

GABRIEL SILVA RAMOS

**DETECÇÃO E REMEDIAÇÃO DE ERROS NA
GENERALIZAÇÃO DE CONCEITOS MATEMÁTICOS POR
MEIO DE SISTEMAS TUTORES INTELIGENTES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Alexandre Ibrahim Direne

CURITIBA

2010

GABRIEL SILVA RAMOS

**DETECÇÃO E REMEDIAÇÃO DE ERROS NA
GENERALIZAÇÃO DE CONCEITOS MATEMÁTICOS POR
MEIO DE SISTEMAS TUTORES INTELIGENTES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Alexandre Ibrahim Direne

CURITIBA

2010

GABRIEL SILVA RAMOS

**DETECÇÃO E REMEDIAÇÃO DE ERROS NA
GENERALIZAÇÃO DE CONCEITOS MATEMÁTICOS POR
MEIO DE SISTEMAS TUTORES INTELIGENTES**

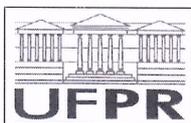
Dissertação aprovada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Informática da Universidade Federal do Paraná, pela Comissão formada pelos professores:

Orientador: Prof. Dr. Alexandre Ibrahim Direne
Departamento de Informática - UFPR

Membro Interno: Prof. Dr. Andrey Ricardo Pimentel
Departamento de Informática - UFPR

Membro Externo: Prof. Dr. Davidson Cury
Departamento de Informática - UFES

Curitiba, 30 de Agosto de 2010

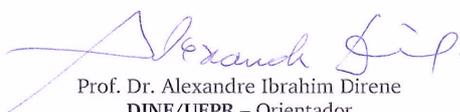


Ministério da Educação
Universidade Federal do Paraná
Programa de Pós-Graduação em Informática

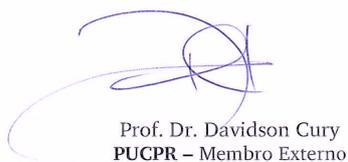
PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, do aluno Gabriel Silva Ramos, avaliamos o trabalho intitulado, *“DETECÇÃO E REMEDIAÇÃO DE ERROS NA GENERALIZAÇÃO DE CONCEITOS MATEMÁTICOS POR MEIO DE SISTEMAS TUTORES INTELIGENTES”*, cuja defesa foi realizada no dia 30 de agosto de 2010, às 14:00 horas, Laboratório Multiterminal do Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação do candidato.

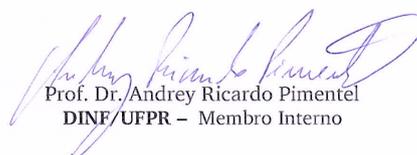
Curitiba, 30 de agosto de 2010.



Prof. Dr. Alexandre Ibrahim Direne
DINF/UFPR – Orientador



Prof. Dr. Davidson Cury
PUCPR – Membro Externo



Prof. Dr. Andrey Ricardo Pimentel
DINF/UFPR – Membro Interno

AGRADECIMENTOS

A lista de pessoas que merecem receber minha imensa gratidão é grande e, portanto, mesmo que tentasse, não conseguiria mencionar todos aqui. O primeiro agradecimento vai para meu grande amigo e orientador, Alexandre Direne, por ter me guiado sempre com imenso entusiasmo e sabedoria durante toda essa trajetória, e por ter acreditado e confiado plenamente na minha capacidade até o fim.

Agradeço do fundo de meu coração aos meus pais, Marcelene e Laertes, a quem dedico este trabalho, por terem me dado a oportunidade de perseguir meus sonhos e objetivos, sempre me incentivando, motivando e acompanhando nos altos e baixos dessa caminhada, e por serem os responsáveis por tudo que sou.

Gostaria de agradecer a todos os meus amigos e colegas, que compartilharam comigo esses momentos difíceis dessa etapa de minha vida, me ajudando a levantar quando estive caído, sempre com muita compreensão e compaixão.

Agradeço também a todas as pessoas envolvidas no projeto CONDIGITAL, pois sem eles, este trabalho não teria sido possível. Mais especificamente, agradeço aos colegas Lourdes Almeida, Luciana Gastaldi, Bárbara Palharini, Karina Pessoa, Márcia Cyrino e Rodolfo Vertuan, docentes e alunos de pós-graduação stricto sensu do Departamento de Matemática da UEL, que ajudaram a especificar os conteúdos que foram transformados em software.

Por fim, gostaria de agradecer a todos os professores e funcionários do Departamento de Informática da UFPR, os quais, direta ou indiretamente contribuíram para a realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS	vi
LISTA DE SIGLAS	vii
RESUMO	viii
ABSTRACT	ix
1 INTRODUÇÃO	1
1.1 Problema central	1
1.2 Objetivos do trabalho	3
1.2.1 Objetivo geral	3
1.2.2 Objetivos específicos	3
1.3 Contexto do projeto	4
1.4 Estrutura da dissertação	5
2 RESENHA LITERÁRIA	7
2.1 Histórico crítico da Inteligência Artificial na Educação	7
2.1.1 Sistemas Tutores Inteligentes em geral	7
2.1.2 Diagnosticadores de erro em conhecimento procedimental	9
2.1.3 Arquiteturas cognitivistas dos Sistemas Tutores Inteligentes	11
2.2 Histórico crítico sobre os Micromundos	13
2.2.1 Falta de apoio à explicação sobre erros	13
2.2.2 Aspectos de representação externa	14
3 FUNDAMENTOS DA SOLUÇÃO ADOTADA	16
3.1 Estudo Empírico	16
3.1.1 Entrevistas com especialistas em Ensino de Matemática	17
3.1.2 Aplicação de um Objeto de Aprendizagem	17

3.2	Conceitos da edição gráfica de fractais	19
3.2.1	Requisitos funcionais para a edição gráfica de fractais	19
3.2.2	Ligação de conceitos gramaticais e analíticos dos fractais	21
3.3	Requisitos funcionais do diagnóstico	21
4	IMPERFEIÇÕES NO PROCESSO DE GENERALIZAÇÃO	25
4.1	Generalização de aspectos cognitivos	25
4.1.1	Erros no processo indutivo humano	25
4.1.2	Caracterização de erros	26
4.1.3	Razão para diagnóstico automático	28
4.1.4	Automação do diagnóstico de erros com remediação	29
4.2	Generalização de aspectos pedagógicos	30
4.2.1	Finalidade contextualizadora de elementos introdutórios	31
4.2.2	Finalidade do encadeamento dos exercícios	31
5	ARQUITETURA FUNCIONALISTA DA FERRAMENTA	34
5.1	Arquitetura da ferramenta SAGA	34
5.2	Especificação dos módulos	35
5.2.1	Interface do Aprendiz	35
5.2.2	Interface do Autor	37
5.2.3	Analisador de Expressões	38
5.2.4	Analisador Aritmético	38
5.2.5	Interpretador de Regras de Produção	38
5.2.6	Controlador de Enunciados	39
5.3	Cenários de aplicação	40
5.3.1	Progressões Geométricas em Fractais	40
5.3.2	Funções de Primeiro Grau	43
6	CONCLUSÃO	46
6.1	Reafirmação da contribuição	46
6.2	Trabalhos Futuros	47

A	OA DE PROGRESSÕES GEOMÉTRICAS EM FRACTAIS	49
B	OA DE FUNÇÕES DE PRIMEIRO GRAU	53
C	OA DE MATEMÁTICA FINANCEIRA	60
	REFERÊNCIAS BIBLIOGRÁFICAS	79

LISTA DE FIGURAS

4.1	Exercício proposto pelo OA de Progressões Geométricas em Fractais	29
5.1	Arquitetura funcionalista da ferramenta SAGA	35
5.2	Interface do Objeto de Aprendizagem	36
5.3	Interface de Autoria para o domínio de Fractais	37
5.4	Fractal Floco de neve de Koch	41
5.5	Fractal dos Hexágonos	42
5.6	Um enunciado do OA de Funções de Primeiro Grau	44
5.7	Exercício de roldanas do OA de Funções de Primeiro Grau	45

LISTA DE SIGLAS

ACT* *Adaptive Character of Thought*

ACT-R *Adaptive Control of Thought–Rational*

C3SL Centro de Computação Científica e Software Livre

CETEPAR Centro de Excelência em Tecnologia Educacional do Paraná

CONDIGITAL Projeto Conteúdos Digitais

FNDE Fundo Nacional de Desenvolvimento da Educação

IA Inteligência Artificial

IFS *Iterated Function System Fractals*

L-system *Lindenmayer System*

LACTEC Instituto de Tecnologia para o Desenvolvimento

MEC Ministério da Educação e Cultura

MCT Ministério da Ciência e Tecnologia

MRE Múltiplas Representações Externas

OA Objeto de Aprendizagem

RE Representação Externa

SAGA Sistema de Aprofundamento por Generalização na Aprendizagem

SEED Secretaria de Educação a Distância

STI Sistema Tutor Inteligente

UFPR Universidade Federal do Paraná

RESUMO

Neste documento são apresentados os aspectos que justificam a necessidade de se aplicar diagnóstico automático ativo seguido de intervenção imediata durante a interação de um aprendiz com um Objeto de Aprendizagem (OA) para o ensino de conceitos matemáticos. Poucos trabalhos foram encontrados na literatura científica sobre OA que permitam a classificação de erros de generalização de conceitos, com a finalidade de enriquecer e precisar o diagnóstico. Conceitos referentes à criação de ferramentas com essas características são estabelecidos, bem como uma arquitetura baseada neles é desenvolvida. Um protótipo baseado em tal arquitetura é construído e aplicado em domínios que envolvem conceitos analíticos, como Progressões Geométricas em Fractais e Funções de Primeiro Grau. Por fim, são apresentadas as perspectivas de pesquisa futura.

ABSTRACT

In this document we present the aspects that justify the need of applying automatic active diagnostic followed by immediate intervention, during the interaction between an apprentice and a Learning Object (LO) focused on the teaching of Math concepts. A few works have been found in the scientific literature about LO that allow the classification of concept generalization bugs, with the objective of improving and precise the diagnostic. Concepts concerning the creation of tools with such characteristics are established, as well as an architecture is developed, based on them. A prototype based on such architecture is built and applied in domains that involve analytic concepts, like Geometric Progressions in Fractals and First Degree Functions. Finally, the perspectives of future research are presented.

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo apresentamos uma visão geral do trabalho de mestrado que culminou na elaboração desta dissertação. Nas seções que se seguem, especificamos o problema central abordado no projeto, destacando os objetivos gerais e específicos a serem alcançados, além de identificarmos o contexto no qual esta pesquisa se originou. Ao final do capítulo descrevemos brevemente a organização dos tópicos abordados nos capítulos seguintes.

1.1 Problema central

A solução de problemas analíticos é constituída de tarefas de alta demanda cognitiva. Alguns psicólogos cognitivistas especializados no conhecimento de Matemática e Física chegam a dizer que tais problemas formam barreiras grandes demais para a faixa etária na qual são aplicados nas escolas [52]. Os iniciantes da Análise enfrentam um grande número de dificuldades na sua aprendizagem. Em uma primeira fase, os aprendizes devem compreender os princípios do conhecimento analítico, tais como: (a) aritmética básica; (b) noção de variáveis; (c) sintaxe de operadores e operandos; (d) propriedades algébricas (*e.g.*, comutatividade, associatividade, distributividade e fatoração); (e) semântica das propriedades algébricas como transformações isoladas assim como a precedência de operadores.

Em uma segunda fase, os aprendizes precisam adquirir a perícia de integrar a sintaxe e a semântica da linguagem analítica para produzir planos de transformação e grafá-los no formato de expressões. Em outras palavras, essa perícia requer a capacidade de combinar conceitos de transformação e verificação de equivalências para manipular os sub-objetivos de uma solução de problema. Por exemplo, ao redigir uma solução para um passo de um exercício de um software educacional, o aprendiz pode precisar de um plano para: (a) deduzir ou copiar a formula da área de um triângulo equilátero; (b) contar a quantidade

de figuras geométricas básicas (*e.g.*, triângulos equiláteros) que ocorrem no passo; (c) determinar que uma variável bem geral (*e.g.*, ordem do passo) é exigida na expressão de resposta; (d) conjugar, em uma única função, o resultado da contagem com outro parâmetro, tal como a própria ordem do passo.

A maior parte dos ambientes interativos de aprendizagem que apoiam o ensino de campos da Matemática limitam sua intervenção ao plano exclusivamente aritmético [52]. Os raros casos de Sistemas Tutores Inteligentes (STI) que tentaram avançar sobre o domínio de Álgebra [10, 12] parecem ter uma atuação restrita quanto às intervenções sobre o aprendiz nos casos de erro.

Para simplificar a tarefa dos iniciantes ao escreverem expressões analíticas, parece ser necessária uma nova abordagem no campo de pesquisa de STI de tal forma a: (a) oferecer problemas gradativamente mais complexos de maneira a formar no aprendiz uma base voltada a exemplos relevantes para seu nível aproximado de competência; (b) dar assistência personalizada diante de um erro do aprendiz por meio do diagnóstico automático ativo seguido de intervenção imediata para desenvolver perícia; (c) promover a visualização de características das expressões matemáticas de maneira a interligar os princípios com as perícias de manipulação algébrica por meio de algum fator dinâmico de orientação gráfica passo-a-passo.

Há várias vantagens instrucionais na aplicação de diagnóstico automático ativo seguido de intervenção imediata para o desenvolvimento de perícias. Uma delas está na detecção e remediação de erros dentro do contexto específico em que eles ocorrem. Diante de tal especificidade, um iniciante tem maiores chances de compreender bem as mensagens automaticamente geradas pelo STI e, a partir delas, desenvolver perícia. Uma outra vantagem está na capacidade de um STI em analisar soluções parciais do aprendiz. A cada evento mínimo de interação (uma tecla digitada, ou um clique de mouse, por exemplo) o STI é capaz de reconhecer um desvio da solução correta. Dessa forma não há a necessidade de esperar que o aprendiz progrida com um erro até completar uma solução errada para então haver intervenção do sistema [5, 6]. Como consequência disso, há uma redução no número de erros que ocorrem concorrentemente, fazendo com que o algoritmo de busca

heurística seja potencialmente mais eficaz em seu diagnóstico, caracterizando mais uma vantagem [42].

Como forma de investigação empírica, a presente pesquisa procurou adotar duas linhas lógicas de projeto e implementação para diversos STI que possuem o mesmo arcabouço arquitetural. A primeira linha toca a estrutura de visualização geométrica como a Representação Externa [3] mais concreta a ser gerada para facilitar a compreensão da dinâmica de aplicação de conceitos algébricos. A segunda linha, que foi a que escolhemos seguir, se relaciona com a combinação entre os recursos de visualização citados e o de diagnóstico inteligente de erros em expressões analíticas para formar um ambiente integrado a ser usado pelo aprendiz desses conceitos analíticos. Vale ainda ressaltar que tais STI se destinam a apoiar o ensino de Matemática escolar do nível médio, pois se amparam em aspectos mais típicos da perícia do que dos princípios de modelagem analítica para gerar explicações sobre erros.

1.2 Objetivos do trabalho

Nesta seção são apresentados os objetivos gerais e específicos que este trabalho pretendeu alcançar durante seu desenvolvimento.

1.2.1 Objetivo geral

De maneira geral, o objetivo principal deste trabalho foi efetuar estudos de pesquisa e desenvolvimento para criar uma ferramenta de software capaz de realizar diagnóstico ativo sobre erros analíticos cometidos por aprendizes do Ensino Médio. Tal diagnóstico visa direcionar o aprendiz a compreender seus erros conceituais, para que possa ser capaz de remedia-los posteriormente.

1.2.2 Objetivos específicos

Tendo em vista o objetivo geral delineado anteriormente, podemos destacar como secundários, ou mais específicos, os seguintes objetivos:

- Elaborar, projetar e conduzir a criação de ferramentas de software de cunho educacional para o ensino e aprendizagem de conceitos matemáticos de nível médio;
- Desenvolver ferramentas de software capazes de realizar diagnóstico passivo sobre erros cometidos por aprendizes durante o uso de software educacional de ensino de conceitos matemáticos;
- Construir um interpretador de regras de produção estratificado para abordar primariamente o diagnóstico de sub-generalização, super-generalização e até erros de miscelânea;
- Disseminar o conhecimento produzido e adquirido em publicações da área de Informática na Educação;

1.3 Contexto do projeto

O presente trabalho é decorrente do projeto CONDIGITAL do grupo do estado do Paraná. Este grupo é financiado pelo Fundo Nacional de Desenvolvimento da Educação (FNDE) por meio do **Edital 001/07 MEC/MCT**. O projeto é uma iniciativa da Secretaria de Educação a Distância (SEED) do Ministério da Educação (MEC) e do Ministério da Ciência e Tecnologia (MCT).

Na Universidade Federal do Paraná (UFPR), este projeto é desenvolvido pelo Centro de Computação Científica e Software Livre (C3SL) em parceria com o Instituto de Tecnologia para o Desenvolvimento (LACTEC), o Centro de Excelência em Tecnologia Educacional do Paraná (CETEPAR) e a Universidade Estadual de Londrina (UEL). Um de seus objetivos principais é o de contribuir para a melhoria e a modernização dos processos de ensino e aprendizagem da área de Matemática (além de outras disciplinas) na rede de escolas públicas, e até mesmo privadas, do Brasil.

O projeto previa o desenvolvimento de quatro Objetos de Aprendizagem (OA) para apoiar o ensino de Matemática para a faixa etária que compreende o último ano do Ensino Fundamental e todo o Ensino Médio. Os OA devem ser utilizados para apoiar atividades laboratoriais nos seguintes sub-domínios: (a) progressões geométricas que ocorrem em

elementos da geometria fractal; (b) funções de primeiro grau, com exemplos sobre o coeficiente de elasticidade de molas e composições de roldanas móveis; (c) matemática financeira, com exemplos de jogos e desafios que envolvem o cálculo de juros simples e compostos; (d) funções cíclicas (ou trigonométricas), com exemplos sobre projeções de sombras e movimento de planetas no espaço. Os três primeiros já encontram-se completamente desenvolvidos, sendo que o último está em estágio avançado de implementação.

Cada um desses domínios compõe um OA. Durante a execução de cada um deles, o aprendiz é levado a resolver exercícios, inserindo como resposta diversas expressões analíticas. A cada exercício resolvido, a tendência é que haja um aumento tanto na complexidade quanto na quantidade de conceitos envolvidos, necessários para concluí-los. Assim, identificou-se a necessidade da existência de um sub-sistema capaz de realizar um acompanhamento do progresso do aprendiz. Tal acompanhamento se traduz em diagnosticar erros conceituais e intervir com a apresentação de mensagens no intuito de auxiliar a remediação desses erros.

O presente trabalho de pesquisa e desenvolvimento destinou-se prioritariamente (mas não exclusivamente) ao desenvolvimento de tal ferramenta. Espera-se que as hipóteses simplificadoras adotadas sejam suficientes para que a ferramenta possa ser reutilizada por todos os OA construídos durante o projeto. Cabe ainda observar que a ferramenta de diagnóstico ativo pode ser ajustada minimamente de maneira para ser utilizado não apenas em Matemática mas também no apoio ao ensino e a aprendizagem de conceitos em áreas como Física e Química.

1.4 Estrutura da dissertação

Esta dissertação está organizada em sete capítulos. Após o presente capítulo introdutório, no Capítulo 2 analisamos a literatura existente que aborda a Inteligência Artificial na educação, explorando conceitos como Sistemas Tutores Inteligentes, seus modelos cognitivos e diagnosticadores de erro, além de analisar alguns aspectos sobre Micromundos, Representações Externas (RE) e Múltiplas Representações Externas (MRE). No Capítulo 3 são apresentados os conceitos fundamentais que fazem parte da solução adotada para o

problema escolhido, além de alguns exemplos do funcionamento esperado da ferramenta. Depois, no Capítulo 4 discorre-se sobre o processo de generalização do aprendizado humano e suas eventuais falhas. Após isso, no Capítulo 5 apresentamos uma arquitetura funcionalista da ferramenta de software construída durante este trabalho, mostrando também alguns cenários de aplicação. Logo em seguida, no Capítulo 6 apresentamos nossas conclusões e prevemos alguns trabalhos futuros. Por fim, existem três anexos contendo documentos de descrição do funcionamento das OA produzidas no projeto CONDIGITAL, e logo após as referências bibliográficas utilizadas neste trabalho.

CAPÍTULO 2

RESENHA LITERÁRIA

Este capítulo procura resumir em duas grandes linhas, e de maneira crítica, o estado da arte sobre tecnologias aplicadas à educação. A primeira linha é a que inclui as abordagens instrucionistas com a aplicação da Inteligência Artificial na construção de sistemas computacionais ativos (onde o aprendiz fica em uma situação predominantemente passiva). Na segunda linha desta resenha, a situação contrária é destacada, na qual os sistemas passivos dão a vez aos aprendizes predominantemente ativos das abordagens construcionistas [57].

2.1 Histórico crítico da Inteligência Artificial na Educação

Nesta seção, os sistemas ativos são enfocados sob três ângulos: *(a)* sistemas tutores inteligentes como forma de aplicar Inteligência Artificial à Educação; *(b)* diagnosticadores de erro em conhecimento procedimental como o campo mais próximo aos do problema central deste trabalho de pesquisa que tenta lidar com erros no conhecimento sobre diferentes campos da Matemática, notadamente o analítico; *(c)* arquiteturas cognitivistas dos Sistemas Tutores Inteligentes, onde tipificamos as principais iniciativas do passado que foram aplicadas na construção dos software que, de alguma forma, lidaram com erros durante o ensino de conhecimentos procedimentais diversos (*e.g.*, Aritmética básica, Programação de Computadores e outros).

2.1.1 Sistemas Tutores Inteligentes em geral

O conceito de Sistema Tutor Inteligente (STI) surgiu da ideia de aplicar técnicas de Inteligência Artificial (IA) em programas com objetivos educacionais. Em geral, um STI é um programa capaz de guiar um aprendiz através de um currículo, geralmente composto por uma série de tarefas, ao mesmo tempo que monitora o progresso do aprendiz.

Para atingir isso, técnicas de IA são aplicadas para decidir sobre a forma de apresentação, sobre quais porções do conteúdo devem ser apresentadas, e também para modelar o conhecimento adquirido pelo aprendiz até dado momento.

No entanto, muito ainda resta aos STI em termos de modelagem do aprendiz, no que diz respeito à representação de conhecimento procedimental, presente em domínios que vão desde a aritmética básica até a programação de computadores, por exemplo. Abordagens anteriores, como os modelos de *overlay* [18] e o diferencial se concentraram em representar o conhecimento correto demonstrado pelo aprendiz, ignorando a possibilidade de o aprendiz deter conhecimentos diferentes dos ideais, e não somente um subconjunto de habilidades a serem ensinadas. Posteriormente, com o surgimento da necessidade de diagnóstico, foram concebidas modelagens procedimentais que, entre outras propriedades, incluíram a representação explícita dos erros (ou *bugs*, em inglês) de aprendizagem.

Apesar da extensa pesquisa realizada até hoje na área de STI, ainda podem ser identificadas diversas limitações nesses sistemas. Dentre essas limitações, as mais importantes podem ser de natureza arquitetural, representacional ou avaliativa. As limitações arquiteturais são originadas da divisão modular do sistema por meio de funções. Isso causa um impacto negativo na representação de relações entre conceitos de solução de problemas elicitados no modelo do especialista (também conhecido como modelo do domínio) e conceitos didáticos e pedagógicos elicitados no modelo pedagógico. Os efeitos negativos dessa modularização já foram relatados em pesquisas famosas, como ocorreu no caso da primeira tentativa de construção do módulo pedagógico *GUIDON* [19] sobre o sistema especialista *MYCIN* [45]. Para relaxar a forte modularização e assim integrar o módulo pedagógico *GUIDON-2* ao módulo de diagnóstico, o sistema especialista precisou ser totalmente remodelado, o que derivou na construção do diagnosticador reversível *NEOMYCIN* [20], que agora dava explicações sobre a origem de suas inferências.

Adicionalmente, as limitações de natureza representacional ocorrem pois os STI sofrem da limitação de escopo. Ela afeta tanto o conjunto de certezas sobre fatos dos seus diversos módulos internos como a profundidade de encadeamento lógico de suas inferências sobre esses fatos. O passado mostrou diversas pesquisas que tentaram enfrentar o problema

do escopo reduzido dos sistemas inteligentes por meio da modelagem de aspectos de pragmática além dos aspectos semânticos [30]. Porém, os progressos científicos atingidos na área de IA em geral parecem ter beneficiado os diálogos de mais curto prazo. Os STI requerem técnicas para a manutenção coerente e completa de diálogos de longo prazo e as técnicas aplicadas para lidar com ampliação do escopo ainda deverão evoluir antes de mostrarem sucesso.

Finalmente, as limitações de natureza avaliativa são decorrentes do fato de os STI ainda enfrentarem grandes problemas com a identificação do que os aprendizes fazem de correto ou de errado na solução de problemas. Como consequência, a mais importante modalidade de avaliação, a formativa [49], não pode ser conduzida com garantias mínimas por um STI até o presente momento. É ela, ao contrário da avaliação somativa, que pode oferecer ao tutor e ao aprendiz (ambos humanos) uma forma de ampliação substancial da escala de benefícios do tutoramento um-para-um. Todavia, poucos STI atingiram algum progresso nesse sentido avaliativo-formativo, tais como o *MR-Tutor* [31], o *Radiology Tutor* [44] e o *RUI* [23, 22, 24, 25].

2.1.2 Diagnosticadores de erro em conhecimento procedimental

Também conhecidos como diagnosticadores de erros (*Bug Finders*), os sistemas de monitoramento automático são ferramentas que têm como objetivo identificar erros cometidos pelo aprendiz durante o processo de solução de problemas e, eventualmente, intervir de maneira inteligente para tentar levar o aprendiz a corrigí-los. São capazes de detectar discrepâncias entre a solução esperada para um dado problema e a fornecida pelo aprendiz, indo além de informá-lo sobre a localização dos erros ao incentivar a descoberta dos motivos que o levaram a cometer tais erros [26, 27].

O fato de os desenvolvedores de STI terem o conhecimento necessário para atuarem também como peritos no domínio de programação de computadores contribuiu para que STI diagnosticadores fossem aplicados amplamente a esse campo. Esses sistemas obtiveram reconhecido sucesso, sendo exemplos de destaque o *LISP Tutor* [9], no ensino da linguagem LISP, e o *PROUST* [33], no ensino da linguagem Pascal. Também foi

atingido sucesso em aplicações de diagnosticadores em subáreas da matemática. Podemos destacar: o *Geometry Tutor* [8], tutor de geometria, o *Cascade* [53, 55, 56], tutor de aritmética básica, e ainda o *Algebra Land* [12], um tutor de conceitos algébricos.

Esses diagnosticadores podem ser classificados em três grandes grupos: (a) os que se baseiam em uma solução de referência (*Specimen Answer*), como o sistema *Laura* [1]; (b) os que realizam análise de especificação (*Specification Analysis*), como o *Mycroft* [28], *Pudsy* [35] e *PROUST* [34]; (c) os que conduzem um diálogo de depuração (*Debugging Dialogue*), como o tutor de Prolog de Shapiro [43]. É importante ressaltar que há muitos anos não se observa nenhum progresso de relevância computacional nesse importante campo educacional, cujos aspectos cognitivos impõem grandes desafios de pesquisa original.

Em uma primeira dimensão complementar às três citadas acima, pode-se dizer que os STI também conseguiram realizar comparações de padrão (*pattern matching*) gramatical de código de programas com os possíveis gabaritos. Esse foi o caso do sistema *PROUST* [46, 32], que diagnosticava erros para programas em linguagem Pascal. Seu conhecimento era representado internamente em termos de conjuntos de objetivos onde cada objetivo era composto por um ou mais planos que poderiam ser conduzidos para que esse objetivo fosse atingido. Finalmente, cada plano era associado a um padrão gramatical, o qual podia ser comparado (ou “casado”) com fragmentos de código em Pascal fornecidos pelo aprendiz. O não casamento com pelo menos um padrão de resposta resultava em uma mensagem de erro gerada pelo *PROUST*.

Em uma segunda dimensão complementar estão os STI capazes de sintetizar códigos com erro. Eles trabalham na expectativa de reproduzir o comportamento do aprendiz e, dessa forma, intervir de maneira explicativa específica. Um dos exemplos mais conhecidos foi o *GREATERP* [9, 6, 5], o qual se constitui no módulo tutor do sistema de ensino da linguagem de programação LISP. Um outro exemplo foi o sistema *GIL* [37]. Todavia, a grande limitação dessa abordagem de síntese artificial de código com erros para atingir o diagnóstico estava na frequente falta de sintonia da geração de explicações em situações de relevância prática em que tais erros do aprendiz eram cometidos.

Na terceira dimensão complementar, vale ressaltar também as abordagens fundamen-

tadas nas diferenças entre a saída esperada de um programa correto e a saída produzida pelo programa potencialmente errado do aprendiz. Dois exemplos de sistema dentro dessa abordagem foram o *BIP* [11] apud [14] e o *PDS6* [43] apud [40]. A grande limitação desses sistemas foi caracterizada pela superficialidade das explicações, mesmo que suas intervenções fossem relatadas como frequentes. Em resumo, a maneira de lidar mais com os efeitos dos erros (na listagem de saída) do que com suas causas criaram barreiras para o sistema lidar com o processo de solução de problema de programação. Vale lembrar aqui que é exatamente o processo de solução de problema de programação que precisa ser decifrado pelo STI para se atingir um aperfeiçoamento da intervenção sobre o erro do aprendiz.

Finalmente, cabe falar sobre a verificação da especificação de um programa. A grande vantagem dessa abordagem foi a possibilidade de avaliar o erro do aprendiz sob uma orientação mais global. Os principais exemplos de STI dessa categoria foram o *TALUS* [39] apud [47], o *AURAC* [29] apud [41] o sistema LAURA [1]. Seja porque a linguagem de programação era simplificada ou alguma outra razão peculiar, a verificação da especificação teve apenas alguns resultados positivos. Sua grande falha estava na falta de parâmetros específicos do erro na explicação gerada pelo STI para o aluno, dado o contexto do problema atacado.

2.1.3 Arquiteturas cognitivistas dos Sistemas Tutores Inteligentes

Para ilustrar parte desses desafios, o foco dos STI baseados na teoria cognitivista ACT* [6], e mais recentemente na ACT-R [7], é evitar ao máximo que o usuário se desvie do comportamento de um aprendiz ideal. Através do uso de estruturas hierárquicas de objetivos e do mecanismo de aprendizado chamado de compilação de conhecimento, os tutores automáticos do padrão ACT* induzem os aprendizes a sucessivas aproximações durante a solução de problemas. Para auxiliar na tarefa de detecção de erros, a base de conhecimento de tais tutores, além das regras ideais a serem transmitidas ao aprendiz, contém também um catálogo de erros conhecidos. Esses erros, da forma em que são tradicionalmente codificados na teoria, ainda não são qualificados como super- ou sub-generalizações

[48] do aprendiz. Pelo fato da detecção do uso correto de conhecimentos procedimentais (produções) ser uma das bases da compilação de conhecimento, é necessário que tal mecanismo de aprendizado ofereça intervenção (*feedback*) imediatamente após detectar um erro significativo. Todavia, tal estratégia de intervenção imediata, apesar de ser capaz de prover ao aprendiz diferentes níveis de explicação, é constituída apenas por mensagens de texto em linguagem natural.

Outro exemplo está na teoria REPAIR [13], que se concentrou na modelagem de erros do aprendiz que também não chegaram a ser classificados como super- ou sub-generalização. Essa teoria pressupõe a criação de analogias, também chamadas de impasses, as quais fazem referências a conhecimento incompleto ou ainda esquecido. Os impasses são originados a partir de perturbações sobre o estado correto de uma habilidade procedimental formalizada empiricamente por meio de estudos com especialistas no domínio específico. Na tentativa de contornar tais impasses, uma ação de recuperação (*repair*) deve então ser realizada, a qual poderia originar novos *bugs* observados em aprendizes. Em sua forma ideal, a repetição encadeada de impasses e recuperações terminaria por revelar o procedimento incorreto utilizado pelo aprendiz, podendo então ser mencionado pelo tutor (humano ou automático) na geração de explicações. Em tutores automáticos do passado, as explicações baseadas no padrão REPAIR também eram puramente textuais.

Já com a teoria STEP, VanLehn se concentrou nas sequências de ensino utilizadas por tutores humanos para modelar procedimentos incorretos. Posteriormente, sistemas como o Sierra [50], que se baseou em aspectos das teorias REPAIR e STEP, foram desenvolvidos com o intuito de fornecer explicações. Ainda, a linha seguida pelo projeto Cascade [51], na qual uma parte do tutor de física Andes [54] baseou-se mais tarde, foi bem sucedida ao incentivar e modelar o efeito da auto-explicação. Isso ocorreu no contexto do aprendizado através de exemplos e solução de problemas voltados para a reflexividade dos conceitos. Porém, mais uma vez, tais sistemas apenas atuaram no plano da geração de explicações textuais.

2.2 Histórico crítico sobre os Micromundos

Nesta seção, é feita uma breve abordagem sobre os micromundos (ou microcosmos) de aprendizagem em contra-ponto ao que foi pesquisado no campo de STI. Essa contradição é importante pois nenhum STI, por mais ativo que seja, até o momento foi considerado mais eficaz do que os micromundos, por mais passivos que esses micromundos sejam. São apresentadas as principais limitações dos micromundos para apoiar o aprendiz na compreensão de erros, quais sejam: (a) falta de apoio à explicação sobre erros; (b) aspectos de representação externa.

2.2.1 Falta de apoio à explicação sobre erros

Os micromundos (ou microcosmos) são ambientes semanticamente ricos que promovem uma imersão do aprendiz em situações cujas simbologias figurativa e operativa [3] se mostram intuitivas ao aprendiz. Devido a isso, tais sistemas possuem uma interface passiva, ou seja, fazem do aprendiz o agente ativo da aprendizagem [38]. Essa suposição assume que a própria dinâmica de interação que o aprendiz impõe sobre o sistema gera a evolução do aparecimento de novos conteúdos que são necessários à aquisição do conhecimento sobre a solução de problemas.

Em outras palavras, o ambiente computacional passivo de um micromundo não gera explicação direta para um aprendiz sobre a causa de um erro que ele cometeu na solução de um problema. Ao contrário, o que se fez em pesquisas da área foi tentar registrar em estados internos do sistema qualquer mudança de configuração provocada por uma entrada do aprendiz, errada ou não. A partir daí, o próprio aprendiz deveria ser capaz de identificar o que está errado para só então tentar refazer um solução.

Uma das maneiras mais originais que se tem notícia do mundo de pesquisa em relação ao registro e apresentação passiva do desempenho de um aprendiz na solução de problemas está no que as autoras chamaram de *modelos abertos de aprendizes* [17] (ou *open learner models* na língua inglesa). Uma forma concreta de se fazer isso foi por meio da criação de medidores de desempenho ou habilidade [15] (*skillometers* na língua inglesa). Em

resumo, esses micromundos tentam oferecer relógios com certas contagens de quantidade de tarefas feitas pelo aprendiz com alguma idéia das que terminaram e de maneira correta (de acordo com alguma medida).

Uma das vantagens de fazer com que o próprio aprendiz tire suas conclusões acerca das causas de ocorrência de seus erros é a de não ser necessário o projeto e a implementação de uma máquina de busca com raciocínio semelhante ao do diagnóstico inteligente [16]. Por outro lado, a desvantagem principal está em determinar, desde a época do projeto do micromundo, quais representações externas (do conhecimento humano) permitem que o ambiente computacional realmente ofereça ao aprendiz a citada riqueza semântica para promover a compreensão do comportamento da interface durante suas mudanças de estado. Uma parte disso está tratado na próxima subseção.

2.2.2 Aspectos de representação externa

Em um campo paralelo que se situa entre as representações em formato computacional e as teorias pré-computacionais da psicologia cognitiva, surgiu a idéia de Representação Externa (RE) [21]. Tal conceito compreende técnicas de representação, organização e apresentação de conhecimento. Diversos tipos de modelos, entre eles os proposicionais e os gráficos, pertencem a essa classificação. Acredita-se que a utilização de REs apropriadas diminui a carga cognitiva exigida na realização de determinadas tarefas, entre outros benefícios, os quais resultam na elevação do desempenho geral do aprendiz. Mais recentemente, pesquisas que fizeram uso da combinação de diferentes REs no processo de aprendizado deram origem ao conceito de Múltiplas Representações Externas (MRE) [2, 4, 3].

A taxonomia de Ainsworth [2, 3] pode ser encarada como uma organização das funções cognitivas que as MRE podem desempenhar quando apoiam a solução de problemas. Ainsworth propõe que qualquer representação seja descrita nos seguintes termos: (1) o mundo representado; (2) o mundo representante; (3) quais aspectos do mundo representado estão sendo representados; (4) quais aspectos do mundo representante são responsáveis pela modelagem; (5) a correspondência entre os dois mundos. Apesar do extenso

material que Ainsworth já produziu, sua taxonomia ainda pode ser muito mais explorada e aplicada de maneiras ainda não vistas. Isso se explica pelo fato de, em geral, os sistemas existentes não possuírem uma base forte para cumprir os diferentes papéis ou funções cognitivas sugeridas. Um exemplo típico de tais limitações está no fraco apoio que os software educacionais dão ao aprendiz na solução interativa de problemas.

Diante do que foi apresentado, constata-se que, até o momento, são raros os trabalhos de pesquisa que abordaram a vinculação entre Representações Internas e Externas. Tal lacuna se caracteriza fundamentalmente pelo fato de não haver tratamento automático nem para a detecção nem para a remediação da sub- nem da super-generalização de conceitos matemáticos por parte de um aprendiz humano. Em especial, no quesito de remediação automática, as MRE seriam as candidatas naturais a reforçar a geração de explicações por parte da máquina. Todavia, ainda há muito a ser explorado nesse sentido pois a literatura dessa subárea parece ser mais escassa ainda.

CAPÍTULO 3

FUNDAMENTOS DA SOLUÇÃO ADOTADA

Neste capítulo apresentamos o problema a ser tratado por este trabalho, assim como o conjunto de conceitos metodológicos que foram utilizados para se obter a solução adotada, descrita nos próximos capítulos. O problema consiste na construção de uma ferramenta de software capaz de realizar diagnóstico ativo durante a interação de um aprendiz com um dos Objetos de Aprendizagem desenvolvidos durante o projeto. Tal ferramenta de software, um diagnosticador e remediador de erros conceituais, é um dos módulos que pode compor um STI.

A finalidade do OA sobre os quais este trabalho se baseia é auxiliar o aprendizado de certos conceitos matemáticos por aprendizes na faixa etária equivalente ao final do Ensino Fundamental e início do Ensino Médio. Cada OA consiste, basicamente, em uma série de exercícios que tratam, na sua maioria, de conceitos analíticos que estão fortemente relacionados a outros conceitos dos planos algébrico e geométrico. Como parte das soluções atingidas e descritas neste trabalho, tais conceitos se aplicam a uma vasta gama de domínios específicos de áreas da matemática. Como será detalhado ao longo desta dissertação, os primeiros exemplos implementados nesses domínios são: (a) Progressões Geométricas em Fractais; (b) Funções de Primeiro Grau; (c) Matemática Financeira. Também se encontra em fase adiantada de implementação as aplicações dos mesmos princípios e ferramentas para o domínio de Funções Cíclicas (*i. e.*, funções trigonométricas).

3.1 Estudo Empírico

Para a obtenção de informações do ambiente prático de ensino de Matemática objetivando a melhor fundamentação deste trabalho, foi conduzido um estudo empírico com a colaboração de especialistas. Nas próximas subseções descrevemos brevemente os contatos e entrevistas realizadas. Logo depois, relatamos também as experiências que obtivemos ao

aplicar um dos primeiros OA desenvolvidos durante o projeto em uma turma de alunos do ensino fundamental.

3.1.1 Entrevistas com especialistas em Ensino de Matemática

No início do desenvolvimento do projeto CONDIGITAL, detectamos que trabalhos semelhantes anteriores nessa área haviam dado enfoque a Representações Externas essencialmente visuais, como gráficos cartesianos, desenhos e animações. Por esta razão, decidimos consultar a equipe de professores da UEL que atuaram como conteudistas no projeto. Essa equipe era composta por Luciana Gastaldi e Lourdes Almeida, especialistas em ensino e aprendizagem da disciplina de Matemática, além de seus orientandos de mestrado e doutorado, Bárbara Palharini, Karina Pessoa e Rodolfo Vertuan.

Durante as entrevistas, nos focamos em tentar extrair informações a respeito das diversas maneiras com que os conceitos a serem ensinados poderiam ser apresentados visualmente ao aprendiz. Destas entrevistas, foram produzidos documentos de descrição mais detalhada do que viria a se tornar cada um dos OA posteriormente produzidos. Tais documentos estão contidos nos Anexos A, B e C, que correspondem aos OA dos domínios de progressões geométricas que ocorrem em elementos da geometria fractal, funções de primeiro grau, e matemática financeira, respectivamente.

Ao final de cada entrevista, tornava-se cada vez mais claro que os conceitos a serem ensinados e os exercícios associados a eles possuíam uma complexidade relativamente grande. Portanto, a presença de um sub-sistema de acompanhamento de erros conceituais acoplado a cada um dos OA poderia ser muito benéfica para o aprendizado. Essa conclusão se tornou ainda mais evidente após uma sessão de aplicação de um dos OA, descrita a seguir.

3.1.2 Aplicação de um Objeto de Aprendizagem

Após o término da fase de implementação do primeiro OA desenvolvido durante o projeto CONDIGITAL, cujo domínio abordado foi o de progressões geométricas em elementos da geometria fractal, decidimos utilizá-lo na realização de um experimento prático. Assim,

levamos o software até o Colégio Marista Santa Maria, localizado em Curitiba, Paraná, para que fosse utilizado por alunos de duas turmas de oitava série do Ensino Fundamental, cada uma com cerca de 20 (vinte) adolescentes.

Cada uma das duas sessões do experimento consistiu em deixar que os alunos, reunidos em duplas, explorassem o OA e tentassem resolver os exercícios propostos, após a leitura do conteúdo introdutório presente na ferramenta. É válido notar que os aprendizes nunca haviam tido contato algum com assuntos como Geometria Fractal e Progressões Geométricas. O embasamento teórico que precedeu a atividade foi apenas uma breve, porém muito esclarecedora, explanação sobre Regularidades, dada pelo professor de Matemática das turmas. Tal base teórica, essencial para a compreensão tanto de Progressões Aritméticas quanto Geométricas, foi fundamental para que os aprendizes prosseguissem nos exercícios.

Não esperávamos que sequer algum dos alunos conseguisse finalizar toda a sequência de exercícios propostos, mesmo porque o tempo da atividade era muito restrito (cerca de 40 minutos). Sendo assim, ficamos surpresos quando uma das duplas conseguiu completar todos os exercícios, sendo portanto uma exceção.

Durante a interação dos aprendizes com o software, tentamos observar e anotar as dificuldades mais comumente encontradas por eles. Importante notar que não foi dado nenhum auxílio, suporte ou dica de nossa parte em questões referentes ao domínio de Matemática que eram eventualmente levantadas por eles. Apenas auxiliamos em questões puramente computacionais e com eventuais dificuldades de interação com o software em si.

O OA utilizado durante esse experimento apresentava apenas um simples sistema de diagnóstico passivo, capaz somente de apresentar dicas pré-estabelecidas após o aprendiz inserir uma resposta considerada errada pela terceira vez consecutiva, num determinado exercício, além de indicar visualmente cada erro cometido. Com base nesse fato, e valendo-se das observações feitas durante a aplicação do software, obtivemos mais uma confirmação de que um sistema de diagnóstico ativo, capaz de diagnosticar e auxiliar na remediação dos erros conceituais dos aprendizes, seria uma adição de grande valia para os OA.

3.2 Conceitos da edição gráfica de fractais

Nesta seção apresentamos os conceitos relacionados tanto a descrição gráfica quanto a uma descrição formal de elementos da geometria fractal. Nas subseções seguintes exploramos os requisitos para uma ferramenta de edição gráfica de fractais, e em seguida relacionamos os conceitos gramaticais e analíticos presentes nessas formas geométricas. Acreditamos que o estabelecimento prévio desses conceitos auxiliaria a elaborar diagnósticos mais precisos para o sistema a ser construído neste trabalho.

3.2.1 Requisitos funcionais para a edição gráfica de fractais

Durante as primeiras fases de desenvolvimento deste projeto de mestrado, desenvolvemos a base conceitual para a construção de uma ferramenta de representação gráfica de fractais. O objetivo de tal ferramenta seria tornar possível a edição manual de determinados parâmetros, os quais corresponderiam às propriedades que definem um fractal. Dessa forma, após a edição, deveria ser possível visualizar uma representação gráfica do fractal correspondente a tais parâmetros. É importante ressaltar que só foi considerada uma determinada categoria de fractais, chamados de Fractais de Sistemas de Funções Iterativas (*Iterated Function System Fractals* ou Fractais *IFS*).

Uma maneira bem difundida de formalizar a definição de um fractal é pela utilização dos Sistemas Lindenmayer. Os *L-systems*, como também são conhecidos, são na realidade um tipo de gramática formal de derivação. Cada sistema é definido por: (a) um conjunto de variáveis; (b) um conjunto de constantes; (c) um axioma, que consiste de uma sequência de constantes e/ou variáveis que definem o estado atual do sistema; (d) um conjunto de regras de produção, ou seja, regras de substituição, que definem que o predecessor deve ser substituído pelos sucessores (símbolos à direita da regra). A diferença básica entre uma gramática formal e um *L-system* é que, no processo de produção (derivação), todas as regras que podem ser aplicadas ao estado atual do *L-system* serão aplicadas ao mesmo tempo, definindo assim um passo do processo de produção. Nas gramáticas formais tradicionais, apenas uma derivação é feita a cada passo do processo.

Considerando então um *L-system* e uma dada iteração do processo de produção, temos um determinado estado do sistema. Tal estado pode então ser traduzido para uma representação geométrica de um fractal (conjunto de pontos e retas) por uma técnica chamada de “Gráficos de Tartaruga” (*Turtle Graphics*). Essa técnica consiste em interpretar uma cadeia de caracteres como uma sequência de comandos de movimentação para uma entidade chamada de Tartaruga. Ao interpretar a sequência de símbolos (variáveis e constantes) presente no axioma de um *L-system* como tal sequência de comandos, o trajeto percorrido pela movimentação da tartaruga define então o conjunto de segmentos de reta que representam geometricamente o fractal. A partir dessa descrição geométrica, pode-se facilmente representar graficamente o fractal.

Os parâmetros de definição e edição de propriedades dos fractais que devem estar presentes na ferramenta são os seguintes: (a) axioma inicial; (b) conjunto de regras de produção; (c) valor do incremento, em graus, a ser somado ao vetor de orientação da Tartaruga após um comando de mudança de direção; (d) número da iteração do fractal. Ao editar-se tanto o axioma inicial quanto o conjunto de regras, deve ser possível definir qualquer quantidade de variáveis, as quais devem ser reconhecidas e adicionadas automaticamente ao conjunto de variáveis do *L-system*. O nome de uma variável deve ser um único caractere alfabético, maiúsculo ou minúsculo. Já o conjunto de constantes é sempre o mesmo e consiste de apenas 4 símbolos: “+”, “-”, “[” e “]”.

Cada caractere presente no axioma, como já foi dito, deve ser interpretado como um comando de movimentação, o que por sua vez irá potencialmente refletir na representação geométrica do fractal. Para esse fim, as variáveis que forem letras minúsculas devem ser interpretadas como comandos de movimentação em linha reta. Nesse caso, a distância percorrida deve ser um valor constante a ser calculado pela ferramenta. Já as variáveis que forem letras maiúsculas não devem ocasionar movimentação alguma, servindo apenas para agregar poder de expressão ao *L-system* e seu processo de derivação.

As constantes, por sua vez, definem outras ações de movimentação, como por exemplo: girar a Tartaruga em torno de seu eixo, no sentido horário ou anti-horário (lembrando que o grau de rotação é um parâmetro da ferramenta, fornecido pelo usuário), que cor-

respondem às constantes “+” e “-”, respectivamente. Outra ação possível é armazenar o estado atual da tartaruga (posição e orientação), através da constante “[”. Posteriormente, o último estado armazenado pode ser carregado como estado corrente, através do comando correspondente à constante “]”. Alguns *L-systems* ainda implementam outros comandos, que permitem, por exemplo, movimentar a Tartaruga sem que isso reflita em um segmento de reta a ser representado geometricamente.

3.2.2 Ligação de conceitos gramaticais e analíticos dos fractais

Os *L-systems*, sendo, em suma, representações gramaticais formais de fractais, devem preservar as propriedades analíticas desses, para serem considerados como representações válidas. Dessa forma, ao analisarmos a sequência de iterações de um determinado fractal, o axioma correspondente a cada uma delas contém uma quantidade de símbolos que, muitas vezes, reflete ou tem alguma relação de proporção com determinadas características geométricas do fractal.

Sabemos que, salvo em alguns casos, medidas como o comprimento do lado, o perímetro e a área dos fractais têm comportamentos de progressões geométricas, quando analisados em termos da sequência de suas iterações. O mesmo comportamento é observado quando analisamos a quantidade de cada um dos símbolos, ou de padrões de símbolos, presentes nos axiomas dos *L-systems* correspondentes. Tais quantidades também formam progressões geométricas, ou somas de progressões geométricas. Esse fato é um forte indicativo de que a natureza intrínseca do processo de derivação dos *L-systems* modela adequadamente o processo recursivo de construção de cada iteração de um fractal.

3.3 Requisitos funcionais do diagnóstico

A ferramenta de software a ser construída é classificada como sendo um diagnosticador e remediador de erros, componente indispensável na arquitetura de um STI. Em suma, suas atribuições gerais são automatizar o diagnóstico imediato de quaisquer erros conceituais cometidos pelo aprendiz durante a interação com o sistema. Em seguida ela deve fornecer

auxílio relevante ao usuário, na forma de mensagens pré-formuladas, sucessivamente mais explicativas, na medida em que haja persistência num mesmo erro. A distinção entre o presente sistema e sistemas similares se dá pela presença de um módulo classificador de erros de generalização.

A partir dos conceitos metodológicos adotados para a formulação da solução adotada para o problema, são apresentados a seguir, de forma breve, alguns cenários do funcionamento esperado da ferramenta ao final de seu desenvolvimento. Em ambos os exemplos, o Objeto de Aprendizagem considerado é o que trata do domínio de Progressões Geométricas em elementos da Geometria Fractal.

Exemplo 1 :

O aprendiz está resolvendo o exercício 2 do OA, cujo enunciado diz que deve-se preencher a tabela com os comprimentos dos lados do fractal Triângulo de Sierpinsky utilizando como referência a variável l , que representa o comprimento do lado do referido fractal no passo 0 de sua construção. Ao preencher a célula referente ao passo n (última linha da tabela), o aprendiz entra com uma expressão como a seguinte:

$$l/2$$

Ao que o sistema apresenta, numa primeira intervenção de remediação a seguinte mensagem: *Você notou que a expressão se refere à n-ésima iteração do Fractal? Sabendo disso, é esperado que a variável n apareça nessa expressão. Tente novamente.* A seguir, numa tentativa de remediar seu erro, o aprendiz fornece a seguinte expressão:

$$l/n$$

Nesse caso, o aprendiz segue a recomendação do sistema ao incluir a variável n na fórmula, mas ainda assim o sistema intervém apresentando sua segunda tentativa

de remediação, com a mensagem: *Na sua tentativa de incluir a variável n na expressão, você pode ter deixado de notar que existe uma regularidade (razão) entre os elementos anteriores dessa coluna. Tente incluir essa razão na sua próxima expressão. Após essa intervenção, espera-se que o aprendiz seja capaz de entrar com a expressão correta, que seria semelhante à fórmula a seguir:*

$$l/2^n$$

Exemplo 2 :

Desta vez o aprendiz está resolvendo o exercício 5 do OA, cujo enunciado pede ao aprendiz que preencha a tabela com expressões que representam o perímetro do fractal Curva de Koch nos diversos passos de sua construção. Ao preencher a célula referente ao passo 1 (segunda linha da tabela), o aprendiz constrói a seguinte expressão:

$$12/l$$

Como ação de remediação imediata, o sistema intervém com a mensagem: *A presença de l no divisor pode resultar em uma diminuição excessiva no valor do perímetro. Tente novamente.* Consequentemente, o aprendiz faz uma nova tentativa, fornecendo a expressão a seguir:

$$12 + l$$

Ao diagnosticar os erros, a segunda intervenção por parte do sistema consiste na mensagem: *Você foi bem sucedido ao retirar a variável l do divisor. Porém, note que a presença de uma soma ou subtração envolvendo l descaracteriza as propriedades de uma progressão geométrica. Tente novamente.* Por fim, espera-se que após essa intervenção, o aprendiz seja capaz de construir uma expressão correta para a célula da tabela em questão, que deve ser semelhante à seguinte expressão:

12 * l

Os exemplos descritos acima, além de apresentarem as funcionalidades básicas da ferramenta, como a classificação dos erros de generalização, também demonstram uma outra capacidade que se espera estar presente na versão final do sistema. Tal capacidade consiste em manter um histórico das intervenções de remediação, a fim de possibilitar uma análise relacional entre as intervenções anteriores e a atual. Dessa forma, é possível incluir às mensagens uma referência positiva caso haja adesão por parte do aprendiz às sugestões do sistema, ou ainda, um maior aprofundamento na explicação, caso haja reincidência num mesmo tipo de erro.

CAPÍTULO 4

IMPERFEIÇÕES NO PROCESSO DE GENERALIZAÇÃO

Neste capítulo continuamos a tratar de conceitos essenciais para a solução adotada, mas agora dando mais atenção a questão do processo de generalização que ocorre durante o aprendizado humano. Nas seções a seguir, tratamos de assuntos relacionados aos erros no processo indutivo humano, e também comentamos acerca da razão de adotarmos a estratégia de realizar diagnóstico automático. Por fim falamos brevemente sobre aspectos pedagógicos relacionados aos Objetos de Aprendizagem.

4.1 Generalização de aspectos cognitivos

Tendo em mãos os dados sobre os principais erros cometidos por aprendizes durante a interação com o Objeto de Aprendizagem, foi possível analisá-los de forma a organizá-los numa taxonomia, separando-os de acordo com o tipo de erro conceitual com o qual está mais relacionado. Nesta seção discorreremos acerca dessa taxonomia, apresentando as definições de cada tipo de erro conceitual, características específicas de erros de aprendizagem, além de fundamentarmos a escolha por executar diagnóstico automático e imediato.

4.1.1 Erros no processo indutivo humano

Assumimos da disciplina de Educação que ao longo do tempo, o aprendizado indutivo humano pode gerar três tipos de erros conceituais, denominados como a seguir:

Sub-generalização : Ocorre quando o aprendiz não consegue classificar determinado elemento como pertencente a determinada classe conceitual à qual ele realmente se enquadra.

Super-generalização : Ocorre quando o aprendiz classifica indevidamente determinado elemento como pertencente a determinada classe conceitual, quando na verdade ele

não pertence.

Miscelânea : Ocorre quando há uma mistura dos dois erros anteriores, caracterizando uma falta de compreensão mais profunda.

Para classificar os erros cometidos pelo aprendiz em uma dentre as categorias apresentadas, faz-se necessária a decomposição em características mais pontuais das expressões fornecidas. Tais características, quando detectadas individualmente ou em conjunto de outras, devem então caracterizar o erro de forma mais precisa.

4.1.2 Caracterização de erros

A fim de caracterizar os erros do aprendiz, foi concebido um certo nível de granularidade de características que são usadas, individualmente ou em grupo, para identificar a classe à qual um erro pertence. A seguir apresentamos a lista de características que foram identificadas como tendo maior relevância no contexto do OA do domínio de Progressões Geométricas em Fractais.

- Falta da variável l ;
- Falta da variável n ;
- Expoente da variável l é diferente do esperado;
- Coeficiente da variável l é diferente do esperado;
- Base da variável n é diferente da esperada;
- A variável l é o operando de uma soma ou subtração quando não deveria ser;
- A variável l encontra-se no expoente;
- A variável n encontra-se na base, ou seja, falta uma exponenciação;
- A variável l está sob uma raiz quadrada quando não deveria estar;
- A variável n está sob uma raiz quadrada quando não deveria estar;

- A variável n está no dividendo, quando não deveria estar;
- A variável l está no divisor, quando não deveria estar.

Ainda, foram identificadas também, no mesmo contexto, outras características. Estas não representam necessariamente nenhum erro se forem consideradas isoladamente, porém, ao serem conjugadas com uma ou mais características da lista anterior, podem ajudar a aumentar a gama de erros conceituais identificados pelo sistema. Tais características são listadas a seguir.

- A variável l está presente;
- A variável l encontra-se na base de uma exponenciação;
- A variável l encontra-se no dividendo;
- Expoente da variável l está correto;
- Coeficiente da variável l está correto;
- A variável n está presente;
- A variável n encontra-se no expoente;
- A variável n encontra-se no divisor;
- Base da variável n está correta;

As características apresentadas aqui devem ser usadas para compor regras com a seguinte estrutura:

```
Se <condições> Então
    <ação de remediação>
```

Nesse formato de regra, as **condições** são um subconjunto das características mencionadas, as quais, daqui pra frente, serão chamadas de **fatos**. Já a **ação de remediação** é essencialmente a apresentação de uma mensagem de auxílio ao aprendiz. O funcionamento dessas regras será visto com um pouco mais de detalhe mais a frente, dentro do contexto de um interpretador de regras.

4.1.3 Razão para diagnóstico automático

Na instância do Objeto de Aprendizagem que trata do tema Progressões Geométricas em Fractais, e mais especificamente em qualquer dos exercícios propostos ao aprendiz, é necessário que seja preenchida uma tabela com uma série de expressões analíticas que representam propriedades diversas acerca do fractal chamado Triângulo de Sierpinsky, por exemplo. Cada linha da tabela representa uma determinada iteração da construção do fractal, em ordem crescente, formando assim uma Progressão Geométrica. Na última linha, pede-se que o aprendiz preencha-a com expressões que generalizam o raciocínio aplicado na construção das expressões anteriores. Em outras palavras, trata-se do termo geral da progressão.

Dessa forma, podemos considerar que cada uma das expressões analíticas de uma mesma coluna da tabela (mesma propriedade ou medida do fractal) são fragmentos da solução de um problema maior, o qual pode ser descrito por um encadeamento do raciocínio analítico com características geométricas e algébricas. Tal raciocínio pode ser afetado por problemas na generalização dos conceitos envolvidos, o que nos leva a crer na necessidade de intervenção imediata ao se detectar tais problemas.

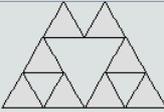
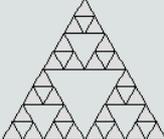
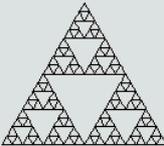
Apresentamos na figura 4.1 um exemplo de erro que pode ocorrer durante a resolução de um exercício. O aprendiz deve construir uma expressão que represente o comprimento do lado da n -ésima iteração do fractal Triângulo de Sierpinsky. Do enunciado, sabe-se que a variável l representa o comprimento da primeira iteração, e n representa o número da iteração.

Uma das expressões que pode ser considerada como correta é a seguinte:

$$l/2^n$$

E um possível erro de generalização poderia levar o aprendiz a entrar com uma expressão como a seguinte:

$$l/2$$

Progressões Geométricas em Fractais		
☉ (clique aqui para aparecer o texto)		
Iteração	Fractal	Lado
2		
3		$\frac{l}{8}$
4		$\frac{l}{16}$
n	figura limite	

← → Exercício 2 A+ A- A Calculadora Menu

Figura 4.1: Exercício proposto pelo OA de Progressões Geométricas em Fractais

Claramente, o que ocorreu nesse caso foi a falta da generalização do expoente no divisor, provavelmente pelo fato de o aprendiz não ter verificado que a sequência de expressões em questão forma uma P.G. de razão $1/2$. Acreditamos que erros de natureza semelhante a essa levam o aprendiz a uma situação difícil de transpor sem auxílio. Por essa razão, faz-se necessária a intervenção planejada para remediar tais erros. Ao classificá-los numa taxonomia que permita identificar suas causas com mais facilidade, é possível então fornecer um *feedback* especializado que auxiliaria o aprendiz a superar suas dificuldades.

4.1.4 Automação do diagnóstico de erros com remediação

Com base no conjunto de fatos (características usadas na identificação de erros), podem ser construídas regras de produção a serem utilizadas por um sistema especialista de encadeamento progressivo para diagnosticar e remediar automaticamente os erros cometidos pelo aprendiz. Esse processo ocorre da forma descrita a seguir.

Inicialmente, a solução do aprendiz é submetida a testes de caráter aritmético, que

analisam valorações possíveis das variáveis. Em caso de falha em qualquer um deles, passa-se para uma fase de análise das características presentes na expressão.

Nessa segunda fase, é associada previamente a cada célula de resposta do exercício um conjunto de fatos possíveis de estarem presentes na solução do aprendiz. A seguir, para cada fato realmente identificado, adiciona-se uma identificação do mesmo a uma base de fatos, inicialmente vazia. Depois, percorre-se a base de regras, verificando-se os fatos presentes na base de fatos contra as antecedentes de cada regra. Caso uma regra que identifica o erro como sub- ou super-generalização seja totalmente satisfeita, sua ação de remediação é ativada e uma mensagem de auxílio é apresentada ao aprendiz.

Caso contrário, o erro é considerado de miscelânea. Então, passa-se a uma terceira fase. Similar à segunda, essa fase também analisa as características da expressão, porém com outro conjunto de fatos e regras, agora associados a valores de incerteza, e com mensagens não tão especializadas.

Ainda na segunda fase, caso o usuário persista em um de seus erros, há um aprofundamento na remediação que tentará conduzi-lo cada vez mais a uma solução de referência.

4.2 Generalização de aspectos pedagógicos

A abordagem pedagógica por trás de cada uma das instâncias do Objeto de Aprendizagem é constituída de uma breve cobertura do assunto específico por meio de páginas introdutórias. Nessa introdução são apresentados conceitos fundamentais, pertencentes ao tema do OA, que o aprendiz deve compreender para que seja capaz de resolver com êxito os exercícios propostos. Esses exercícios são apresentados logo na sequência das páginas introdutórias, sendo organizados em ordem crescente de dificuldade típica de alunos da faixa etária do Ensino Médio. Discutiremos ao decorrer desta seção a finalidade de cada um desses aspectos pedagógicos dos Objetos de Aprendizagem.

4.2.1 Finalidade contextualizadora de elementos introdutórios

A introdução, composta por uma série de páginas com conteúdo introdutório, possui um caráter essencialmente contextualizador. Na medida em que apresenta conceitos fundamentais para a compreensão de determinada sub-área da Matemática por parte do aprendiz, também cumpre o papel de prepará-lo para enfrentar os exercícios. Um aspecto relevante da introdução é o fato de estarem resumidos diversos aspectos anteriormente ensinados pelo professor em sala de aula.

A seguir mostramos um exemplo de um trecho típico da introdução do OA de Progressões Geométricas em Fractais:

“Uma progressão geométrica é uma sequência de números onde cada termo, a partir do segundo, é igual ao anterior multiplicado por uma constante (seja q), chamada razão. A letra q foi escolhida por ser a inicial da palavra quociente. Se dividirmos a quantidade de lados de uma iteração pela quantidade de lados da iteração anterior, sempre obteremos a razão $q = 4$ na Curva de Koch. Para este caso, como o número de lados aumenta em cada iteração, podemos dizer que a quantidade de lados deste fractal, de uma iteração para outra, forma uma P.G. crescente de razão 4.”

Pode-se perceber claramente o caráter contextualizador do texto, mais especificamente quando é citada a relação entre a razão (conceito de Progressões Geométricas) e os lados das iterações (conceitos da Geometria Fractal). Também é fácil notar que o conhecimento é apresentado de forma resumida, o que remete a uma apresentação anterior à atividade laboratorial.

4.2.2 Finalidade do encadeamento dos exercícios

Foi definido que após a apresentação do conteúdo introdutório, cada OA conteria uma sequência de exercícios a serem resolvidos pelo aprendiz. De maneira geral, pode-se dizer esses exercícios são bastante complexos. Devido a isso, o esperado é que haja uma alta

frequência de erros cometidos pelos aprendizes. Isso se justifica pois o propósito de tais exercícios é de serem aplicados ao nível médio de ensino do currículo escolar formal.

Abaixo apresentamos dois exemplos de enunciados presentes no OA de Progressões Geométricas em Fractais:

“Agora você realizará um preenchimento semelhante ao exercício anterior, porém sem escolher o tamanho do menor lado da iteração 0, que foi pré-fixado com o valor l . Utilize apenas números inteiros e o sinal de divisão do teclado virtual se for necessário criar frações.”

“Se você entendeu bem a progressão geométrica com que o tamanho do lado muda de acordo com a iteração, será fácil trabalhar agora com o perímetro do mesmo Fractal. Para isso, preencha o perímetro em cada uma das iterações por meio de uma expressão que é função de alguns valores. Neste exercício, você também não poderá utilizar números com a vírgula decimal.”

Uma característica que fica evidente quando analisamos os enunciados é a de que eles possuem diversas variáveis, o que contribui para o aumento da complexidade do exercício. Outras características, verificadas mais facilmente durante a resolução dos exercícios são a de que eles possuem soluções exatas, e que tais soluções podem ser expressas sob várias formas. Pode-se constatar também que eles possuem uma ligação direta com os conceitos da introdução, o que pode ajuda o aprendiz a relacionar os conceitos de forma a aplicá-los da maneira esperada.

Ao analisar especificamente o desenvolvimento dos exercícios do OA de Progressões Geométricas em Fractais, pode-se verificar que o aprendiz deve preencher tabelas com expressões analíticas que representam certas propriedades e medidas geométricas de várias iterações (passos) de um Fractal. Ao final de cada tabela, há uma linha que corresponde a uma iteração genérica, ou seja, o aprendiz deve entrar com uma expressão analítica que representa uma certa medida ou propriedade para qualquer iteração. Normalmente isso é feito com a introdução de uma variável n na fórmula, a qual representa a quantidade de

iterações da figura do fractal. Em outras palavras, o aprendiz deve ser capaz de generalizar certos conceitos a fim de produzir a expressão esperada ao final de cada exercício.

Durante a interação com o Objeto de Aprendizagem, o aprendiz pode acabar cometendo erros derivados da incorreta generalização de conceitos matemáticos, e por vezes pode ter certa dificuldade de identificar a causa de determinado erro. Assim, faz-se necessária a presença de um componente de software que seja capaz de diagnosticar, classificar e remediar tais erros, a fim de auxiliar o usuário do software no seu processo de aprendizado.

CAPÍTULO 5

ARQUITETURA FUNCIONALISTA DA FERRAMENTA

Com o objetivo de incorporar os conceitos apresentados nos capítulos 3 e 4, a ferramenta de software denominada **Sistema de Aprofundamento por Generalização na Aprendizagem (SAGA)** foi projetada e construída como parte deste trabalho de pesquisa e desenvolvimento. Neste capítulo apresentamos a arquitetura funcionalista proposta para a referida ferramenta, além de indicarmos como ela se relaciona com os Objetos de Aprendizagem do projeto CONDIGITAL.

A ferramenta SAGA pode ser descrita como sendo um diagnosticador e remediador de erros, um módulo que pode compor a arquitetura de um STI. Mais especificamente, suas atribuições gerais são automatizar o diagnóstico imediato de quaisquer erros conceituais cometidos pelo aprendiz durante a interação com o sistema. Em seguida ela deve fornecer auxílio relevante ao usuário, na forma de mensagens pré-formuladas, sucessivamente mais explicativas, na medida em que haja persistência num mesmo erro.

5.1 Arquitetura da ferramenta SAGA

De acordo com o diagrama apresentado na Figura 5.1, três módulos principais compõem a arquitetura funcionalista do SAGA. São eles: o Analisador de Expressões, o Analisador Aritmético e o Interpretador de Regras de Produção. Ainda, ligadas a este último módulo, estão presentes a Base de Regras e a Base de Fatos, que exercem papéis essenciais na arquitetura. Tais módulos são detalhados na próxima seção.

O diagrama também deixa claro que há uma interação direta do SAGA com a Interface do Aprendiz, parte integrante do Objeto de Aprendizagem. Há também a Interface de Autoria, que se comunica com um módulo chamado Controlador de Enunciados. Tais componentes também são detalhados mais a frente.

Na implementação da ferramenta SAGA foi utilizada a linguagem de programação

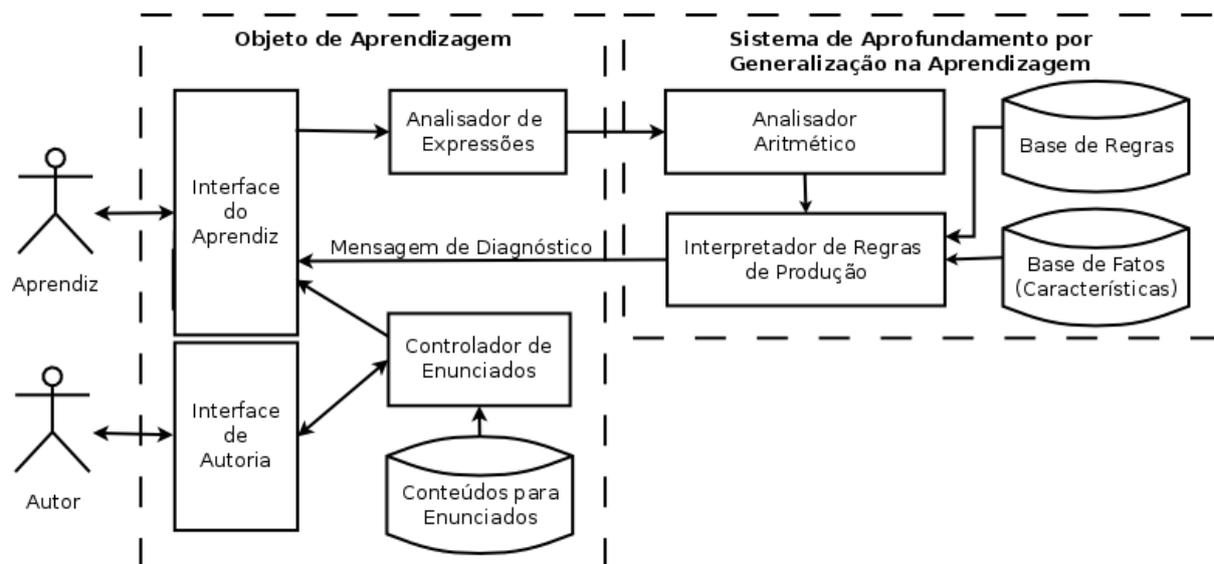


Figura 5.1: Arquitetura funcionalista da ferramenta SAGA

Java, seguindo os preceitos do paradigma de Programação Orientada a Objetos. A escolha de tal linguagem deu-se prioritariamente pela facilidade de integração com os Objetos de Aprendizagem, que à época estavam sendo desenvolvidos nessa mesma linguagem. Visto que os OA foram projetados para execução em ambiente *Web*, o sistema desenvolvido conseqüentemente também possui essa característica. Sua execução, portanto, não está restrita a configurações específicas de hardware ou sistema operacional, necessitando apenas de um navegador padrão de Internet.

5.2 Especificação dos módulos

Nesta seção detalhamos o funcionamento dos principais módulos da ferramenta SAGA. O relacionamento e a comunicação entre os módulos, e entre estes e seus depósitos de dados, também são descritos de forma a tentar deixar claro o funcionamento do sistema de uma forma geral.

5.2.1 Interface do Aprendiz

A Interface do Aprendiz é um módulo integrante do Objeto de Aprendizagem. Ela é a responsável pela apresentação dos conteúdos introdutórios e também dos exercícios propostos. Cabe também a esse módulo receber e armazenar, mesmo que temporariamente,

as soluções e respostas aos exercícios fornecidas pelo aprendiz, além de tratar todo e qualquer evento de interação com o OA.

O controle de interação e apresentação do conteúdo é feito pela ferramenta de software desenvolvida durante o trabalho de mestrado de Diego Marczal [36], denominada CARRIE. Além de outras funcionalidades, tal controlador provê um Teclado Virtual, necessário para a entrada sistematizada de expressões analíticas, fornecidas pelo aprendiz durante a solução de exercícios.

Na Figura 5.2 pode-se visualizar um exemplo de Interface do Aprendiz. A interface em questão pertence ao OA do domínio de Progressões Geométricas em Fractais. É possível perceber que ela é composta por: título do domínio; área de apresentação de conteúdos, enunciados e resolução de exercícios; painel de controle de paginação e acesso a funções especiais.

Progressões Geométricas em Fractais

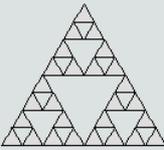
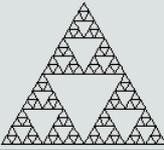
○ ↑ (clique aqui para esconder o texto)

Exercício 2

Agora você realizará um preenchimento semelhante ao exercício anterior, porém sem escolher o tamanho do menor lado da iteração 0, que foi pré-fixado com o valor l . Utilize apenas números inteiros e o sinal de divisão do teclado virtual se for necessário criar frações.

Dica:

- Note que agora o teclado virtual também ativou a tecla especial l (tamanho do lado na iteração 0).

Iteração	Fractal	Lado
3		$\frac{l}{8}$
4		$\frac{l}{16}$
n	figura limite	

← → Exercício 2 A+ A- A Calculadora Menu

Figura 5.2: Interface do Objeto de Aprendizagem

5.2.2 Interface do Autor

A Interface do Autor é um importante componente do Objeto de Aprendizagem. Apesar de não ter sido projetada para execução integrada ao ambiente de aprendizado, sua relevância para o sistema como um todo permanece alta.

Através dessa Interface, o autor (*i.e.* especialista em um domínio) é capaz de projetar enunciados para um determinado domínio da disciplina de Matemática. Tais enunciados serão posteriormente utilizados pela Interface do Aprendiz, durante a interação do aprendiz com o OA. É importante notar que a Interface do Autor é apenas uma camada de apresentação (ou *frontend*) para o Controlador de Enunciados, descrito mais a frente.

A título de exemplo, pode-se visualizar na Figura 5.3 uma Interface do Autor para o domínio de Progressões Geométricas em Fractais. Através dela, o autor é capaz de elaborar um Fractal para ser utilizado posteriormente por um enunciado no OA de mesmo domínio.

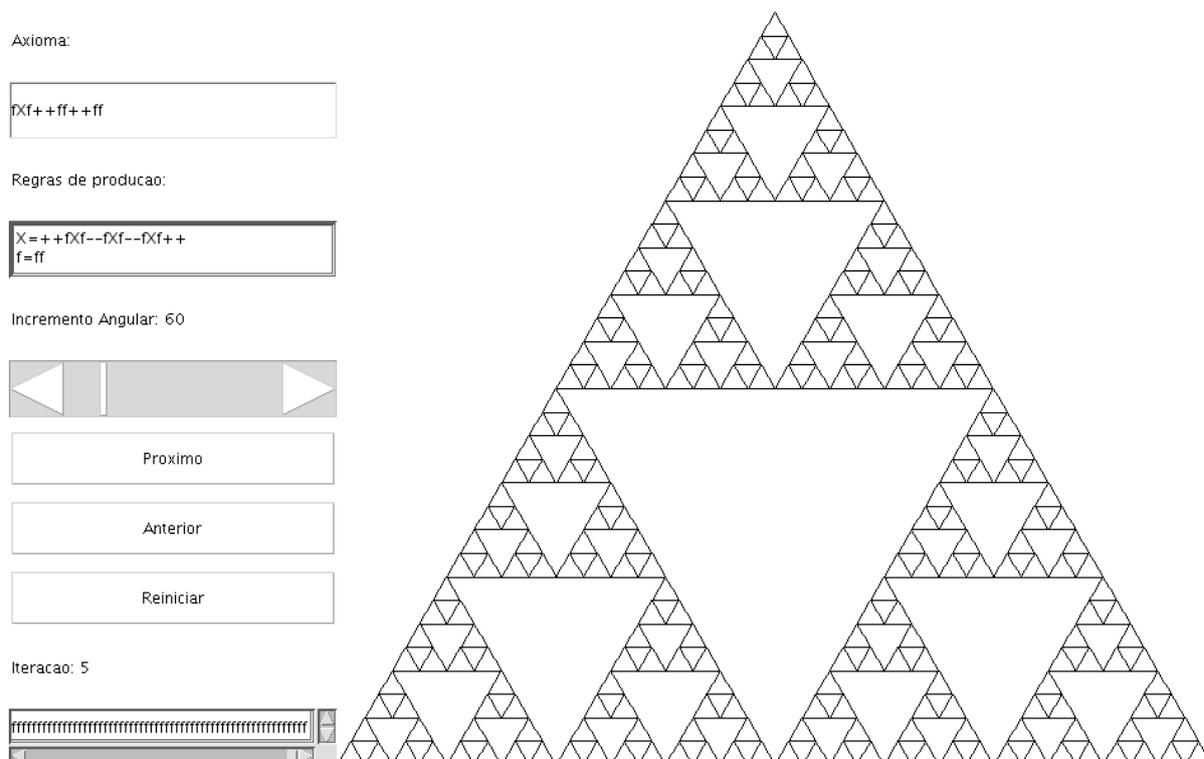


Figura 5.3: Interface de Autoria para o domínio de Fractais

5.2.3 Analisador de Expressões

O Analisador de Expressões é um módulo situado na fronteira entre o Objeto de Aprendizagem e a ferramenta SAGA. Seu funcionamento pode ser resumido pelos seguintes passos: (a) receber da Interface do Aprendiz uma expressão analítica; (b) analisar sintaticamente a expressão analítica, identificando variáveis, constantes, operadores e níveis de precedência; (c) caso a expressão esteja sintaticamente bem estruturada, elaborar uma representação interna (estruturada como uma árvore) e repassá-la ao Analisador Aritmético.

5.2.4 Analisador Aritmético

O Analisador Aritmético é o módulo responsável por realizar as primeiras verificações sobre as expressões analíticas fornecidas pelo aprendiz como tentativas de solução a um dos exercícios propostos. Após receber a descrição da expressão, elaborada pelo Analisador de Expressões, o comportamento do Analisador Aritmético consiste das seguintes etapas: (a) identificar as variáveis presentes na expressão; (b) substituir cada uma das variáveis por valores pré-estabelecidos; (c) calcular o valor resultante da expressão a cada substituição de variáveis; (d) verificar se todos os resultados obtidos conferem com as soluções de referência para determinado exercício.

A partir do resultado da última etapa de verificação, o módulo é capaz de decidir, com alto grau de confiabilidade, se a expressão fornecida está correta. Ou seja, não é necessário nenhuma manipulação algébrica de normalização, o que poupa tempo computacional. Assim, otimiza-se o tempo de resposta do diagnóstico final a ser entregue pelo sistema.

5.2.5 Interpretador de Regras de Produção

O Interpretador de Regras de Produção é o módulo principal da ferramenta SAGA. Ele é responsável por encontrar uma classificação adequada para o erro conceitual do aprendiz e, de acordo com tal classificação, é executada uma ação de remediação correspondente. Essa ação é caracterizada pela elaboração de uma mensagem de auxílio, a qual é enviada

à Interface do Aprendiz para exibição.

Sendo, em suma, um sistema especialista de encadeamento progressivo, o processo de diagnóstico ocorre durante a execução dos seguintes passos: (a) receber do Analisador Aritmético a representação interna da expressão analítica com erro; (b) identificar as características presentes na expressão, ao compará-las com cada uma das características previstas na Base de Fatos; (c) com base nas características identificadas, procura-se casar um subconjunto delas à antecedente de alguma das regras armazenadas na Base de Regras; (d) aplicar a ação de remediação descrita na regra escolhida, enviando a mensagem de diagnóstico à Interface do Aprendiz.

A Base de Regras está organizada de maneira que cada uma das regras corresponde a uma das classificações da taxonomia de erros conceituais. Portanto, a informação sobre qual tipo de erro conceitual, dentre sub-generalização, super-generalização ou miscelânea, pode começar a ser inferida pela repetição da aplicação desse processo.

5.2.6 Controlador de Enunciados

O Controlador de Enunciados é parte integrante do Objeto de Aprendizagem. Suas duas funções principais são as seguintes: prover os enunciados de exercícios para a Interface do Aprendiz; e possibilitar ao autor a elaboração de tais enunciados.

Ao interagir com o Controlador de Enunciados, através da Interface do Autor, ele é capaz de projetar e elaborar elementos que compõem os enunciados de exercícios do OA. Tais elementos podem ser de diversas naturezas, como componentes gráficos (*i.e.* imagens, animações, diagramas), regras lógicas para a execução de um jogo, ou apenas simples elementos textuais. Todos esses elementos são armazenados numa base, denominada Conteúdos para Enunciados, para posterior recuperação durante a execução do OA pelo aprendiz.

5.3 Cenários de aplicação

Nesta seção descrevemos a aplicação da arquitetura da ferramenta SAGA em dois domínios dos OA do projeto CONDIGITAL: Progressões Geométricas em Fractais e Funções de Primeiro Grau.

5.3.1 Progressões Geométricas em Fractais

No início do desenvolvimento deste trabalho, foi desenvolvida uma ferramenta de edição gráfica de fractais que corresponde ao Controlador de Enunciados, na arquitetura apresentada anteriormente. O desenvolvimento de tal ferramenta foi alcançado ao adotarmos como domínio as Progressões Geométricas em elementos da Geometria Fractal, ao mesmo tempo que utilizamos os conceitos apresentados na Seção 3.2.

Com essa ferramenta, o autor é capaz de descrever formalmente um fractal, fornecendo os parâmetros de um *L-system*. Esses sistemas foram descritos com mais detalhes na seção 3.2.1. Após completar a descrição de um determinado fractal e verificar sua correspondência visual na ferramenta, é possível incluí-lo para que seja usado nos enunciados dos exercícios do OA de Progressões Geométricas em Fractais.

A princípio, estavam previstos três fractais a serem utilizados no OA. O primeiro, denominado Triângulo de Sierpinsky, possui a descrição formal apresentada a seguir. Sua visualização na ferramenta pode ser observada na Figura 5.3.

Axioma inicial: `fXf++ff++ff`

Regras de produção:

`X=++fXf--fXf--fXf++`

`f=ff`

Incremento angular: 60 graus

O segundo Fractal, denominado Floco de neve de Koch está representado na figura 5.4. Como descrição formal, ele possui os seguintes parâmetros:

Axioma inicial: `f--f--f`

Regras de produção:

$$f=f+f--f+f$$

Incremento angular: 60 graus

Axioma:

f--f--f

Regras de producao:

f=f+f--f+f

Incremento Angular: 60



Proximo

Anterior

Reiniciar

Iteracao: 4

+f--f+f+f+f--f+f--f+f--f+f+f--f+f

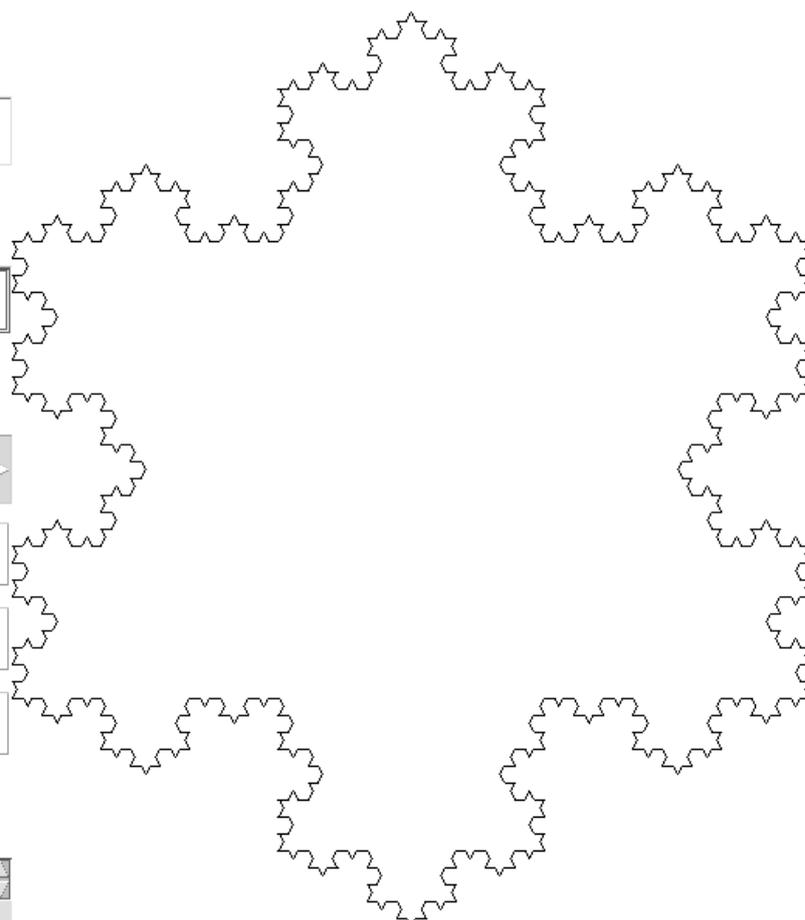


Figura 5.4: Fractal Floco de neve de Koch

O terceiro Fractal previsto, conhecido como Fractal dos Hexágonos, acabou por não permanecer na versão final do OA. Sua representação gráfica pode ser observada na Figura 5.5, enquanto que sua descrição formal é a seguinte:

Axioma inicial: fX-fX-fX-fX-fX

Regras de produção:

$$X=++fX-fX-fX-fX-f+++$$

$$f=fff$$

Incremento angular: 60 graus

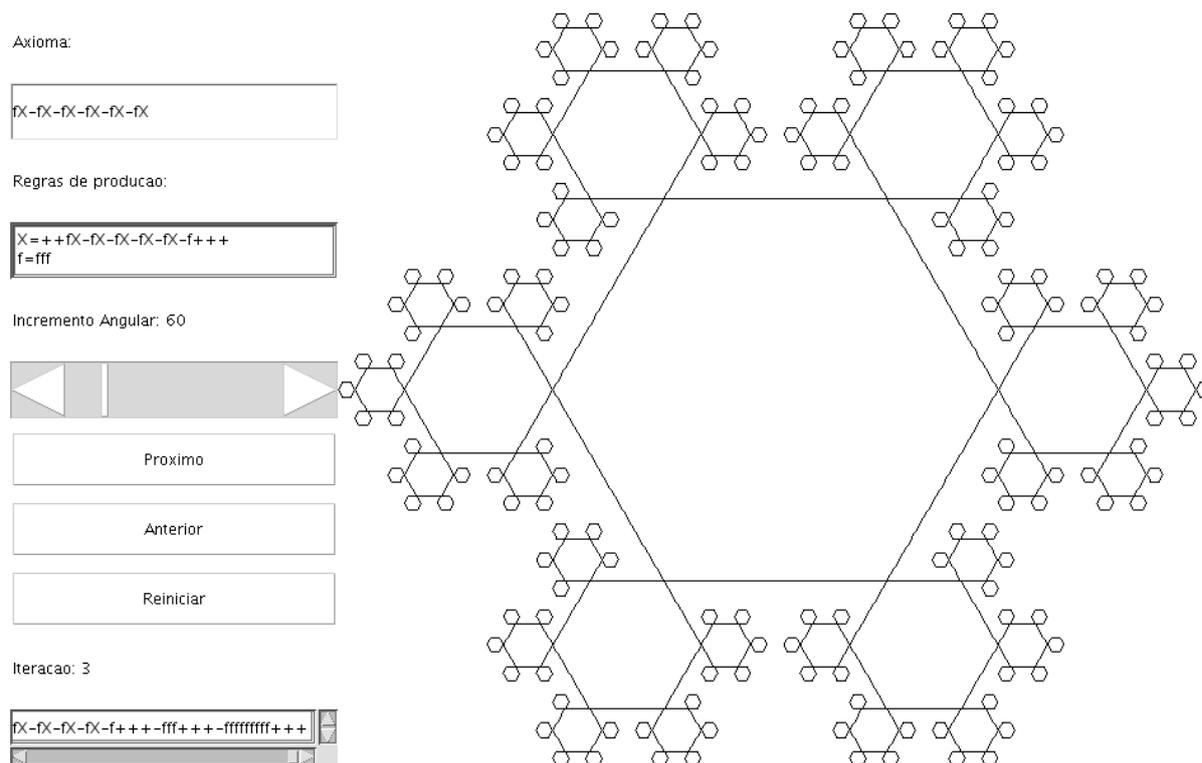


Figura 5.5: Fractal dos Hexágonos

Durante este trabalho, foi desenvolvido também um protótipo de ferramenta, baseado na arquitetura funcionalista do SAGA e nos requisitos funcionais descritos na seção 3.3, capaz de realizar diagnóstico ativo de erros conceituais do aprendiz. Esse protótipo foi projetado para a integração com o OA do domínio de Progressões Geométricas em Fractais. Foram implementados os módulos principais da arquitetura, como o Analisador de Expressões, o Analisador Aritmético e o Interpretador de Regras de Produção. A Base de Fatos e a Base de Regras foram elaboradas a partir dos principais erros conceituais identificados durante o estudo empírico, os quais estão relacionados no Capítulo 4.

Abaixo, listamos alguns exemplos de funcionamento do protótipo. Os três exemplos consideram que o aprendiz está interagindo com OA durante a construção da expressão referente ao perímetro do Fractal Triângulo de Sierpinsky, na iteração referente ao termo genérico da Progressão (passo n). O diagnóstico feito pela ferramenta, e refletido nas mensagens de auxílio exibidas ao aprendiz, considerou que o aprendiz teve erros de sub-generalização, super-generalização e um erro de miscelânea, respectivamente no primeiro,

segundo e terceiro exemplo. A resposta correta para o exercício considerado é a expressão

$$l/2^n$$

Exemplo 1: Solução fornecida pelo aprendiz: $1/n$. Mensagem de diagnóstico: “Você notou que a expressão se refere à n -ésima iteração do Fractal? Sabendo disso, é esperado que a variável n apareça nessa expressão. Tente novamente.”

Exemplo 2: Solução fornecida pelo aprendiz: $l/\sqrt{2 * n}$ Mensagem de diagnóstico: “A presença da operação raiz quadrada envolvendo n faz pouco sentido quando se trata do cálculo do perímetro. Tente novamente.”

Exemplo 3: Solução fornecida pelo aprendiz: l/n Mensagem de diagnóstico: “Você notou que existe uma regularidade (razão) entre os elementos anteriores dessa coluna? Tente incluir essa razão na sua próxima expressão.”

Uma das limitações do protótipo é a incapacidade de fazer raciocínio baseado em incerteza. Em outras palavras, não é possível abduzir a causa de um determinado erro com base na estatística de ocorrências anteriores de erros semelhantes. Outra limitação está no fato de o protótipo considerar apenas um passo de tentativa de solução, para cada exercício proposto, sem relacionar eventos separados. Isso quer dizer que não há um acompanhamento progressivo do aprendizado por meio da construção de um Modelo do Aprendiz.

5.3.2 Funções de Primeiro Grau

Está em desenvolvimento uma ferramenta de autoria, similar àquela descrita na subseção anterior, mas dessa vez aplicada ao domínio de Funções de Primeiro Grau. Através de tal ferramenta, um autor seria capaz de elaborar elementos de enunciados para o OA desse domínio. Assim, essa ferramenta seria correspondente ao módulo Controlador de Enunciados, descrito na arquitetura do SAGA.

Durante o desenvolvimento da OA do domínio de Funções de Primeiro Grau, foram

ligados manualmente no código os elementos que compõem os enunciados dos exercícios propostos. Tais elementos se resumem à descrição do comportamento, e animações correspondentes, de sistemas de molas e pesos e de sistemas de roldanas e polias. Um exemplo de enunciado pode ser verificado na Figura 5.6. Já um exemplo de exercício que introduz Múltiplas Representações Externas, nas formas de animação, gráfico cartesiano e expressão analítica, pode ser observado na Figura 5.7.

Funções de Primeiro Grau - Exercício 6

Instruções **Animação/Simulação**

Preencha cada campo da tabela com o valor ou expressão que corresponde à quantidade de moedas no prato dourado de maneira a equilibrá-lo perfeitamente com o prato azul. Em seguida, trace a reta da função de primeiro grau que relaciona as quantidades de moedas em cada prato e depois responda às questões relativas ao gráfico que serão apresentadas aos poucos.

Cuidado: nesta atividade, o peso do prato azul equivale ao peso de uma moeda, mas não considere isso na quantidade de moedas dele que já está na primeira coluna da tabela.

Dicas:

- (1) Antes de iniciar o preenchimento, utilize a animação na aba ao lado para observar a relação entre as quantidades de moedas nos pratos de maneira a deixar o sistema em equilíbrio;
- (2) Depois disso, para preencher um campo qualquer da tabela, siga cuidadosamente a ordem dada pela

x = Moedas do Prato Azul	y = Moedas do Prato Dourado
0	4
1	Clique aqui para responder
2	

y 20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

0 1 2 3 4 x

Exercício 6

Figura 5.6: Um enunciado do OA de Funções de Primeiro Grau

A aplicação do restante dos módulos da arquitetura SAGA para o domínio mencionado também encontra-se em fase de desenvolvimento. Devido à similaridade de conceitos fundamentais entre esse domínio e o domínio anteriormente abordado, a maior parte do esforço de adaptações deve ficar por conta da construção de novas Bases de Fatos e de Regras, o que inclui a elaboração de mensagens de auxílio especializado para cada tipo de erro conceitual identificado.

Funções de Primeiro Grau – Exercício 6 A A⁻ A⁺

Instruções **Animação/Simulação**

Moedas do prato dourado:

Moedas do prato azul:

Peso ↓

Peso dos pratos:
dourado = nenhuma moeda
azul = 1 moeda

x = Moedas do Prato Azul	y = Moedas do Prato Dourado
3	
4	20 Correto
n	$n \times 4 + 4$ Correto

Questão A: A reta indica qual tipo de função do primeiro grau?

←
→
Exercício 6
📁
📝
🧮
❌

Figura 5.7: Exercício de roldanas do OA de Funções de Primeiro Grau

CAPÍTULO 6

CONCLUSÃO

Neste último capítulo reafirmamos brevemente a contribuição deste trabalho. Por fim, tentamos identificar e descrever de maneira sucinta algumas propostas de trabalho que podem seguir na linha da pesquisa efetuada.

6.1 Reafirmação da contribuição

Neste trabalho apresentamos a importância de tratar o apoio pedagógico ao aprendizado de conceitos analíticos de forma mais profunda. Levantamos casos de ferramentas de software que tentaram avançar do plano aritmético para os domínios de Matemática e Física, mas com restrições a respeito das intervenções sobre o aprendiz nos casos de erro. Passamos então à revisão da bibliografia que trata da construção de software semelhante, tentando identificar possíveis abordagens promissoras, ao passo que incorporávamos a evolução e obstáculos encontrados em projetos similares. Identificamos então as vantagens de se construir um software específico para diagnosticar e posteriormente auxiliar na remediação de erros de generalização de conceitos matemáticos, atuando nos planos analítico, algébrico e geométrico.

Realizamos estudos empíricos, envolvendo entrevistas com especialistas e aplicação de protótipos de OA com aprendizes reais, o que nos levou a identificar os erros conceituais mais comumente apresentados. Com base nisso, estabelecemos conceitos e requisitos para a construção de ferramentas capazes de realizar o diagnóstico e auxiliar na posterior remediação dos erros conceituais. Um diferencial de tal arquitetura de software é a classificação dos erros de generalização de conceitos matemáticos em três classes distintas, o que teorizou-se facilitar o processo de elaboração de intervenções relevantes ao aprendiz, a fim de auxiliá-lo no seu processo de remediação.

Por fim, foram construídas ferramentas baseadas na arquitetura proposta, sendo am-

bas aplicadas ao domínio de Progressões Geométricas em Fractais. A primeira delas, manuseada pelo autor, permite a formulação de fractais para uso em enunciados do OA. A segunda é uma implementação do software diagnosticador e remediador, integrado à Interface do Aprendiz do OA. Com essa última ferramenta, acreditamos ter contribuído para um primeiro passo na identificação das causas de erros conceituais durante o aprendizado de conceitos analíticos.

6.2 Trabalhos Futuros

Durante a elaboração deste trabalho, identificamos algumas lacunas que consideramos importantes para a continuidade do projeto. Uma dessas lacunas trata das limitações existentes nas explicações puramente textuais fornecidas pelo processo de remediação. Tais limitações podem exigir uma demasiada carga cognitiva por parte do aprendiz, no que tange à interpretação da simbologia dos termos típicos do dicionário léxico, por exemplo.

Uma solução possível para essas limitações estaria no uso de MRE nas intervenções do sistema. Uma certa forma de MRE já estaria presente no OA. Por exemplo, durante a análise do valor encontrado pela convergência do somatório das áreas de triângulos em uma iteração alta do Fractal Triângulo de Sierpinsky, o aprendiz deve relacionar uma determinada expressão não somente com outras na mesma coluna da tabela (ver Figura 4.1). Na verdade, ele deve relacionar a expressão também com a representação gráfica do fractal e com expressões presentes nas outras colunas, as quais, por sua vez, representam sub-expressões utilizadas na fórmula da área total dos triângulos menores. Tal análise vai além do uso puro e simples de termos técnicos e inclui a dinâmica das iterações do fractal. Um raciocínio semelhante poderia ser utilizado para criar MRE no processo de remediação.

Outra lacuna identificada refere-se à limitação da ferramenta de software construída, no que tange à quantidade de passos de interação com o aprendiz durante o processo de diagnóstico. Atualmente, a ferramenta só é capaz de realizar um único passo de interação, ao receber a solução fornecida pelo aprendiz, processá-la e informar uma mensagem de diagnóstico. Ao estender a quantidade de passos de interação, fazendo uso de um modelo

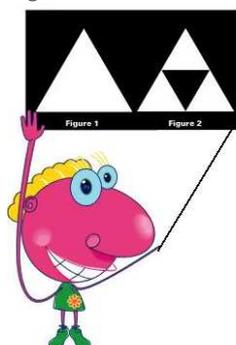
de aprendiz, acredita-se que o diagnóstico possa se tornar cada vez mais preciso. Nessa mesma linha, ao introduzir um modelo de aprendiz, pode-se pensar em realizar uma forma de raciocínio com incertezas, ao analisar-se estatisticamente ou probabilisticamente as causas de um determinado erro, utilizando um processo de inferência abdutiva.

ANEXO A

OA DE PROGRESSÕES GEOMÉTRICAS EM FRACTAIS

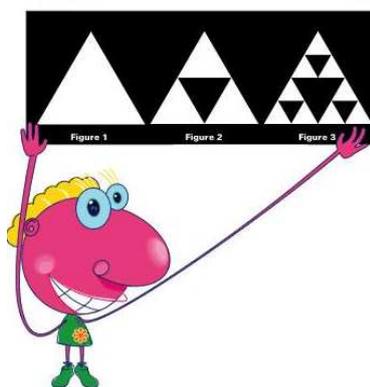
Quadro 1: Triângulo de Sierpinsky

- Inicia-se com um triângulo equilátero.
Aparece escrito: iteração 0: I_0
- a partir deste triângulo equilátero, a cada clique, faz-se os seguintes passos:
 - o Cada um dos lados do triângulo é dividido ao meio;
 - o Os pontos médios de cada um dos lados do triângulo são unidos e forma-se um novo triângulo tendo estes pontos como vértices;
 - o Este triângulo central deve ser retirado;



Aparece escrito: Iteração 1.

- o Repete-se o procedimento em cada um dos triângulos restantes.



Aparece escrito: iteração 2.

E assim por diante...

Quadro 2: Cálculo do Perímetro do Triângulo de Sierpinsky

Aparece um triângulo equilátero, com a indicação Iteração I_0 e com duas lacunas para serem preenchidas: Comprimento do lado e Perímetro.

triângulo

Comprimento do lado $L=$

Perímetro $P=$

Caso haja erro aparecer a mensagem: Perímetro é a soma dos comprimentos dos lados de uma figura plana.

Resposta correta:

Considerar-se-ão corretas as respostas que forem desta forma

Comprimento do lado $L= l$

Perímetro $P=3l$

Ou seja:

Comprimento do lado $L= 1$

Perímetro $P=3$

Ou

Comprimento do lado $L= 2$

Perímetro $P=3 \cdot 2=6$

E assim por diante.

Quadro 3: aparece um triângulo equilátero, com um triângulo vazado no meio, com a indicação Iteração I_1 e com duas lacunas para serem preenchidas: Comprimento do lado e Perímetro.

triângulo

Comprimento do lado $L=$

Perímetro $P=$

Caso haja erro aparecer a mensagem: Perímetro é a soma dos comprimentos dos lados de uma figura plana. Verifique que neste caso, há um buraco no meio, aumentando o número de lados.

Resposta correta:

Considerar-se-ão corretas as respostas que forem desta forma

Comprimento do lado $L = \frac{1}{2}l$

Perímetro $P = 9 \cdot \frac{1}{2}l$

Logo após ser digitada a resposta correta, aparecer a primeira e a segunda resposta, uma abaixo da outra.

Quadro 4: aparece um triângulo equilátero, com um triângulo vazado no meio, e mais três outros, conforme figura, com a indicação Iteração I_2 e com duas lacunas para serem preenchidas: Comprimento do lado e Perímetro.

triângulo

Comprimento do lado $L =$

Perímetro $P =$

Resposta correta:

Considerar-se-ão corretas as respostas que forem desta forma

Comprimento do lado $L = \frac{1}{4}l$

Perímetro $P = 3 \cdot 9 \cdot \frac{1}{4}l$

Logo após ser digitada a resposta correta, aparecer a primeira, a segunda e a terceira resposta, uma abaixo da outra.

Quadro 5: aparece um triângulo equilátero, conforme figura, com a indicação Iteração I_3 e com duas lacunas para serem preenchidas: Comprimento do lado e Perímetro.

triângulo

Comprimento do lado $L =$

Perímetro $P =$

Resposta correta:

Considerar-se-ão corretas as respostas que forem desta forma

Comprimento do lado $L = \frac{1}{8}l$

Perímetro $P = 3 \cdot 3 \cdot 9 \cdot \frac{1}{8}l$

E assim por diante.

O último quadro I_n deverá ter como resposta:

$$\text{Comprimento do lado } L = \frac{1}{2^n} l$$

$$\text{Perímetro } P = 3^{n+1} \cdot \frac{1}{2^n} l$$

E deverão aparecer todas as respostas em uma tabela.

Quadro 6: Aparece a pergunta: o que acontece com o perímetro se este procedimento for repetido muitas vezes, ou seja, se n ficar muito grande?

Pretende-se que o aluno perceba que:

$$\lim_{n \rightarrow \infty} 3^{n+1} \cdot \frac{1}{2^n} l = 3 \cdot \lim_{n \rightarrow \infty} \left(\frac{3}{2}\right)^n l = \infty$$

Sugestão: reorganizar os termos e sugerir que o gráfico da exp seja feito.

ANEXO B

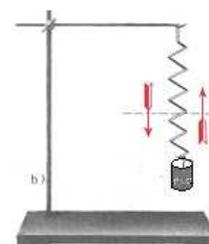
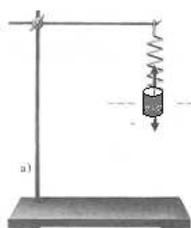
OA DE FUNÇÕES DE PRIMEIRO GRAU

Experimento : Molas

Atividade 1: Alongamento de Molas

Apresentar, no início da atividade, várias molas, de mesmo comprimento, com constantes elásticas distintas, para o aluno escolher uma. A partir de diferentes escolhas, o aluno deve descobrir que quanto mais forte a mola, menor a sua constante elástica e, portanto, menor o alongamento. Observação: as constantes elásticas devem ser valores entre 0,5 e 3.

No primeiro quadro, deve aparecer um recipiente (porta-filmes) preso em uma mola, no qual devem ser colocadas moedas de mesmo peso. A cada moeda, mede-se o alongamento que a mola sofreu. (este alongamento é obtido a partir da diferença entre o comprimento final e o comprimento inicial da mola).



Número de moedas (n)	Alongamento (A(n))
0	0
1	K.1
2	K.2
...	...
n	K.n

O objetivo do exercício é que o aluno encontre uma fórmula para cada tipo de mola. O Alongamento da Mola, de acordo com a constante elástica, vai gerar uma função do tipo

$$Y=ax,$$

Nesta atividade, Y é o Alongamento ($A(n)$), a é a constante elástica da mola (K) e x é o número de moedas (n).

$$A(n)= K.n$$

Chamar a atenção para:

- tem-se duas variáveis: n , que é a variável independente e $A(n)$ que é a variável dependente, ou seja, o Alongamento depende do número de moedas;
- neste exemplo, n é uma **variável discreta** e $A(n)$ é uma variável contínua. Uma variável discreta é aquela que assume valores inteiros e positivos. (É aquela que pode ser contada). Uma **variável contínua** é aquela que pode assumir valores não inteiros. (É aquela que pode ser medida).
- para cada valor da variável independente n , tem-se um único valor para a variável dependente ($A(n)$) a ele correspondente. Esta é uma condição necessária e suficiente para caracterizar uma função.

Definição de função: Dados dois conjuntos A e B , dizemos que f é uma função de A em B , se para cada elemento x de A (variável independente) corresponder um único elemento y de B (variável dependente), tal que $f(x)=y$.

Notação: $f: A \rightarrow B$

$$x \rightarrow f(x)$$

Atividade 2: Construção de Gráficos do Alongamento de Molas

Pedir que o aluno construa os gráficos, no mesmo plano cartesiano, das diferentes funções obtidas na Atividade 1.

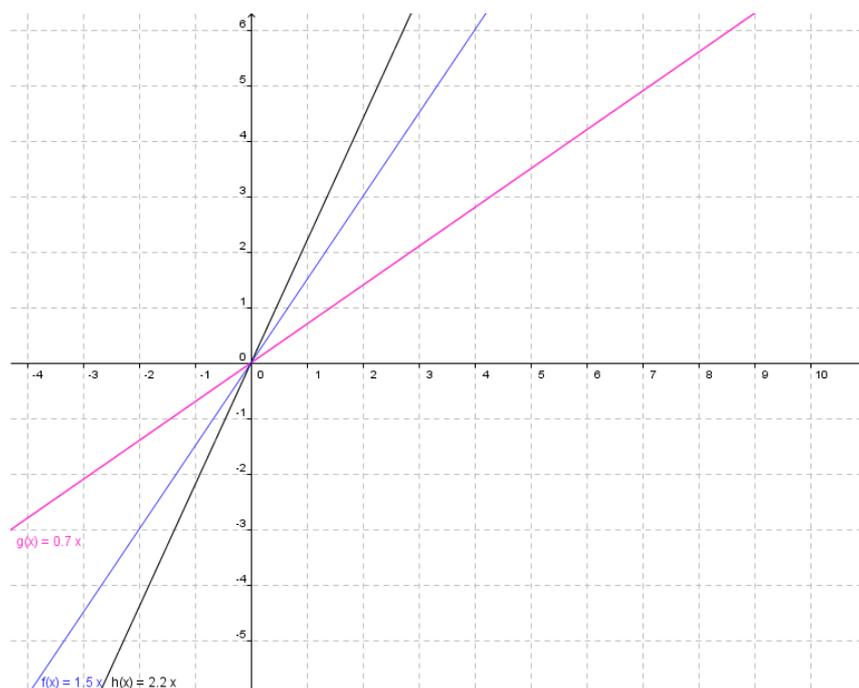
Recomendamos o software Geogebra por ser livre e muito eficiente.

Daremos, a seguir, exemplos de 3 funções de molas com constantes elásticas distintas.

$$A_1(n)=1,5.n$$

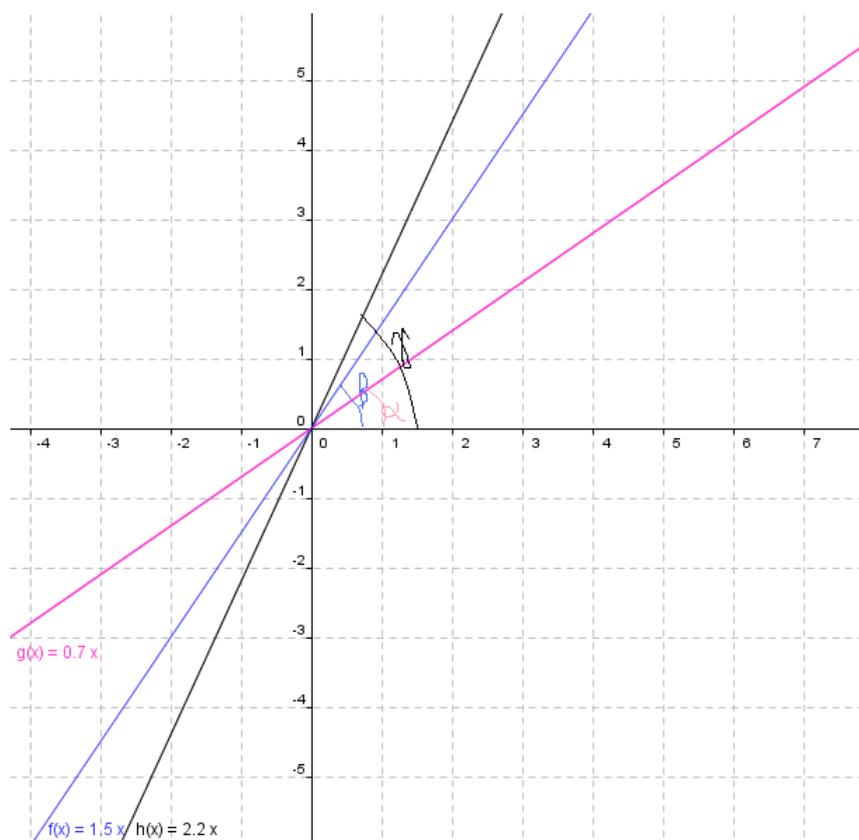
$$A_2(n)=0,7.n$$

$$A_3(n)=2,2.n$$



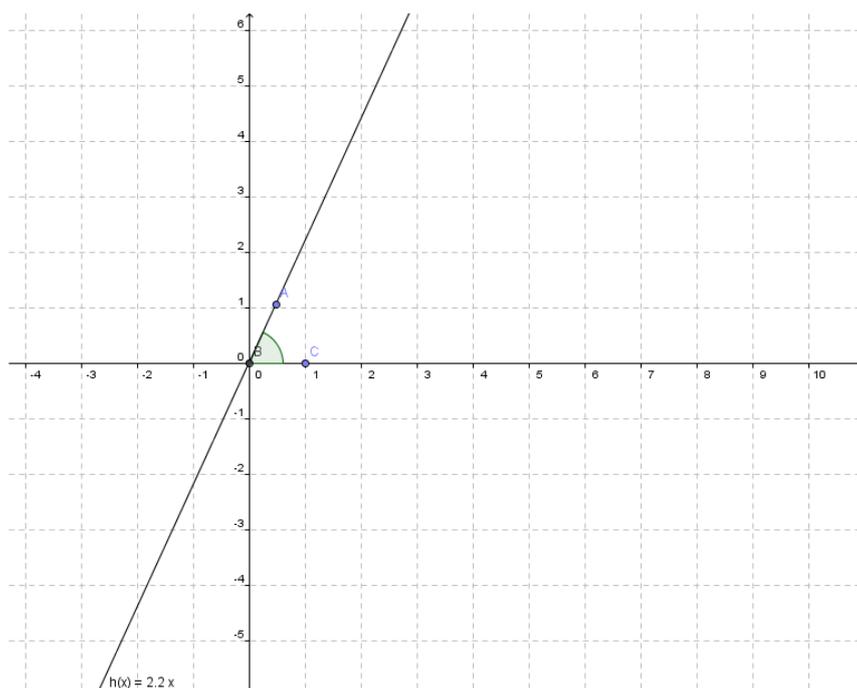
Pedir para que o aluno compare a inclinação das retas obtidas (que representam diferentes alongamentos) com as constantes elásticas. Pedir que ele clique no gráfico da reta que representa a mola mais fraca. Deve concluir que quanto maior a inclinação da reta, maior a constante elástica e mais fraca a mola. A esta inclinação dá-se o nome de coeficiente angular da reta.

Observe no gráfico a seguir os ângulos α, β, γ que as retas formam com o eixo x. Calcule o valor da tangente desses ângulos e os compare.



Definição de Coeficiente Angular da Reta:

Seja a reta $Y=ax$ representada abaixo e seja α o ângulo que esta forma com o eixo x.



Colocar o ângulo α e $\frac{\Delta y}{\Delta x}$.

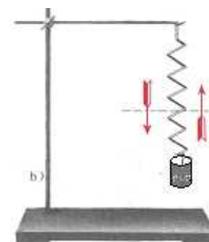
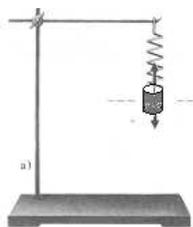
O número a , chamado coeficiente angular de Y é a tangente do ângulo que a reta forma com a direção horizontal e é dado por $a = \operatorname{tg} \alpha = \frac{\Delta y}{\Delta x}$.

Observação: se o coeficiente angular for positivo, a reta representará uma função crescente, se for negativo, representará uma função decrescente.

Atividade 3: Comprimento de Molas

Apresentar, no início da atividade, várias molas, com a mesma constante elástica K (por exemplo, $k=0,5$) e comprimentos distintos, para o aluno escolher uma. Observação: os comprimentos das molas devem variar entre 3cm e 10cm.

No primeiro quadro, deve aparecer um recipiente (porta-filmes) preso em uma mola, no qual devem ser colocadas moedas de mesmo peso. A cada moeda, mede-se o comprimento total da mola, o qual deve ser escrito da seguinte forma: $C=C_0+A(n)$.



Número de moedas (n)	Comprimento (C(n))
0	
1	C_0
	C_0+k
2	
	C_0+2K
n	
	$C_0+K.n$

O objetivo do exercício é que o aluno encontre uma fórmula para cada mola. O Comprimento Total da Mola, de acordo com o Comprimento inicial, vai gerar uma função do tipo

$$Y=ax + b,$$

Nesta atividade, Y é o Comprimento Total (C(n)), a é a constante elástica da mola (K), x é o número de moedas (n) e b é o comprimento inicial da mola (C_0).

$$C(n)= K.n + C_0$$

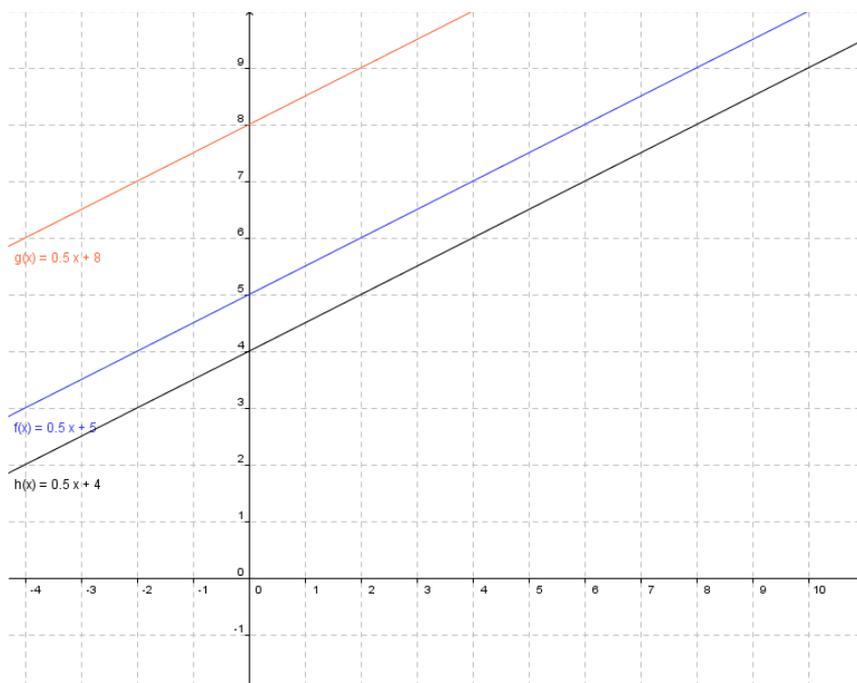
Atividade 4: Construção de Gráficos do Comprimento de Molas

Pedir que o aluno construa os gráficos, no mesmo plano cartesiano, das diferentes funções obtidas na Atividade 3. Daremos, a seguir, exemplos de 3 funções de molas com comprimentos iniciais distintos.

$$C_1(n)=0,5.n + 5$$

$$C_2(n)=0,5.n + 8$$

$$C_3(n) = 0,5.n + 4$$



Pedir para que o aluno compare os pontos nos quais os gráficos interceptam o eixo y com os diferentes comprimentos iniciais das molas. Deve concluir que o ponto em que o gráfico intercepta o eixo y corresponde ao comprimento inicial da mola. A este valor dá-se o nome de coeficiente linear da reta.

Definição de Coeficiente Linear da Reta:

Seja a reta $Y = ax + b$ representada abaixo e seja $(0, b)$ o ponto em que esta intercepta o eixo y. Ao valor b dá-se o nome de Coeficiente Linear.

Chamar a atenção do aluno para o fato de que estas retas são paralelas (por apresentarem o mesmo coeficiente angular). Os distintos coeficientes lineares imprimem uma translação da reta na direção do eixo y.

À função do tipo $Y = ax + b$, com a e b constantes reais e $a \neq 0$, dá-se o nome de função do primeiro grau.

Optamos aqui por não reforçar a distinção feita no Ensino Médio entre as Funções Afim e Linear, a qual provoca obstáculos didáticos no Ensino Superior, no qual todas as funções do primeiro grau são consideradas funções lineares.

ANEXO C

OA DE MATEMÁTICA FINANCEIRA

Introdução sobre o assunto

De modo geral, a matemática financeira utiliza conceitos matemáticos aplicados à análise de dados financeiros. Voltada para a área comercial, a matemática financeira é utilizada na análise de alternativas de investimento ou financiamento de bens de consumo.

Alguns dos conceitos inerentes à matemática financeira são os conceitos de **desconto**, **juros**, **juros simples** e **juros compostos**, bem como os conceitos de **capital**, **período** e **montante**.

Em cada conceito em vermelho deve haver um link para a respectiva definição que está mais a frente neste documento.

3

Idéias Gerais sobre o simulador

- **Interface:** jogo em que os alunos devem realizar 5 tarefas sobre Matemática Comercial e Financeira.
- **Objetivo do jogo:** Utilizar os conceitos de Matemática Comercial e Financeira ao realizar as tarefas.
- **Contexto:** o centro comercial de uma cidade.

Na página inicial do simulador deve ser apresentada uma cidade em sua vista superior. Para iniciar o jogo, à tarefa 1 aparece. Somente após a realização da mesma é que o jogo apresenta a tarefa 2 e assim sucessivamente. A cada tarefa, parte da cidade em que se situam os estabelecimentos envolvidos em sua execução, são ampliados como por uma lente de aumento. Ao clicar em um dos três estabelecimentos (os quais oferecem opções relacionadas à tarefa), surge uma nova "janela" mostrando o produto ou serviço oferecido e a forma de pagamento.

4

- *Junto a cada tarefa, deve aparecer um ícone de ajuda, o qual o aluno pode "escolher" consultar para visualizar conceitos de juros simples, composto, desconto e etc., e uma barra de ferramenta de cálculo (tipo planilha do Excel ou calculadora científica), caso o aluno julgue necessário.*
- *O aluno deve realizar os cálculos referentes às formas de pagamento oferecidas em cada estabelecimento e digitar a resposta num local indicado pelo jogo. Ao digitar as três respostas corretas o jogo pede que o aluno decida pelo estabelecimento que responde ao problema apresentado na tarefa. Se alguma resposta estiver incorreta, o jogo avisa e dá uma nova chance (máximo de 2 novas chances). Caso o aluno, mesmo visualizando as três respostas corretas, decida por um estabelecimento em que o produto é mais caro: game over. Caso utilize as três chances e não encontre os valores esperados: game over.*

5

- *Para cada estabelecimento consultado na tarefa n, o jogo armazenará um total de 10 (ou quanto acharem conveniente) valores diferentes nas formas de pagamento, o que implica em respostas diferentes para cada vez que o aluno executar a tarefa 1, por exemplo, ao iniciar o jogo novamente. É interessante que o próprio jogo selecione tais valores dentre os armazenados, de modo que em uma sala de aula, por exemplo, grupos de alunos em máquinas distintas possam realizar as mesmas tarefas, obtendo respostas diferentes.*
- *Ao finalizar o jogo, executando corretamente todas as tarefas, surge uma comemoração na tela, bem como o tempo em que o aluno realizou as atividades. Este tempo fica armazenado no site ou na máquina em que o jogo foi executado, de modo a compor um ranking com os 10 melhores tempos.*

6

Tarefas

Descreveremos, a seguir, cada uma das tarefas que idealizamos. As primeiras são para introdução dos conceitos de desconto, juros simples e juros compostos. As outras são para aplicação dos conceitos já apresentados. Nestas atividades de aplicação é que surgirão 3 opções de estabelecimento para que o aluno escolha, dentre elas, onde o preço de determinado produto ou serviço é mais barato.

7

1ª Situação

Você pretende comprar um rádio cujo o valor é R\$699,00. No entanto, você tem R\$500,00 para dar de entrada. Você decide pedir um desconto para pagamento à vista e acaba ganhando um desconto de 30%. Você conseguirá pagar o rádio com o dinheiro que já tem ou faltará dinheiro para isso? Se faltar, quanto faltará? Se conseguir pagar com o dinheiro que tem, sobrá quanto?



RESPOSTA: *Para o pagamento à vista com desconto, o rádio sairá por R\$489,30 . Isso significa que o dinheiro que tenho é suficiente para pagar o rádio e que sobrá a quantia de R\$209,70.*

8

Na situação acima é possível identificar o conceito de **Desconto**

Desconto é o abatimento dado ao valor de um produto, geralmente negociado para pagamento do produto à vista, ou seja, para pagamento integral do produto no ato da compra do mesmo.

9

2ª Situação

Você pretende investir R\$500,00 e resgatar o dinheiro 3 meses depois. Para isso, tem de decidir entre duas instituições financeiras que oferecem as condições destacadas abaixo. Em qual delas, resgatará uma quantia maior? Qual o valor que resgatará em cada instituição?

<p>Instituição Financeira 1</p> <p>Juros de 2% ao mês, calculado sempre sobre o valor inicial do investimento.</p>		<p>Instituição Financeira 2</p> <p>Juros de 2% ao mês, calculado sobre o montante do mês anterior.</p> <p><i>Obs. Montante é o capital inicial (investimento inicial neste caso) acrescido de juros.</i></p>
----------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

RESPOSTA: Resgatará uma quantia maior na instituição financeira 2, pois enquanto na instituição financeira 1 resgatará R\$740,94, na instituição 2 resgatará R\$741,78.

10

Na situação acima temos o conceito de **Juros**, que representam uma compensação em dinheiro, pelo uso de um capital financeiro, em um determinado intervalo de tempo.

Na instituição financeira 1, como em cada período de tempo o juro gerado é constante e é calculado sobre o valor do investimento inicial (na situação – 699 reais), temos os chamados **JUROS SIMPLES**.

Na instituição financeira 2, como após cada período de tempo, os juros gerados são incorporados ao capital do início do período, a partir do qual incidirão novos juros no período seguinte, temos os chamados **JUROS COMPOSTOS** ou juros sobre juros.

11

3ª Situação

Este mês você precisa comprar uma geladeira. Algumas lojas da cidade estão oferecendo desconto na compra deste produto à vista. A partir de uma pesquisa pelas lojas da cidade você selecionou três planos de pagamento. Escolha a partir das opções abaixo a opção mais vantajosa.

Loja 1	Loja 2	Loja 3
Preço do produto R\$ 1500,00	Preço do produto R\$ 1399,00	Preço do produto R\$ 1699,00
À vista com desconto de: 30%	À vista com desconto de: 15%	À vista com desconto de: 40%

RESPOSTA:

12

Neste primeiro caso, a resposta correta é... Loja 3.

Valores que devem substituir as opções apresentadas em cada loja, a cada vez que o jogo for iniciado. O valor da geladeira e a porcentagem de desconto é alterada para cada nova jogada, como segue:

Loja 1		Loja 2		Loja 3		Respostas
Valor à vista	Desconto	Valor à vista	Desconto	Valor à vista	Desconto	
899,00	10%	1200,00	20%	1099,00	30%	Loja 3
699,00	20%	899,00	30%	599,00	5%	Loja 1
1299,00	12,5%	1500,00	27%	1469,00	30%	Loja 3
1120,00	5%	1200,00	15%	1400,00	20%	Loja 2
1900,00	30%	2200,00	40%	1699,00	15%	Loja 2

Em cada nova jogada, o simulador poderia sortear uma das opções. 13

4ª Situação

Você precisa fazer um curso de inglês – nível básico – por 6 meses e decide pesquisar condições de pagamento em três escolas diferentes.

Descubra em qual destas escolas o pagamento a prazo implica em um maior aumento monetário em relação ao valor à vista.

Escola 1	Escola 2	Escola 3
Condições de Pagamento: À vista: R\$ 600,00	Condições de Pagamento: À vista: R\$ 800,00	Condições de Pagamento: À vista: R\$ 500,00
À prazo: Entrada de R\$ 110,00 mais 5 parcelas fixas com taxa de 8% sobre o valor da entrada.	À prazo: Entrada de R\$ 150,00 mais 4 parcelas fixas com taxa de 10% sobre o valor da entrada.	À prazo: Entrada de R\$ 220,00 mais 3 parcelas fixas com taxa de 30% sobre o valor da entrada.

RESPOSTA:

14

Neste primeiro caso, a resposta correta é... Instituição 2.

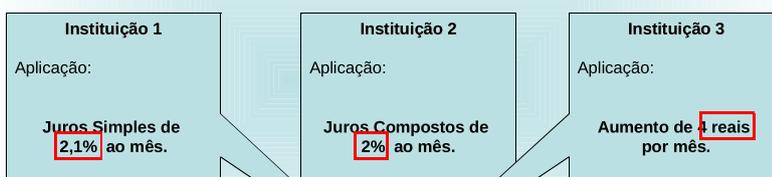
Valores que devem substituir as opções apresentadas em cada escola de inglês, a cada vez que o jogo for iniciado. O número de parcelas e a taxa de juros permanecem e os valores à vista e de entrada para pagamento a prazo são alterados conforme segue:

Escola 1		Escola 2		Escola 3		Respostas
à vista	Entrada	à vista	Entrada	à vista	Entrada	
630,00	115,50	840,00	157,50	525,00	200,00	Escola 3
660,00	20,00	880,00	140,00	550,00	225,00	Escola 2
690,00	20,00	920,00	330,00	575,00	330,00	Escola 1
720,00	132,00	960,00	180,00	600,00	126,00	Escola 3
570,00	104,50	760,00	100,00	475,00	275,00	Escola 2

Em cada nova jogada, o simulador poderia sortear uma das opções. 15

5ª Situação de aprendizagem

Você quer aplicar uma quantia de 200 reais durante 3 meses e precisa saber qual aplicação é mais vantajosa, dentre as oferecidas por três instituições.



RESPOSTA:

Neste primeiro caso, a resposta correta é... Instituição 1.

Valores que devem substituir as opções apresentadas em cada instituição, a cada vez que o jogo for iniciado. O tempo de 3 meses deve ser mantido. Já a taxa de juros e o valor do capital inicial destacados, devem mudar como segue:

	Instituição 1			Instituição 2		Instituição 3	Resposta
	Capital Inicial	Taxa de Juros Simples	Taxa de Juros Compostos	Aumento de x Reais por mês			
opção 1	500 reais	12%	11%	58 reais		Instituição 2	
opção 2	1000 reais	12%	10%	121 reais		Instituição 3	
opção 3	250 reais	6%	5%	14,50 reais		Instituição 1	
opção 4	100 reais	1%	1,2%	0,67 reais		Instituição 2	
opção 5	400 reais	7%	7%	30,30 reais		Instituição 3	

Em cada nova jogada, o simulador sorteia uma das opções. 17

6ª Situação

Julia aplicou 1256 reais durante um ano. Após tal período com juros de 1,2% ao mês a aplicação rendeu 193,29 reais. Qual tipo de juro incidiu sobre a aplicação de Julia?

RESPOSTA: Juros Compostos

Valores que devem substituir as opções apresentadas em cada instituição, a cada vez que o jogo for iniciado.

	Valor depositado	Taxa de juros	Juros	Resposta
opção 1	1256,00	1,2%	193,29	Compostos
opção 2	850,00	3,2%	326,4	Simple
opção 3	724,00	1,8%	156,38	Simple
opção 4	368,00	3,0%	156,68	Compostos
opção 5	1500,00	2,3%	414,00	Simple
opção 6	2250,00	1,5%	440,14	Compostos

Em cada nova jogada, o simulador sorteia uma das opções. 19

7ª Situação

Para comprar uma TV, você decide fazer um financiamento de R\$1000,00 em um banco que cobra uma taxa de juros de 1% ao mês. Essa importância será amortizada através do SAC – Sistema de Amortizações Constantes – em 5 prestações mensais. Quanto você pagará ao final de tudo?

RESPOSTA:

Valores que devem substituir o total financiado, a taxa de juros por mês e o período de financiamento.

	Valor Financiado	Taxa de juros	Período	Resposta
opção 1	2000 reais	2%	5 meses	2120 reais
opção 2	800 reais	2%	4 meses	840 reais
opção 3	600 reais	5%	4 meses	675 reais
opção 4	480 reais	5%	3 meses	528 reais
opção 5	300 reais	1%	3 meses	306 reais

Em cada nova jogada, o simulador poderia sortear uma das opções.

Conceitos

(Ajuda)

Incluimos, a seguir, conceitos que são fundamentais e que os alunos irão consultar caso julguem necessário.

Desconto

O desconto é o abatimento dado a um valor monetário em determinadas condições. De modo geral, é expresso por um percentual aplicado sobre o valor inicial do produto.

23

Juros

Os juros representam uma compensação em dinheiro, pelo uso de um capital financeiro, em um determinado intervalo de tempo.

24

Juros Simples

Quando em cada período de tempo o juro gerado é constante e é calculado somente sobre o capital inicial.

25

Juros Compostos

Quando após cada período de tempo, os juros gerados são incorporados ao capital do início do período, a partir do qual incidirão novos juros no período seguinte, proporcionando o que chamamos "juros sobre juros" ou "juros compostos".

26

Capital

Valor aplicado através de alguma operação financeira. Bem inicial investido ou valor inicial financiado.

27

Período

Intervalo de tempo (dia, mês, bimestre, trimestre, semestre, ano) no qual o investimento / financiamento foi ou será efetuado. Datas de início ou de término da operação.

28

Montante

É o capital inicial acrescido de juros.

29

Amortização

É o processo de pagamento de uma dívida por meio de pagamentos periódicos, para tal, o valor total do capital empregado na operação é separado em parcelas que podem ter o mesmo valor ou não, as quais são chamadas amortizações e correspondem a um dos elementos utilizados no cálculo de cada prestação a ser paga.

Conheça um sistema de amortização – SAC ([fazer um link aqui](#)).

Prestações

As prestações a serem pagas por quem empresta o capital são obtidas acrescentando a cada uma das amortizações, juros referentes ao saldo devedor que ainda resta a ser pago para a liquidação total da dívida. O saldo devedor de cada período é obtido retirando-se do saldo devedor anterior, a amortização correspondente a este período.

30

SAC – Sistema de Amortizações Constantes

Como o próprio nome diz, as amortizações neste sistema devem ser constantes, ou seja, possuir o mesmo valor. Deste modo, para obter o valor de cada uma delas, basta dividir o valor total do capital empregado na operação financeira pela quantidade de períodos que levará para a quitação do empréstimo.

Por exemplo, se emprestarmos 300 reais para serem pagos em 3 meses, inicialmente dividimos 300 por 3, obtendo 100 reais – que é o valor de cada amortização e não o de cada prestação.

Para obter o valor da prestação, calcula-se também os juros que incidirão sobre cada parcela. Os juros são calculados sempre em cima do saldo devedor. No exemplo, tomando uma taxa de juros de 10% ao mês, temos:

No primeiro mês: como deve-se os 300 reais emprestados, calcula-se 10% destes 300, isto é, 30 reais, que correspondem aos juros pagos na primeira parcela. Logo, a primeira parcela será de 100 reais (a primeira amortização) + 30 reais (juros da primeira parcela) – 130 reais.

No segundo mês: como já foram pagos 100 reais da dívida inicial (amortização presente na primeira parcela), o saldo devedor, neste momento é de 200 reais. Assim, calcula-se 10% destes 200, isto é, 20 reais, que correspondem aos juros pagos na segunda parcela. Logo, a segunda parcela será de 100 reais (a segunda amortização) + 20 reais (juros da segunda parcela) – 120 reais.

No terceiro e último mês: como já foram pagos 200 reais da dívida inicial (amortização presente nas duas primeiras parcelas), o saldo devedor, agora, é de 100 reais. Calcula-se 10% destes 100, obtendo 10 reais. Como antes, a terceira parcela será composta da amortização final de 100 reais (fim da dívida) e dos 10 reais correspondentes ao saldo devedor (juros da terceira parcela) – 110 reais.

Finalmente, o valor final pago pelo empréstimo será de $130+120+110$ reais = 360 reais.

Percebe-se que no SAC as amortizações são constantes e os juros e as parcelas são decrescentes.

31

Formulário

em Ajuda

32

Fórmula para o cálculo do valor de venda de um determinado produto após ser dado um desconto sobre o seu valor de custo.

$$V = C. (1 - i)$$

sendo

V = valor de venda

C = valor de custo

i = taxa percentual de desconto oferecida (escrita na forma decimal)

33

Fórmula para o cálculo de desconto.

$$D = |V - C|$$

sendo

C = valor de custo

V = valor de venda

34

Fórmula para o cálculo de juros simples

$$J = c.i.n$$

sendo

J = juros

c = capital inicial

i = taxa de juros (escrita na forma decimal)

n = tempo

35

Fórmula para o Cálculo do Montante obtido a juros simples.

$$M = C.(1+i.n)$$

sendo

M = Montante após n períodos

C = Capital inicial

i = taxa de juros (escrita na forma decimal)

n = tempo

36

Fórmula para o Cálculo do Montante obtido a juros compostos

$$M = C \cdot (1+i)^n$$

sendo

M = Montante após n períodos

C = Capital inicial

i = taxa de juros (escrita na forma decimal)

n = tempo

37

Ajuda com relação a execução do jogo

Ajuda
Esta parte deve ser implementada pelos responsáveis técnicos
pela tradução do jogo em linguagem computacional.

38

BIBLIOGRAFIA

- [1] A. Adam e J. P. Laurent. Laura, a system to debug student programs. *Artificial Intelligence*, 15(1-2):75 – 122, 1980.
- [2] S. Ainsworth. A functional taxonomy of multiple representations. *Computers and Education*, 33(2/3):131–152, 1999.
- [3] S. Ainsworth. DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, 16(3):183–198, 2006.
- [4] S. Ainsworth e N. V. Labeke. Using a multi-representational design framework to develop and evaluate a dynamic simulation environment. Dynamic Information and Visualisation Workshop, julho de 2002.
- [5] J. R. Anderson. Acquisition of cognitive skill. *Psychological Review*, 89(89):369–403, 1982.
- [6] J. R. Anderson. *The Architecture of Cognition*. Harvard University Press, 1983.
- [7] J. R. Anderson. *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., 1993.
- [8] J. R. Anderson, D. F. Boyle, e G. Yost. The geometry tutor. *IJCAI-85*, páginas 1–7, 1985.
- [9] J. R. Anderson e B. J. Reiser. The lisp tutor: it approaches the effectiveness of a human tutor. *BYTE*, 10(4):159–175, 1985.
- [10] R. Azevedo. Beyond intelligent tutoring systems: Using computers as meta-cognitive tools to enhance learning? *Instructional Science*, 30(1):31–45, 2002.
- [11] A. Barr, M. Beard, e R. C. Atkinson. The computer as tutorial laboratory: the stanford bip project. *International Journal on the Man-Machine Studies*, 8(5):567–596, 1976.

- [12] J. S. Brown. Process versus product : A perspective on tools for communal and informal electronic learning. *Journal of Computing Research*, 1(2):179–201, 1985.
- [13] J. S. Brown e K. VanLehn. Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science*, 4:379–426, 1980.
- [14] P. Brusilovsky. Adaptive hypermedia: From intelligent tutoring systems to web-based education. *International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science*,, páginas 1–7. Springer, 2000.
- [15] S. Bull. Supporting learning with open learner models. *4th Hellenic Conference with International Participation: Information and Communication Technologies in Education, Atenas*, 2004.
- [16] S. Bull e J. Kay. Student models that invite the learner in: The SMILI open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120, 2007.
- [17] S. Bull e H. Pain. Did I say what I think I said, and do you agree with me? inspecting and questioning the student model. *Proceedings of World Conference on Artificial Intelligence in Education (AIED-95)*, páginas 124–131, 1995.
- [18] J.R. Carbonell. AI in CAI: An artificial intelligence approach to computerassisted instruction. *IEEE Transactions on Man-Machine Systems*, 11:190–202, 1970.
- [19] W. J. Clancey. *Knowledge-based Tutoring: the GUIDON Program*. MIT Press, 1987.
- [20] W. J. Clancey e R. Letsinger. NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. *IJCAI-81*, páginas 829–835, 1981.
- [21] R. Cox e P. Brna. Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education*, 6(2/3):239–302, 1995.
- [22] D. Cury, A. Direne, e N. Omar. Modelos baseados em oráculos para a aprendizagem de conceitos visuais. *Revista Brasileira de Informática na Educação*, 2:18–29, 1998.

- [23] A. Direne. Authoring intelligent systems for teaching visual concepts. *International Journal of Artificial Intelligence in Education*, 8(1):71–78, 1997.
- [24] A. Direne, D. Scott, F. Silva, L. Bona, , M. Castilho, M. Sunyé, e L. García. Acquiring expertise in medical radiology through long-term interactions. *Proceedings of the 21th IEEE International Symposium on Computer-Based Medical Systems (CBMS2008)*, páginas 403–408, 2008.
- [25] A. Direne, D. Scott, F. Silva, L. Bona, , M. Castilho, M. Sunyé, e L. García. Authoring adaptive tutoring systems for complex visual skills. *Proceedings of the Nineth IEEE International Conference on Advanced Learning Technologies (ICALT-2009)*, páginas 160–164, 2009.
- [26] B. du Boulay e C. Sothcott. Computers teaching programming: an introductory survey of the field. *Artificial intelligence and education; vol. 1: learning environments and tutoring systems*, 1:345 – 372, 1987.
- [27] B. du Boulay e C. Sothcott. *Intelligent systems for teaching programming*. Elsevier Science Publishers, 1988.
- [28] I. P. Goldstein. Summary of mycroft: a system for understanding simple picture programs. *Artificial Intelligence*, 6:249–288, 1975.
- [29] T. Hasemer. An empirically-based debugging system for novice programmers. Relatório Técnico 6, Human Cognition Research Laboratory, Open University., 1983.
- [30] A. Herzig, D. Longin, e J. Virbel. Towards an analysis of dialogue acts and indirect speech acts in a bdi framework. Massimo Poesio e David Traum, editors, *Proc. 4th Workshop on the Semantics and Pragmatics of Dialogue (GOTALOG 2000)*, 2000.
- [31] N. Jeffery, M. Sharples, B. A. Teather, D. Teather, B. du Boulay, G. du Boulay, e A. Direne. Statistical modelling and intelligent tutoring of visual concepts for mr imaging of the head. *Proceedings of World Conference on Artificial Intelligence in Education (AI-ED 93)*, páginas 562–563, 1993.

- [32] W. L. Johnson. *Intention-based Diagnosis of Novice Programming Errors*. Research Notes in Artificial Intelligence, Morgan Kaufmann, 1986.
- [33] W. L. Johnson e E. Soloway. Proust: an automatic debugger for pascal programs. *BYTE*, 10(4):179–190, 1985.
- [34] W. L. Johnson e E. Soloway. Proust: Knowledge-based program understanding. *IEEE Transactions on Software Engineering*, 11:267–275, 1985.
- [35] F. J. Lukey. Understanding and debugging programs. *International Journal of Man-Machine Studies*, 12(2):189 – 202, 1980.
- [36] D. Marczal. Um arcabouço que enfatiza a retroação a contextos de erro durante o acesso a conteúdos educacionais. Dissertação de Mestrado, PPGInf - Universidade Federal do Paraná, 2010.
- [37] D. C. Merrill, B. J. Reiser, M. Ranney, e J. G. Trafton. Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *Journal of the Learning Sciences*, 2(3):277 – 305, July de 1992.
- [38] T. Murray, L. Winship, R. Bellin, e M. Cornell. Toward glass box educational simulations: Reifying models for inspection and design. *Workshop External representations in AIED, San Antonio, Texas*, 2001.
- [39] W. R. Murray. *Automatic Program Debugging for Intelligent Tutoring Systems*. Morgan Kaufmann, 1988.
- [40] W. R. Murray. Automatic program debugging for intelligent tutoring systems. *Computational Intelligence*, 3(1):1–16, 2007.
- [41] H. A. Ramadhan. Programming by discovery. *Journal of Computer Assisted Learning*, 16:83–93, 2000.
- [42] B. Reiser. Causal models in programming. *Proceedings of World Conference on Artificial Intelligence in Education*, 1987.

- [43] E. Y. Shapiro. *Algorithmic Program Debugging*. MIT Press, Cambridge, MA, 1982.
- [44] M. Sharples, N. Jeffery, B. du Boulay, B. Teather, D. Teather, e G. du Boulay. Structured computer-based training in the interpretation of neuroradiological images. *International Journal of Medical Informatics*, 60(3):263–280, 2000.
- [45] E. Shortliffe. *Computer Based Medical Consultations: MYCIN*. Elsevier, 1976.
- [46] E. M. Soloway e W. L. Johnson. Remembrance of blunders past: a retrospective on the development of PROUST. *Proc. of the 6th Cognitive Science Society Conference*, 1984.
- [47] J. S. Song, S. H. Hahn, K. Y. Tak, e J. H. Kim. An intelligent tutoring system for introductory c language course. *Computers & Education*, 28(2):93 – 102, 1997.
- [48] B. A. Teather, M. Sharples, N. Jeffery, D. Teather, B. du Boulay, A. I. Direne, e G. H. du Boulay. Statistical modelling and structured image description for intelligent tutoring in MR images of the head. *Rivista di Neuroradiologia*, 7:29–35, 1994.
- [49] D. Teather, B. A. Teather, K. M. Wills, G. H. du Boulay, et al. Evaluation of computer advisor in the interpretation of CT images of the head. *Neuroradiology*, 30:511–577, 1988.
- [50] K. VanLehn. Learning one subprocedure per lesson. *Artificial Intelligence*, 31:1–40, 1987.
- [51] K. VanLehn. Cascade: A simulation of human learning and its applications. & H. Pain S. P. Brna, S. Ohlsson, editor, *Artificial Intelligence in Education, 1993: Proceedings of AI-ED 93*. Charlottesville, VA: Association for the Advancement of Computing in Education, 1993.
- [52] K. VanLehn. Analogy events: How examples are used during problem solving. *Cognitive Science*, 22(3):347–388, 1998.
- [53] K. VanLehn. Rule learning events in the acquisition of a complex skill: An evaluation of cascade. *Journal of the Learning Sciences*, 8(1):71–125, 1999.

- [54] K. VanLehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, e M. Wintersgill. The andes physics tutoring system: Lessons learned. *In International Journal of Artificial Intelligence and Education*, 15(3):1–47, 2005.
- [55] K. VanLehn, S. Siler, C. Murray, e W. Baggett. What makes a tutorial event effective? M. A. Gernsbacher & S. Derry, editor, *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, páginas 1084–1089, Hillsdale, NJ, 1998. Erlbaum.
- [56] K. VanLehn, S. Siler, C. Murray, T. Yamauchi, e W. B. Baggett. Why do only some events cause learning during human tutoring? *Cognition and Instruction*, 21(3):209–249, 2003.
- [57] E. Wenger. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, 1987.