

# UM AMBIENTE WEB PARA O APOIO AO ENSINO DE PROGRAMAÇÃO EVOLUCIONÁRIA

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

**Fernando Henrique Silva Fragoso**  
**Orientador: Wellington Pinheiro dos Santos**



UNIVERSIDADE  
DE PERNAMBUCO

**FERNANDO HENRIQUE SILVA  
FRAGOSO**

**UM AMBIENTE WEB PARA O APOIO  
AO ENSINO DE PROGRAMAÇÃO  
EVOLUCIONÁRIA**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, novembro de 2009.**

# Agradecimentos

Agradeço primeiramente aos meus pais, por não terem poupado esforços para me proporcionar a educação que tenho hoje, e que foi fundamental para que eu me tornasse o que sou.

Ao professor Wellington, por estar sempre presente apoiando a realização deste trabalho e pelo incentivo e conselhos que me levaram a não desistir no meio do caminho.

Agradeço a todos os amigos e companheiros que de alguma forma ajudaram direta ou indiretamente na concretização desta monografia.

# Resumo

Este trabalho apresenta uma proposta de ensino de Programação Evolucionária, sub-área da Computação Evolucionária, utilizando conceitos de educação à distância e um sistema de gerenciamento de aprendizado. Foi desenvolvido um curso em um ambiente web, sob a plataforma Moodle, que pretende facilitar o processo de aprendizagem da tecnologia, utilizando uma abordagem multidisciplinar e recursos diferenciados para que este processo se torne tanto eficiente quanto prazeroso para os estudantes. Funcionando como elo de ligação entre a teoria e a prática, também foi desenvolvido e integrado ao curso um software que simula o efeito das mutações da programação evolucionária na otimização de algumas funções pré-determinadas.

# Abstract

This work presents a proposal of a web environment to aid at learning Evolutionary Programming, a subset of Evolutionary Computation, using concepts of distance education and learning management systems. To achieve these goals, we developed a web-based learning system using Moodle, to help learners to apprehend concepts and algorithms on evolutionary programming using a multidisciplinary approach and several additional educational resources. We tried to make learning an affordable and joyful experience for students. We also developed a software simulator to model the effect of mutations on the performance of evolutionary programming algorithms at the optimization of predetermined benchmark functions.

# Sumário

<b>Resumo</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Sumário</b>	<b>iii</b>
<b>Índice de Figuras</b>	<b>v</b>
<b>Índice de Tabelas</b>	<b>vi</b>
<b>Tabela de Símbolos e Siglas</b>	<b>vii</b>
<b>Capítulo 1 Introdução</b>	<b>8</b>
1.1 Objetivos e metas	8
1.2 Metodologia e estratégia de ação	9
1.3 Organização do trabalho	10
<b>Capítulo 2 Princípios de programação evolucionária</b>	<b>11</b>
2.1 História	12
2.2 Algoritmos evolucionários	13
2.2.1 Componentes dos algoritmos evolucionários	14
2.3 Programação evolucionária	15
2.3.1 Representação	16
2.3.2 Mutação	16
2.3.3 Recombinação	17
2.3.4 Seleção dos pais	17
2.3.5 Seleção dos sobreviventes	18
2.4 Conclusão	18
	iii

<b>Capítulo 3 Sistemas de gerenciamento de aprendizado</b>	<b>19</b>
3.1 Características dos sistemas de gerenciamento de aprendizado	20
3.2 Exemplos de sistemas	21
3.2.1 Projeto Amadeus	22
3.2.2 Blackboard	23
3.2.3 Moodle	24
<b>Capítulo 4 Desenvolvimento</b>	<b>26</b>
4.1 Requisitos para a construção do curso	26
4.2 Estrutura do curso	27
4.2.1 Módulo I – Introdução	28
4.2.2 Módulo II – Algoritmos Evolucionários	29
4.2.3 Módulo III – Programação Evolucionária	29
4.3 Ferramentas utilizadas	29
4.4 Ambiente desenvolvido	32
<b>Capítulo 5 Conclusão e Trabalhos Futuros</b>	<b>38</b>
5.1 Dificuldades encontradas	38
5.2 Trabalhos futuros	39
<b>Bibliografia</b>	<b>40</b>

# Índice de Figuras

<b>Figura 1.</b>	Pseudocódigo de algoritmo evolucionário .....	14
<b>Figura 2.</b>	Pseudocódigo do algoritmo da programação evolucionária .....	15
<b>Figura 3.</b>	Ferramentas de um LMS .....	20
<b>Figura 4.</b>	Tela principal do Projeto Amadeus .....	22
<b>Figura 5.</b>	Tela principal do Blackboard LMS .....	23
<b>Figura 6.</b>	Tela principal do Moodle .....	24
<b>Figura 7.</b>	Cada máquina virtual Java interagindo com o seu respectivo sistema operacional.....	30
<b>Figura 8.</b>	Comparação entre um sistema síncrono e um sistema assíncrono com AJAX	31
<b>Figura 9.</b>	Tela inicial.....	32
<b>Figura 10.</b>	Recurso de texto do curso sendo visualizado. ....	33
<b>Figura 11.</b>	Recurso de vídeo. ....	34
<b>Figura 12.</b>	Diagrama de Classes .....	35
<b>Figura 13.</b>	Simulador de Programação Evolucionária .....	36
<b>Figura 14.</b>	Resultados da simulação .....	37



# Índice de Tabelas

<b>Tabela 1.</b> Evolução x Solução de problemas [2] .....	11
<b>Tabela 2.</b> Tabela estrutural do curso.....	28

# Tabela de Símbolos e Siglas

(Dispostos por ordem de aparição no texto)

CE – Computação Evolucionária

PE – Programação Evolucionária

LMS – Learning Management System

J2EE – Java2 Platform Enterprise Edition

JSP – Java Server Pages

HTML - HyperText Markup Language

# Capítulo 1

## Introdução

A Computação Evolucionária, subárea da Inteligência Artificial, usa os princípios do processo natural da evolução, proposto por Darwin [1], para otimização de problemas usando os algoritmos evolucionários.

Devido a sua multidisciplinaridade, o tema exige uma atenção especial no seu ensino, o uso de uma ferramenta de gerenciamento de aprendizagem (LMS) pode facilitar bastante o processo de aprendizado da Computação Evolucionária.

Entre os vários dialetos de Computação Evolucionária existentes, foi escolhido a Programação Evolucionária, que tem como base o uso de operadores de mutação para encontrar melhores soluções que otimizem uma função [2]. Ao contrário dos Algoritmos Genéticos, outro dialeto que tem sido tema de várias pesquisas com aplicações nas mais diversas áreas, a Programação Evolucionária, que pode ser tão poderosa quanto, não é muito conhecida e vem sendo pouco utilizada, principalmente no Brasil.

### 1.1 Objetivos e metas

O projeto tem como objetivo geral desenvolver, utilizando as ferramentas de um LMS, um ambiente web para auxiliar o ensino de Programação Evolucionária.

Para que esse objetivo seja alcançado é preciso subdividi-lo em objetivos específicos:

- Estudo em Computação Evolucionária e seus dialetos, para que o conteúdo possa ser repassado com mais propriedade;
- Estudo dos LMS, na visão do desenvolvedor, para que possam ser desenvolvidos *plugins* e para que as ferramentas de ensino dos mesmos sejam bem utilizadas;

- Definição dos requisitos do curso, com base no estudo de outros, juntamente com o conhecimento adquirido em Computação Evolucionária, modelar um roteiro que possa auxiliar ao máximo o aprendizado do tema;
- Desenvolvimento do curso, obedecendo aos requisitos que foram previamente definidos;
- Teste do ambiente de aprendizado, utilizando como voluntários estudantes de computação que não estão habituados com o tema para que se matriculem no curso e ao final respondam um questionário.

## **1.2 Metodologia e estratégia de ação**

Após pesquisas dentre os diversos sistemas de gerenciamento de aprendizado existentes no mercado, como o Blackboard e Amadeus, o Moodle foi escolhido para ser utilizado no experimento.

O Moodle é um dos LMS mais utilizados hoje em dia em todo o mundo, entre seus diferenciais existe o fato de ser gratuito e de código aberto, o que o torna completamente customizável. Facilitando ainda mais a adaptação do seu código, o Moodle possui uma grande comunidade de usuários que participa da sua manutenção, seja sugerindo modificações, novas funcionalidades ou reportando eventuais erros nas versões. Outra vantagem deste LMS é que seu criador possui formação em educação, o que facilitou o sistema a se tornar cada vez mais amigável também ao professor.

É utilizada também uma simulação que tem o objetivo auxiliar o aprendiz a compreender o conceito da programação evolucionária. Essa simulação foi desenvolvida com tecnologia Java (J2EE, Servlets, JSP), que foi escolhida por ser gratuita e independente de plataforma, e por ter execução na *web*, acessível a todos. O ambiente de desenvolvimento utilizado foi o Netbeans IDE 6.5.1.

O questionário final, que terá a função de teste do curso e fonte de sugestões para sua melhoria, será desenvolvido com a própria ferramenta de testes *online* do Moodle.

## **1.3 Organização do trabalho**

O segundo capítulo mostra os fundamentos da Programação Evolucionária desde a teoria da evolução proposta por Darwin que inspirou a técnica; fala sobre o surgimento dos outros estudos que vieram a formar campo que chamamos hoje de Computação Evolucionária; mostra os algoritmos evolucionários, base de todas as técnicas, e apresenta seus elementos básicos; e ao fim explica o funcionamento da Programação Evolucionária, apresentando as suas diferenças em relação às outras técnicas.

O capítulo seguinte apresenta o conceito de Sistemas de Gerenciamento de Aprendizagem, mostrando como esse tipo de sistema é utilizado hoje em dia nas instituições de ensino ao redor e sua importância tanto sozinho quanto funcionando como um complemento às aulas presenciais. Também são apresentadas as características de alguns desses sistemas.

O quarto capítulo mostra todos os aspectos do desenvolvimento do curso produzido com este trabalho. Os critérios utilizados para que o curso obtivesse o desempenho esperado, as ferramentas utilizadas na criação do curso e seus módulos, a descrição da estrutura desenvolvida e de cada um dos módulos e as telas do sistema.

O quinto e último capítulo apresenta as conclusões gerais do projeto, descreve as principais dificuldades que foram encontradas durante a sua realização e sugere algumas idéias de trabalhos futuros.

# Capítulo 2

## Princípios de programação evolucionária

Segundo a teoria da evolução por seleção natural de Charles Darwin, as características hereditárias que são favoráveis à sobrevivência de uma espécie vão, ao longo das gerações, tornando-se mais evidentes nas populações, tendo em vista que os indivíduos mais aptos possuem mais chances de reprodução. Por outro lado, as características desfavoráveis vão diminuindo, tendendo a desaparecer depois de várias gerações [1].

Inspirando-se nessa teoria tão aceita pela comunidade científica, se apresenta a computação evolucionária, ramo da inteligência computacional, que se propõe a resolver problemas de tentativa e erro fazendo analogia aos princípios da evolução encontrada na natureza. A utilização de um princípio que já tem milhões de anos de sucesso, torna a computação evolucionária uma técnica bastante promissora.

<b>Evolução</b>	<b>Solução de problemas</b>
Ambiente	Problema
Indivíduos	Solução candidata
Aptidão	Qualidade

**Tabela 1.** Evolução x Solução de problemas [2].

A técnica utiliza uma população de soluções, geradas aleatoriamente ou a partir de alguma técnica, que passam por um processo iterativo no qual vão evoluindo e formando novas gerações de soluções. A cada geração, as novas soluções são testadas e as melhores, que possuem um maior nível de aptidão (ou

*fitness*), são usadas para geração de novas soluções, enquanto as menos aptas vão sendo descartadas do grupo [2].

Para geração das novas populações são utilizados operadores de mutação, recombinação e seleção. Na seleção, são escolhidos os indivíduos mais aptos para gerarem descendentes, essa reprodução pode ser assexuada, onde apenas um indivíduo é capaz de gerar descendentes, ou sexuada, onde um par de indivíduos são necessários para gerar uma nova geração. Na recombinação, os descendentes recebem parte do material genético de cada um dos seus antecedentes, que são escolhidos de forma aleatória. Já com os operadores de mutação os indivíduos recebem mudanças aleatórias no seu material genético.

## 2.1 História

Alan Turing foi o primeiro a pensar em aplicar os princípios de Darwin para automatização da solução problemas, em 1948, quando propôs “*genetical and evolutionary search*”, onde usa a idéia de hereditariedade, mutações e seleção natural na escolha da melhor solução para um dado problema. Somente em 1962, Bremermann fez experimentos reais em otimização através de evolução e recombinação. Assim, como Bremermann, nos anos 60 vários outros pesquisadores utilizaram os princípios propostos por Turing em seus experimentos. Nos Estados Unidos, Fogel, Owens e Walsh propuseram a programação evolucionária (*evolutionary programming*) enquanto Holland desenvolvia um algoritmo genético (*genetic algorithm*). Já na Alemanha, Rechenberg e Schwefel trabalhavam nas estratégias evolutivas (*evolution strategy*) [2][3][4].

Somente nos anos 90, foi cunhado o termo *computação evolucionária*, e assim ficou conhecido o campo de pesquisa que abrange todas as áreas surgidas nos anos 60 e posteriormente [2].

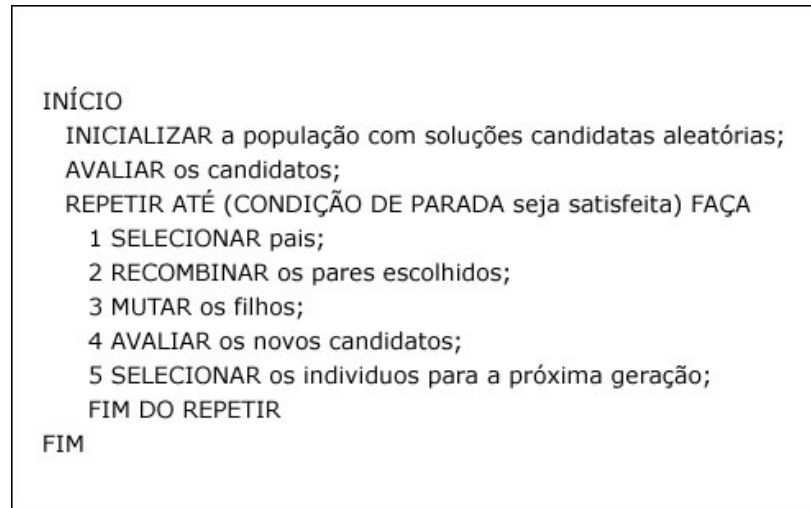
## 2.2 Algoritmos evolucionários

Os algoritmos envolvidos nas sub-áreas da Computação Evolucionária são denominados algoritmos evolucionários. Apesar de todos terem suas peculiaridades, eles possuem uma idéia básica em comum: uma população de indivíduos passando por processos de seleção natural, onde só os mais aptos sobrevivem, aumentando assim a qualidade das gerações subseqüentes.

Dada uma função a ser maximizada, são gerados aleatoriamente alguns elementos pertencentes ao seu domínio, que formarão a primeira geração de soluções candidatas. Essas soluções são testadas e as melhores são responsáveis pela formação de uma nova geração por meio de operadores de recombinação e/ou operadores de mutação. Esse processo é repetido até que se obtenha uma solução com qualidade satisfatória ou que aconteça algum critério de parada pré-estabelecido [2].

Enquanto o processo de seleção natural vai melhorando a qualidade das soluções já existentes, os operadores de mutação e recombinação diversificam os indivíduos, o que faz com que as soluções não fiquem presas a pontos (mínimos ou máximos) locais. A união desses fatores tende a melhorar a aptidão das soluções a cada nova geração, se aproximando cada vez mais de valores ótimos para a função. A Figura 1 mostra um esquema geral de algoritmo evolucionário em pseudocódigo [5].





**Figura 1.** Pseudocódigo de algoritmo evolucionário.

### 2.2.1 Componentes dos algoritmos evolucionários

Os componentes mais importantes que compõem os algoritmos evolucionários são: a representação dos indivíduos, a função de avaliação, a população, o mecanismo de seleção dos pais, os operadores de mutação e recombinação e o mecanismo de seleção dos indivíduos sobreviventes. Todos esses componentes devem ser especificados para se definir um algoritmo evolucionário particular, além de uma condição de parada.

O primeiro passo é fazer a ligação entre os fenótipos, os indivíduos do mundo real, e os genótipos, que são a representação dos indivíduos codificados no algoritmo. Os fenótipos, ou soluções candidatas, podem ser mapeados nos genótipos, ou cromossomos, como um vetor de números inteiros ou binários, por exemplo. O termo “representação” é usado tanto para a codificação, o mapeamento do fenótipo para o genótipo, quanto para a decodificação, o caminho inverso. A função de avaliação (ou *fitness*) é a base para a seleção, ela define os requerimentos para que um indivíduo seja considerado mais ou menos apto. A população é o conjunto de soluções candidatas e forma a unidade da evolução. A especificação de uma população é basicamente o número de indivíduos que farão parte da sua composição. O mecanismo de seleção dos pais consiste em escolher

probabilisticamente indivíduos com boa aptidão para que eles se tornem geradores de novos indivíduos. Uma boa seleção de pais é um fator importante para a melhora dos indivíduos ao longo das gerações. Os operadores de mutação atuam sobre um indivíduo, efetuando pequenas mudanças no seu código e gerando um novo indivíduo levemente modificado. Estes operadores são estocásticos, baseados em números aleatórios para geração dos novos valores para os indivíduos. Já a recombinação age sobre dois indivíduos, mesclando suas características e gerando novos indivíduos. A recombinação, assim como a mutação, ocorre baseada em valores randômicos. Por último acontece a seleção dos sobreviventes, essa etapa define quais indivíduos irão se manter na população e quais serão descartados. Esse processo ocorre baseado na aptidão de cada indivíduo e é feito depois dos processos de recombinação e mutação [2].

## 2.3 Programação evolucionária

Originalmente proposto por Lawrence J. Fogel em 1966, este dialeto da computação evolucionária, que será objeto de estudo deste trabalho, tem como único operador de variação empregado, a mutação.

```
INICIO
  GERAÇÃO t=0;
  INICIALE a população com n indivíduos;
  AVALIE a aptidão da população;
  REPITA ENQUANTO (t final não é alcançada) FAÇA
    OPERE mutação sobre a população;
    AVALIE a aptidão da população;
    SELECIONE n indivíduos para a nova população;
    t = t+1;
  FIM REPITA
FIM
```

**Figura 2.** Pseudocódigo do algoritmo da programação evolucionária

O primeiro experimento da técnica feito por Fogel foi representar indivíduos de uma população por máquinas de estados finitos tendo como objetivo descrever o comportamento programas computacionais, daí o nome “programação evolucionária”. No experimento uma sequência de símbolos era apresentado à população de máquinas de estado finito, e cada uma delas recebia um valor de *fitness* de acordo com o desempenho obtido na previsão dos símbolos de entrada e saída. As melhores máquinas eram escolhidas para se tornarem pais para geração seguinte. A Figura 2 representa um pseudocódigo do algoritmo da Programação Evolucionária.

Em anos posteriores, a Programação Evolucionária foi usada na solução de diversos problemas de engenharia incluindo planejamento e roteamento de tráfico, detecção de câncer, planejamento militar, sistemas de controle, identificação e processamento de sinais, aprendizado em jogos, entre outros. A programação evolucionária pode ser aplicada em qualquer área em que os algoritmos evolucionários são aplicáveis [3] [4].

### 2.3.1 Representação

A programação evolucionária é frequentemente usada para otimizar funções da forma  $f : R^n \rightarrow R$ , e sua representação é feita através de uma coleção de  $\eta$  indivíduos, onde cada indivíduo possui um vetor  $\chi$  onde  $[\chi_1, \chi_2, \dots, \chi_\eta] \in R^n$  e um vetor  $\sigma$  com os desvios padrão para os valores em  $\chi$ . Logo, a representação geral dos indivíduos na programação evolucionária se dá por  $[\chi_1, \dots, \chi_\eta, \sigma_1, \dots, \sigma_\eta]$  [2].

### 2.3.2 Mutação

A operação de mutação é um dos pontos mais importante em um algoritmo de programação evolucionária, pois é ela a responsável pela busca de novos conjuntos de soluções. Várias abordagens têm sido criadas, utilizando diferentes operadores de mutação como o operador Gaussiano, o de Lévy e o de Cauchy, e também diferentes técnicas, como a mutação *single-point* e a mutação de estratégia mista [6].

Usando como exemplo o operador de mutação mais amplamente associado com o que veio a ser conhecido como programação evolucionária clássica, ou CEP, a mutação transforma um cromossomo  $[\chi_1, \dots, \chi_n, \sigma_1, \dots, \sigma_n]$  em  $[\chi'_1, \dots, \chi'_n, \sigma'_1, \dots, \sigma'_n]$ , onde

$$\sigma'_i = \sigma_i \cdot (1 + \alpha \cdot N(0,1))$$

$$\chi'_i = \chi_i + \sigma'_i \cdot N_i(0,1)$$

$N(0,1)$  representa uma distribuição gaussiana aleatória distribuída com média 0 e desvio padrão [2].

### 2.3.3 Recombinação

Os algoritmos de programação evolucionária não utilizam recombinação, apenas mutação. Desde os anos 90 vêm sendo discutidas as vantagens de um algoritmo que utiliza apenas mutação ao invés de uma união entre recombinação e mutação. Fogel e Atmar compararam os resultados de algoritmos de programação evolucionária com e sem o uso de recombinação em várias funções e concluíram que houve melhora de performance nas versões sem recombinação.

Atualmente, tanto a programação evolucionária quanto os algoritmos genéticos (abordagem da computação evolucionária que faz o uso da recombinação) são temas de pesquisas que vêm mostrando que cada técnica tem suas vantagens e desvantagens de acordo com o tipo do problema abordado [2].

### 2.3.4 Seleção dos pais

Diferentemente dos outros dialetos, na programação evolucionária a seleção dos pais não caracteriza um problema, pois nela cada membro da população dá origem a um novo membro por mutação. Nos algoritmos genéticos, por exemplo, os indivíduos que irão gerar crias são escolhidos de forma estocástica, baseado na aptidão [2].

### 2.3.5 Seleção dos sobreviventes

Ao final do processo de mutação, é a vez do operador de seleção, que descarta metade dos indivíduos através de um processo estocástico, onde cada indivíduo é comparado com um número pré-determinado de outros indivíduos, em cada comparação os indivíduos vão ganhando pontuações, que vão atribuir a qualidade da solução candidata. Quanto maior o número de comparações, mais determinístico fica o processo e mais recursos computacionais são utilizados. Os melhores indivíduos segundo o processo de seleção fará parte da nova população em um processo iterativo [2].

## 2.4 Conclusão

A computação evolucionária obtém seu sucesso por se basear em um estudo de caso que já funciona a milhões de anos, a evolução natural das espécies. Na analogia com problemas computacionais, uma população de soluções evolui ao longo das gerações com o objetivo de melhorar sua aptidão. Ao final desse processo iterativo, a solução que possuir a melhor aptidão, ou *fitness*, será a solução escolhida para o problema.

Na Programação Evolucionária, dialeto que é o tema de estudo deste trabalho, não é usada a recombinação de material genético para a geração de um novo indivíduo, mas somente operadores de mutação que agem em cada indivíduo da população, gerando um novo.

# Capítulo 3

## Sistemas de gerenciamento de aprendizado

Está cada vez mais difundido nas universidades e organizações empresariais o uso da internet como meio de entrega de informações e conhecimento para o aprendizado a distância. Entre os benefícios dos sistemas de gerenciamento de aprendizado (Learning Management Systems, LMS) estão a flexibilidade na distribuição das informações, assim como redução de custos de viagens, tendo em vista que a internet proporciona a aproximação, de forma síncrona ou assíncrona, entre pessoas que estão geograficamente distantes entre si e dos grandes centros urbanos.

Existem também algumas barreiras que ainda impedem o crescimento dessa forma de ensino, como os custos em infra-estrutura, treinamento da equipe de desenvolvimento, e a própria necessidade do usuário de se adaptar à nova forma de aprendizado. Além das barreiras técnicas também existe a desconfiança e o preconceito da sociedade, que está há muito tempo habituada com o sistema tradicional de ensino [7] [8].

Além de auxiliar no ensino à distância, os LMS são utilizados para complementar o ensino presencial, diminuindo significativamente o tempo dedicado às aulas expositivas e oferecendo oportunidades de aprendizagem mais completas e eficazes. Um estudo publicado em 2005 mostra que na maioria dos países desenvolvidos, aproximadamente 75% das universidades usam LMS como complemento às aulas presenciais [9].

## 3.1 Características dos sistemas de gerenciamento de aprendizado

Os sistemas de gerenciamento de aprendizado consistem de um ambiente integrado que oferece facilidade e suporte para o desenvolvimento da atividade de aprendizado, tanto no modo síncrono como no assíncrono, além de possibilitar a interação colaborativa entre os usuários e oferecer mecanismos de controle e monitoramento.

A interatividade é a palavra chave para o sucesso de um LMS, por meio das ferramentas que a aplicação pode oferecer (como fóruns, *chats*, listas de discussão e troca de mensagens entre os usuários), os aprendizes e gestores podem sempre estar trocando conhecimento e informação. A Figura 3 mostra algumas ferramentas integradas a um LMS [10].



Figura 3. Ferramentas de um LMS.

Além de oferecer serviços que visam a interação, um sistema de gerenciamento de aprendizado robusto deve centralizar e automatizar a administração e prover um ambiente amigável, que possua uma curva de

aprendizado suave que não atrapalhe o aprendiz na absorção do conteúdo principal. Uma outra característica crucial é a sua portabilidade e possibilidade de reuso.

Na configuração dos LMS o gestor do curso define os objetos de aprendizagem e estratégias de gerenciamento. Os objetos de aprendizagem são pequenas partes do conteúdo que são armazenados num repositório de dados. Essa estrutura do curso definida por meio de objetos de aprendizagem possibilita a reusabilidade do objeto em outros cursos, o que proporciona uma redução do tempo de desenvolvimento do curso e nos custos de manutenção.

## 3.2 Exemplos de sistemas

A escolha de um LMS é uma das decisões mais importantes quando se deseja iniciar um projeto de educação a distância ou um complemento para o ensino presencial.

Existem vários pontos que devem ser abordados no momento de escolher o sistema que irá ser utilizado. Por exemplo, deve ser levado em consideração o número de usuários que irá utilizar o sistema e a previsão de crescimento desse número. Outro fator importante é a escolha dos recursos que serão necessários ao projeto, para que o sistema possa ser escolhido na medida certa, sem que haja desperdícios.

A escolha de um LMS comercial ou um de código aberto, vai implicar na necessidade ou não de uma equipe de infra-estrutura e suporte técnico. No caso de um LMS proprietário a manutenção do sistema é de responsabilidade da empresa contratada, no caso dos LMSs de código aberto é preciso uma equipe ambientada com o sistema escolhido para que possam ser feitas as alterações que atendem necessidades específicas.

Quanto ao custo, os LMSs de código aberto levam grande vantagem, já que são disponibilizados gratuitamente para *download* e adaptação. Além disso vários deles são mantidos por uma comunidade de desenvolvedores voluntários. Em sistemas comerciais o custo-benefício depende muito do contrato feito entre as



partes. Algumas empresas, por exemplo, garantem a hospedagem e manutenção e cobra de acordo com o número de alunos matriculados nos cursos, outras cobram apenas pela licença de uso do sistema.

É preciso verificar todas essas variáveis para garantir a escolha correta de um sistema de gerenciamento de aprendizagem. A seguir, serão apresentados alguns sistemas existentes no mercado.

### 3.2.1 Projeto Amadeus

O projeto Amadeus foi desenvolvido pelo Centro de Informática (Cin), da Universidade Federal de Pernambuco (UFPE). Ele consiste em um misto de ensino a distância e ensino presencial, conceito conhecido como *blended learning*, e apresenta-se como o primeiro LMS se segunda geração.

Essa plataforma, totalmente livre, tem como objetivo ampliar as possibilidades de trabalho dos professores e proporcionar formas criativas de relacionamento, comunicação e colaboração entre os participantes. O Amadeus LMS procura oferecer um espaço ao aluno para construir e obter conhecimento, oferecendo ferramentas para tornar o processo mais interessante.



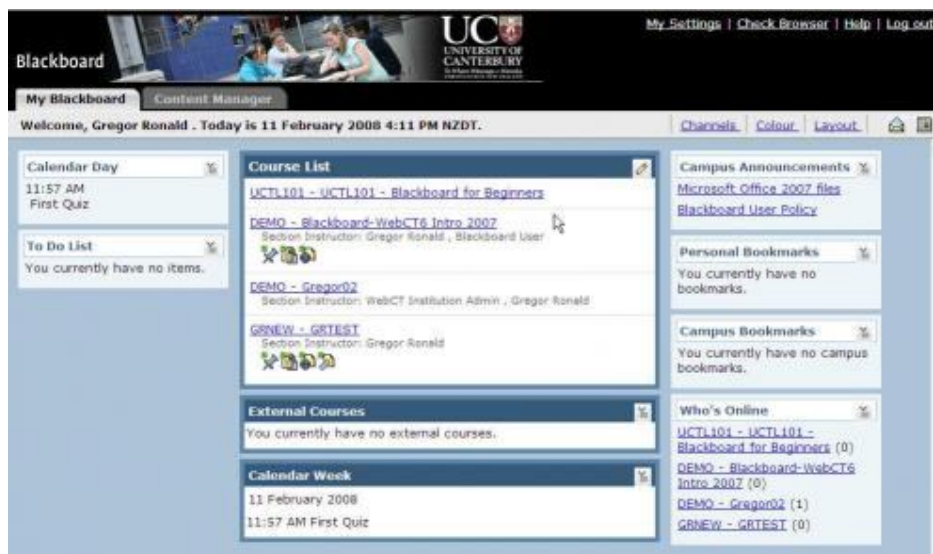
Figura 4. Tela principal do Projeto Amadeus

Entre as novidades, estão as mensagens de texto enviadas aos alunos sempre que o professor inclui algum material novo. Além disso está previsto a inclusão do padrão da TV Digital Brasileira no sistema.

A plataforma é utilizada pelo SENAI do Rio Grande do Norte, pela Universidade Federal de Pernambuco e a Pontifícia Universidad Católica (PUC) de Valparaíso, no Chile, já analisa a possibilidade de uso do *software* em países hispânicos. Além disso, existe um projeto de cooperação já iniciado com universidades alemãs.

### 3.2.2 Blackboard

O Blackboard é um *software* proprietário desenvolvido pela Blackboard Inc., empresa norte americana que desenvolve sistemas e provê serviços voltados para educação *online*. A Figura 5 mostra a tela inicial do Blackboard LMS.



**Figura 5.** Tela principal do Blackboard LMS

A plataforma, desenvolvida em Java, possui larga adoção em instituições de ensino e hoje é líder de mercado entre os *softwares* proprietários de *e-learning*. O motivo é a sua facilidade de utilização, flexibilidade pedagógica, amplitude de funções, tanto para o ensino a distância quanto no apoio ao ensino presencial.

Mais de 2600 instituições, em cerca de 72 países, utilizam o Blackboard como LMS. No Brasil, o sistema é utilizado, por exemplo, pelo SENAC de São Paulo, pelo Instituto de Educação Superior de Brasília (IESB), pela Fundação Getúlio Vargas (FGV-SP), entre várias outras.

### 3.2.3 Moodle

O Moodle, criado em 2001 pelo educador e cientista da computação Martin Dougiamas, é um dos LMS mais utilizado da atualidade. Hoje ele possui mais de 40 mil sites registrados, com 30 milhões de usuários divididos em 3 milhões de cursos, e em mais de 200 países (de acordo com a base de dados dinâmica do próprio *site* oficial, em outubro de 2009)

O grande motivo do sucesso do Moodle está no fato do mesmo ser um *software* livre, sob licença GNU-GPL, totalmente gratuito e com uma comunidade de colaboradores voluntários que sugerindo modificações e reportam defeitos. Assim a plataforma se mantém em constante avanço, sempre com a cooperação que é a essência do *software* livre. A comunidade de colaboradores é a responsável também pelo grande número de idiomas disponíveis para o Moodle, cerca de 75. A Figura 6 mostra uma das telas do Moodle.



Figura 6. Tela principal do Moodle

Desenvolvido utilizando a linguagem PHP, o Moodle funciona, sem a necessidade de modificações, em vários sistemas operacionais como Unix, Windows, Solaris e Mac OS. Além disso, utiliza uma completa abstração de base de dados, podendo assim ser utilizado com qualquer banco de dados existente.

O forte embasamento na Pedagogia Construcionista é um diferencial do Moodle. O sistema baseia-se na idéia de que a aprendizagem é particularmente eficaz quando se dá construindo algo para que outros experimentem. Isso se mostra desde a interface, que ao contrário de outros sistemas não é orientada por ferramentas, e sim por semanas ou tópicos, onde as ferramentas são artefatos para auxiliar no objetivo principal, a aprendizagem do conteúdo.

# Capítulo 4

## Desenvolvimento

O ensino da Computação evolucionária exige uma atenção especial, principalmente devido a sua multidisciplinaridade, neste capítulo descreveremos a construção de um curso utilizando uma ferramenta de gerenciamento de aprendizagem que tem o objetivo de apoiar estudantes e professores na transmissão de conhecimento a respeito do tema.

Inicialmente foi pensado em desenvolver completamente um sistema *web* que auxiliasse no processo de ensino de uma tecnologia, mas após uma pesquisa de levantamento de requisitos, foi decidido utilizar um LMS já existente para ser a base do curso, para evitar reescrita de código. O sistema escolhido foi o Moodle, por ser de código aberto e por sua comunidade de desenvolvedores.

### 4.1 Requisitos para a construção do curso

Nesta seção serão descritos os requisitos que foram levados em consideração no intuito de que o curso produzido conseguisse da melhor forma possível obter a atenção do aluno e o fazer se interessar por essa maneira de aprender e interagir.

A utilização de uma linguagem clara e direta é o ponto principal do curso. Mesmo não sendo indicado para pessoas totalmente leigas em computação, os textos procuram ser de fácil entendimento para proporcionar uma leitura mais agradável e menos cansativa. As imagens também tem grande importância na realização desse objetivo, quebrando uma possível impressão de um texto pesado por parte do aluno, o que pode desencorajar a leitura, dificultando assim um bom aproveitamento do curso.

Questionários são utilizados como forma de fixação do conteúdo, tanto com questões de múltipla escolha, como em questões discursivas que exigem uma reflexão sobre o tema apresentado por parte do aluno. Ainda podem ser utilizadas

várias ferramentas nativas do LMS que facilitam a comunicação e integração entre professores e alunos, como fóruns, *chats* e outras.

Para exemplificar o uso da Programação Evolucionária, a técnica que o curso dá ênfase, é utilizado um simulador que examina o desempenho de alguns operadores de mutação (o gaussiano, o de Cauchy, o de Lévy e o de ponto único) agindo sobre funções de variadas dimensões. O usuário pode também especificar algumas variáveis e examinar a mudança dos resultados em cada caso. O simulador tem o objetivo de ser uma ponte entre a teoria apresentada nos módulos do curso e as possíveis aplicações práticas da técnica de Programação Evolucionária.

## 4.2 Estrutura do curso

O curso procura ter uma abordagem atemporal e não necessitar de uma supervisão constante de um tutor. Para satisfazer esses dois requisitos, foi escolhida para o curso uma estrutura dividida em módulos, que funcionam como estágios de aprendizagem na formação do conhecimento.

Cada módulo é basicamente composto por um documento de texto contendo a teoria do assunto tratado no módulo, alguns *links* interessantes que possam adicionar algo de novo ou até mesmo buscar abordagens diferentes ao assunto que foi apresentado e um pequeno questionário para que o estudante teste o conhecimento obtido com a realização do módulo. A tabela 2 mostra o conteúdo de cada módulo do curso.

Nas próximas subseções listaremos os módulos do curso, explicando um pouco sobre o assunto abordado em cada um deles.

Módulo	Título	Conteúdo
I	Introdução	<ul style="list-style-type: none"> <li>- Teoria da evolução proposta por Charles Darwin</li> <li>- História da Computação Evolucionária</li> </ul>
II	Algoritmos Evolucionários	<ul style="list-style-type: none"> <li>- Visão geral dos algoritmos evolucionários e seus componentes</li> </ul>
III	Programação Evolucionária	<ul style="list-style-type: none"> <li>- História da Programação Evolucionária</li> <li>- Primeiras aplicações</li> <li>- Particularidades em seu algoritmo</li> <li>- Simulador de mutações</li> </ul>

**Tabela 2.** Tabela estrutural do curso

#### 4.2.1 Módulo I – Introdução

O primeiro módulo do sistema visa situar o estudante no universo que fará parte de todo o curso, ele possui a fundamentação teórica dos princípios básicos da Computação Evolucionária.

O conteúdo teórico do módulo inicia com uma breve explicação sobre a história do naturalista inglês Charles Darwin e sua teoria da evolução, descrita no seu livro “A Origem das Espécies” de 1859. Em seguida, o módulo conta com o histórico do surgimento da Computação Evolucionária, desde as primeiras idéias de usar o conceito de Darwin na resolução de problemas computacionais até o surgimento das técnicas que temos hoje, como por exemplo a Programação Evolucionária.

#### **4.2.2 Módulo II – Algoritmos Evolucionários**

Este módulo é o primeiro contato do estudante com o funcionamento real da tecnologia. Nele serão apresentados os algoritmos evolucionários, que são a base de todas as técnicas de Computação Evolucionária.

São apresentados os componentes dos algoritmos (a representação dos indivíduos, a função de avaliação, o mecanismo de seleção dos pais, os operadores de mutação e recombinação e o mecanismo de seleção dos indivíduos sobreviventes) e cada um deles é explicado em detalhes, fazendo sempre a ligação entre o mundo real e a seus representantes codificados no algoritmo.

#### **4.2.3 Módulo III – Programação Evolucionária**

Este módulo que apresenta a técnica de Computação Evolucionária que é abordada no curso, a Programação Evolucionária.

Inicialmente é apresentado um histórico da técnica e o porque do surgimento do nome. Cada item do algoritmo é revisto apresentando suas particularidades na Programação Evolucionária, sempre sendo feito uma comparação com as outras técnicas e listando as vantagens e desvantagens em cada caso.

Neste módulo é também apresentado o simulador construído, para que o estudante teste na prática os efeitos das mutações na Programação Evolucionária.

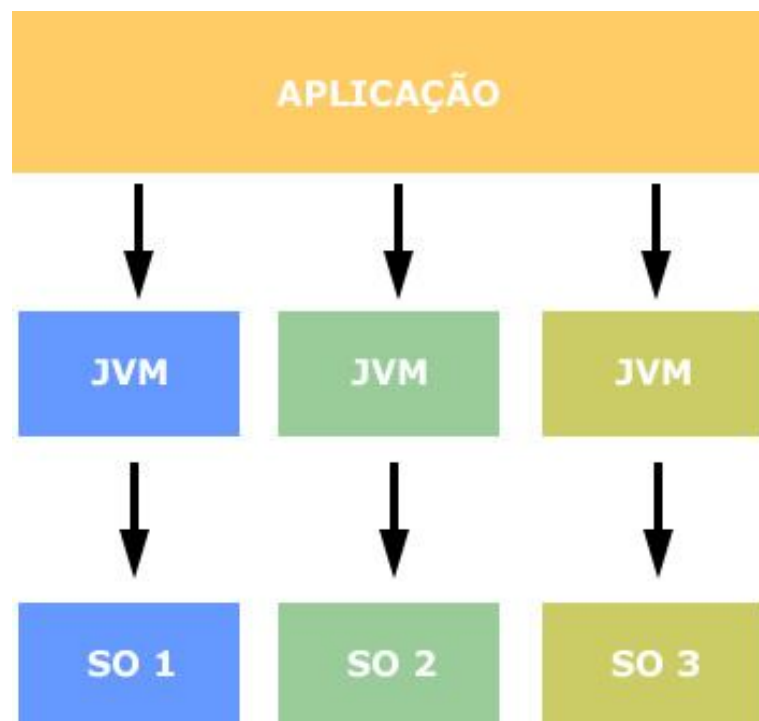
### **4.3 Ferramentas utilizadas**

Como sistema de gerenciamento de aprendizagem, o Moodle foi escolhido principalmente devido à sua popularidade, à sua independência de plataforma e por ser um software livre, com o código totalmente aberto.

O Moodle é um sistema com uma modularidade muito grande, ele possui suporte para vários bancos de dados do mercado como MySQL, Postgre, SQL Server e Oracle.



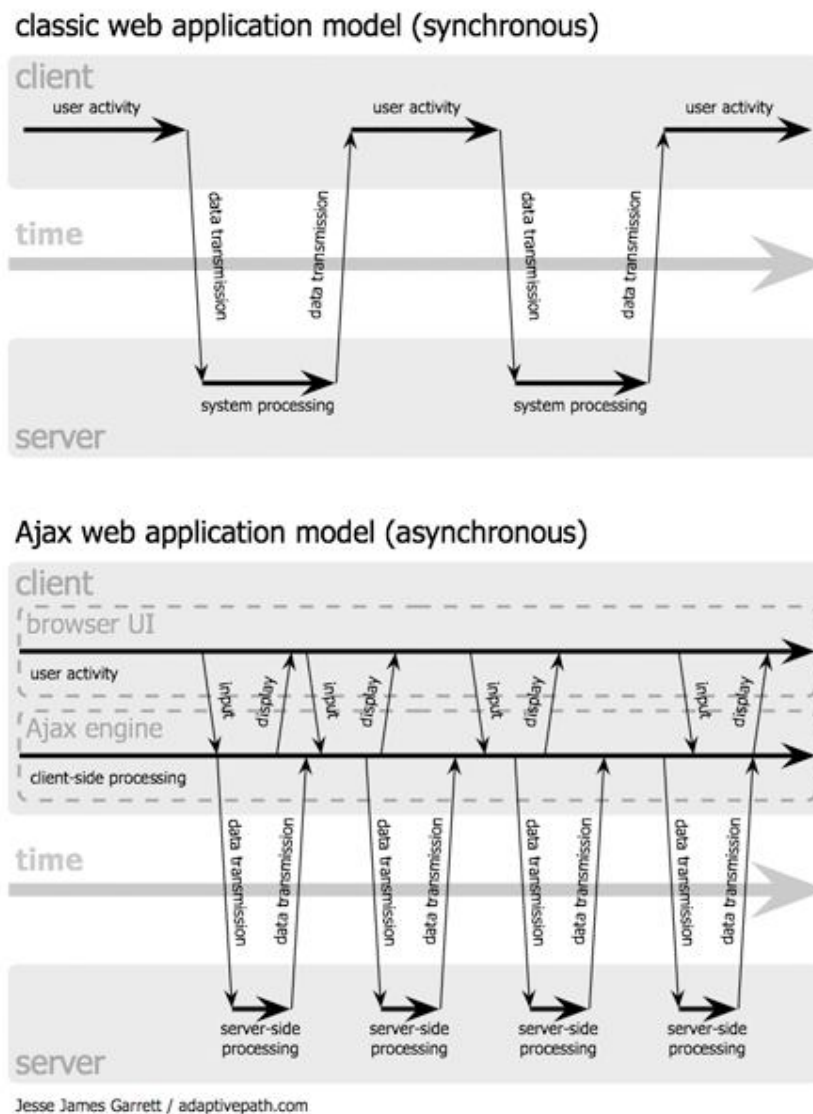
Para a implementação do simulador, foi usada a linguagem de programação Java J2EE, versão 6, por motivos bem parecidos com a escolha do LMS: como o fato de possuir uma imensa comunidade de desenvolvedores, além de também ser independente de plataforma e gratuita. A linguagem Java foi criada em 1992, é uma linguagem orientada à objetos e tem como principal diferencial a sua máquina virtual, que interage com o sistema operacional, deixando o programador se preocupar apenas com a implementação das suas regras de negócio, e não com a interação do seu programa com os sistemas operacionais. Foram utilizadas no projeto páginas JSP, que são arquivos que possuem tanto código HTML quanto código Java. Assim a página mostra os resultados das simulações feitas pelo o usuário e ao mesmo tempo compõe a interface da aplicação. A figura 7 mostra a interação entre as máquinas virtuais Java e diferentes sistemas operacionais.



**Figura 7.** Cada máquina virtual Java interagindo com o seu respectivo sistema operacional

Para obter uma interface mais amigável ao usuário, foi utilizado AJAX em todas as requisições do sistema. O AJAX, *Asynchronous Javascript and XML*,

consiste em uma forma de utilizar tecnologias como javascript e XML, que são executadas pelo *browser* do usuário, fazendo com que as páginas possam fazer requisições ao servidor sem a necessidade de recarregar toda a página. AJAX não é uma tecnologia, e sim uma integração de várias tecnologias existentes. A figura 8 mostra o fluxo de uma aplicação web síncrona e uma assíncrona utilizando AJAX.



**Figura 8.** Comparação entre um sistema síncrono e um sistema assíncrono com AJAX

Para auxiliar na utilização do AJAX no sistema, foi usado o *framework* JQuery, ele tem a função de ajudar o desenvolvedor a evitar a incompatibilidade de

código entre os navegadores existentes, além de reduzir a quantidade de código necessária para efetuar ações mais comuns de um sistema *web*. O JQuery prega também a reutilização de código, por meio de *plugins*, que hoje já são milhares espalhados na rede evitando a reescrita desnecessária de código.

A plataforma utilizada no desenvolvimento foi o NetBeans IDE 6.5.1, que é um ambiente de desenvolvimento integrado, gratuito e de código aberto, como quase todas as ferramentas que auxiliaram no desenvolvimento deste trabalho. O NetBeans foi também desenvolvido na linguagem Java e possui versões para vários sistemas operacionais, como Windows, Linux, Mac OS e Solaris. Ele possui suporte para várias linguagens além do Java, como por exemplo Ruby e PHP.

## 4.4 Ambiente desenvolvido

Nesta seção o desenvolvimento do curso de Computação Evolucionária será descrito em detalhes. Cada módulo criado e funcionalidade implementada será analisada, sempre apresentando as telas do sistema e listando as dificuldades e problemas encontrados durante a concepção, e mostrando como estes foram resolvidos. A figura 9 mostra a tela inicial do curso desenvolvido.



Curso de Computação Evolucionaria

Você acessou como Fernando Fragoso: Estudante ()

EngComp POLI UPE > CCE101

Participantes

Atividades

Buscar nos Fóruns

Administração

Meus cursos

Programação

Fórum de notícias

1 Módulo I - Introdução

Neste módulo veremos uma introdução ao conceito de computação evolucionária.

Iniciaremos com uma rápida explicação sobre a teoria da evolução proposta por Darwin, seguiremos com a abstração dos conceitos evolucionários e sua aplicação na resolução de problemas computacionais. Por fim veremos um pouco da história do surgimento do termo Computação Evolucionária e suas sub-áreas.

Curso de Computação Evolucionária - Introdução

Vídeo - Seleção natural fácil de entender

2 Módulo II - Algoritmos Evolucionários

Neste módulo entraremos na parte conceitual da Computação Evolucionária.

Seremos apresentados aos chamados Algoritmos Evolucionários, a base de todas as técnicas de Computação Evolucionária. Veremos, um por um, todos os componentes que compõem esses algoritmos.

Curso de Computação Evolucionária - Algoritmos Evolucionários

3 Módulo III - Programação Evolucionária

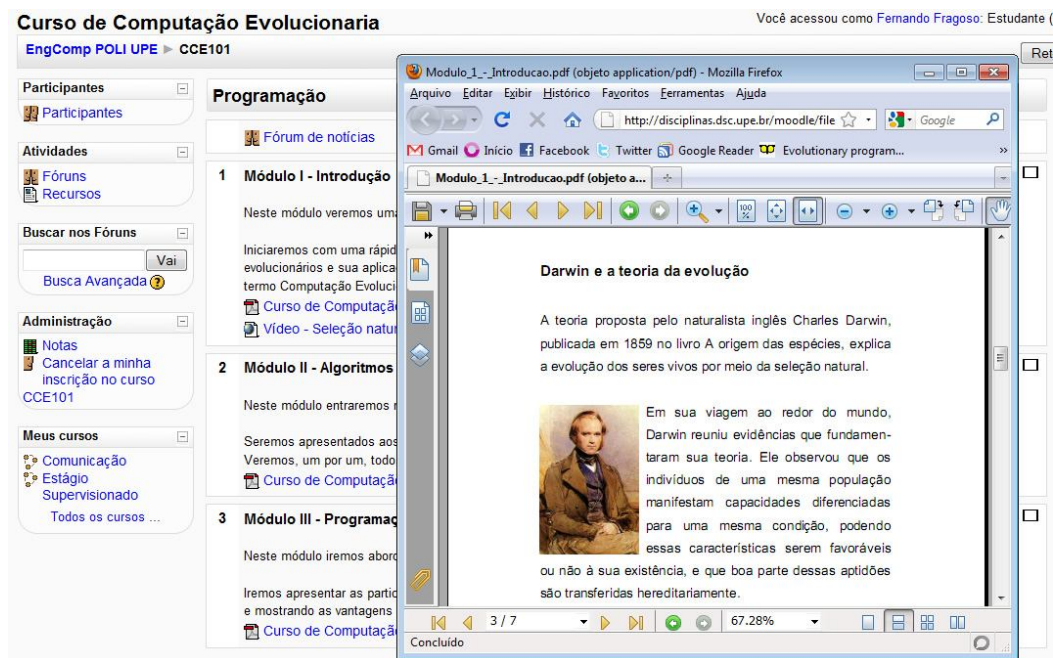
Neste módulo iremos abordar uma das técnicas de Computação Evolucionária, a Programação Evolucionária.

Iremos apresentar as particularidades da Programação Evolucionária em cada componente, sempre comparando com outras técnicas e mostrando as vantagens e desvantagens entre elas.

Curso de Computação Evolucionária - Programação Evolucionária

Figura 9. Tela inicial

A Figura 9 mostra a tela inicial do curso, onde tem-se uma visão geral das ferramentas do sistema e dos módulos existentes no curso. Cada módulo possui um pequeno texto demonstrando os objetivos que pretendem ser alcançados pelo estudo do mesmo. Além disso nessa tela são listados os recursos e atividades disponíveis em cada módulo, como documentos, vídeos, *links* interessantes e questionários.



**Figura 10.** Recurso de texto do curso sendo visualizado.

A Figura 10 mostra a visualização de um recurso de texto de um dos módulos do curso. Cada módulo possui um documento em formato PDF, que tem o intuito de ser uma apostila apresentando o seu conteúdo teórico. As apostilas possuem uma linguagem simples, com um espaçamento adequado e o uso de imagens sempre que possível para evitar de que o texto tenha uma leitura cansativa e desencoraje o estudante. As apostilas podem ser salvas no computador do usuário ou impressas facilmente para visualização *offline*.

A Figura 11 apresenta mais um recurso utilizado pelo curso, neste caso um vídeo externo. Os módulos possuem sempre *links* interessantes que possam mostrar abordagens diferentes ou novos pontos de vista em relação ao texto base do módulo, aumentando as fontes e descentralizando o conhecimento obtido pelo estudante matriculado no curso.

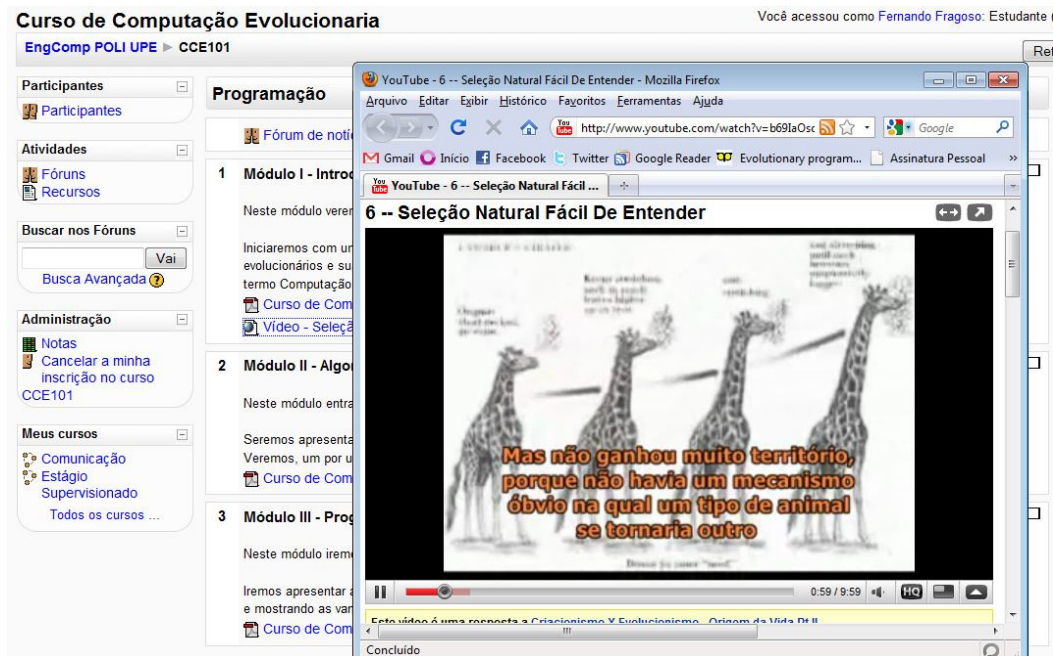


Figura 11. Recurso de vídeo.

O simulador de programação evolucionária é mais um recurso interessante que deve ser analisado pelos estudantes. Ele pretende ser uma demonstração do uso dos conceitos apresentados durante todo o curso de forma prática. O programa consiste em uma interface com 4 tipos de algoritmos de mutação e 16 funções de teste, parâmetros como o tamanho da população, quantidade de gerações e número de comparações durante a seleção podem ser modificados e seu comportamento estudado em cada uma das simulações. A figura 12 mostra o diagrama de classes do simulador.

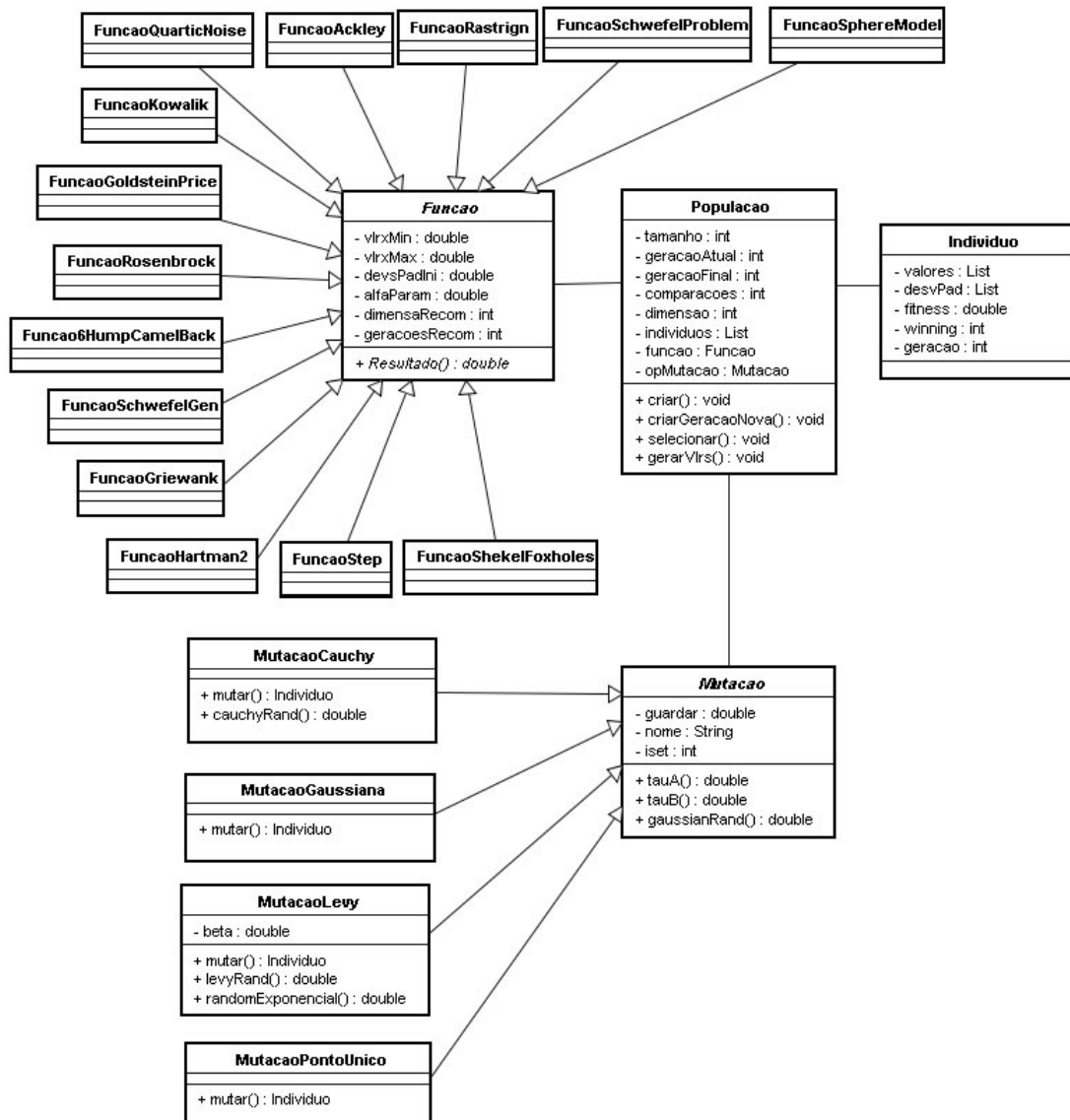
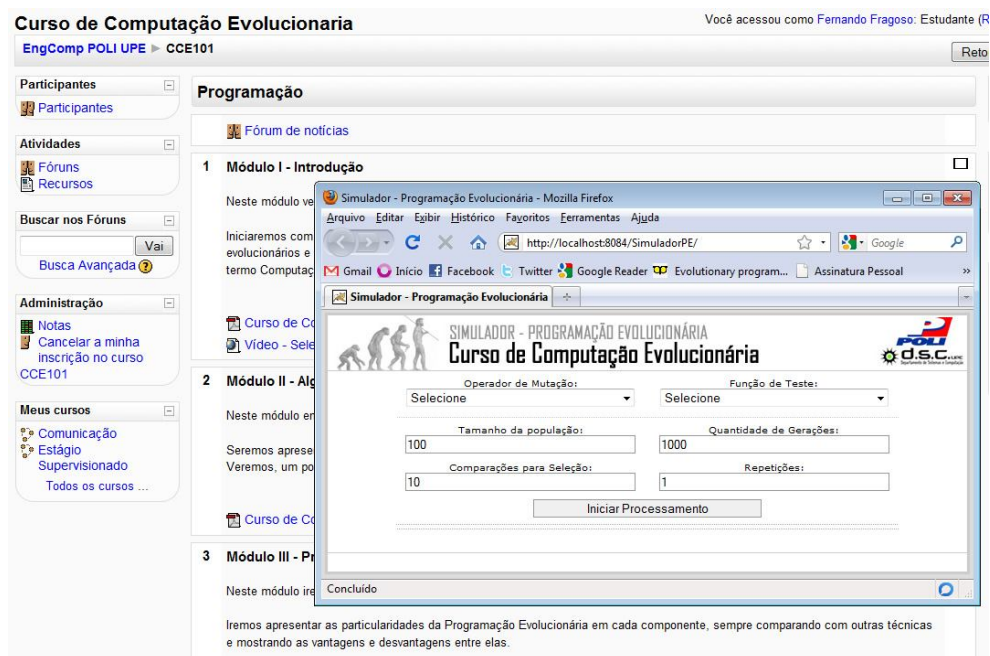


Figura 12. Diagrama de Classes

A proposta inicial seria transformar o simulador em um *plugin* para o Moodle, porém foram encontrados problemas na hora da integração, principalmente por causa da linguagem utilizada no simulador, o Moodle foi desenvolvido em PHP e o simulador em JAVA, e isso tornou a integração uma tarefa complicada.

Outro caminho analisado foi implementar o simulador em PHP para tornar a integração como *plugin* do Moodle mais simples, mas a linguagem se tornou um impecílio, algumas operações matemáticas não eram realizadas com a mesma eficiência o que trazia distorções nos resultados que impossibilitavam o trabalho do simulador. Assim, foi decidido manter o simulador em Java e não integrá-lo ao Moodle.



**Figura 13.** Simulador de Programação Evolucionária

A figura 13 apresenta a interface do simulador de programação evolucionária. Apesar de não estar integrado ao Moodle, ele apresenta uma identidade visual compatível com todo o resto do curso, o que faz com que a idéia de unidade não se perca.

A figura 14 mostra o programa apresentando o resultado de uma simulação de teste efetuada com o operador *gaussiano* de mutação agindo sobre a função de Ackley, uma população de 100 indivíduos se desenvolvendo por 1500 gerações, utilizando 10 comparações na hora da seleção dos indivíduos.

**Curso de Computação Evolucionária** Você acessou como **Fernando Fragoso**: Estudante (Re)

EngComp POLI UPE > CCE101 Reton

**Participantes**

Participantes

**Atividades**

Fóruns  
Recursos

**Buscar nos Fóruns**

Vai

Busca Avançada ?

**Administração**

Notas  
Cancelar a minha inscrição no curso CCE101

**Meus cursos**

Comunicação  
Estágio Supervisionado  
Todos os cursos ...

**Programação**

Fórum de notícias

**1 Módulo I - Intro**

Neste módulo ver

Iniciaremos com evolucionários e e termo Computaçã

Curso de Co

Vídeo - Sele

**2 Módulo II - Alg**

Neste módulo ent

Seremos apresen

Veremos, um por

Curso de Co

**3 Módulo III - Pr**

Neste módulo ire

Iremos apresenta

Simulador - Programação Evolucionária - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:8084/SimuladorPE/

Gmail Início Facebook Twitter Google Reader Evolutionary program... Assinatura Pessoal

**Simulador - Programação Evolucionária**

Operador de Mutação: Gaussiano Função de Teste: Ackley

Tamanho da população: 100 Quantidade de Gerações: 1500

Comparações para Seleção: 10 Repetições: 3

Iniciar Processamento

Função	Operador de Mutação	População	Gerações	Média
Ackley	Gaussiano	100	1500	17.0480
Ackley	Gaussiano	100	1500	15.2445
Ackley	Gaussiano	100	1500	15.9412

Concluído

**Figura 14.** Resultados da simulação



# Capítulo 5

## Conclusão e Trabalhos Futuros

Este capítulo apresenta as conclusões gerais acerca do projeto realizado, listadas as dificuldades que foram encontradas durante o seu desenvolvimento e propõe algumas idéias de caminhos que podem ser seguidos para dar continuidade ao trabalho aqui iniciado.

O curso de Computação Evolucionária que foi desenvolvido demonstrou a grande utilidade que podem ter os Sistemas de Gerenciamento de Aprendizado mesmo quando são utilizados para auxiliar no ensino de temas multidisciplinares e complexos. Porém, para que um curso obtenha êxito é preciso ser desenvolvido com cautela, sempre tentando prender a atenção do aluno, utilizando recursos diferenciados, como animações e vídeos, ao invés de apenas textos, o que transformaria o aprendizado na frente do computador uma experiência fatigante ao aluno.

Também é importante acrescentar a importância do *feedback* do tutor no processo de aprendizagem, que deve estar sempre presente, participando ativamente dos fóruns e *chats* que as ferramentas LMS normalmente disponibilizam. O uso desse tipo de sistema de ensino se torna muito mais efetivo quando é utilizado como complemento às aulas presenciais tradicionais ao invés de apenas funcionar como uma ação isolada de educação à distância.

### 5.1 Dificuldades encontradas

Antes mesmo dos problemas de adaptação à ferramenta e implementação do simulador, uma grande dificuldade encontrada foi na definição dos requisitos para que o curso obtivesse uma boa aceitação entre os estudantes. Educação à distância é uma área de pesquisa ainda recente, e entrar nesse universo se mostrou uma grande mudança de paradigma e foi a principal dificuldade encontrada.

O processo de desenvolvimento do simulador também apresentou dificuldades, tendo em vista que a idéia inicial seria implementá-lo como um *plugin* para obter uma integração completa ao Moodle, mas a linguagem em que ele foi escrito (Java) dificultou bastante essa idéia. A tentativa de mudança da linguagem do simulador para PHP para que esse fosse compatível com o Moodle e facilitasse a integração também não foi bem sucedida, dessa vez devido a matemática necessária para o simulador não ter sido bem executada pelo PHP, gerando inconsistências em números com ponto flutuante.

## 5.2 Trabalhos futuros

Para dar continuidade ao trabalho, novos módulos podem ser implementados, assim como foi feito com o módulo de programação evolucionária, as outras sub-áreas da Computação Evolucionária também podem ser apresentadas, como por exemplo os algoritmos genéticos e a programação genética.

O problema da integração do simulador de programação evolucionária, ou de novos simuladores, pode ser resolvido com um tempo maior disponível para pesquisa. A integração completa do simulador à ferramenta poderia promover o compartilhamento dos resultados das simulações entre os estudantes do curso, assim conseguindo de forma colaborativa fazer estudos sobre eficiência dos operadores de mutação em cada função de teste, e assim talvez chegar a resultados interessantes.

# Bibliografia

- [1] Darwin, C. **A Origem das Espécies**. Porto: Lello & Irmãos (2003) 572 p.
- [2] Eiben, A. E.; Smith J. E. **Introduction to Evolutionary Computing**. 1 ed. Berlin: Springer (2003) 299 p. (Natural computing series)
- [3] Fogel, L. J. **Biotechnology: Concepts and Applications**, Prentice-Hall, Englewood, (1963) 826p.
- [4] L. J. Fogel and D. B. Fogel **Artificial intelligence through evolutionary programming**, Final Report (1986), U.S. Army Research Institute.
- [5] Yao, X.; Liu, Y; Lin, G. **Evolutionary programming made faster**. IEEE Transactions on Evolutionary Computation 3 (2)(1999) 82-102
- [6] Ji, M. Tang, H. Guo, J. **A single-point mutation evolutionary programming**. Information Processing Letters 90 (2004) 293-299
- [7] Rumble, G. A. **Gestão dos Sistemas de Ensino à Distância**. UNESCO: Instituto Internacional de Planejamento de Educação, IIPÉ, Paris, 1993
- [8] Ellis, R. K. **Field Guide to Learning Management Systems**, 2009, ASTD Learning Circuits
- [9] Coates H. **Leveraging LMSs to enhance Campus-Based student engagement**, Educause Quarterly, number 1, 2005, p. 66
- [10] Lisboa, E. S.; Jesus, A. G.; Varela, A. M. **LMS em Contexto Escolar: estudo sobre o uso do Moodle pelos docentes de duas escolas no Norte de Portugal**, 2009, Educação, Formação & Tecnologias, Vol 2, pp. 44-57

