

Walisson Cardoso Gomes

**Bachome: uma Ferramenta de Auxílio no
Ensino da Seleção Natural
Controlada por Algoritmo Genético**

Belém-Brasil

2017

Walisson Cardoso Gomes

**Bachome: uma Ferramenta de Auxílio no
Ensino da Seleção Natural
Controlada por Algoritmo Genético**

Trabalho de conclusão de curso apresentado à banca da faculdade de computação da UFPA como requisito para obtenção do grau de Bacharel em Ciência da Computação.

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO

Orientador: Claudomiro de Souza de Sales Junior
Coorientador: Reginaldo Cordeiro dos Santos Filho

Belém-Brasil

2017

*Dedico este trabalho às pessoas que não desistem de tentar fazer o mundo um lugar
melhor*

Agradecimentos

Primeiramente, meus sinceros agradecimentos aos meus orientador, coorientador e também amigos, Claudomiro de Souza de Sales Junior e Reginaldo Cordeiro dos Santos Filho pela ajuda e orientação, sem os quais, este trabalho não seria possível.

Agradeço à minha família, de forma especial aos meus pais, pelo apoio incondicional durante toda a minha carreira acadêmica.

Agradeço aos companheiros do Laboratório de Eletromagnetismo Aplicado (LEA), onde aprendi e estou aprendendo os caminhos para ser um pesquisador.

Por fim, mas não menos importante, agradeço à todos os meus amigos de longa data que estiveram comigo nos ensino fundamental e médio no Colégio de Ensino Icoaraciense (CEI). Aos meus companheiros de turma que se tornaram amigos para a vida e à direção do colégio que me recebeu com grande carinho quando precisei de permissão para validar a proposta deste trabalho com seus alunos.

*“Bem-aventurado o homem que acha sabedoria, e o homem que adquire conhecimento;
Porque é melhor a sua mercadoria do que artigos de prata, e maior o seu lucro que o ouro
mais fino.”*

(Bíblia Sagrada, Provérbios 3:13,14)

Resumo

A teoria sintética da evolução é uma das principais áreas de estudo da biologia moderna e tem aplicações em diversos campos práticos, a citar, agricultura, medicina e algoritmos de otimização. A seleção natural é um dos mecanismos centrais da evolução, responsável pela seleção de características de adaptação em indivíduos e a compreensão básica de seu funcionamento é primordial no entendimento da teoria como um todo. Entretanto, uma parcela significativa dos estudantes nos diferentes graus de escolaridade não têm um conhecimento funcional do assunto. Desta maneira, faz-se interessante a construção de um modelo computacional capaz de auxiliar o estudante a assimilar os conceitos chave envolvidos no processo de evolução das espécies. Este trabalho propõe a criação de uma ferramenta para o auxílio do ensino da evolução por meio da simulação do processo de seleção natural utilizando algoritmos genéticos. A aplicação proposta evolui uma população através do algoritmo genético e representa os seus indivíduos de forma gráfica como uma colônia de bactérias, possibilitando a adição de elementos de variação e de seleção na simulação. A metodologia empregada tem por objetivo demonstrar o funcionamento da seleção natural por um viés prático de tal forma que alunos do ensino básico possam entender os preceitos e consequências do processo de seleção natural e alunos do ensino superior se beneficiam com a interface interativa da aplicação para com o algoritmo genético. A aplicação foi experimentada com diferentes cenários em sala de aula para um grupo de alunos do nono ano e os resultados demonstram que a utilização da ferramenta conduz uma discussão frutífera para diagnosticar e corrigir conhecimentos mal formados dos alunos sobre o tema.

Palavras-chave: Algoritmos Genéticos. Seleção Natural. Ensino. Simulação.

Abstract

The synthetic theory of evolution is one of the main areas of study in modern biology and has applications in several practical fields, such as agriculture, medicine and optimization algorithms. The natural selection is one of the central mechanisms of evolution, responsible for the selection of characteristics of adaptation in individuals and a basic comprehension of its functioning is primordial on the understanding of the theory as a whole. However, a significant proportion of students in the different schooling levels do not have a functional knowledge of it. In this way, the development of a computational model capable of helping students to assimilate the concepts involved in the process of species evolution is interesting. This work proposes the creation of a tool to help evolution teaching by simulating the process of natural selection using genetic algorithms. The proposed application evolves a population through the genetic algorithm and represents its individuals in a graphical way as a colony of bacteria, allowing the addition of elements of variation and selection in the simulation. The objective of the applied methodology is to show natural selection functioning in a practical way such that elementary school students understand the precepts and consequences of the natural selection process and the students of higher education benefit from an interactive interface of the application to the genetic algorithm. The application was evaluated with different scenarios in a classroom with a group of students on the ninth grade and the results demonstrate that the use of the tool leads to a fruitful discussion to diagnose and correct malformed knowledge of the students about the theme.

Keywords: Genetic Algorithms. Natural Selection. Teaching. Simulation.

Lista de ilustrações

Figura 1 – <i>Workflow</i> básico de uma AG.	21
Figura 2 – Probabilidade de Seleção de acordo com o <i>fitness</i>	23
Figura 3 – Processo de cruzamento de um ponto de corte.	24
Figura 4 – Combinação de formatos e cores foram nove tipos diferentes de indivíduos.	28
Figura 5 – Exemplo de configuração populacional representado na interface gráfica.	29
Figura 6 – Diagrama de Casos de Uso para a aplicação Bachome.	32
Figura 7 – Componentes da construção da aplicação Bachome.	33
Figura 8 – Diagrama das classes que compõem o AG.	34
Figura 9 – Diagrama das classes que compõem o pacote UI	35
Figura 10 – Apresentação dos indivíduos da população. Este caso de simulação inclui a configuração da população várias gerações após inserção de mutação que ocasiona em representação por diferentes cores.	36
Figura 11 – Painel de configuração para os aspectos gerais da simulação.	36
Figura 12 – Painel de configuração para as propriedades do AG.	37
Figura 13 – Painel de configuração para as propriedade do AG com representação real ativada.	37
Figura 14 – Configuração populacional após a aplicação de diversas operações de mutação e seleção nos instantes de tempo t_i . Em a), divisão da população por formato e em b), divisão por cor.	40
Figura 15 – Configuração da população nos instantes de tempo a) t_1 , b) t_2 , c) t_3 , d) t_4 , e) t_5 , f) t_6 , g) t_7 , h) t_8	42
Figura 16 – Configuração populacional após aplicação de antibiótico e seleção por recursos.	43

Lista de tabelas

Tabela 1	– Aptidão de quatro indivíduos para a função de avaliação $f(x) = 2^x$	23
Tabela 2	– Operadores implementados e suas variantes.	26
Tabela 3	– Vantagens oferecidas pelo formato.	28
Tabela 4	– Vantagens oferecidas pela cor.	29
Tabela 5	– Lista de requisitos funcionais.	30
Tabela 6	– Lista de requisitos não funcionais.	31
Tabela 7	– Efeitos da adição de mutação e seleção.	40

Lista de abreviaturas e siglas

AE	Algoritmo Evolucionário
AG	Algoritmo Genético
DNA	<i>deoxyribonucleic acid</i>
SGA	<i>Simple Genetic Algorithm</i>
PH	Potencial Hidrogeniônico
CEI	Colégio de Ensino Icoaraciense
UFPA	Universidade Federal do Pará

Lista de símbolos

α	Letra grega Alpha
\sum_i^n	Somatório de i até n

Sumário

1	Introdução	12
1.1	Trabalhos Relacionados	13
1.2	Motivação	15
1.3	Justificativa	15
1.4	Objetivos	16
1.5	Estrutura do trabalho	17
2	Fundamentação Teórica	18
2.1	Biologia Evolutiva	18
2.2	Algoritmos Genéticos	20
2.2.1	Representação	22
2.2.2	Função de Avaliação	22
2.2.3	Seleção de Pais	23
2.2.4	Operador de Cruzamento	24
2.2.5	Operador de Mutação	25
2.2.6	Seleção de Sobreviventes	25
2.2.7	Operadores Disponíveis	26
3	Descrição da Ferramenta	27
3.1	Lista de Requisitos	30
3.2	Casos de Uso	32
3.3	Implementação	32
3.4	Interface Gráfica	35
4	Resultados	38
4.1	Cenários de Simulação	38
4.2	Validação com Professor	43
4.3	Validação em Sala de Aula	44
5	Conclusão	46
	Referências	48

1 Introdução

Seleção Natural é um dos mecanismos centrais no processo de evolução das espécies. No entanto, como apontam diversos estudos, os conceitos e consequências que permeiam esta área são frequentemente mal entendidos por alunos do ensino regular e superior. Em (GREGORY, 2009), um estudo sobre os equívocos mais comuns cometidos por estudantes é apresentado. Dentre eles podemos citar: o errôneo entendimento sobre o surgimento de mudanças em uma espécie como a necessidade ou o desejo dos indivíduos de mudarem para se adaptar ao meio e que apenas as mudanças benéficas são passadas para os descendentes. Gregory traz ainda em seu trabalho (GREGORY, 2009, pp. 164–166) um quadro-resumo de diversos estudos mostrando o alto grau de equívocos demonstrados por estudantes em relação à seleção natural e evolução. A quantidade de estudantes que aprendem incorreções como as citadas é significativa.

A teoria da evolução é uma disciplina complexa que requer o conhecimento de conceitos básicos da biologia, como DNA, seleção das espécies, especiação, e a capacidade de conectá-los através de diferentes escalas de tempo, espaço e organização biológica (SPETH et al., 2009). Acrescido a isto, frequentemente os estudantes são instruídos por pessoas com baixo conhecimento científico do assunto, ou aprendem conceitos errôneos passados por meios de comunicação em massa, ou mesmo são instruídos por pessoas com qualificação, como professores, mas a transmissão de tais conceitos relacionados à evolução apresentam algumas imprecisões ou falta de objetividade (MOORE et al., 2002).

É importante que o aluno aprenda corretamente os conceitos intrínsecos da evolução. Esta é uma área central e unificadora na biologia, um importante tópico da grade curricular em diversos países. Ainda mais, Gregory argumenta que muitos dos estudantes têm dificuldade em aceitar a teoria da evolução por causa do entendimento incompleto da mesma (GREGORY, 2009, p. 163). Fornecer um aprendizado adequado de forma a fazê-los compreender a lógica básica da teoria, ajudaria em sua difusão.

Dada a problemática encontrada, surge a oportunidade de utilizar uma ferramenta computacional interativa de simulação para auxiliar o estudante em seu aprendizado e/ou assistir o professor em suas aulas à respeito da seleção natural e evolução das espécies. O software *Bachome* (nome formado por segmentos de *Bacteria's Home*, ou Lar de Bactérias, em português), da qual o presente trabalho trata, expressa alguns dos conceitos chave da seleção natural em ambiente gráfico. O software simula o desenvolvimento de uma população de micro-organismos cujo comportamento é descrito por um algoritmo genético, uma classe de algoritmos evolucionários.

Algoritmos evolucionários (AEs) são algoritmos inspirados no processo de seleção

natural e evolução. Neste tipo de algoritmos, um determinado ambiente é preenchido com uma população e seus indivíduos competem por sobrevivência e reprodução (EIBEN; SMITH, 2003, p. 13). Algoritmos genéticos (AGs), por sua vez, são a classe mais popular de AEs e largamente conhecidos como algoritmos de otimização (EIBEN; SMITH, 2003, p. 99). Em suma, AGs são uma representação do processo de evolução através da seleção natural, onde o indivíduo mais apto, ao final, é tido como uma solução à um problema específico. Além disso, são amplamente utilizados na academia e, conseqüentemente, o público alvo desta aplicação é expandido também para alunos do ensino superior da área de computação que estejam interessados em aplicar para disciplinas que incluam o aprendizado de algoritmos genéticos.

1.1 Trabalhos Relacionados

Não é novidade que diferentes tecnologias da informação têm sido aplicadas com objetivos educacionais. Em (BARAB; DEDE, 2007), os autores argumentam que experiências semelhantes a jogos virtuais, como jogos eletrônicos e simulações, fornecem um forte sentimento de engajamento e oportunidade para aprendizado. Expõem ainda que mesmo quando estes meios são inicialmente desenvolvidos com objetivo de entreter, apresentam um cenário rico de discussões e questionamentos. Aproveitando as vantagens e metodologias de tecnologias baseadas em jogos é possível criar uma mecânica de ensino voltada para o aprendizado científico profunda e atraente. Por exemplo, em (HERRERO et al., 2014) os autores investigam como o jogo *Spore* pode ser utilizado como uma ferramenta de aprendizado em aulas de biologia.

O jogo *Spore* permite que os jogadores desenvolvam as suas próprias formas de vida que evoluirão desde seres unicelulares até espécies complexas interplanetárias. O jogo oferece diversas fases representando o processo de evolução da espécie ao longo de diferentes estágios de tempo e Herrero et al. (2014, p. 28) exemplifica como o jogo pode ser utilizado em uma aula de biologia sob a supervisão de um professor para engajar os estudantes no ensino de evolução natural. Os autores ressaltam que as falhas do jogo em representar fielmente conceitos como variação, aleatoriedade e o processo natural de seleção podem ser supridas com o acompanhamento de um professor e virar tópico de discussão.

Vale mencionar que existem diversos trabalhos de objetivo educacional que tentam simular o processo de seleção natural através de um ambiente interativo onde, a partir de escolhas de cenários e características de indivíduos, o usuário pode observar as alterações em uma população sendo representada. Yamanoi e Iwasaki (2015) descrevem um simulador denominado *Origami Bird Simulator* baseado no recurso de ensino *Origami Bird*, na qual os estudantes aprendem sobre seleção natural criando e atirando pássaros feitos de papel.

Os pássaros que atingirem maior distância são considerados mais aptos à sobrevivência e reprodução. O simulador cria um ambiente virtual para este protocolo e adiciona o conceito de especiação.

Em (JONES; LAUGHLIN, 2010) os autores descrevem um modelo gratuito e online, chamado *PopGen Fishbowl*, que simula a seleção natural e outros aspectos da evolução natural como migração e mutação. O ambiente simula o progresso de uma população de peixes em um pequeno aquário e possibilita a alteração de parâmetros como tamanho da população, frequência dos alelos, tamanho da descendência e mortalidade.

Os autores em (SPETH et al., 2009) expõem a criação de uma metodologia para o ensino da evolução usando o Avida-ED. Avida-ED é um software educacional para o aprendizado da evolução em cursos do ensino básico. É um ambiente construído usando como base o conhecido software Avida, um ambiente de evolução de formas de vida digital que competem por recursos computacionais.

As três aplicações descritas, *Origami Bird Simulator*, *PopGen FishBowl* e Avida-ED, têm variados cenários: a primeira explora pássaros, a segunda peixes e a terceira seres virtuais semelhantes a vírus. Esses cenários são criados com o objetivo de contextualização e para tornar o aprendizado prático. Este trabalho propõe a criação de uma ferramenta de simulação de seleção natural e evolução que segue um molde semelhante, com a adição de novos aspectos para a melhora da simulação e adição de elementos de configuração que serão descritos ao longo deste trabalho. O simulação desenvolvida tem como temática uma população de micro-organismos chamados de bactérias e são observados sob um microscópio. A aplicação aqui tratada tem o funcionamento de sua simulação baseado em algoritmos genéticos, um modelo mais apropriado para simular o comportamento de populações ao longo do tempo.

Existem ainda algumas aplicações menos formais que buscam descrever o mesmo processo e são disponibilizadas gratuitamente na internet. Podem ser citadas as aplicações *Wolves and Bunnies*¹, *Who Wants to Live a Million Years?*², *Natural Selection* por *Virtual Lab*³, *Natural Selection* por SEPUP⁴ e *Bugging Out in Old England*⁵ em língua inglesa e Aves e Insetos⁶. A pesquisa bibliográfica ajudou a revelar o quão raras as aplicações do gênero nacionais são. Desse forma, este trabalho contribui na ampliação de material de referência educacional de qualidade em língua portuguesa brasileira.

¹ Disponível em: <<https://phet.colorado.edu/en/simulation/natural-selection>>

² Disponível em: <<http://www.sciencechannel.com/games-and-interactives/charles-darwin-game>>

³ Disponível em: <www.mhhe.com/biosci/genbio/virtual_labs/BL_12/BL_12.html>

⁴ Disponível em: <http://sepuplhs.org/high/sgi/teachers/evolution_act11_sim.html>

⁵ Disponível em: <<http://sciencenetlinks.com/interactives/evolution.html>>

⁶ Disponível em: <<http://www.ib.usp.br/evolucao/inic/aves.htm>>

1.2 Motivação

As aplicações de simulação dos trabalhos citados, *Origami Bird Simulator*, *PopGen FishBowl* e *Avida-ED*, possuem diferentes mecanismos para o desenvolvimento das populações neles representadas, porém são todos inspirados em estratégias evolucionárias e tratam aspectos como cruzamento e mutações, características dos algoritmos evolucionários. Algoritmos genéticos são um tipo de algoritmos evolucionários e mostra-se como um diferencial da aplicação aqui desenvolvida, uma vez que algoritmos genéticos foram criados justamente para simular o processo de evolução através da seleção natural. Imagina-se que o mesmo possa fornecer um modelo adequado de desenvolvimento de uma população e suprir os requisitos na representação de um processo de evolução natural simulado por computador. Com isto em mente, o trabalho de desenvolvimento do programa *Bachome* utiliza-se das manipulações de população de um AG para modelar o processo de desenvolvimento de uma espécie. As variações e microevoluções são repassadas aos usuários por meio de uma interface amigável e de fácil entendimento, ao passo que o usuário pode também alterar as configurações do ambiente e assim interagir com a mesma.

Ao questionar-se sobre as alterações na população, os estudantes estarão também refletindo sobre o processo de seleção natural e como os indivíduos sobrevivem de acordo com a pressão do meio. Concomitantemente, o AG fornece alterações aleatórias e seleção dirigida, aspectos básicos da evolução.

Desenvolver uma aplicação baseada em AG torna-se também um estímulo para uso por alunos dos cursos de graduação em ciência da computação, pois o código em linguagem de programação Java, aberto e modular, fornece a possibilidade de uso durante todo um curso de computação evolucionária. A aplicação pode ser utilizada inicialmente para recordação e apreensão de conceitos da biologia evolutiva e, posteriormente, para inspeção, modificação e desenvolvimento de código.

Biologia evolucionária, algoritmos genéticos e a aplicação *Bachome* serão tratados em mais detalhes adiante neste trabalho.

1.3 Justificativa

Ao passo que jogos virtuais comerciais que exploram modelos científicos podem ser entusiasmantes e imersivos, seu foco é primordialmente entretenimento. Por isso, mesmo quando têm um meio de suportar educacionalmente o jogador, geralmente apresentam dificuldades em manter a fidedignidade ao tópico que se aninham, quando estão fortemente relacionados com o ensino de tópicos mais científicos. Como exemplo, temos o jogo *Spore* que suportava em seu plano de *marketing* uma precisão científica dúbia, e mostrava-se mais interessado em oferecer uma jogabilidade divertida (BOHANNON, 2008). O jogo

mostrava incoerências, tais como não mostrar microevoluções, mas, ao contrário, mudanças completas e escolhidas pelos jogadores de uma vez. O jogo falha em demonstrar as variações e aleatoriedade, conceitos essenciais para a evolução (HERRERO et al., 2014, 28). É necessário o acompanhamento de um professor para que tais problemas sejam explicitados.

No âmbito das simulações, que são notadamente voltadas para o aprendizado, temos uma redução de interatividade, mas ganho em foco. Os trabalhos citados cumprem seu papel, no entanto apresentam alguns problemas em comum, principalmente em relação a quantidade de serviços por eles oferecidos. No *Origami Bird Simulator* e no *PopGen FishBowl* os algoritmos utilizados para controlar a simulação podem ser tidos como AEs, utilizando uma espécie de dominância de genes, mutação e cruzamento. Porém, este funcionamento mostra-se como uma caixa preta, e nenhuma tentativa é feita para fornecer controle sobre estes aspectos internos. O Avida-ED é baseado no software Avida, e, ao contrário, é uma ferramenta relativamente complexa, especialmente quando consideramos alunos dos ensinos fundamental e médio. Seria mais adequado dividir a manipulação das aplicações em níveis, onde alunos do ensino básico acessam a parte mais básica das configurações e alunos do ensino superior podem compreender o restante. Além de se diferenciar pelo uso do AG, que simula o próprio processo de seleção natural e evolução de forma abstrata, uma das propostas do Bachome é fornecer este tipo de separação, onde a configuração do AG está disponível para manipulação, mas apenas os usuários interessados precisam entrar em contato com elas.

A pesquisa bibliográfica revelou ainda que existem raras simulações e aplicações interativas em Português brasileiro. Isso dificulta o entendimento dos alunos que não dominam o idioma inglês. A aplicação aqui proposta tem toda a sua interface no idioma português, incluindo o documento de ajuda para os usuários. A mesma figura como objeto de produção intelectual nacional e, mais especificamente, da Universidade Federal do Pará.

1.4 Objetivos

Este trabalho tem como objetivo geral desenvolver a ferramenta de software aqui denominada Bachome, suas principais características e sua utilização como uma ferramenta de ensino. O software deve ser capaz de utilizar um algoritmo genético para evoluir uma colônia de “bactérias” e representar o processo evolutivo de forma gráfica e compreensível para o usuário. Ao final deste trabalho, tem-se uma descrição da realização de experimentos comprobatórios das qualidades da aplicação desenvolvida.

Como objetivos específicos e consequentes contribuições, temos:

- Utilizar um algoritmo genético para evoluir uma população de “bactérias”. Bactérias

são a metáfora utilizada como representação da população apresentada ao usuário. A apresentação da população será atualizada a cada iteração do AG.

- Fornecer ambientação gráfica para imersão do usuário. O ambiente gráfico é colorido, e visualmente interessante, fornecendo fácil diferenciação entre as classes de Bactérias-indivíduo utilizadas e a configuração populacional.
- Permitir ao usuário interagir com diversos aspectos da aplicação por meio da aplicação gráfica. Inclui-se como objetivo permitir ao usuário escolher um cenário para a população e os fatores de pressão do ambiente. Os usuários mais avançados se beneficiarão podendo escolher também aspectos relacionados ao algoritmo genético como tipo de cruzamento, mutação e seleções de sobrevivência e cruzamento.
- Disponibilizar alteração manual no código da aplicação. Os usuários mais avançados podem alterar, inserir e retirar partes de interesse da aplicação. Com codificação em linguagem de programação Java, as classes podem ser alteradas facilmente seguindo a organização lógica da aplicação.
- Fornecer documentação de ajuda para que os usuários possam entender cada parte da aplicação.
- Também é objetivo deste trabalho fornecer um ambiente em língua Portuguesa brasileira. Especialmente para alunos cuja única língua de domínio é o Português. Dessa forma, fornecer uma aplicação com este parâmetro possibilita entendimento completo sobre todos os seus aspectos para alunos no território nacional.
- Validar a ferramenta. O presente trabalho trata de criar cenários teste, detalhá-los e aplicar em ambiente escolar sob a supervisão de um professor. Os alunos participaram de um experimento sob a metodologia descrita adiante neste trabalho para verificar a eficácia da ferramenta como instrumento de apoio.

1.5 Estrutura do trabalho

Este trabalho está organizado da seguinte forma: O capítulo 2 traz a fundamentação teórica necessária para o entendimento das premissas e dos conceitos básicos da biologia evolutiva, para o leitor não habituado, e estes conceitos são em seguida aplicados na descrição de algoritmos genéticos e nos operadores de população empregados. O capítulo 3 traz o detalhamento da ferramenta, esboça aspectos como organização, modelagem e componentes utilizados. O capítulo 4 mostra os resultados de uma bateria de testes realizados sobre a aplicação e, logo após, é descrita a aplicação de uma metodologia de validação em sala de aula para a ferramenta. Por fim, o trabalho se encerra com a conclusão e trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo serão expostos definições e conceitos a serem utilizados durante o restante dos capítulos. Primeiramente é feita uma rápida revisão sobre a teoria da evolução, onde são abordados temas que serão úteis para entendimento do contexto geral da aplicação e que servem como base para os algoritmos genéticos. Em seguida, são apresentados os principais conceitos para o entendimento geral de AGs e dos módulos implementados na aplicação.

2.1 Biologia Evolutiva

A ideia prevaiente na época de Darwin era a fixidez das espécies, embora naturalistas e filósofos já houvessem questionado esta ideia anteriormente (RIDLEY, 2006). Um dos estudiosos que trouxe à tona esta discussão foi Jean-Baptiste Lamarck, especialmente com seu trabalho denominado *Philosophie Zoologique* de 1809. Ridley (2006) conta que neste trabalho Lamarck argumenta que as espécies mudam lentamente com o tempo, talvez até o ponto em que seus indivíduos tornem-se diferentes o suficiente para serem considerados uma nova espécie. Ele desenvolveu um mecanismo em duas partes para explicar estas mudanças: a primeira parte sugere uma “força interna” desconhecida que leva o indivíduo a produzir uma prole levemente diferente e a segunda diz que as características adquiridas em vida são passadas para a prole, geralmente características associadas a esforço e desejo de mudança.

Durante uma viagem, Charles Darwin estudava os pássaros das ilhas Galápagos. Ele supunha que os tentilhões do arquipélago deveriam todos pertencer à uma espécie, mas em suas observações constatou que cada ilha tinha sua única e distinta espécie. Ficava fácil imaginar então que os tentilhões deveriam ter um ancestral comum. Bastava criar uma teoria que explicasse porque as espécies mudam. Ele considerou utilizar as teorias que existiam, inclusive a de Lamarck, mas descartou-as, pois demonstrou que nenhuma podia explicar o conceito de adaptação.

Darwin encontrou a solução para o processo de adaptação e descreveu em sua teoria da seleção natural. Ridley (2006, p. 34) explica que

Devido à luta pela vida, formas que são mais bem-adaptadas à sobrevivência deixam uma progênie maior e automaticamente aumentam em frequência de uma geração para a outra. Como o ambiente muda ao longo do tempo (por exemplo, do úmido para o árido), diferentes formas de uma espécie estarão mais bem-adaptadas a ele do que as formas do passado. As formas mais bem-adaptadas terão sua frequência aumentada, enquanto as formas mal-adaptadas terão sua frequência diminuída.

Uma das críticas de cunho mais popular que era feita à teoria da evolução, e que prevalece ainda hoje, era taxar a evolução como sendo um processo aleatório. No entanto, vê-se que apesar das alterações na descendência serem de cunho aleatório, a seleção natural não o é, tornando a evolução um processo dirigido. Este é um dos erros mais comuns concebidos por estudantes (GREGORY, 2009).

As ideias de Darwin foram resumidas no clássico livro *On the Origin of Species* (DARWIN, 1859). Essas teorias foram, em geral, bem aceitas pela comunidade acadêmica, mas a teoria da seleção natural carecia de um mecanismo que explicasse a hereditariedade satisfatoriamente. Os modelos apresentados até então não eram convincentes e o de Darwin, um modelo de miscigenação dos progenitores, também recebia fortes críticas. Tal problema foi apenas superado quando a hereditariedade mendeliana passou a ser utilizada (RIDLEY, 2006).

A hereditariedade mendeliana, mesmo sem o conhecimento de material genético, mostrava como e em que proporção as características dos pais eram passadas para a prole seguindo o conceito de genes e dominância. Dessa forma, era possível explicar a passagem de características para a descendência. Na década de 1920 as pesquisas sobre a genética mendeliana ampliaram-se, e alguns anos depois originou-se o que é atualmente conhecido como neodarwinismo, ou teoria sintética da evolução, ou ainda síntese moderna (RIDLEY, 2006). A teoria da evolução que conhecemos hoje, surge desta união entre os conceitos de hereditariedade mendeliana e a evolução de Darwin.

Com o avanço da genética, descobriu-se que a hereditariedade, passagem de características entre progenitores e sua descendência, em quase todos os seres vivos é causada pelas moléculas de DNA (sigla para Ácido desoxirribonucleico, do inglês, *Deoxyribonucleic acid*). O DNA carrega as informações para a construção de novos corpos e está presente em quase todas as células do corpo e em todas as células reprodutivas.

O DNA é composto de uma sequência de unidades, chamadas de nucleotídeos. A porção total de nucleotídeos pode ser separada em genes, que contém informações para codificação de proteínas, e porções não-codificadoras. Não necessariamente cada gene é formado por porções contíguas de nucleotídeos, mas podem estar espalhados pelo código genético. A característica principal é que um gene, ou um grupo deles, fornece informação para a codificação de uma ou mais proteínas. As proteínas, por sua vez, em uma forma simplificada, constroem, regulam, mantêm e defendem o corpo (RIDLEY, 2006).

Quando uma célula reproduz, seu DNA é replicado. Entretanto alguns erros podem ocorrer durante o processo físico de cópia. Estes erros são chamados de mutações e o DNA resultante da mutação pode codificar para uma forma de proteína diferente da original. Quando mutações ocorrem na produção de gametas, essas alterações são repassadas para os descendentes podendo ocasionar em diferenças entre eles e os progenitores.

O descendente em um processo de reprodução sexuado recebe parte do material genético de cada pai, gerando indivíduos que podem expressar características de ambos. Associado à isso, a mutação adiciona variabilidade à população. A variação entre os indivíduos de uma espécie decorre, portanto, pelo surgimento espontâneo de mutações.

Um erro muito comum é acreditar que durante o processo de evolução, uma mutação é responsável por uma alteração significativa ou que a criação de estruturas é resultado de um avanço único. Por exemplo, o surgimento da primeira asa em um ancestral dos pássaros não se deu pela alteração causada por uma única mutação, mas um processo gradual de aquisição. Ocorrem transformações graduais de entidades primordiais, simples o suficiente para terem surgido por acaso. As alterações sucessivas são simples o suficiente em relação às anteriores para acontecerem a esmo, mas ao final a acumulação dos passos simples forma um produto complexo. O processo cumulativo, no entanto, não é aleatório uma vez que é dirigido pela sobrevivência na seleção natural (DAWKINS, 1991).

Muitos argumentam que em estágio inicial, a proto-asa, seria na verdade um empecilho para o pássaro, e portanto, uma desvantagem no processo de seleção natural. No entanto, a proto-asa poderia ter alguma utilidade em estágios iniciais como amortecimento de quedas e, com seu desenvolvimento, ajudaria em atividades como planar e posteriormente, voar.

A evolução é uma temática bem difundida, mas, ainda assim, é uma área do conhecimento que poucos dominam (MOORE et al., 2002). Todos os conceitos rapidamente citados nesta seção se entrelaçam em uma teoria consolidada e forte para explicar a origem das espécies. Auxiliar o estudante no aprendizado correto deste tema representa uma melhora nos seus conhecimentos acadêmicos e auxilia na quebra de preconceitos adquiridos devido a errônea compreensão dos mecanismos relacionados à evolução e passados culturalmente.

2.2 Algoritmos Genéticos

Em 1960 algoritmos genéticos foram inventados por John Holland, e posteriormente desenvolvidos por ele e seus estudantes. Ao contrário de estratégias e programação evolutivas, o objetivo inicial de Holland não era criar um algoritmo para resolver problemas específicos, mas estudar o fenômeno da adaptação, como ele acontece na natureza e desenvolver formas pelas quais adaptação natural poderia ser importada para dentro de computadores (MITCHELL, 1996).

A estratégia de algoritmos genéticos consistia em, dadas uma população de cromossomos, constituídos de “genes” 0’s e 1’s (representação binária), gerar uma nova população com um mecanismo baseado na seleção natural para escolha de indivíduos para reprodução,

juntamente com operadores de cruzamento, mutação e inversão¹.

A maioria das aplicações de algoritmos genéticos empregam indivíduos do tipo haploide (que possuem uma única fita de DNA e, portanto, não há dominância), particularmente, indivíduos com um único cromossomo (MITCHELL, 1996). A configuração do cromossomo determina o genótipo do indivíduo.

Os algoritmos genéticos, assim como os algoritmos evolucionários, utilizam um *workflow* básico para o seu funcionamento representado no digrama de atividades da Figura 1. O *loop* compreendido entre as tarefas **Seleção de Pais** e **Avaliar indivíduos** representa uma iteração do modelo. Por fim, a decisão **Finalizar?** é definida pelo critério de parada estipulado. Um critério muito comum é definir um número limite de iterações (SRINIVAS; PATNAIK, 1994).

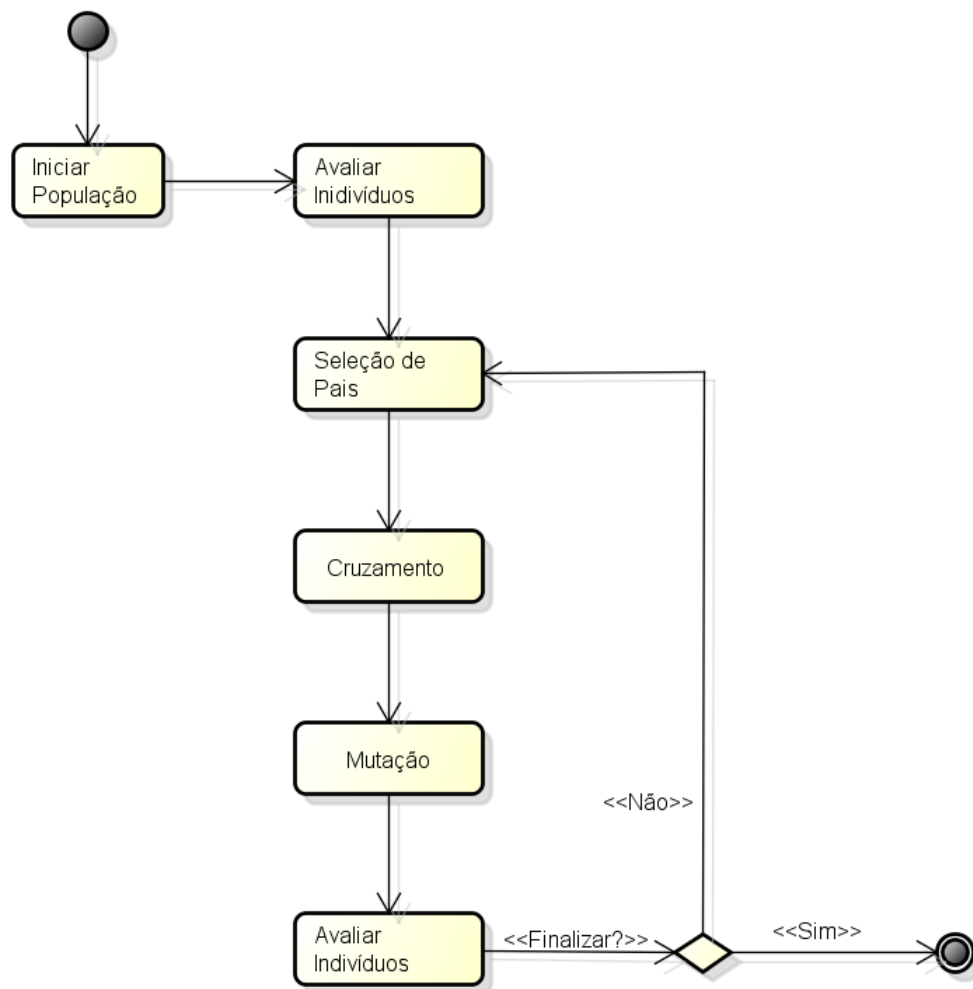


Figura 1 – *Workflow* básico de uma AG.

¹ Inversão é um operador inspirado por operações similares na campo da genética, onde inverte-se parte de um cromossomo, como, por exemplo, a posição de dois alelos. Este operador tem por objetivo manter partes do cromossomo que forneçam vantagem competitiva. Entretanto, ele não oferecia muita vantagem na performance do algoritmo e acabou sendo esquecido para a maior parte dos estudos (MITCHELL, 1996).

O algoritmo genético canônico, ou algoritmo genético simples (do inglês *simple genetic algorithm* (SGA)), tem tipos de operadores bem definidos para cada etapa do *workflow*: inicialização de população com representação binária, cruzamento de um ponto de corte, mutação por inversão de gene, seleção de pais proporcional à adaptação dos indivíduos e seleção geracional de sobreviventes. No entanto, com o passar dos anos, diversas variações destes operadores foram desenvolvidas (PAVAI; GEETHA, 2016; SONI; KUMAR, 2014). Com o objetivo explanatório, as subseções seguintes tratam em mais detalhes os componentes e operadores comuns de um algoritmo genético, assim como explicitam os operadores e configurações implementados na aplicação Bachome.

2.2.1 Representação

A população forma a unidade de evolução do algoritmo. A função da população é armazenar um conjunto de soluções para um problema simultaneamente. (EIBEN; SMITH, 2003). Terminologias como “solução candidata”, “fenótipo”, ou simplesmente “indivíduo” são todas utilizadas para designar uma solução específica da população. A solução pode ser obtida decodificando-se o genótipo do indivíduo para o contexto do problema de acordo com a representação utilizada. Para tanto, a operação **Inicializar População** trata de atribuir uma solução aleatoriamente gerada em cada cromossomo de acordo com a representação utilizada.

AGs têm por característica utilizarem como representação de indivíduos palavras (cromossomos) construídas sobre um alfabeto finito, onde cada palavra representa uma solução para o problema. No trabalho original de Holland, apenas genes 0's e 1's eram utilizados, mas representações do tipo inteiras, reais ou outro tipo de alfabeto podem ser manipuladas. Este trabalho utiliza representação real, ou seja, cada gene é um valor compreendido no intervalo $[0,1]$. Esta codificação foi escolhida pois cada indivíduo da aplicação possui algumas características, chamadas níveis de resistência, no domínio dos reais. O mapeamento entre uma solução e o problema torna-se direto.

2.2.2 Função de Avaliação

A avaliação forma a base para a seleção. Do ponto de vista do problema a ser resolvido, ela representa o contexto de evolução. Sua função então é designar uma métrica de avaliação para cada indivíduo da população de acordo com sua qualidade como solução (EIBEN; SMITH, 2003). Esta métrica fornece a aptidão do indivíduo, ou *fitness*.

Por exemplo, se o problema consiste em encontrar uma solução que maximize a função $f(x) = 2x$, então uma métrica de avaliação poderia ser assinalar a cada indivíduo i da população o valor de *fitness* f_i , onde $f_i = f(x_i)$ e x_i é o valor do cromossomo de i após a decodificação para o domínio do problema por uma função $d(i)$. Se a população tem

quatro indivíduos I_1 , I_2 , I_3 e I_4 , cujos valores após decodificação são, respectivamente, 4, 3, 2 e 1, o valor de *fitness* resultante é expresso na Tabela 1.

Tabela 1 – Aptidão de quatro indivíduos para a função de avaliação $f(x) = 2^x$

Indivíduo	$d(I_i)$	<i>fitness</i>
I_1	4	16
I_2	3	8
I_3	2	4
I_4	1	2

A função de avaliação é intrínseca ao problema a ser resolvido. No caso da aplicação Bachome, envolve um conjunto de fatores como o meio escolhido pelo usuário e competição por recursos. Os indivíduos melhor adaptados recebem um maior *fitness*. Esse tipo de avaliação foi denominado “Natural”.

2.2.3 Seleção de Pais

A seleção de pais faz o sorteio dos indivíduos que gerarão os indivíduos da próxima geração. Este mecanismo simula o processo de seleção natural, onde pais mais aptos tendem a gerar mais descendentes. Para tanto, a probabilidade de que um indivíduo i da população será escolhido depende da proporção de seu *fitness* comparado com o restante da população (LINDEM, 2008).

O SGA tem como mecanismo de seleção a proporcionalidade do *fitness*, ou roleta viciada, onde o *fitness* de todos os n indivíduos é somado e cada indivíduo i , com *fitness* f_i , tem uma probabilidade $P(i) = f_i / \sum_{j=1}^n f_j$ de ser sorteado. Por exemplo, utilizando os indivíduos da Tabela 1 em uma roleta, os segmentos representados na Figura 2 demonstram a proporcionalidade da probabilidade de seleção de cada um.

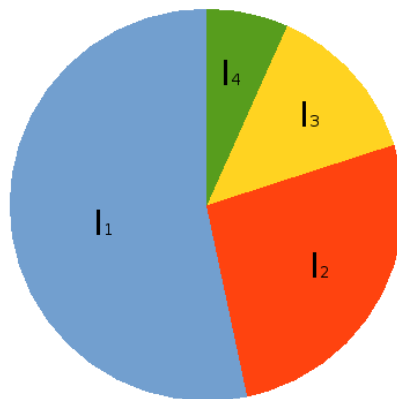


Figura 2 – Probabilidade de Seleção de acordo com o *fitness*.

Seleção por torneio é um operador que não requer um conhecimento global da população, nem uma medida confiável de qualidade. Este operador funciona como segue:

dados k indivíduos selecionados as acaso, retorna o que tem maior aptidão. Segundo [Eiben e Smith \(2003, p. 86\)](#), a seleção por torneio é o método mais utilizado em AGs devido sua simplicidade e possibilidade de controle da pressão de seleção pelo parâmetro k .

Os dois métodos, seleção por roleta e torneio, são disponibilizados na aplicação para escolha do usuário e o método inicial é a seleção por torneio.

2.2.4 Operador de Cruzamento

Baseados no processo de cruzamento genético, ou *crossover*, este operador tem por objetivo recombinar blocos de construção de genótipos em novos genótipos. A recombinação dos pais acontece segundo uma probabilidade P_c , chamada probabilidade de cruzamento ou taxa de cruzamento ([SRINIVAS; PATNAIK, 1994](#)). A forma mais simples de cruzamento é o cruzamento de um ponto utilizado no SGA. Neste método, dois pais selecionados têm seu cromossomo dividido em uma posição aleatória e material “genético” é trocado para formar dois novos indivíduos. A Figura 3 evidencia o processo descrito utilizando como exemplo dois cromossomos com representação binária. Na figura à esquerda, dois cromossomos pais são escolhidos e um ponto de corte é sorteado entre os alelos três e quatro. À direita, dois cromossomos filhos são gerados pela troca dos segmentos.

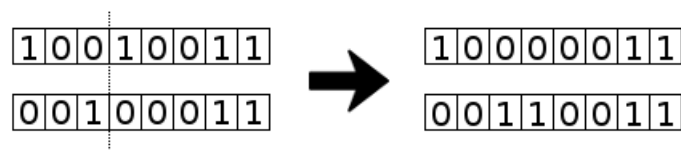


Figura 3 – Processo de cruzamento de um ponto de corte.

O cruzamento aritmético total é utilizado para cruzamento em representação real. Trata de realizar uma soma ponderada dos alelos dos pais de acordo com um peso α , gerado por distribuição uniforme. O primeiro filho tem cada alelo $a_i = \alpha x_i + (1 - \alpha) y_i$ e o segundo filho tem cada alelo $b_i = \alpha y_i + (1 - \alpha) x_i$, onde x_i e y_i são os valores de alelos nas posições i do primeiro e segundo pais, respectivamente.

Os cruzamentos de um ponto de corte, padrão da aplicação, e aritmético total foram implementados na aplicação. Além destes, um terceiro operador, chamado replicação, foi implementado cuja única função é clonar um pai selecionado pela operador de seleção de pais para a população seguinte, sem sofrer cruzamento, apenas mutação. A intenção aqui é reproduzir o comportamento de replicação, reprodução assexuada, que alguns micro-organismos possuem.

2.2.5 Operador de Mutação

O objetivo da mutação, assim como na evolução, é introduzir variabilidade e inovação na população. Lindem (2008, p. 78) ressalta que a mutação ocorre segundo uma probabilidade P_m baixa, caso contrário o algoritmo se parecerá muito com técnicas de busca aleatória.

O procedimento padrão de mutação do SGA é a inversão de gene, que consiste em trocar o valor de um gene escolhido aleatoriamente de 0 para 1, ou vice-versa. Para representações reais, um sistema semelhante é utilizado, chamado mutação uniforme, que consiste em atribuir um novo valor real, dentro do intervalo da representação, obtido de forma aleatória. No caso da aplicação aqui tratada, no intervalo $[0,1]$. Outra forma de alterar o valor do gene é atribuindo um desvio de seu valor segundo uma distribuição Gaussiana, com o objetivo de reduzir a discrepância entre os valores original e mutado do gene. Esse tipo de mutação é geralmente referenciado como mutação Gaussiana.

Foram implementados para o operador de mutação na aplicação os três tipos de mutações e, por padrão, a mutação uniforme é o operador inicial.

2.2.6 Seleção de Sobreviventes

Seleção de sobreviventes, ou módulo de população, é semelhante ao processo de seleção de sobreviventes na natureza. Considerando a população anterior formada pelos pais e a população criada pela sua descendência, a seleção de sobreviventes consiste em eliminar parte da população por diferentes estratégias.

A estratégia geracional, utilizada pelo SGA, apenas substitui a população de pais pela população de filhos. Esta estratégia é simples e direta, mas não demonstra grande verosimilhança com ambientes biológicos reais. O AG da aplicação Bachome acrescenta uma forma de seleção personalizada que soma mais de um fator para escolher os indivíduos a perecerem: quantidade de indivíduos no meio, que simula competição por recursos, tempo de vida, e pressão seletiva do ambiente escolhido pelo usuário.

Na operador de seleção de sobreviventes nomeado “Natural”, o *fitness* é determinado por $\sum_i^n gene(i)factor(i) + Aleat[0,0.1]$, onde *gene(i)* representa o valor no intervalo $[0,1]$ de cada gene *i* de um indivíduo, *factor(i)* é 0, se o fator de seleção *i* está inativo e 1, caso contrário, e *Aleat* $[0,0.1]$ representa uma distribuição de variável aleatória no intervalo $[0,1]$, inserindo um pequena fator de aleatoriedade no configuração do *fitness*. Dessa forma, os genes correspondem, individualmente a um fator de seleção diferente e em proporções variáveis. Uma vez que a quantidade de indivíduos na população corrente ultrapassa o limiar P_l , apenas os P_l indivíduos mais aptos (com maior *fitness*) sobrevivem. Os indivíduos que alcançaram o limiar de idade I_i têm seu *fitness* rebaixado, impreterivelmente, à zero.

2.2.7 Operadores Disponíveis

Com o objetivo de sumarizar as informações apresentadas, a Tabela 2 resume os operadores implementados na aplicação e as suas versões ativadas por padrão na aplicação. Cada um dos itens representados por “usuário” fazem referência à uma classe que o usuário pode preencher para ter seu operador próprio definido.

Tabela 2 – Operadores implementados e suas variantes.

Operador	Implementações	Padrão
<i>Fitness</i>	Natural Usuário	Natural
Seleção de Pais	Roleta Torneio Usuário	Torneio
Seleção de Sobreviventes	Geracional Natural Usuário	Natural
Cruzamento	Um ponto de corte Replicação Aritmético Usuário	Um ponto de corte
Mutação	Inversão Uniforme Gaussiana Usuário	Inversão

3 Descrição da Ferramenta

A ferramenta Bachome (Bacteria's Home) tem por objetivo auxiliar no aprendizado da evolução natural das espécies com enfoque no processo de seleção natural. A ferramenta, em nenhum aspecto, busca substituir a figura do professor em sala ou fornecer toda a base teórica do assunto, mas trazer um contexto prático para o aprendizado. Para tanto, a aplicação simula uma colônia de bactérias sendo observada por intermédio de um microscópio. Alterações na população são ocasionadas por adição de dois tipos de mutação, representadas por cor e formato, e a seleção dos indivíduos pode ser feita alterando características do cenário da simulação.

As formas de seleção de indivíduos pelo ambiente envolvem limitação de recursos (precisam competir por alimento com diferenciação de aptidão nesta tarefa) e a presença de predadores (indivíduos precisam fugir e conter mecanismos de defesa). Há também três opções para a seleção artificial: aplicação de antibiótico, desregulação de temperatura e desregulação de PH (potencial hidrogeniônico). O antibiótico é um tóxico capaz de eliminar bactérias atuando sobre etapas do seu metabolismo, eliminando a maioria dos indivíduos, mas causando a seleção das bactérias resistentes a ele. Este é um assunto amplamente divulgado, uma vez que é um dos principais fatores causadores do crescente aparecimento de bactérias resistentes a este tipo de medicamento. Por outro lado, bactérias podem ser sensíveis a variações ambientais, tais como adição e redução de temperatura, e aumento e redução da acidez do meio. Indivíduos que suportam melhor estas variações de temperatura e PH, não tem o seu metabolismo grandemente afetado e, portanto, têm maiores chances de sobrevivência. Cada um destes fatores favorece um tipo específico de indivíduo na população, ou seja, favorece a classe de indivíduos mais bem adaptada. A utilização de formatos e cores diferentes para cada classe é um aspecto ilustrativo e didático empregado na aplicação.

Existem três tipos de formatos que um indivíduo pode assumir: formato alongado com presença de flagelo, formato esférico e formato triangular. Apesar de os formatos serem baseados em classes de bactérias reais, é tratado aqui de forma meramente ilustrativa e nenhum estudo é realizado ou diferenciação é feita em relação a este tema. Da mesma forma, existem três cores que os indivíduos podem assumir: cinza, verde e vermelho. A Figura 4 ilustra cada um dos nove tipos de indivíduos formados pela combinação de formas e cores. Cada uma dessas classes é adaptada para um tipo de ambiente. Por exemplo, bactérias do formato triangular têm vantagens quando o acesso a recursos alimentares é limitado no sistema e bactérias vermelhas são mais resistentes à variação da temperatura. A Tabela 3 resume as vantagens competitivas oferecidas pelo formato de cada indivíduo e a Tabela 4 lista as vantagens dos indivíduos relacionados à representação de sua cor.

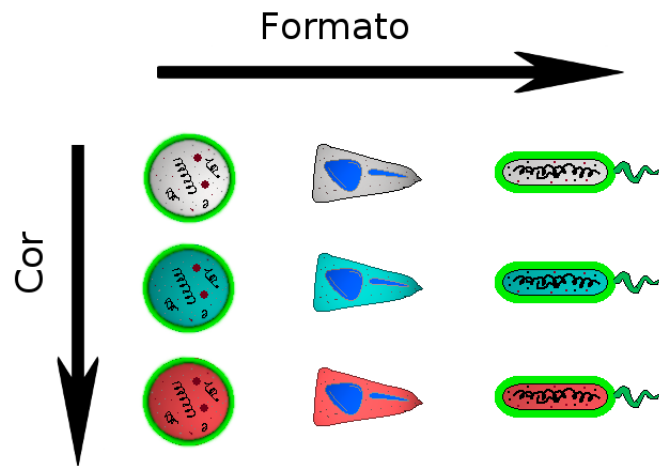


Figura 4 – Combinação de formatos e cores foram nove tipos diferentes de indivíduos.

Devido aos três tipos de formatos e três tipos de cores diferentes, têm-se um total de nove possíveis tipos de indivíduos. Pelas definições apresentadas nas Tabelas 3 e 4 pode-se perceber também que as vantagens oferecidas não são auto-excludentes, ou seja, um indivíduo pode ter vantagens competitivas levando em consideração a sua cor e o seu formato. Dois fatores de seleção que estão sempre presentes, indiferentemente à caracterização de cada indivíduo, são relacionados à quantidade de indivíduos vivos na população e as suas idades. Populações tem potencial para crescer exponencialmente, mas geralmente se mantêm estável depois que atingem um certo tamanho devido à limitação de recursos naturais (GREGORY, 2009). Quando a população alcança um limiar $maxPop$, os indivíduos menos aptos são impreterivelmente eliminados. Da mesma forma, existe um limiar de idade $maxAge$ que determina quando um indivíduo deve ser eliminado por exceder seu tempo de vida.

Tabela 3 – Vantagens oferecidas pelo formato.

Formato	Implicação	Favorecimento
Alongado	Indivíduo rápido	Favorecido quando há seleção com predadores. A bactéria é capaz de fugir com mais facilidade de predadores (anti-corpos e outras bactérias).
Triangular	Maior eficácia para obtenção de recursos	Favorecido quando há seleção por recursos. É mais eficiente na busca por recursos, garantindo sobrevivência a longo prazo.
Circular	Nenhuma vantagem	Esta é a coloração original dos indivíduos (sem aplicação de mutação) e não representa vantagem em qualquer cenário.

Tabela 4 – Vantagens oferecidas pela cor.

Cor	Implicação	Favorecimento
Vermelho	Resistente a variação de temperatura	Esse tipo de indivíduo tem resistência maior quanto à variação da temperatura. A cor vermelha é utilizada para dar indícios desta informação.
Verde	Resistente a Antibiótico	Quando a opção de adição de antibiótico é ativada, este tipo de indivíduo é favorecido pela sua resistência ao químico.
Cinza	Resistente a variação de PH	Este é o formato original dos indivíduos (sem aplicação de mutação) e representa um indivíduo com maior resistência à mudanças do PH do meio.

A aplicação atualiza a interface gráfica em intervalos de tempo t e, em cada uma dessas iterações, o AG é responsável por gerar novos indivíduos e selecionar os sobreviventes. O AG utiliza as informações colhidas na interface para configurar seus operadores e o módulo de avaliação utiliza os fatores de seleção selecionados pelo usuário para calcular o *fitness* dos indivíduos no contexto do AG em cada iteração. A Figura 5 mostra um exemplo de exibição de população na interface criada. No exemplo desta figura, a cor cinza é a cor predominante entre os indivíduos e os três formatos estão presentes.



Figura 5 – Exemplo de configuração populacional representado na interface gráfica.

Considerando os detalhes iniciais e objetivos da ferramenta, este capítulo tem por objetivo detalhar a construção e os aspectos mais relevantes da mesma por meio da análise dos requisitos e detalhes de sua construção.

3.1 Lista de Requisitos

Um conjunto de requisitos foram reunidos para serem implementados. Os requisitos cobrem as funcionalidades mais importante da aplicação Bachome e foram considerados para o levantamento dos casos de uso. Os requisitos foram pensados levando em consideração os aspectos positivos e negativos das aplicações anteriormente analisadas neste trabalho. Também foram incluídos aspectos únicos, como a adição da possibilidade de configuração do algoritmo genético para uso de estudantes interessados em AGs. Os requisitos foram avaliados e alguns dos requisitos finais foram sugeridos pelo professor Claudomiro, orientador deste trabalho e também professor da disciplina de computação evolucionária na Universidade Federal do Pará (UFPA). A Tabela 5 lista os requisitos funcionais do sistema e a Tabela 6 faz o mesmo para os requisitos não funcionais. Cada uma das tabelas traz o identificador (ID), uma breve descrição do requisito e a sua respectiva prioridade.

Tabela 5 – Lista de requisitos funcionais.

ID	Descrição	Prioridade
RF 01	O sistema deve ser capaz de exibir graficamente os indivíduos que estão incluídos na população do algoritmo genético a cada iteração.	Alta
RF 02	O sistema deve oferecer a possibilidade de alterar a representação de indivíduos utilizada no AG.	Alta
RF 03	O sistema deve oferecer as opções de aplicar pressão de seleção por competição por recursos e presença de predação no ambiente.	Alta
RF 04	O sistema deve oferecer a escolha de aplicar pressão de seleção por adição de antibióticos, variação de temperatura e variação de PH.	Alta
RF 05	O sistema deve oferecer a possibilidade de incluir mutações que ocasionem em mudança de formato nos indivíduos.	Alta
RF 06	O sistema deve oferecer a possibilidade de incluir mutações que ocasionem em mudança de cor nos indivíduos.	Alta
RF 07	O sistema deve oferecer a opção de salvar o histórico da simulação.	Baixa
RF 08	O sistema deve exibir graficamente a quantidade de indivíduos de cada tipo na população.	Baixa
RF 09	O sistema deve oferecer a possibilidade de parar e retomar a execução da simulação.	Baixa
RF 10	O sistema deve oferecer a possibilidade de ajustar a pressão em cada item de seleção ativo.	Média

RF 11	O sistema deve oferecer a possibilidade de escolher a seleção de pais utilizada no AG.	Alta
RF 12	O sistema deve oferecer a possibilidade de selecionar o operador de cruzamento para o AG.	Alta
RF 13	O sistema deve oferecer a possibilidade de selecionar o operador de mutação para o AG.	Alta
RF 14	O sistema deve permitir alterar as probabilidades de cruzamento e mutação, P_c e P_m em tempo de execução.	Média
RF 15	O sistema deve possibilitar a seleção do tipo de <i>fitness</i> a ser utilizado pelo AG.	Alta
RF 16	O sistema deve permitir a escolha da seleção de sobreviventes para o AG.	Alta
RF 17	O sistema deve oferecer a entrada de operadores definidos pelo usuário.	Média

Tabela 6 – Lista de requisitos não funcionais.

ID	Descrição	Prioridade
RFN 01	O sistema deve manter uma interface que contenha, no máximo, 30 itens sendo mostrados simultaneamente.	Média
RFN 02	Poucos cliques devem ser necessários para acessar as configurações da simulação.	Alta
RFN 03	Poucos cliques devem ser necessários para acessar as configurações do AG.	Alta
RFN 03	A interface gráfica deve estar disponível em português.	Alta
RFN 04	O algoritmo genético deve estar fracamente acoplado e coeso dentro da aplicação.	Alta
RFN 05	A interface gráfica deve estar fracamente acoplada e coesa dentro da aplicação.	Alta
RFN 06	A aplicação deve estar pronta para execução com suas configurações iniciais.	Alta
RFN 07	O sistema deve ser portátil para computadores com diferentes sistemas operacionais.	Média
RFN 08	O sistema deve economizar recursos de hardware quando executando.	Média
RFN 09	A janela da aplicação deve ser redimensionável.	Baixa
RFN 10	O sistema deve implementar modificações de configuração na simulação tão logo estejam disponíveis.	Alta
RFN 11	O vocabulário da interface do sistema deve ser acessível para alunos do ensino básico.	Alta

RFN 12	A estrutura do algoritmo genético deve ser feita de modo a facilitar a sua expansão por parte do ator Universitário .	Alta
RFN 13	O sistema não deve permitir que o usuário selecione operadores de cruzamento, mutação e avaliação incompatíveis com a representação utilizada.	Alta

3.2 Casos de Uso

O diagrama de casos de uso ilustrado na Figura 6 foi desenvolvido para listar as funcionalidades principais que o sistema deveria conter. O usuário representado na figura pode ser um aluno do ensino básico que executa a aplicação por vontade própria e estaria interessado em testar a aplicação dos diferentes operadores de seleção e mutação da simulação, pode ser um tutor mostrando a simulação para um grupo de alunos e usar as operações disponíveis para auxiliar em sua explanação à respeito da seleção natural, ou ainda um estudante do nível superior, interessado em investigar o funcionamento do AG e, portanto, operar nas opções disponíveis para alteração de módulos e funções de avaliação.

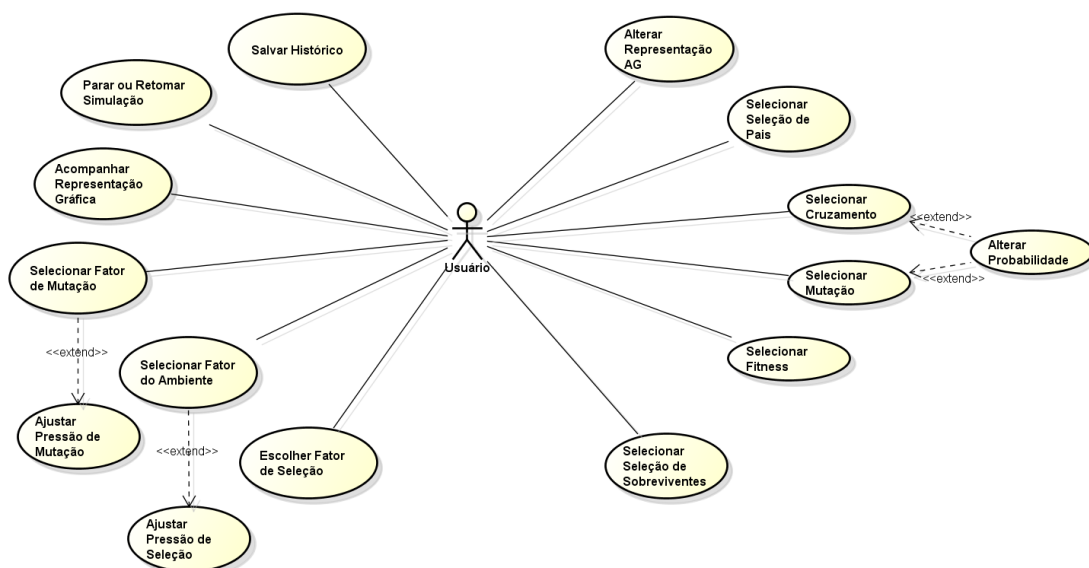


Figura 6 – Diagrama de Casos de Uso para a aplicação Bachome.

3.3 Implementação

Com base na série de itens listados como requisitos, foi desenvolvida uma estrutura que pode ser segmentada em três partes: algoritmo genético, interface gráfica e classe de controle. A Figura 7 ilustra essa divisão lógica. Objetiva-se delimitar claramente a função de cada item, favorecendo a reutilização e alteração do código, uma vez que uma das

funções deste trabalho é servir de base para alterações e experimentações por parte dos alunos do ensino superior.

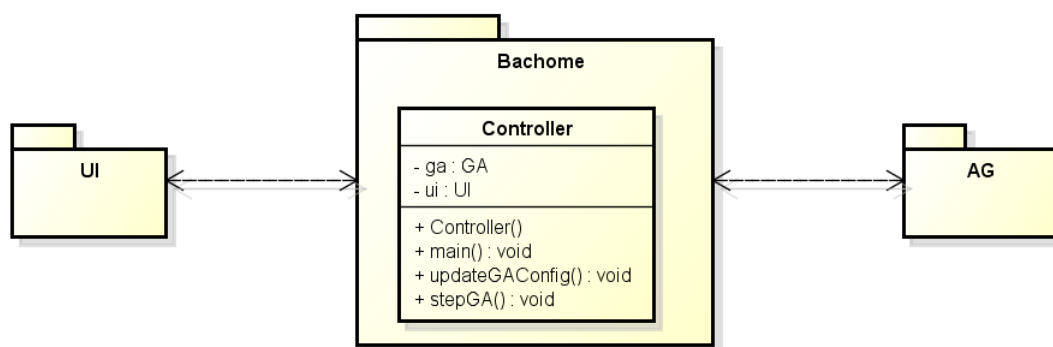


Figura 7 – Componentes da construção da aplicação Bachome.

A classe de controle organiza o fluxo de execução da aplicação e constrói uma ponte entre o AG e a interface gráfica. O AG tem seu funcionamento de forma quase isolada utilizando apenas as informações da interface gráfica repassadas para função de avaliação através da classe de controle, responsável também por recuperar as informações de população do AG, organizá-las e repassar para exibição na classe UI. Dessa forma, a construção segue o padrão de projetos do tipo *façade*, facilitando a alteração e substituição dos módulos e a comunicação com os elementos de cada pacote.

A Figura 8 apresenta as classes que compõem o pacote do algoritmo genético na aplicação. A classe **GA** é a responsável pela comunicação para com a classe de controle e fornece todo o interfaceamento de comunicação necessário. Vale ressaltar que cada operador do AG é representado por uma interface seguindo o padrão de projetos *strategy*. O objetivo é tornar possível alternar entre os diferentes tipos de operadores em tempo de execução e facilitar a implementação de novos tipos sob a interface (GAMMA et al., 2000). Os operadores citados na seção 2.2 estão implementados para cada uma destas interfaces e, acrescido a isto, também estão disponíveis classes com prefixo “*user*” que podem ser preenchidas para a adição de qualquer operação desejada pelo usuário.

O AG possui uma população alvo do processo de evolução, que por sua vez contém indivíduos com um único cromossomo e cada cromossomo têm uma porção de genes. Um ID único é assinalado para cada indivíduo. Esta operação é opcional, mas foi implementada em ordem de facilitar a identificação de indivíduos presentes em gerações passadas. O gene, por fim, pode ser real ou binário, no entanto, apenas uma das representações pode ser utilizada por execução. Por padrão, a representação real é utilizada, mas a binária pode ser selecionada, desde que acompanhada por implementações de operadores condizentes.

Algoritmos genéticos dependem fortemente da utilização de números aleatórios. Portanto, é interessante utilizar um gerador de números pseudo-aleatórios com maior

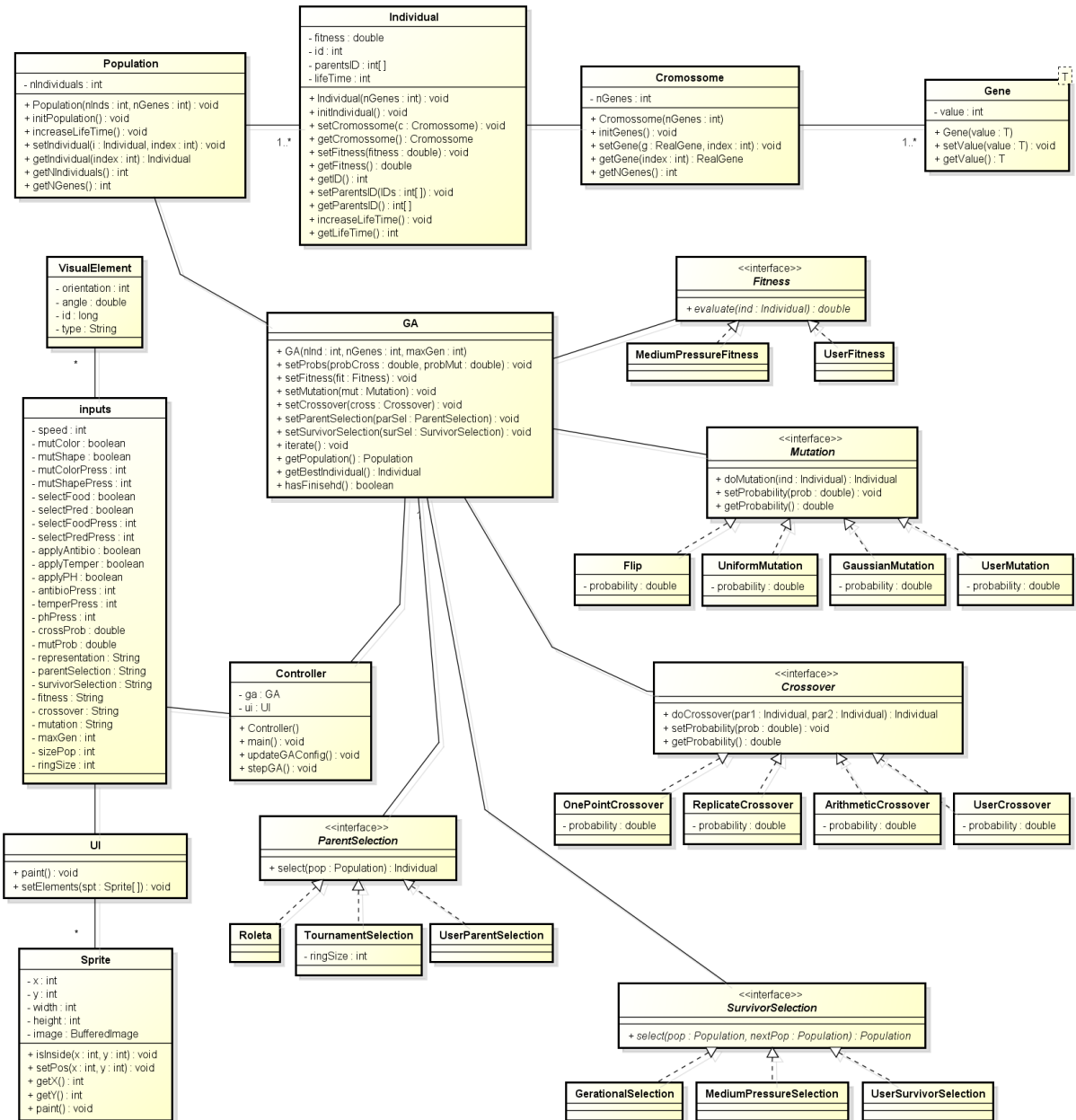


Figura 8 – Diagrama das classes que compõem o AG.

qualidade do que o gerador padrão da linguagem Java. Uma versão eficiente do gerador *Mersenne Twister* (MATSUMOTO; NISHIMURA, 1998) foi empregada para quaisquer operações com números aleatórios na aplicação. A classe **MersenneTwistFast**, de código livre, foi anexada sob padrão de projeto *Singleton*.

A Figura 9 exibe as classes presentes no pacote de interfaceamento gráfico criado para a aplicação. A classe **Inputs** armazena todas as entradas obtidas da interface. Os dados são inseridos na classe por meio de métodos *set* e, quando os dados sofrem alteração, uma variável de alerta é atualizada para que a classe **Controller**, ilustrada na figura 7, possa descarregar as novas informações por meio da interface de métodos *get*.

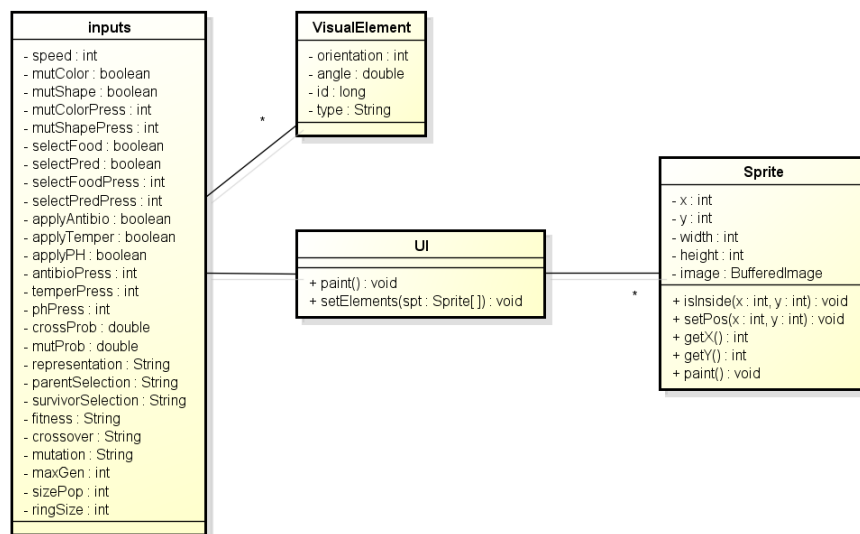


Figura 9 – Diagrama das classes que compõem o pacote **UI**.

A classe **Controller**, a cada iteração, atualiza a lista de itens para mostrar em tela (**VisualElement**) com base na população corrente. A interface gráfica se encarrega de criar elementos visuais de acordo com o tipo explicitado por cada objeto **VisualElement**. Se um indivíduo estava na simulação na iteração passada, sua posição na janela é mantida, caso contrário, o novo indivíduo é alocado em uma posição aleatória do ambiente.

3.4 Interface Gráfica

A Figura 10 demonstra a interface principal da aplicação. Nela são apresentados: um fundo verde, com algumas ilustrações em verde claro, e a representação visual da população do AG. A população inicial da simulação é representada apenas por bactéria circulares de cor cinza. Na simulação ilustrada na Figura 10, foi adicionado o fator de mutação que resulta no incremento de variabilidade por cor.

No canto superior esquerdo da Figura 10 é possível notar a presença de um menu do tipo *drop down* intitulado **Editar**. Esta aba fornece acesso para os painéis de configuração de elementos da simulação e configuração do AG. Estes painéis podem ser conferidos, respectivamente, pelas Figuras 11 e 12.

No painel de configuração de elementos da simulação da Figura 11, as barras deslizantes são ativadas somente quando os seus respectivos elementos de seleção e mutação estão ativos. Por padrão, as mutações e seleções estão desativadas e o único controle exercido é a o tamanho limítrofe da população.

No painel de configuração das propriedade do algoritmo genético, representado na Figura 12, são oferecidas opções para alterar os módulos utilizados pelo AG e as

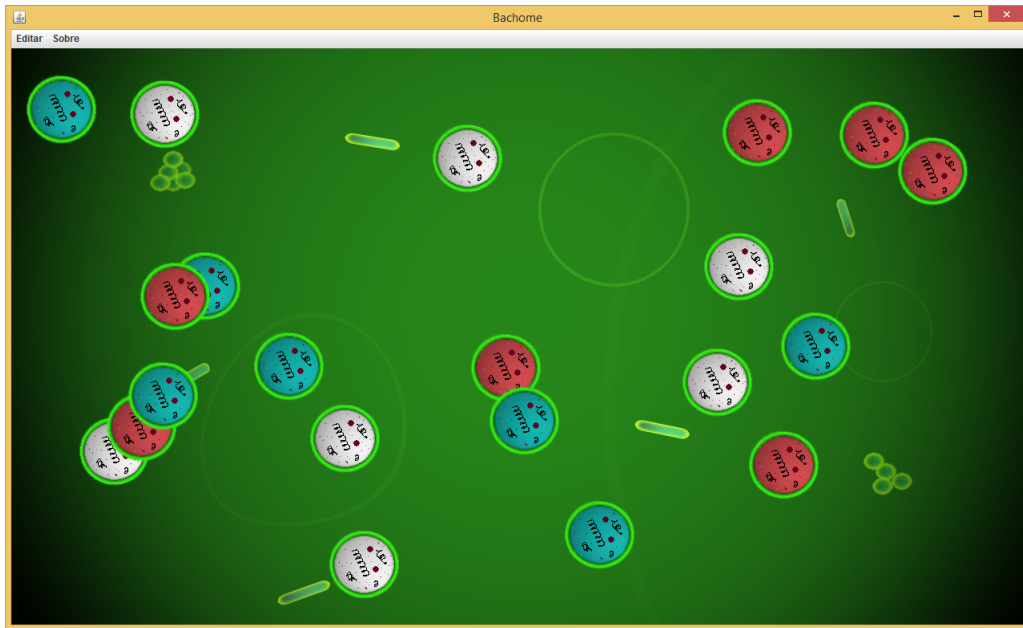


Figura 10 – Apresentação dos indivíduos da população. Este caso de simulação inclui a configuração da população várias gerações após inserção de mutação que ocasiona em representação por diferentes cores.

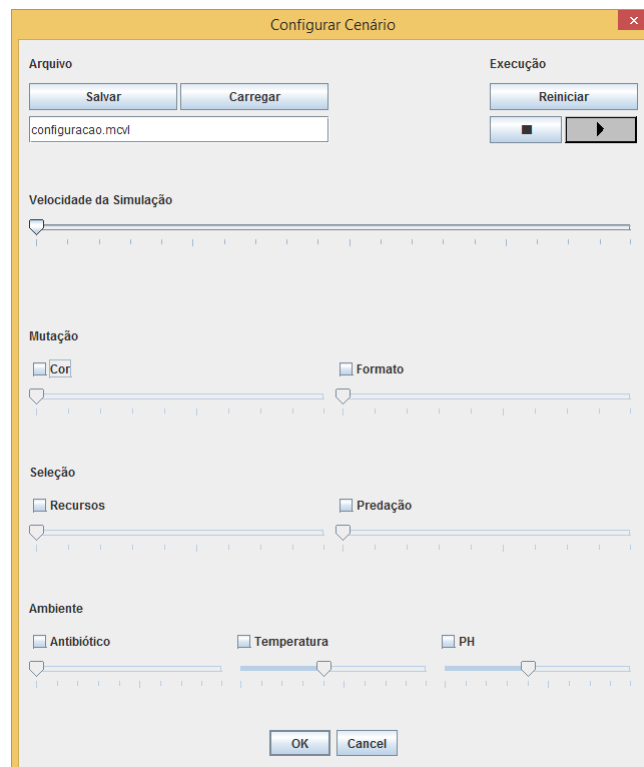


Figura 11 – Painel de configuração para os aspectos gerais da simulação.

configurações que resultem em conflitos são bloqueadas (**RFN 13**). Por exemplo, quando o algoritmo genético está sendo executado sob codificação binária, apenas os elementos passíveis de aplicação neste tipo de representação são selecionáveis nos menus *drop down* de

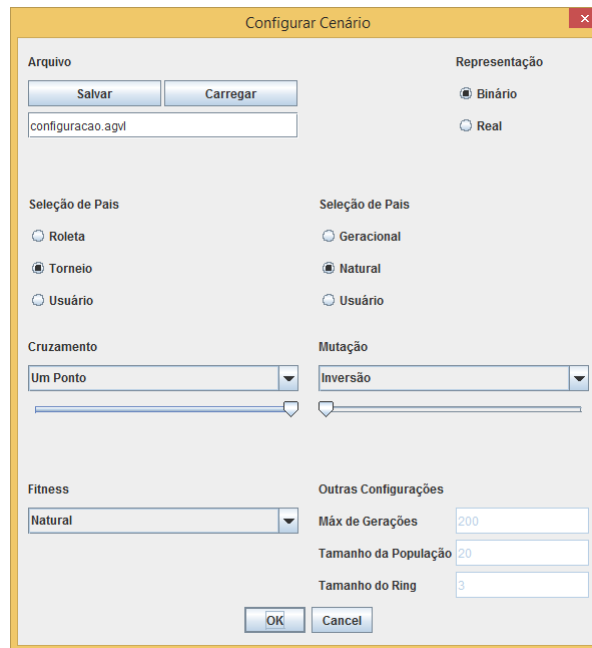


Figura 12 – Painel de configuração para as propriedades do AG.

cruzamento, mutação e avaliação. O mesmo ocorre para a seção **Outras Configurações**, onde tamanho da população, por exemplo, não é editável quando a seleção dos pais **Natural** é escolhida. A Figura 13 mostra a seleção da representação do tipo real. Com esta escolha, ficam disponíveis cruzamento aritmético e mutação gaussiana, como ilustrado na figura. Além disso, foram escolhidos, seleção de pais por roleta, seleção de sobreviventes do tipo geracional e *fitness* disponível para criação pelo usuário.

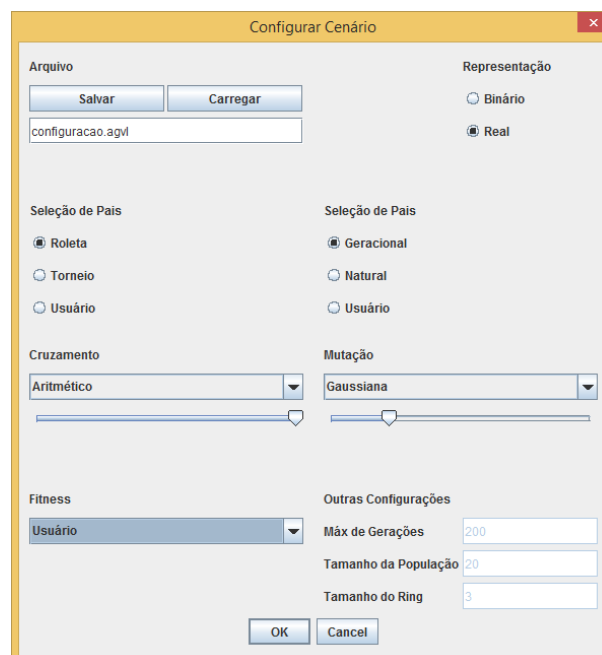


Figura 13 – Painel de configuração para as propriedade do AG com representação real ativada.

4 Resultados

Com o objetivo de avaliar as funcionalidades implementadas na ferramenta, uma metodologia de avaliação em três etapas foi desenvolvida. A primeira etapa consiste em avaliar a desenvoltura do algoritmo em evoluir a população verificando o resultado da aplicação das opções disponibilizadas ao usuário. A segunda etapa consiste na realização de uma entrevista com um professor de biologia dos ensinos fundamental e médio para obtenção de opiniões e críticas. Por fim, a terceira etapa é a apresentação da ferramenta em ambiente de sala de aula para um grupo de estudantes do ensino fundamental sob a supervisão de um professor. As seções seguintes tratam de descrever os resultados obtidos pela aplicação desta metodologia no uso da aplicação Bachome.

4.1 Cenários de Simulação

Para que a aplicação seja bem sucedida na representação do processo de evolução a partir da seleção natural, alguns conceitos básicos precisam ser explorados ([SPETH et al., 2009](#)):

1. A origem da variação: os alunos devem ser capazes de compreender a origem da variação entre as espécies como um processo ocasionado por mudanças aleatórias no código genético. A ideia de variação surgindo pela vontade dos indivíduos para se adaptar ao meio deve ser combatida para evitar, entre outros problemas, a dificuldade na compreensão dos itens que seguem.
2. Hereditariedade: partindo da ideia de variações adquiridas no código genético, o aluno precisa compreender que as variações são repassadas para a prole por meio dos mecanismos de hereditariedade. É também importante entender que indivíduos mais adaptados tendem a gerar uma prole maior, principalmente devido ao maior tempo de vida e sucesso reprodutivo. Isso dará maior probabilidade de difusão de características benéficas em relação ao meio.
3. Adaptação: dada a variabilidade encontrada na população, o aluno deve apreender que os indivíduos são favorecidos pela sua adaptação ao meio, que a pressão de seleção e competição intra-espécie é variável de acordo com as condições ambientais na qual os indivíduos estão inseridos e aqueles que melhor adaptados estiverem, tendem a sobreviver e continuar o ciclo evolutivo.

4. Variação da população: resume as ideias apresentadas, demonstrando que essas mudanças de população ocorrem pela progressiva variação dos indivíduos, que por sua vez, variam pela soma dos fatores genético e ambiental.

Partindo de todo o arcabouço teórico e técnicas aplicados à ferramenta, descritos neste trabalho, a aplicação é capaz de endereçar cada um destes requisitos:

1. A origem da variação: o algoritmo genético introduz variações aleatórias no genoma dos indivíduos a serem gerados em iterações subsequentes. As variações causadas simulam a alteração de alelos da prole e podem ser tanto alterações benéficas quanto prejudiciais ou de caráter indiferente. Essas alterações são exibidas de forma ilustrativa para o usuário por cores e formatos específicos.
2. Hereditariedade: o algoritmo genético simula o processo de hereditariedade trocando o material genético de indivíduos da população corrente para criar indivíduos que estarão na próxima geração. Vale ainda ressaltar que o algoritmo genético gera maior prole para indivíduos considerados mais aptos. Este processo fica claro para o usuário quando um tipo de indivíduo aparece repetidamente na interface.
3. Adaptação: se dá pelas funções de *fitness* e subsequente seleção de sobreviventes, onde os indivíduos com maior aptidão para o cenário corrente têm maiores chances de sobreviver para a próxima iteração. Indivíduos que se mantêm estáticos na aplicação representam os sobreviventes, fazendo-se perceber as características de seleção atuais. Similarmente, é notório o aspecto geral da população quando uma característica se sobressai sobre as demais.
4. Variação de população: representada graficamente na aplicação pelas mudanças de formato e cor.

Para comprovar experimentalmente os argumentos apresentados, um experimento de teste dos itens da simulação tomou lugar. Objetivando verificar o efeito de cada um dos fatores presentes na aplicação isoladamente, primeiramente inicializa-se a simulação e as opções de variabilidade e seleção são sequencialmente habilitadas. As consequências da adição de cada fator são condizentes com as premissas apresentadas, como pode ser conferido na Figura 14. A população é representada de duas formas, a Figura 14a mostra a evolução da população quanto ao formato de seus indivíduos e a Figura 14b se refere no tocante à cor. Primeiramente é acrescentada variabilidade na simulação através da adição de mutações de cor e formato, em seguida seleção por recursos, nenhuma seleção e, posteriormente, predação. Foram também aplicadas seleções por adição de antibiótico, variação de temperatura e variação de PH. Cada uma destas operações é realizada na ordem apresentada nos instantes de tempo t_i , onde i varia de um a oito. Os fatores de

mutação aplicados são acumulativos e as operações de seleção são excludentes, ou seja, a partir do momento em que uma operação de mutação é adicionada como fator, ela não é retirada, porém para as operações de seleção, apenas uma ou nenhuma está ativa em cada intervalo de tempo.

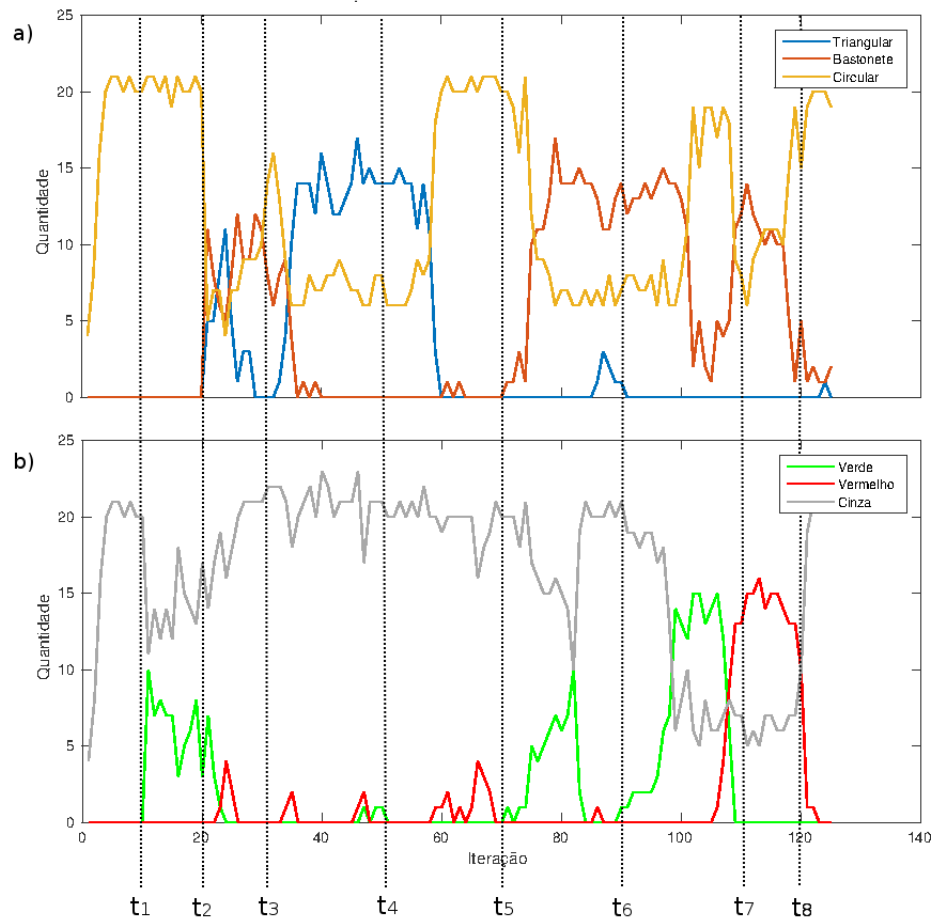


Figura 14 – Configuração populacional após a aplicação de diversas operações de mutação e seleção nos instantes de tempo t_i . Em a), divisão da população por formato e em b), divisão por cor.

Os fatores experimentados nos instantes t_i na Figura 14, são apresentados e suas consequências descritas na Tabela 7. A Figura 15 mostra a população nos instantes t_i sendo representada na interface gráfica.

Tabela 7 – Efeitos da adição de mutação e seleção.

Tempo	Operação	Efeito
t_1	Permitir mutação	Depois da adição deste tipo de mutação, observou-se o aparecimento de indivíduos com a cor verde no gráfico de diferentes cores.

t_2	Permitir mutação representada por diferentes formatos.	Com a adição deste tipo de mutação passaram a existir indivíduos com os três tipos de formatos na população na Figura 14a.
t_3	Aplicação de seleção por recursos.	Por definição, bactérias com formato triangular são mais eficazes na busca por recursos para sobrevivência. No momento que a operação foi realizada não haviam bactérias com este formato na população na Figura 14a, mas depois de algumas iterações, pode-se perceber o surgimento de bactérias deste tipo. Isso implica que um indivíduo com formato triangular surgiu por causa de uma mutação e em poucas iterações a sua descendência se tornou a mais numerosa na população devido à sua vantagem seletiva.
t_4	Retirada do fator de seleção.	Sem fatores de seleção, a população manteve-se estável por um período e, por um evento aleatório, a porção de bactérias com formato triangular passou a morrer.
t_5	Aplicação de seleção por predação.	Por definição, bactérias de formato alongado são mais rápidas e tem maiores chances de sobrevivência na presença de predadores. Isso explica o rápido crescimento deste tipo de indivíduo na Figura 14a.
t_6	Aplicação de antibiótico.	Por definição, bactérias representadas pela cor verde têm resistência à antibióticos e, devido este fator, pode-se observar um aumento exponencial da população de bactérias verdes em relação às demais.
t_7	Variação da temperatura.	Bactérias representadas pela cor vermelha são, por definição, resistentes à variação de temperatura, contribuindo para a sobrevivência e, conseqüentemente, crescimento populacional deste tipo de indivíduo observando o gráfico da Figura 14b.
t_8	Variação do PH.	Por fim, bactérias representadas pela cor cinza sofrem considerável aumento populacional pela sua resistência à variação de PH do meio. A maioria das bactérias cinza na população no instante t_8 eram também bactérias de formato circular, o que fez este tipo de formato sobressair-se concomitantemente, apesar de esta característica ser considerada irrelevante para o fator de seleção atual.

A aplicação é capaz de agregar mais de um fator de seleção simultaneamente. A Figura 16 mostra a configuração populacional 20 iterações depois da aplicação de

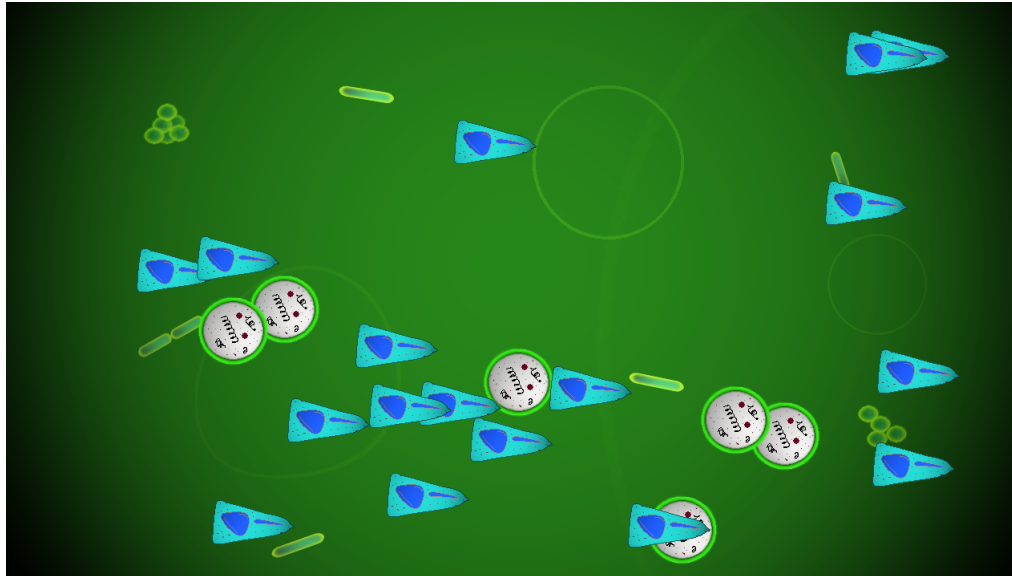


Figura 16 – Configuração populacional após aplicação de antibiótico e seleção por recursos.

4.2 Validação com Professor

Para validar a proposta em ambiente prático, a aplicação foi experimentada com um grupo de sete alunos do nono ano do ensino fundamental sob a supervisão do professor Osiris no Colégio de Ensino Icoaraciense (CEI, s.d.). De antemão, uma pequena entrevista informal foi conduzida com este professor para que fosse possível identificar pontos positivos e negativos da aplicação e verificar quão profundamente o assunto poderia ser esboçado para o grupo. O professor forneceu as seguintes opiniões sobre os alunos:

- O conhecimento dos estudantes é básico, mas suficiente para entender os principais conceitos passados na aplicação. A maior dificuldade de assimilação estaria relacionado com a genética e hereditariedade.
- Os estudantes têm problemas para entender a diferenciação entre seleção natural e evolução.
- Eles conhecem vários exemplos práticos no contexto, mas cometem erros ao tentar explicá-los.
- O assunto desperta-lhes bastante interesse.

O professor forneceu também algumas opiniões sobre a aplicação:

- A apresentação da aplicação é interessante e estimulante.
- Seria possível conduzir aulas sobre evolução usando a aplicação como base.

- Achou interessante o uso de fatores de seleção relacionados à bactérias, e isso poderia trazer uma discussão multi-temática para a aula.
- No entanto, considerou um pouco difícil de entender o que está acontecendo no começo da simulação. As informações são passadas de forma rápida e sem grandes preparativos.

Dadas as considerações fornecidas pelo professor, foi possível ajustar o planejamento da apresentação da ferramenta para melhor atender aos alunos.

4.3 Validação em Sala de Aula

O teste de apresentação da ferramenta para alunos foi feito em sala de aula no dia 21 de fevereiro de 2017, no turno da manhã. A simulação foi executada utilizando um único *laptop* de forma que todos os alunos observassem os mesmos acontecimentos. Os alunos foram dispostos em formato de semi-círculo para observar a tela enquanto o autor deste trabalho operava a ferramenta. Esta configuração foi feita para gerar um clima de conforto e aflorar discussões provenientes de alunos. A apresentação teve duração de 30 minutos e consistia em realizar ciclos de períodos curtos de exposição de alguma tarefa na aplicação, indagação aos estudantes sobre as possíveis causas do fenômeno demonstrado e explicação da teoria juntamente com a correção dos erros presentes nas respostas dadas pelos estudantes. Os comentários eram feitos pelo autor deste trabalho e pelo professor Osiris, que também supervisionava o experimento.

Um fluxo lógico construtivo foi seguido durante a apresentação: primeiramente foi introduzido o conceito de mutação, depois cada um dos fatores de seleção e, por fim, um exemplo de combinação de fatores de seleção, tal qual a metodologia aplicada na seção 4.1. Cada etapa era concluída quando os alunos demonstravam que estavam compreendendo a situação corrente. A estrutura de apresentação empregada tinha os seguintes objetivos:

1. Construir um ensino interessante e participativo.
2. Fomentar o pensamento crítico-analítico dos estudantes na tentativa de explicar um fenômeno pelos fatos evidenciados.
3. Habituar o aluno ao crescente nível de variabilidade da população no ambiente para fazê-lo perceber a origem comum das variações na mutação.
4. Utilizar a lógica de exemplos passados para indicar a solução de problemas novos.
5. Construir um ensino gradativamente mais difícil com base nos ensinamentos anteriores.

6. Sanar as dúvidas e corrigir pensamentos incorretos sobre a seleção natural aplicada à evolução.

À princípio o conceito de mutações foi apresentado sem menção ao processo de seleção natural. Quando os estudantes haviam entendido a natureza aleatória da sua ocorrência, a seleção por recursos foi colocada em prática. Quando questionados sobre o porque de bactérias com formato triangular estarem muito mais presentes na população, tiveram diversas ideias como, por exemplo, que as outras ficaram “fracas” ou que as triangulares “mataram” as demais. Alguns alunos sugeriram que as bactérias mudaram de formato porque precisavam. Com esta etapa foi possível perceber as maiores dificuldades apresentadas em interpretar a situação corrente. Com a aplicação da seleção por predação, as ideias dos estudantes começaram a entrar em sintonia e relacionar a sobrevivência com a adaptação. Um dos estudantes expôs: “Essas daqui (bactérias com flagelo) começaram a evoluir muito e as redondas começaram a sumir”.

Durante todo o experimento os estudantes continuamente relacionavam a seleção introduzida com a resistência de bactérias a antibióticos. Ficou claro que este exemplo mais popular é o de mais fácil assimilação para eles. Quando a seleção por antibiótico foi aplicada, os estudantes foram capazes de identificar corretamente o processo de mutação com o surgimento de bactérias resistentes no meio.

O experimento revelou que os alunos aprendem em sala diversos exemplos práticos da seleção natural, mas falham em compreender o desencadeamento lógico do processo que leva às consequências conhecidas. O grupo citou exemplos como a introdução da gripe no Brasil pela chegada dos portugueses e o desaparecimento dos ancestrais do *Homo Sapiens*. Um dos estudantes chegou a perguntar se os casos relacionados à pessoas que tem imunidade à AIDS são um processo de evolução ou de seleção natural.

Ao final do experimento, os estudantes conseguiram explicar razoavelmente bem a última situação apresentada, onde a população era submetida a seleção por predação simultaneamente com a aplicação de variação de temperatura, relacionando os indivíduos de cor vermelha com formato alongado como os mais bem adaptados ao meio.

O experimento mostra que a aplicação tem potencial para ser utilizada como uma ferramenta de suporte no ensino da seleção natural, podendo ser fonte para uma série de discussões pertinentes ao estudo da evolução em sala de aula. O uso da ferramenta acrescido a um ensino interativo, é uma solução interessante para tornar o aprendizado mais eficaz.

5 Conclusão

Este trabalho descreveu uma aplicação para simular em computador o processo de seleção natural que tem por objetivo principal o auxílio no ensino da teoria sintética da evolução. A aplicação descrita simula este processo biológico pela utilização de um algoritmo genético para evoluir uma população que é continuamente exibida para o usuário com a utilização de uma interface gráfica.

A interface gráfica construída fornece a visualização da população a cada iteração do AG e é possível acompanhar as transformações sofridas pela inserção de variabilidade, hereditariedade e pressão de seleção do meio. O usuário faz uso das opções para configurar elementos de seleção e mutação para a população, tornando assim a experiência interativa e imersiva. A interface torna possível também a alteração dos elementos do algoritmo genético.

O programa criado pode ser utilizado para instrução de alunos em todos os níveis de escolaridade. As diferentes possibilidades de uso do ambiente da simulação fornecem apoio ao ensino da evolução para alunos do ensino básico e o interfaceamento para configuração do AG fornece utilidade também para alunos que estejam interessados em aprender sobre este tema, incluindo o acesso à código fonte de fácil extensibilidade.

O trabalho também demonstrou a realização de um conjunto de testes para validar a ferramenta. A ferramenta simulou com fidedignidade o processo de adaptação e seleção natural através da aplicação de diferentes tipos de mutações e variados ambientes de seleção. Através dos três tipos de formato, circular, alongado e triangular, e das três diferentes colorações, vermelho, verde e cinza, das bactérias, foi possível identificar qual era o tipo sendo momentaneamente favorecido na população, ou seja, o tipo mais bem adaptado ao meio. Dessa forma, a aplicação cumpriu o requisito de representação do processo de seleção dirigida. A aplicação foi também instrumentada em sala de aula e constatou-se que a aplicação pode ser utilizada com o acompanhamento de um professor para gerar cadeias de discussão pertinentes e aprendizado investigativo.

O professor avaliou que a dinâmica apresentada na interface não é de simples entendimento inicial e alguns alunos tiveram dificuldade em relacionar os acontecimentos demonstrados com a teoria da evolução. Contudo, com o passar de algumas iterações, os alunos começaram a capturar a essência do que estava sendo exibido, engajando-se de forma positiva. Despertou grande interesse nos alunos ao desafiar-los a propor explicações concisas para os fenômenos. Futuramente esforços serão aplicados ao melhoramento da apresentação da aplicação para tornar a introdução dos elementos mais compassada e eliminar a dificuldade inicial de entendimento.

Em geral, como trabalhos futuros, objetiva-se adicionar a língua inglesa como opção de idioma secundário, melhorar a demonstração da relação entre pais e sua prole, aperfeiçoar a aplicação no quesito visual, trabalhar com mais detalhes os conceitos relacionados à especiação, microevoluções e implementar avanço mais suave no desenvolvimento dos indivíduos, explicitando as bactérias das populações anterior e corrente. A imersão pode ainda ser melhorada com adição de animações e movimentação na interface.

Com tudo que foi explicitado, conclui-se que o trabalho alcançou seus objetivos iniciais, através de simulações e da validação em sala de aula, e deixa margens para aprimoramentos. Com as atividades previstas como trabalhos futuros, é possível criar um software aberto de alta qualidade técnica e científica, contribuindo para a academia, para a educação na área da biologia e para a produção científica nacional. Objetiva-se ainda a escrita de um artigo científico sobre o assunto.

Referências

- BARAB, S.; DEDE, C. Games and immersive participatory simulations for science education: An emerging type of curricula. *Journal of Science Education and Technology*, Springer Netherlands, v. 16, n. 1, p. 1–3, 2 2007. ISSN 1059-0145. Citado na página 13.
- BOHANNON, J. ‘spore’ documentary spawns protest by scientists who starred in it. *Science*, American Association for the Advancement of Science, v. 322, n. 5901, p. 517a–517a, 2008. ISSN 0036-8075. Disponível em: <<http://science.sciencemag.org/content/322/5901/517a>>. Citado na página 15.
- CEI. *COLÉGIO CEI*. s.d. <<http://www.portalcei.com.br/>>. Acesso em: 22 de janeiro de 2017. Citado na página 43.
- DARWIN, C. *On the Origin of Species by Means of Natural Selection*. London: Murray, 1859. Citado na página 19.
- DAWKINS, R. *O relojoeiro cego: a teoria da evolução contra o desígnio divino*. [S.l.]: Companhia das Letras, 1991. ISBN 9788535901610. Citado na página 20.
- EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. [S.l.]: SpringerVerlag, 2003. ISBN 3540401849. Citado 3 vezes nas páginas 13, 22 e 24.
- GAMMA, E. et al. *Padrões de Projeto: soluções reutilizáveis de software orientado a objetos*. [S.l.]: Bookman, 2000. ISBN 8573076100. Citado na página 33.
- GREGORY, T. R. Understanding natural selection: Essential concepts and common misconceptions. *Evolution: Education and Outreach*, v. 2, n. 2, p. 156–175, 2009. ISSN 1936-6434. Disponível em: <<http://dx.doi.org/10.1007/s12052-009-0128-1>>. Citado 3 vezes nas páginas 12, 19 e 28.
- HERRERO, D. et al. Evolution and natural selection: learning by playing and reflecting. *Journal of New Approaches in Educational Research*, v. 3, n. 1, p. 26–33, 2014. ISSN 2254-7339. Disponível em: <<https://naerjournal.ua.es/article/view/v3n1-4>>. Citado 2 vezes nas páginas 13 e 16.
- JONES, T. C.; LAUGHLIN, T. F. Popgen fishbowl: A free online simulation model of microevolutionary processes. *The American Biology Teacher*, University of California Press Journals, v. 72, n. 2, p. 100–103, 2010. ISSN 0002-7685. Disponível em: <<http://abt.ucpress.edu/content/72/2/100>>. Citado na página 14.
- LINDEM, R. *Algoritmos Genéticos (2a edição)*. [S.l.]: BRASPORT, 2008. ISBN 9788574523736. Citado 2 vezes nas páginas 23 e 25.
- MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, ACM, v. 8, n. 1, p. 3–30, jan. 1998. ISSN 1049-3301. Disponível em: <<http://doi.acm.org/10.1145/272991.272995>>. Citado na página 34.
- MITCHELL, M. *An introduction to genetic algorithms*. [S.l.]: MIT Press, 1996. (A Bradford book). ISBN 0262133164. Citado 2 vezes nas páginas 20 e 21.

- MOORE, R. et al. Undergraduates' understanding of evolution: ascriptions of agency as a problem for student learning. *Journal of Biological Education*, v. 36, n. 2, p. 65–71, 2002. Citado 2 vezes nas páginas 12 e 20.
- PAVAI, G.; GEETHA, T. V. A survey on crossover operators. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 49, n. 4, p. 72:1–72:43, dez. 2016. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/3009966>>. Citado na página 22.
- RIDLEY, M. *Evolução*. [S.l.]: Artmed, 2006. ISBN 9788536306353. Citado 2 vezes nas páginas 18 e 19.
- SONI, N.; KUMAR, T. Study of various mutation operators in genetic algorithms. *IJCSIT) International Journal of Computer Science and Information Technologies*, v. 5, n. 3, p. 4519–4521, 2014. Citado na página 22.
- SPETH, E. B. et al. Using aida-ed for teaching and learning about evolution in undergraduate introductory biology courses. *Evolution: Education and Outreach*, v. 2, n. 3, p. 415–428, 2009. ISSN 1936-6434. Disponível em: <<http://dx.doi.org/10.1007/s12052-009-0154-z>>. Citado 3 vezes nas páginas 12, 14 e 38.
- SRINIVAS, M.; PATNAIK, L. M. Genetic algorithms: a survey. *Computer*, v. 27, n. 6, p. 17–26, June 1994. ISSN 0018-9162. Citado 2 vezes nas páginas 21 e 24.
- YAMANOI, T.; IWASAKI, W. M. Origami bird simulator: A teaching resource linking natural selection and speciation. *Evolution: Education and Outreach*, v. 8, n. 1, p. 14, 2015. ISSN 1936-6434. Disponível em: <<http://dx.doi.org/10.1186/s12052-015-0043-6>>. Citado na página 13.