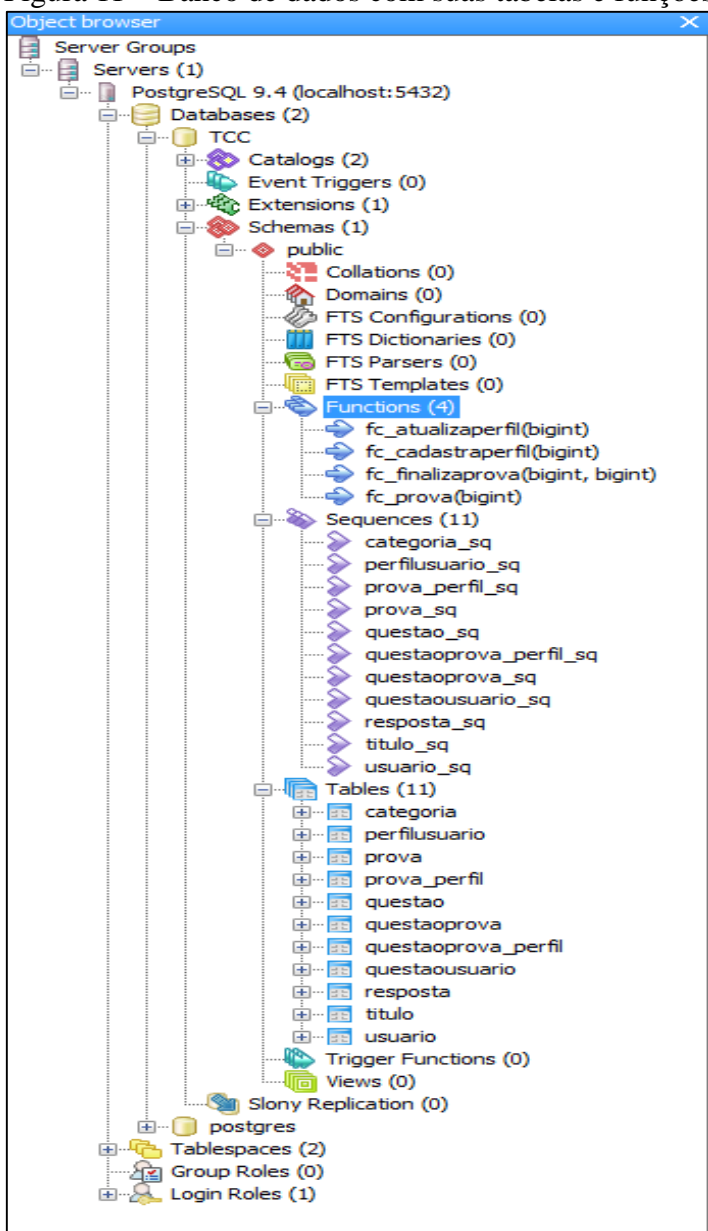
Após cada avaliação ser resolvida, suas questões foram marcadas para não se repetirem nos próximos genomas a serem testados (apenas as categorias podem se repetir). As respostas foram passadas ao usuário junto de sua pontuação na prova, sendo que a cada nova avaliação que o usuário queira fazer, uma nova população será iniciada do zero a partir das estatísticas de todas as respostas anteriores.

8 DESENVOLVIMENTO

Esta seção traz informações sobre o processo de desenvolvimento do software que utiliza a metodologia descrita na seção anterior para a criação de avaliações.

Inicialmente foi criado um banco de dados no PostgreSQL utilizando de sua interface, o pgAdmin III. Foram implementadas as tabelas necessárias para o funcionamento do software e as funções para a criação e atualização do perfil, criação da prova seguindo as regras do AG proposta anteriormente e a finalização das provas, conforme a Figura 11.

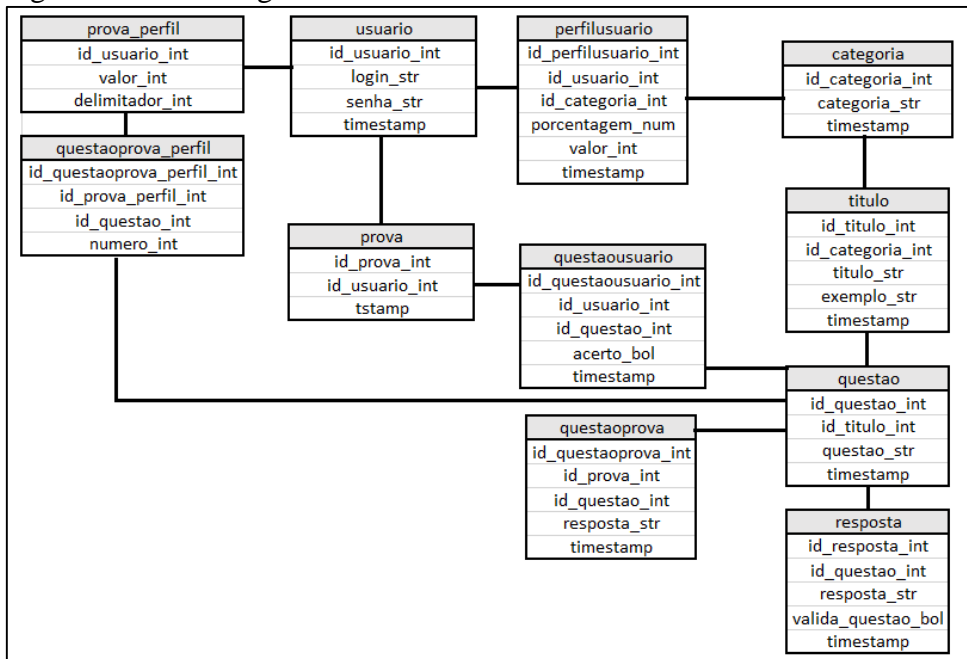
Figura 11 – Banco de dados com suas tabelas e funções.



Fonte: Elaborado pelo autor.

As tabelas são ligadas, seguindo a modelagem na Figura 12, sendo que as tabelas `prova_perfil` e `questao_prova_perfil` foram utilizadas apenas para o AG, aonde foram armazenadas as populações e suas informações, e ao finalizar replica as informações para as tabelas `prova` e `questao_prova` respectivamente.

Figura 12 – Modelagem do banco de dados.



Fonte: Elaborado pelo autor.

Juntamente com cada tabela criada, uma *sequence*, (que se trata de um campo sequencial) foi criada para o campo Id de cada tabela evitando assim registros com identificadores duplicados, conforme Figura 13.

Figura 13 – Exemplo de tabela junto de sua *sequence*.

```

-- Sequence: categoria_sq
CREATE SEQUENCE categoria_sq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 1
  CACHE 1;
ALTER TABLE categoria_sq
  OWNER TO postgres;

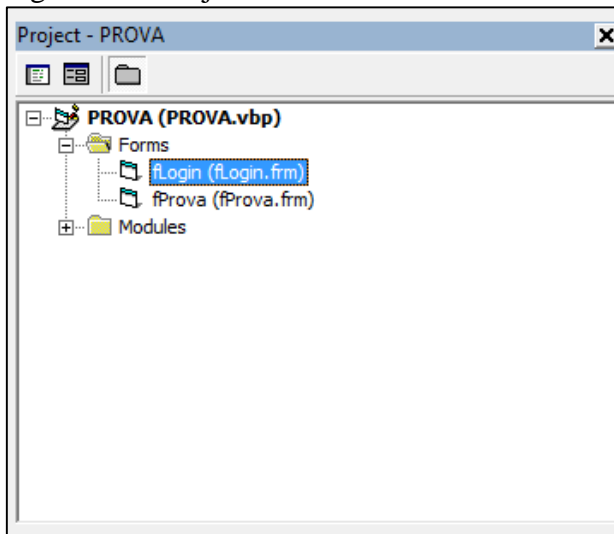
-- Table: categoria
CREATE TABLE categoria
(
  id_categoria_int bigint NOT NULL DEFAULT nextval('categoria_sq'::regclass),
  categoria_str character varying(60),
  "timestamp" timestamp without time zone,
  CONSTRAINT categoria_pkey PRIMARY KEY (id_categoria_int)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE categoria
  OWNER TO postgres;
  
```

Fonte: Elaborado pelo autor.

Com todas as tabelas criadas as mesmas foram alimentadas com as informações de categorias, títulos, questões e respostas para dar início ao desenvolvimento da ferramenta no Visual Basic 6 (VB6).

Ao iniciar o VB6, foi criado um projeto “PROVA” e dentro do mesmo criadas duas telas, fLogin, para o usuário cadastrar seu login e utilizá-lo para as provas, e fProva, aonde será mostrada a prova e as perguntas a serem respondidas como demonstrado na Figura 14.

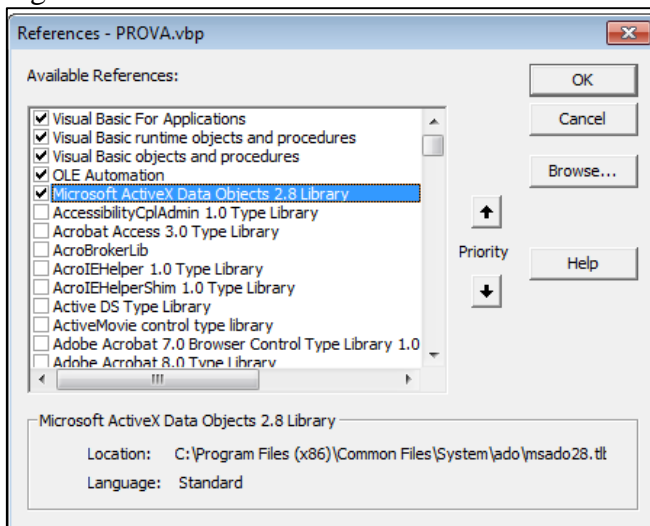
Figura 14 – Projeto VBP.



Fonte: Elaborado pelo autor.

Após sua criação, foi referenciada a biblioteca “Microsoft ActiveX Data Objects 2.8 Library”, conforme Figura 15, para que fosse possível declarar variáveis ADODB que são utilizadas no projeto.

Figura 15 – Referências.



Fonte: Elaborado pelo autor.

Referenciada esta biblioteca, foi implementado o módulo de conexões, onde são criadas as variáveis globais utilizadas em ambas as telas e a rotina de conexão com o banco de dados, demonstrado na Figura 16.

Figura 16 – Módulo de conexão.

```

Public g_conexao      As New adodb.Connection
Public g_rs           As New adodb.Recordset
Public g_strSQL       As String
Public g_IDCliente    As Integer

Public Sub pcConectaBanco()

    Dim strConexao As String
    'Set g_conexao = New ADODB.Connection

    With g_conexao
        'Dados de conexão com o base de dados
        strConexao = "DRIVER={PostgreSQL ANSI};dsn=PostgreSQL" & _
            ";database=TCC" & _
            ";server=localhost" & _
            ";PORT=5432" & _
            ";Username=postgres" & _
            ";uid=postgres" & _
            ";pwd=Gf@781213" & _
            ";TextAsLongVarchar=0;BI=12;"

        .ConnectionString = strConexao
        .CursorLocation = adUseClient

        .Open

    End With

End Sub

```

Fonte: Elaborado pelo autor.

Após a criação do módulo, iniciou-se o processo de design da tela de login, exemplificado na Figura 17.

Figura 17 – Tela de login.

Fonte: Elaborado pelo autor.

Rotinas foram criadas para a validação do login, ou seja, caso o mesmo não seja encontrado é possível cadastrar um novo, e juntamente é criado o perfil do usuário para cada categoria com valor zerado utilizando a função do banco de dados, indicando que nenhum tipo de conhecimento foi validado, já que nenhum prova foi respondida, conforme Figura 18.

Figura 18 – Função de criação de perfis.

```
-- Function: fc_cadastraperfil(bigint)
CREATE OR REPLACE FUNCTION fc_cadastraperfil(id_usuario bigint)
  RETURNS text AS
$BODY$
DECLARE
  registro      RECORD;
BEGIN

  EXECUTE 'Insert into perfilusuario (id_usuario_int,
                                     id_categoria_int,
                                     porcentagem_num,
                                     valor_int,
                                     timestamp)

          Select ' || id_usuario || ', c.id_categoria_int, null, 0, current_timestamp
          From categoria c
          Where c.id_categoria_int not in
          (Select p.id_categoria_int
          from perfilusuario p
          where p.id_usuario_int = ' || id_usuario || ');

  RETURN '';

END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION fc_cadastraperfil(bigint)
  OWNER TO postgres;
```

Fonte: Elaborado pelo autor.

Criado o login e o perfil inicial, o software irá chamar a função de criação de provas no banco de dados e retornará as informações necessárias para a segunda tela, a de prova, sendo elas o Id da prova, título da questão, um exemplo caso exista, a questão em si e a resposta dada pelo usuário, conforme Figura 19.

Figura 19 – Tela da prova Inicio.

1

Titulo: Fill in a/an, the or nothing into the gaps.

Exemplo: She is ____ really nice person. you must meet her. (key = a)

Questão: There is ____ famous university in Oxford.

Resposta:

Pergunta Anterior

Proxima Pergunta

Fonte: Elaborado pelo autor.

Ao clicar nos botões Próxima Pergunta ou Pergunta Anterior, é salva a resposta escrita e refeita a seleção da prova utilizando o Id da mesma e, então, o sistema avança para a próxima pergunta, assim caso o usuário queira voltar para pergunta anterior, a resposta estará disponível para alteração.

O botão de Pergunta Anterior é desabilitado na pergunta 1, enquanto que na pergunta 10 o botão Próxima Pergunta é desabilitado e o botão Finalizar Prova irá aparecer, conforme visto na Figura 20, onde é realizado as funções de finalização da prova e de atualização do perfil.

Figura 20 – Tela da prova Fim.

10

Titulo: Write down the following ordinals. (1st, 2nd etc.)

Exemplo:

Questão: 3rd = ____

Resposta:

third

Pergunta Anterior

Finalizar Prova

Proxima Pergunta

Fonte: Elaborado pelo autor.

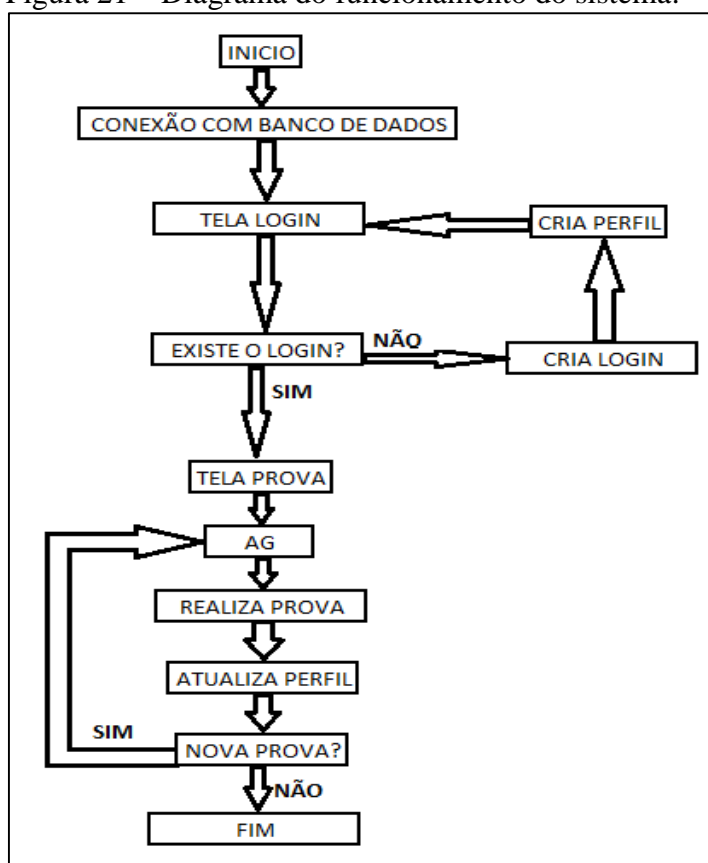
Ao concluir a prova será informada a nota calculada na mesma e se o usuário deseja continuar realizando provas. Caso o processo continue, uma nova prova é feita utilizando a função `fc_prova` do banco de dados (Apêndice A) onde é verificado se já foram realizadas as primeiras 5 provas diagnósticas, ou se a prova que deve retornar é uma de caráter formativo, utilizando-se do AG.

O procedimento que executa o AG é realizado no banco de dados, agilizando o processo de retorno das informações, considerando o número de cinco gerações, valor este definido após serem realizados testes e que serão mais detalhados na próxima seção.

Os perfis traçados para cada categoria recebem um valor de acordo com a porcentagem de acerto de tal categoria pelo usuário, sendo eles acima de 50% o valor da categoria é 1, abaixo de 50%, -1, e se for igual a 50% ele será 0. Estes valores são utilizados no AG para validar as melhores provas, que são as com valor total mais próximas de zero, ou seja, que possuam um melhor equilíbrio entre o conteúdo que o usuário sabe e que ele não sabe.

O funcionamento do sistema foi realizado seguindo o diagrama representado na Figura 21, aonde é demonstrado o início, a chamada das rotinas desenvolvidas e o fim do mesmo.

Figura 21 – Diagrama do funcionamento do sistema.



Fonte: Elaborado pelo autor.

9 RESULTADOS

Os resultados obtidos a partir do desenvolvimento deste sistema, serviram de base para se determinar os melhores parâmetros que definem como o perfil deve ser construído e quantas gerações devem ser consideradas no processo de criação das provas.

Durante os testes para traçar o perfil do usuário foram criados 10 usuários teste e foram geradas provas até que cada um deles tivesse, pelo menos uma pergunta de cada categoria respondida. Os resultados se encontram na Tabela 1.

Tabela 1 – Quantidade de provas por perfil.

Perfis	Quantidade de Provas
Perfil 1	4
Perfil 2	5
Perfil 3	3
Perfil 4	2
Perfil 5	5
Perfil 6	4
Perfil 7	6
Perfil 8	3
Perfil 9	5
Perfil 10	2

Fonte: Elaborado pelo autor.

Tendo como base os resultados obtidos, chegou-se à conclusão de que cinco provas seriam o número mais adequado para se traçar o perfil do usuário com dez categorias, tendo em vista que mesmo que uma categoria não tenha sido respondida, a mesma ainda é considerada porém com o valor zero, ou seja, não se sabe se o usuário possui conhecimentos ou não naquela categoria.

Após a definição da quantidade de provas, foi necessária determinar a quantidade de gerações que seriam consideradas para que boas provas fossem geradas. Neste teste utilizou-se cinco perfis criados com porcentagens pré-determinadas de cada categoria e foram feitos testes utilizando uma quantidade de gerações aleatória. Também foram cruzados os dados de cada perfil, a fim de avaliar o método proposta nesta investigação. O perfil 1 foi criado considerando 40% das categorias positivas, 50% negativas e 10% nulas. A Tabela 2 mostra o processo de evolução da geração de provas a cada nova geração. Conforme descrito, uma prova ideal é aquela avaliada em 0 (ponto de equilíbrio). Observa-se que para o perfil 1 apenas na quinta geração o sistema foi capaz de gerar uma prova com nota 0.

Tabela 2 – Perfil 1 (40% Positivo, 50% Negativo, 10% nulo).

Indivíduos		
GERACAO - 1	Ranque Prova	Nota
5	1	-3
1	2	-3
3	3	-4
4	4	-6
2	5	-8
GERACAO - 2		
7	1	-1
5	2	-3
1	3	-3
3	4	4
6	5	-5
GERACAO - 3		
8	1	-1
7	2	-1
9	3	-3
5	4	-3
1	5	-5
GERACAO - 4		
11	1	-1
10	2	-1
8	3	-1
7	4	-1
9	5	-3
GERACAO - 5		
13	1	0
12	2	0
11	3	-1
10	4	-1
8	5	-1

Fonte: Elaborado pelo autor.

O perfil 2 tem os seguintes valores nas categorias: 50% positivas, 30% negativas e 20% nula. Conforme pode ser visto na Tabela 3, foi gerada uma prova com nota zero logo na segunda geração. Mas vale observar que conforme as gerações iam sendo geradas, a quantidade de provas ideais aumenta e a nota geral entre todas as provas ficam melhores. No final do processo (5ª geração) duas ótimas provas foram criadas.

Tabela 3 – Perfil 2 (50% positivo, 30% negativo, 20% nulo).

Indivíduos		
GERACAO - 1	Ranque Prova	Nota
5	1	-3
3	2	5
2	3	5
1	4	5
4	5	6
GERACAO - 2		
6	1	0
7	2	2
5	3	-3
3	4	5
2	5	6
GERACAO - 3		
6	1	0
7	2	2
5	3	-2
8	4	-3
9	5	5
GERACAO - 4		
6	1	0
5	2	0
7	3	2
10	4	-3
11	5	5
GERACAO - 5		
6	1	0
5	2	0
13	3	2
12	4	-2
7	5	3

Fonte: Elaborado pelo autor.

Para o perfil 3 foram considerados os seguintes valores: 50% das categorias positivas e 50% negativas. O intuito deste perfil foi mostrar como o AG se comporta para um indivíduo totalmente equilibrado. Conforme Tabela 4, uma prova equilibrada (nota 0) foi criada na segunda geração, e ao chegar na quarta geração observa-se que 4 provas poderiam ser consideradas equilibradas.

Tabela 4 – Perfil Equilibrado.

Indivíduos			
GERACAO - 1	Ranque Prova		Nota
4	1		-2
3	2		-2
2	3		-2
1	4		-2
5	5		6
GERACAO - 2			
6	1		0
4	2		-2
3	3		-2
2	4		-2
7	5		-4
GERACAO - 3			
9	1		0
6	2		0
3	3		0
8	4		-2
4	5		-2
GERACAO - 4			
11	1		0
10	2		0
9	3		0
6	4		0
3	5		-2
GERACAO - 5			
13	1		0
12	2		0
11	3		0
10	4		0
9	5		2

Fonte: Elaborado pelo autor.

O Perfil 4 foi criado com ênfase em categorias negativas, tendo 30% positivas e 70% negativas. A Tabela 5 mostra uma geração inicial predominantemente de valores negativos, porém ao chegar na quarta geração todas as provas tem valor ótimo.

Tabela 5 – Perfil Predominante Negativo.

Indivíduos			
GERACAO - 1	Ranque Prova		Nota
4	1		-2
1	2		-2
5	3		-4
3	4		-4
2	5		-4
GERACAO - 2			
7	1		0
4	2		-2
1	3		-2
6	4		-4
5	5		-4
GERACAO - 3			
8	1		0
7	2		0
9	3		-2
4	4		-2
1	5		-2
GERACAO - 4			
11	1		0
10	2		0
9	3		0
8	4		0
7	5		0
GERACAO - 5			
13	1		0
12	2		0
11	3		0
10	4		0
9	5		0

Fonte: Elaborado pelo autor.

O último perfil, predominantemente positivo tem os seguintes valores: 80% positivo e 20% negativo. Observa-se na Tabela 6 que inicialmente as provas eram muito ruins, havendo um desequilíbrio, favorecendo muito as categorias positivas (aquelas que o usuário tinha conhecimento). Entretanto na terceira geração, duas ótimas provas foram geradas.

Tabela 6 – Perfil Predominante Positivo.

Indivíduos			
GERACAO - 1	Ranque Prova	Nota	
5	1	4	
4	2	4	
1	3	4	
3	4	8	
2	5	8	
GERACAO - 2			
6	1	2	
5	2	4	
4	3	4	
1	4	4	
7	5	8	
GERACAO - 3			
9	1	0	
6	2	0	
8	3	4	
5	4	4	
4	5	6	
GERACAO - 4			
11	1	0	
10	2	0	
9	3	0	
6	4	0	
8	5	4	
GERACAO - 5			
13	1	0	
12	2	0	
11	3	0	
10	4	0	
9	5	2	

Fonte: Elaborado pelo autor.

10 CONSIDERAÇÕES FINAIS

Tendo em vista os resultados obtidos chega-se à conclusão de que houve uma melhora significativa na geração das provas. A aplicação do método proposto, baseado em Algoritmos Genéticos, proporcionou a geração automática de provas equilibradas e com caráter mais formativo, contemplando tanto os conteúdos positivos (que o usuário domina) quanto os negativos (aqueles em que o usuário apresenta dificuldades).

Outro ponto importante a ser visto é que mesmo com as limitações da base usada, tendo apenas 474 questões divididas em 10 categorias, e por meio dos AGs foi possível obter um bom resultado. Entretanto, como trabalhos futuros sugere-se o aumento dos parâmetros de comparação, considerando uma base maior de indivíduos (provas), de perfis de usuários, de gerações e de questões. Deste modo acredita-se que os resultados podem ser melhores, porém deve ser observada a performance de tal sistema com quantidades muito grandes de informações e testes.

Este trabalho mostra que o uso do AG num sistema de avaliações pode ajudar tanto a professores, pois seria possível realizar provas individuais e com caráter formativo, quanto aos alunos, pois poderiam estar em constante teste.

REFERÊNCIAS

- AVALIAÇÃO Diagnóstica. **Centro de Políticas Públicas e Avaliação da Educação**, c2014a: Disponível em: <<http://www.portalavaliacao.caedufjf.net/pagina-exemplo/tipos-de-avaliacao/avaliacao-diagnostica/>>. Acesso em: 9 abr. 2014.
- AVALIAÇÃO Formativa. **Centro de Políticas Públicas e Avaliação da Educação**, c2014b: Disponível em: <<http://www.portalavaliacao.caedufjf.net/pagina-exemplo/tipos-de-avaliacao/avaliacao-formativa/>>. Acesso em: 9 abr. 2014.
- AVALIAÇÃO Somativa. **Centro de Políticas Públicas e Avaliação da Educação**, c2014c: Disponível em: <<http://www.portalavaliacao.caedufjf.net/pagina-exemplo/tipos-de-avaliacao/avaliacao-somativa/>>. Acesso em: 9 abr. 2014.
- AVALIAÇÃO Interna. **Centro de Políticas Públicas e Avaliação da Educação**, c2014d: Disponível em: <<http://www.portalavaliacao.caedufjf.net/pagina-exemplo/tipos-de-avaliacao/avaliacao-interna/>>. Acesso em: 9 abr. 2014.
- AVALIAÇÃO Externa. **Centro de Políticas Públicas e Avaliação da Educação**, c2014e: Disponível em: <<http://www.portalavaliacao.caedufjf.net/pagina-exemplo/tipos-de-avaliacao/avaliacao-externa/>>. Acesso em: 9 abr. 2014.
- BOM, M. Oportunidades são melhores para bilíngues. **UB House**, c2014. Disponível em: <<http://ubhouse.com.br/oportunidades-sao-melhores-para-bilingues/>>. Acesso em: 26 fev. 2014.
- BUTTIGNON, K.; SILVA, M. H. da.; GARCIA, R. P. C. **Uma Reflexão sobre o Ensino a Distância, via Internet, no Brasil**. 2002. 63 f. Monografia (Especialização em Informática Empresarial) - Faculdade de Engenharia da Universidade Estadual Paulista, Guaratinguetá, 2002. Disponível em <<http://www.feg.unesp.br/ceie/Monografias-Texto/CEIE0204.pdf>>. Acesso em: 15 mar. 2014.
- GATTI, B. A. O professor e a avaliação em sala de aula. **Estudos em Avaliação Educacional**, São Paulo, n. 27, p. 97-114, jan/jun. 2003. Disponível em: <<http://www.submit.10envolve.com.br/uploads/ba6dbaf3b94f764ef3bce2a19d1ee9e1.pdf>>. Acesso em: 1 mar. 2014.
- GONÇALVES, R. M. A Necessidade de incentivar a aprendizagem da língua inglesa desde a infância. **Revista Don Domênico**, Guarujá, n. 2, out. 2009. Disponível em: <http://www.faculdedondomenico.edu.br/novo/revista_don/artigo2_ed2.pdf>. Acesso em: 1 mar. 2014.
- ISHIMOTO, M. G. T. Avaliação Como Meio de Aprendizagem. **Webartigos.com**, 2006. Disponível em: <<http://www.webartigos.com/artigos/avaliacao-como-meio-de-aprendizagem/32895/>> Acesso em: 9 abr. 2014.
- LINDEN, R. **Algoritmos genéticos: uma importante ferramenta da inteligência computacional**. 2. ed. Rio de Janeiro: Brasport, 2008.

LUCAS, D. C. **Algoritmos Genéticos: Uma Introdução**. Rio Grande do Sul: UFRGS, 2002. 47 f. Apostila. Disponível em <<http://www.inf.ufrgs.br/~alvares/INF01048IA/ApostilaAlgoritmosGeneticos.pdf>>. Acesso em: 15 mar. 2014.

LUGER, G. F. **Inteligência Artificial: estruturas e estratégias para a solução de problemas complexos**. 4. ed. Porto Alegre: Bookman, 2004.

MORAES, D. A. F. de. Prova: instrumento avaliativo a serviço da regulação do ensino e da aprendizagem, **Estudos em Avaliação Educacional**, São Paulo, v. 22, n. 49, p. 238-258 maio/ago. 2011. Disponível em: <<http://www.fcc.org.br/pesquisa/publicacoes/eae/arquivos/1636/1636.pdf>>. Acesso em: 1 mar. 2014.

PILLATI, A.; SANTOS, M. E. M. dos. O domínio da língua inglesa como fator determinante para o sucesso profissional no mundo globalizado. **Secretariado Executivo em Revist@**, Passo Fundo, v. 4, 2008. Disponível em: <<http://perguntaserespostas.com.br/seer/index.php/ser/article/view/1766/1174>>. Acesso em: 26 fev. 2014.

PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. 7. ed. Porto Alegre: Bookman, 2011.

ROCHA, R. M. **Avaliação de competências como instrumento de desenvolvimento e capacitação: um estudo em uma organização financeira**. 2012, 107 f. Dissertação (Mestrado em Administração) - Faculdade de Ciências Empresariais da Universidade Fundação Mineira de Educação e Cultura, Belo Horizonte, 2012. Disponível em: <<http://www.fumec.br/anexos/cursos/mestrado/dissertacoes/completa/marcelo-ribeiro-rocha.pdf>>. Acesso em: 1 mar. 2014.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. 2. ed. Rio de Janeiro: Campus, 2004.

SOARES, G. L. **Algoritmos genéticos: estudo, novas técnicas e aplicações**. 1997. 137 f. Dissertação (Pós-Graduação em Engenharia Elétrica) - Universidade Federal de Minas Gerais, Belo Horizonte, 1997. Disponível em <http://pessoal.utfpr.edu.br/mariano/arquivos/VAS1997_1.pdf>. Acesso em: 15 mar, 2014.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2012.

APÊNDICE A – SQL PARA A REALIZAÇÃO DO ALGORITMO GENÉTICO

```

-- Function: fc_prova(bigint)

-- DROP FUNCTION fc_prova(bigint);

CREATE OR REPLACE FUNCTION fc_prova(id_usuario bigint)
  RETURNS SETOF record AS
$BODY$
DECLARE
registro RECORD;
cont integer;
i integer;
n integer;
sql text;
qtdteste double precision;
tamanhoteste double precision;

BEGIN

--Verifica a quantidade de provas concluidas pelo usuario
cont := (Select count(*) from prova where finalizada_bol = true and id_usuario_int = id_usuario);

--Caso seja menor ou igual a 5 a prova ainda é de carater Diagnostico
If cont <= 5 then

  --Faz o insert na tabela prova
  EXECUTE 'Insert into prova (id_usuario_int, timestamp)
    values (' || id_usuario || ', current_timestamp)';

  --Faz o insert na tabela com as questoes da prova
  EXECUTE 'Insert into questaprova (id_prova_int, id_questao_int, timestamp)
    (select currrval("prova_sq"), t.id_questao_int, current_timestamp
      from (
        SELECT *
        FROM questao
        where id_questao_int not in
          (select id_questao_int
            from questausuario
            where id_usuario_int = ' || id_usuario || ')
        ORDER BY random() limit 10
      ) t
      order by id_questao_int)';

  --Maior que cinco, AG para prova formativa
  else

    --Rotina de algoritmo genetico
    Execute 'delete from questaprova_perfil
      where id_prova_perfil_int in
        (select id_prova_perfil_int
          from prova_perfil
          where id_usuario_int = ' || id_usuario || ')';

    Execute 'delete from prova_perfil
      where id_usuario_int = ' || id_usuario || ';';

    For n in 1..5 loop
      --Faz o insert na tabela prova
      EXECUTE 'Insert into prova_perfil (id_usuario_int) values (' || id_usuario || ')';

      EXECUTE 'alter sequence numero_sq restart;';

      raise notice 'Sequene alterada';
    end loop;
  end if;
END;

```

```

--Faz o insert na tabela com as questoes da prova
EXECUTE 'Insert into questaoprova_perfil (id_prova_perfil_int, id_questao_int, numero_int)
(select currval("prova_perfil_sq"), t.id_questao_int, nextval("numero_sq"))
from (
SELECT *
FROM questao
where id_questao_int not in
(select id_questao_int
from questaousuario
where id_usuario_int = ' || id_usuario || ')
AND id_questao_int not in
(select qp.id_questao_int
from questaoprova_perfil qp
inner join prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
where pp.id_usuario_int = ' || id_usuario || ')
ORDER BY random() limit 10
)t
order by id_questao_int)';

End loop;

for i in 1..4 loop

raise notice '%', 'GERACAO - ' || i;
Execute 'update prova_perfil set delimitador_int = null::integer
where id_usuario_int = ' || id_usuario || ';

Execute 'update prova_perfil set valor_int =
(select sum(pf.valor_int)
FROM questaoprova_perfil qp
INNER JOIN questao q on q.id_questao_int = qp.id_questao_int
INNER JOIN titulo t on t.id_titulo_int = q.id_titulo_int
INNER JOIN categoria c on c.id_categoria_int = t.id_categoria_int
INNER JOIN perfilusuario pf on pf.id_categoria_int = c.id_categoria_int
WHERE pf.id_usuario_int = ' || id_usuario || '
AND qp.id_prova_perfil_int = prova_perfil.id_prova_perfil_int)';

Execute 'update prova_perfil set delimitador_int = 1
WHERE id_prova_perfil_int =
(SELECT p.id_prova_perfil_int
FROM prova_perfil p
WHERE p.id_usuario_int = ' || id_usuario || '
ORDER BY regexp_replace( p.valor_int::varchar, '^[^0-9]*', '')::integer,
p.id_prova_perfil_int DESC LIMIT 1 OFFSET 0)';

Execute 'update prova_perfil set delimitador_int = 2
WHERE id_prova_perfil_int =
(SELECT p.id_prova_perfil_int
FROM prova_perfil p
WHERE p.id_usuario_int = ' || id_usuario || '
ORDER BY regexp_replace( p.valor_int::varchar, '^[^0-9]*', '')::integer,
p.id_prova_perfil_int DESC LIMIT 1 OFFSET 1)';

Execute 'update prova_perfil set delimitador_int = 3
WHERE id_prova_perfil_int =
(SELECT p.id_prova_perfil_int
FROM prova_perfil p
WHERE p.id_usuario_int = ' || id_usuario || '
ORDER BY regexp_replace( p.valor_int::varchar, '^[^0-9]*', '')::integer,
p.id_prova_perfil_int DESC LIMIT 1 OFFSET 2)';

FOR registro IN EXECUTE '
SELECT id_prova_perfil_int,
valor_int,
coalesce(delimitador_int, 0) as delimitador_int
from prova_perfil
where id_usuario_int = ' || id_usuario || '
order by regexp_replace( valor_int::varchar, '^[^0-9]*', '')::integer,
id_prova_perfil_int DESC ;'

LOOP
raise notice '%', registro.id_prova_perfil_int || ' - ' || registro.delimitador_int || ' - ' || registro.valor_int
END LOOP;

```

```

Execute 'delete from questaoprova_perfil
      where id_prova_perfil_int in
            (select id_prova_perfil_int
             from prova_perfil
             where coalesce(delimitador_int, 0) = 0
             and id_usuario_int = ' || id_usuario || ');

Execute 'delete from prova_perfil
      where coalesce(delimitador_int, 0) = 0
      and id_usuario_int = ' || id_usuario || ';

--Faz o insert na tabela prova
EXECUTE 'Insert into prova_perfil (id_usuario_int) values (' || id_usuario || ');

--Faz o insert na tabela com as questoes da prova
EXECUTE 'Insert into questaoprova_perfil (id_prova_perfil_int, id_questao_int, numero_int)
      (SELECT currval("prova_perfil_sq"), t.id_questao_int, t.numero_int
       FROM (
         (SELECT qp.id_questao_int, qp.numero_int
          FROM questaoprova_perfil qp
          INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
           AND pp.delimitador_int = 1
           AND id_usuario_int = ' || id_usuario || '
          ORDER BY numero_int limit 3)

        UNION

         (SELECT qp.id_questao_int, qp.numero_int
          FROM questaoprova_perfil qp
          INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
           AND pp.delimitador_int = 2
           AND id_usuario_int = ' || id_usuario || '
          ORDER BY numero_int limit 4 offset 3)

        UNION

         (SELECT qp.id_questao_int, qp.numero_int
          FROM questaoprova_perfil qp
          INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
           AND pp.delimitador_int = 1
           AND id_usuario_int = ' || id_usuario || '
          ORDER BY numero_int limit 3 offset 7)
        ) t
       ORDER BY id_questao_int)';

--Faz o insert na tabela prova
EXECUTE 'Insert into prova_perfil (id_usuario_int) values (' || id_usuario || ');

--Faz o insert na tabela com as questoes da prova
EXECUTE 'Insert into questaoprova_perfil (id_prova_perfil_int, id_questao_int, numero_int)
      (SELECT currval("prova_perfil_sq"), t.id_questao_int, t.numero_int
       FROM (
         (SELECT qp.id_questao_int, qp.numero_int
          FROM questaoprova_perfil qp
          INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
           AND pp.delimitador_int = 2
           AND id_usuario_int = ' || id_usuario || '
          ORDER BY numero_int limit 3)

        UNION

         (SELECT qp.id_questao_int, qp.numero_int
          FROM questaoprova_perfil qp
          INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
           AND pp.delimitador_int = 1
           AND id_usuario_int = ' || id_usuario || '
          ORDER BY numero_int limit 4 offset 3)

        UNION

         (SELECT qp.id_questao_int, qp.numero_int
          FROM questaoprova_perfil qp
          INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
           AND pp.delimitador_int = 2
           AND id_usuario_int = ' || id_usuario || '
          ORDER BY numero_int limit 3 offset 7)
        ) t
       ORDER BY id_questao_int)';

```

```

) t
ORDER BY id_questao_int);

--Faz o elemento de mutação na prova delimitada como terceira
EXECUTE 'UPDATE questaoprova_perfil set id_questao_int =
(SELECT id_questao_int
FROM questao
where id_questao_int not in
(select id_questao_int
from questaousuario
where id_usuario_int = ' || id_usuario || ')
AND id_questao_int not in
(select qp.id_questao_int
from questaoprova_perfil qp
inner join prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
where pp.id_usuario_int = ' || id_usuario || ')
ORDER BY random() limit 1)
WHERE id_questaoprova_perfil_int = (
SELECT qp.id_questaoprova_perfil_int
FROM questaoprova_perfil qp
INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
WHERE pp.id_usuario_int = ' || id_usuario || '
AND pp.delimitador_int = 3
ORDER BY random() limit 1)';

END LOOP;

Execute 'update prova_perfil set delimitador_int = null::integer where id_usuario_int = ' || id_usuario || '
';
Execute 'update prova_perfil set valor_int =
(select sum(pf.valor_int)
FROM questaoprova_perfil qp
INNER JOIN questao q on q.id_questao_int = qp.id_questao_int
INNER JOIN titulo t on t.id_titulo_int = q.id_titulo_int
INNER JOIN categoria c on c.id_categoria_int = t.id_categoria_int
INNER JOIN perfilusuario pf on pf.id_categoria_int = c.id_categoria_int
WHERE pf.id_usuario_int = ' || id_usuario || '
AND qp.id_prova_perfil_int = prova_perfil.id_prova_perfil_int);
';

Execute 'update prova_perfil set delimitador_int = 1
WHERE id_prova_perfil_int =
(SELECT p.id_prova_perfil_int
FROM prova_perfil p
WHERE p.id_usuario_int = ' || id_usuario || '
order by regexp_replace( p.valor_int::varchar , '[^0-9]*', '')::integer,
p.id_prova_perfil_int DESC LIMIT 1 OFFSET 0);';

Execute 'update prova_perfil set delimitador_int = 2
WHERE id_prova_perfil_int =
(SELECT p.id_prova_perfil_int
FROM prova_perfil p
WHERE p.id_usuario_int = ' || id_usuario || '
order by regexp_replace( p.valor_int::varchar , '[^0-9]*', '')::integer,
p.id_prova_perfil_int DESC LIMIT 1 OFFSET 1);';

Execute 'update prova_perfil set delimitador_int = 3
WHERE id_prova_perfil_int =
(SELECT p.id_prova_perfil_int
FROM prova_perfil p
WHERE p.id_usuario_int = ' || id_usuario || '
order by regexp_replace( p.valor_int::varchar , '[^0-9]*', '')::integer,
p.id_prova_perfil_int DESC LIMIT 1 OFFSET 2);';

raise notice 'FINAL';

FOR registro IN EXECUTE 'SELECT id_prova_perfil_int, valor_int, coalesce(delimitador_int, 0) as delimitador_int
from prova_perfil
where id_usuario_int = ' || id_usuario || '
order by regexp_replace( valor_int::varchar , '[^0-9]*', '')::integer,
id_prova_perfil_int DESC;'

LOOP
raise notice '%', registro.id_prova_perfil_int || ' - ' || registro.delimitador_int || ' - ' || registro.valor_int;
END LOOP;

```

```

--Faz o insert na tabela prova
EXECUTE 'Insert into prova (id_usuario_int, timestamp) values (' || id_usuario || ', current_timestamp)';

--Faz o insert na tabela com as questoes da prova a partir do melhor individuo
EXECUTE 'Insert into questaoprova (id_prova_int, id_questao_int, timestamp)
(select currval('prova_sq'), t.id_questao_int, current_timestamp
from (
    SELECT qp.id_questao_int
    FROM questaoprova_perfil qp
    INNER JOIN prova_perfil pp on pp.id_prova_perfil_int = qp.id_prova_perfil_int
    AND pp.delimitador_int = 1
    AND id_usuario_int = ' || id_usuario || '
) t
order by id_questao_int)';

end if;

RETURN query ( SELECT p.id_prova_int,
q.id_questao_int,
q.questao_str,
t.exemplo_str,
t.titulo_str,
qp.resposta_str,
qp.Id_questaoprova_int
FROM prova p
inner join questaoprova qp on qp.id_prova_int = p.id_prova_int
inner join questao q on q.id_questao_int = qp.id_questao_int
inner join titulo t on t.id_titulo_int = q.id_titulo_int
where p.id_prova_int = currval('prova_sq')
Order by qp.Id_questaoprova_int
);

END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100
ROWS 10;
ALTER FUNCTION fc_prova(bigint)
OWNER TO postgres;

```