

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS FÍSICAS E MATEMÁTICAS
CURSO DE MATEMÁTICA



MATRIX 97- UM SISTEMA PARA O ESTUDO DE MATRIZES

IVANETE ZUCHI

Monografia apresentada ao Curso de
Matemática, para obtenção do grau de
licenciatura em Matemática.

ORIENTADORA: PROF^A CLEIDE REGINA LENTZ PALADINI

Florianópolis
1997

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS FÍSICAS E MATEMÁTICAS
CURSO DE MATEMÁTICA



03738617

***MATRIX 97- UM SISTEMA PARA
O ESTUDO DE MATRIZES***

IVANETE ZUCHI

Monografia apresentada ao Curso de
Matemática, para obtenção do grau de
licenciatura em Matemática.

ORIENTADORA: PROF^ª CLEIDE REGINA LENTZ PALADINI

Florianópolis
1997

2 00907

Banca Examinadora:

Orientadora: Prof^a Cleide R. L. Paladini

Prof^a Mirian Buss Gonçalves

Prof. Samuel Aniceto Zacchi

**Aos meus Pais,
Zeferino e Anselma** que
mesmo distantes sempre
estiveram presentes em
meu coração.

Agradecimentos

À Professora **Cleide R.L. Paladini**, orientadora deste trabalho, pelo estímulo, perseverança, apoio e principalmente pela confiança depositada durante a realização deste trabalho.

A **todos os Professores** que contribuíram para minha formação profissional.

A minha grande amiga **Marivete Machado**, pelo incentivo e apoio incondicional durante o período de *graduação*.

Aos **meus familiares** pela confiança e estímulo em mim depositados.

As **amigas** que dividem, não somente um apto, mas que juntas compartilhamos ideais, o meu muito obrigado pelo incentivo e pela paciência.

Aos **amigos do curso**, o estudo deixou muito mais do que conhecimento, ficou a amizade conquistada.

A **Deus**, pela luz que sempre iluminou o meu caminho.

A **Banca Examinadora** pelas discussões, sugestões e correções do trabalho.

Ao **GEIAAM**(Grupo de Inteligência Artificial aplicada à Matemática), pelo ambiente de realização do trabalho.

Enfim, **a todos** que de uma maneira ou outra contribuíram e incentivaram esta jornada, o meu muito obrigado.

Sumário

1 Introdução	08
2 Informática e Educação	11
2.1. História do computador educacional	11
2.2 O computador e a criança	13
2.3 Quando usar	14
2.4 Como usar	15
3 Inteligência Artificial e Sistemas Especialistas	17
3.1 Inteligência Artificial	17
3.1.1 Histórico	17
3.2 Sistemas Especialistas	18
3.2.1 Comparação entre conhecimento especializado humano e artificial	19
3.2.2 Estrutura de um Sistema Especialista	19
4 Shell Kappa	22
4.1 Uso da shell Kappa	22
4.2 Estrutura da Shell Kappa	22
4.3 Tipos de raciocínios	23
4.3.1 Processos de domínio	24
5 Matrix 97-Um sistema para o estudo de matrizes	27
5.1 Introdução	27
5.2 Objetivos	28
5.2.1 Objetivos gerais	28
5.2.2. Objetivos específicos	28
5.3 Estrutura	29
5.4 Desenvolvimento	31
6 Uma Análise do Matrix 97	37
6.1 Desenvolvimento X Inteligência Artificial	37
6.2 Desenvolvimento X Sistemas Especialistas	41
6.3 Desenvolvimento X Shell Kappa	42
6.3.1 Descreve-se algumas regras	42
6.3.2 Descreve-se algumas funções	45
6.3.3 Descreve-se alguns métodos	50
6.4. Slots	53
6.4.1 Slots com valores numéricos	53
6.4.2 Slots com valores: texto	54
6.5 Interface da Shell Kappa	55
7 Conclusões	58
Anexos	59
8 Referências Bibliográficas	79

Resumo

Este trabalho apresenta o desenvolvimento do **Matrix 97**- Um sistema para o estudo de matrizes. Este sistema foi desenvolvido usando técnicas de Inteligência Artificial aplicadas ao ensino de Matemática. A ferramenta computacional para o desenvolvimento do mesmo foi a *Shell Kappa*. Um dos principais objetivos de seu desenvolvimento é o de proporcionar um ambiente motivador para a aprendizagem deste conteúdo.

Introdução

1. Introdução

Em 94.1 a autora do presente trabalho iniciou o curso de Licenciatura em Matemática e em agosto de 95 foi contemplada com uma bolsa de Iniciação Científica, através da qual integrou-se no grupo GEIAAM (Grupo de Inteligência Artificial Aplicada à Matemática) sob a orientação da Prof^a Cleide Regina Lentz Paladini, integrante do mesmo. Naquele momento o grupo não tinha um laboratório e então sempre se reunia em salas disponíveis do CFM onde eram realizados seminários sobre Inteligência Artificial, Sistemas Especialistas, Softwares Educacionais e o tão polêmico assunto Informática na Educação. Tudo era extremamente novo e ao mesmo tempo era um assunto fascinante, pois, estava sendo lançado o desafio: produzir.

A autora iniciou o projeto de pesquisa estudando Inteligência Artificial, Sistemas Especialistas, Softwares educacionais, propostas pedagógicas e a ferramenta utilizada para o desenvolvimento dos softwares: Shell Kappa. As referências, principalmente da Shell Kappa são todas em inglês, e esta era leiga...teve que trabalhar dobrado! No início de 96 ao grupo integrou-se um novo bolsista, aluno da Produção Mecânica, com o qual trabalhou mais diretamente. Tinham como objetivo desenvolver e implementar computacionalmente um software sobre semelhança de triângulos, esta ficou responsável pela estrutura do sistema e proposta pedagógica e ele com a implementação computacional. Trabalhavam no laboratório de Matemática e em seguida no laboratório obtido em parceria com o Reesc. Começou então a programar, como resultado foi desenvolvido o sistema *Tal 1.0*. O segundo trabalho foi realizado em parceria com outra bolsista do curso de Matemática. Desta vez, a bolsista desenvolveu a proposta pedagógica de um sistema sobre o Teorema de Tales e a autora a implementação computacional. Resultou o Sistema *Tales 1.0*. Estes trabalhos foram publicados no Cricte'96, Cricte'97, VI e VII Semanas de Pesquisa da UFSC, em 96 e 97 respectivamente, e na Fenasoft 97.

Como o trabalho de conclusão de curso deve ser aplicado no ensino de 1^o e 2^o graus, a autora propôs desenvolver um sistema Especialista para o Ensino de Matrizes: O **Matrix 97**. O seu desenvolvimento utilizaria recursos de I.A e todos os conhecimentos adquiridos durante o período de participação no GEIAAM/MTM.

Com o advento de novos recursos tecnológicos, abrem-se perspectivas que tornam mais eficientes as potencialidades do emprego da informática em várias áreas. Uma delas é a educação, que vem exigindo grandes investimentos em torno de inovações no ensino. Em particular, tem-se verificado que a Inteligência Artificial reúne um conjunto de ferramentas e estratégias que podem tornar extremamente atrativo e eficiente o processo ensino-aprendizagem. A Inteligência Artificial tem sido definida como a parte da ciência da computação que envolve o projeto de sistemas computacionais que exibem características da Inteligência humana, ou seja, são capazes de aprender novas informações, entender

Bowie, 1986). No desenvolvimento do sistema **Matrix 97** procurou-se, sempre que possível, aplicar estes recursos.

Uma das idéias que motivou este sistema é a possibilidade de que ele venha a constituir-se em uma ferramenta auxiliar para a aprendizagem deste conteúdo. Com efeito, uma das principais características dos sistemas especialistas refere-se ao fato de que o sistema fornece explicações ao usuário sobre o que está sendo feito e porque se está fazendo, criando-se, assim, efetiva interação sistema/usuário. Esta especificidade diferencia os sistemas especialistas dos demais programas computacionais ou de outros sistemas de informações e pode-se empregá-la, com vantagens, no processo ensino-aprendizagem. Desta forma este sistema especialista, tem como objetivo auxiliar os alunos do segundo grau na aprendizagem deste conteúdo específico de Matemática. Sua estrutura envolve três elementos básicos: alguns dados históricos do processo, desenvolvimento e operações entre matrizes. O apoio computacional requerido pelo sistema inclui a *shell* KAPPA e desenvolve-se numa interface clara, atraente e amigável ao usuário.

Informática e Educação

2. Informática e Educação

2.1. História do Computador Educacional

- Década de quarenta

- O mundo estava em guerra.
- Cálculos complexos tinham que ser feitos sob a pressão de tempo e com máxima precisão possível, para que fossem criadas poderosas armas ou para que fossem descobertos códigos secretos do lado inimigo.

Foi neste contexto nada harmonioso que tiveram origem os computadores enormes que faziam cálculos, onde as entradas de dados eram feitas com cartões perfurados um a um. Estas máquinas foram projetadas por matemáticos que, com certeza, não imaginavam que estas tornar-se-iam instrumentos de pessoas comuns que as utilizariam para diversos fins.

- Após a Guerra

- O computador deixou de ser privilégio de alta ciência e do exército;
- Entrou num mundo mais amplo dos negócios, da pesquisa industrial e universitária.

- Início da década de sessenta

- Surgiu a idéia de usar o computador também na educação, apesar da tecnologia não ser muito desenvolvida naquela época.
- A interação com o aluno era muito diferente do que temos hoje, sem desenhos, cores, sons e ações.
- Foi desenvolvido o sistema LOGO, que até hoje é considerado um modelo de *software* educacional. Desenvolvido no MIT, o Instituto de Tecnologia de Massachussets, por Seymour Papert, um matemático que sempre esteve preocupado com a maneira pela qual as pessoas aprendem, já havendo, inclusive, estudado com Piaget na década de cinquenta.

Papert em contato com os computadores no MIT começou a imaginar como poderia “roubar a tecnologia dos laboratórios para dá-las às crianças”.

- Outra linha de desenvolvimento de sistemas para educação foi seguida por Patrick Suppes, o mentor dos programas de exercícios CAI, *Computer Aided Instruction*. Estes programas, dotados de gráficos dinâmicos, basicamente eram programas de perguntas e respostas, sendo que o computador dava ao estudante uma pergunta e este lhe dava uma resposta, cabendo ao computador retornar respondendo se esta estava correta ou não.

- Outro participante da cultura educacional da informática foi John Kemeny um dos criadores do BASIC. Ele via os estudantes como programadores de computador, tornando este último uma ferramenta que auxilia a aprendizagem ao invés de um protótipo do professor que auxilia a instrução.

Não é relevante aqui a defesa de qualquer um destes sistemas. Observa-se que a conclusão da pesquisa realizada por Bossuet, em meados na década de oitenta, em uma escola de Aiguelong-França, as crianças se vêem motivadas a utilizar qualquer um destes sistemas informáticos. E, nesta mesma linha, Chaves argumenta que qualquer forma adequada de utilização do computador na educação(CAI, LOGO, BASIC, Aplicativos, Jogos e Simulações) pode trazer bons resultados pedagógicos.

- Início dos anos oitenta

Com o advento do microcomputador, o número de pessoas que se dedicavam ao estudo do uso dos computadores na educação chegou a milhões, principalmente professores interessados em aplicar esta nova tecnologia em sala de aula. Para tanto, foi necessário determinar algumas diretrizes que norteassem quando e de que forma utilizá-las.

2.2. O computador e a Criança

A questão do uso dos computadores na escola é de suma importância atualmente. Apresentamos neste contexto motivos pelos quais os computadores, estão cada vez mais invadindo lares e escola, trazendo mudanças na vida das crianças e também no currículo escolar que deve ser adaptado para tornar o uso deste poderoso meio de comunicação o mais proveitoso possível no processo ensino-aprendizagem:

- As crianças parecem particularmente fascinadas por esta tecnologia e estão sempre dispostas a usá-las;
- Os pais ficam ansiosos se seus filhos não usam esta poderosa ferramenta, achando que eles não estão se preparando adequadamente para a vida futura;
- As escolas, mesmo numa época de orçamento limitado, vem comprando computadores, muitas sem a preocupação de como usá-lo em sala de aula.

Uma interpretação possível para o fascínio que as crianças têm pelo computador, seria por causa do dinamismo visual deste, adicionado a uma participação ativa delas junto ao mesmo.

Dean Brown, pioneiro no desenvolvimento da tecnologia para computadores, denominou o computador como a mais espantosa invenção devido a sua combinação excepcional de características: “ele é dinâmico, interativo e programável”.

Mas apesar de todas as vantagens que estas ferramentas computacionais podem proporcionar, algumas restrições devem ser consideradas quanto ao seu uso.

Além de saberem usar os programas que o computador oferece, todos os jovens deveriam sair da escola com um conhecimento básico sobre: a estrutura e funcionamento destas máquinas.

Sem este conhecimento básico, torna-se difícil manter o controle de uso das mesmas. As máquinas estão ficando cada vez mais complexas e com isso as pessoas passam a ficar inertes frente à incompreensão de seu funcionamento. O papel da escola neste caso é o de ensinar o princípio de funcionamento destas máquinas para que as mesmas possam ser usadas criticamente.

2.3. Quando usar

De acordo com os ideais de Piaget, a realidade apresenta-se a cada um de forma diferente, de acordo com o nível de desenvolvimento do indivíduo. Um adulto e uma criança observando uma mesma situação, assimilarão aspectos diferentes dado suas diferenças quanto à estrutura mental.

Como caracterizar então, o desenvolvimento intelectual do indivíduo? Para Piaget, este pode ser dividido em três grandes épocas que coincidem aproximadamente com os três períodos escolares:

- **A primeira época, chamada “estágio sensório-motor”**, se caracteriza pelo imediatismo das crianças em responder perguntas, coincidindo com o período pré escolar.
- **A segunda época, chamada de “estágio das operações concretas”**, corresponde ao primeiro grau e é o período onde a criança desenvolve a lógica concreta.
- **Finalmente, há o “estágio formal”** que cobre o segundo grau e o resto da vida, onde o pensamento é dirigido por princípios de lógica, dedução e indução.

Papert propôs ensinar programação às crianças antes mesmo de elas atingirem o último estágio, mais especificamente após os quatro anos de idade. Para ele, com esta experiência, o desenvolvimento cognitivo delas seria facilmente alcançado, acelerando assim a passagem do pensamento infantil para o pensamento adulto lógico-formal.

Acredita-se que o computador possa ser utilizado, para fins de aprendizagem, por qualquer indivíduo, desde que a metodologia para esta prática seja adequada a ele, isto é, apresente-se de tal forma, que considere sua capacidade cognitiva bem como o desenvolvimento de suas estruturas mentais.

2.4. Como Usar

O que deve ser ressaltado aqui é que para a obtenção de bons resultados com a utilização de computadores na educação, deve-se ter definido:

- A metodologia a ser utilizada;
- Quais os objetivos se pretende alcançar e
- Qual será o papel do professor frente ao uso do computador pelos alunos.

Uma dificuldade encontrada é que muitas instituições de ensino tem insistido em ensinar computação como uma disciplina estanque, não enfocando a multidisciplinariedade dos conteúdos a serem estudados que proporcionaria uma visão geral do conhecimento.

Quanto ao papel do professor, Papert diz que este deve dar apoio ao educando para que este construa suas estruturas cognitivas; evidentemente, sem o currículo tradicional e sem a falsa visão de que o professor é o único detentor do conhecimento. Este apoio se resumiria a não dar resposta e sim estimular o aluno a pensar.

Para que se passe a usar a informática na educação, é importante colocar os equipamentos correspondentes à disposição dos professores e alunos. Não importa aqui a quantidade de recursos disponíveis, o que importa é que os conceitos e as técnicas específicas desta nova tecnologia sejam acessíveis aos futuros usuários dessas ferramentas. Para que isto seja possível, tanto os profissionais da informática quanto as escolas de formação de professores devem mudar suas concepções e se adaptar.

Como afirmou Paulo Freire “Os homens se educam entre si mediatizados pelo mundo”. E não é este o objetivo que se pretende atingir com o uso do computador na educação? Que os indivíduos interajam ao mesmo tempo entre si e com a máquina, tornando-a mais proveitosa possível para o seu aprendizado individual? E isto é também o que pensa Piaget sobre o conhecimento, quando diz que este é construído pela interação com os objetos e pessoas do ambiente em que se vive, ou ainda quando diz que “é adaptando-se às coisas que o pensamento se organiza e é organizando-se que se estruturam as coisas”.

Inteligência Artificial e Sistemas Especialistas

3. Inteligência Artificial e Sistemas Especialistas

3.1. Inteligência Artificial

3.1.1. Histórico

As correntes de pensamento que se cristalizaram em torno da IA já estavam em gestação desde os anos 30. No entanto, oficialmente a IA nasceu em 1956 com uma conferência de verão de, Dartmouth College USA. Na proposta desta conferência, escrita por John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Shannon e submetida a fundação Rockefeller, consta a intenção dos autores em realizar “um estudo durante dois meses, sobre o tópico *inteligência artificial*”. Ao que tudo indica, esta parece ser a primeira menção oficial à expressão “*Inteligência Artificial*”.

Desde os seus primórdios a IA gerou polêmica, a começar pelo seu próprio nome, considerado presunçoso por alguns, até quando da definição de seus objetivos e metodologias. O desconhecimento dos princípios que fundamentaram inteligência, por um lado, e dos limites práticos da capacidade de processamento de computadores, por outro lado, levou periodicamente a promessas exageradas e às correspondentes decepções.

Dada a impossibilidade de uma definição formal precisa para a IA, visto que para tanto seria necessário definir, primeiramente, a própria Inteligência, foram propostas algumas definições operacionais:

- “Uma máquina é inteligente se ela é capaz de solucionar uma classe de problemas que requerem inteligência para serem solucionados por seres humanos”.
- “Inteligência Artificial é a parte da ciência da computação que compreende o projeto de sistemas computacionais que exibam características associadas, quando presentes no comportamento humano, à inteligência”.
- Ou ainda “Inteligência Artificial é o estudo das faculdades mentais através do uso de modelo computacionais”.
- Outros se recusam a propor uma definição para o termo e preferem estabelecer os objetivos da IA: “tornar os computadores mais úteis e compreender os princípios que tornam a inteligência possível”.

Existem duas linhas principais de pesquisa para a construção de sistemas inteligentes:

- A linha Conexionista
- A linha Simbólica.

A linha conexionista visa a modelagem da inteligência humana através da simulação dos componentes do cérebro, isto é, de seus neurônios e de suas integrações. Esta proposta foi formalizada inicialmente em 1943, quando o neuropsicólogo McCulloch e o lógico Pitts propuseram um primeiro modelo matemático para um neurônio. Um primeiro modelo da Rede Neuronal, isto é, um conjunto de neurônios interligados, foi proposto por Rosenblatt. Este modelo era chamado de Perceptron e possui muitas limitações do ponto de vista das propriedades matemáticas de redes artificiais de neurônios. Durante muito tempo esta linha de pesquisa não foi muito ativa, mas o advento dos microprocessadores pequenos e baratos tornou praticável a implementação de máquinas de conexão composta de milhares de microprocessadores, o que, aliado à solução de alguns problemas teóricos importantes, deu um novo impulso às pesquisas na área. O modelo conexionista deu origem à área de redes neurais.

A linha simbólica segue a tradição lógica e os princípios desta linha de pesquisa são apresentados no artigo "Physical Symbol Systems" de Newell. O sucesso dos Sistemas Especialistas (SE), *Expert Systems*, a partir da década de 70, estabeleceu a manipulação simbólica de um grande número de fatos especializados sobre um domínio restrito como o *paradigma corrente para a construção de sistemas inteligentes do tipo simbólico*.

3.2. Sistemas Especialistas

Sistemas Especialistas, sistemas baseados em conhecimento são aplicações de engenharia do conhecimento, uma das três subespecialidades do extenso domínio da Inteligência Artificial. As outras duas incluem robótica e consulta em linguagem natural. Segundo analogia de J. Kaplan, da *Tecknowledge Inc.*, "Inteligência Artificial é engenharia de conhecimento explicada pela Matemática.

Sistemas especialistas são frequentemente empregados como um auxiliar inteligente ou consultor para usuários humanos. Podem ter usos na resolução de problemas de rotina, liberando os peritos para problemas não usuais. Eles podem também levar conhecimento especializados a localizações onde os peritos não estão disponíveis ou torná-las acessíveis quando os serviços especializados sejam dispendiosos.

Diferente dos programas convencionais, os sistemas especialistas podem resolver problemas que exijam julgamentos semelhantes aos que as pessoas utilizam em seus trabalhos diários. Um sistema especialista pode oferecer explicações para as suas conclusões.

Embora sistemas especialistas e peritos reais possam em alguns casos desempenhar tarefas idênticas, as características de ambos são criticamente diversas. Mesmo havendo

algumas vantagens evidentes dos sistemas especialistas, eles não poderão substituir os peritos em todas as situações devido a algumas limitações inerentes. A tabela a seguir resume esta comparação.

3.2.1. Comparação entre conhecimento especializado humano e Artificial

C. HUMANO	C. ARTIFICIAL
Perecível	Permanente
Difícil de transferir	Fácil de transferir
Difícil de documentar	Fácil de documentar
Imprevisível	Consistente
Caro	Razoável
Discriminatório	Imparcial
Individualizado	Social
Criativo	Sem inspiração
Adaptável	Inflexível
Enfoque amplo	Enfoque restrito
Senso comum	Técnico

3.2.2. Estrutura de um Sistema Especialista

Um Sistema Especialista (SE) apresenta em geral uma arquitetura com três módulos:

- Uma base de conhecimento;
- Uma memória de trabalho e
- Um Motor de Inferência.

A base de regras e a memória de trabalho formam a base de conhecimento do SE, onde está representado o conhecimento sobre o domínio.

O motor de inferência é o mecanismo de controle do sistema que avalia e aplica regras de acordo com as informações da memória de trabalho.

A memória de trabalho deve respeitar um Método de Representação de Conhecimento, isto é, uma linguagem formal e uma descrição matemática de seu significado. A lógica de primeira ordem é um exemplo típico de formalismo de representação de conhecimento.

A base de regras passa a conter condições que representam “perguntas” a representação de conhecimento da memória do trabalho. Estas perguntas podem ser de diferentes tipos, mas em geral envolvem variáveis a serem instanciadas e eventualmente alguns tipos de inferência.

Outra característica comum nos SE's atuais é a existência de um mecanismo de raciocínio incerto que permita representar a incerteza a respeito do conhecimento do domínio.

O motor de inferência controla a atividade do sistema. Esta atividade ocorre em ciclos, cada ciclo consistindo em três fases:

- Correspondência de dados, onde as regras que satisfazem a descrição da situação atual são selecionadas.
- Resolução de conflito, onde as regras que serão realmente executadas são escolhidas, entre as regras que foram selecionadas na primeira fase, e ordenadas.
- Ação a execução propriamente dita das regras.

Para a construção de um SE, em qualquer tipo de software, a interface com o usuário final é fundamental para o sucesso do mesmo. Em particular, um SE, além de apresentar uma interface ergonomicamente bem projetada, deve ainda levar em conta o grau de familiarização do usuário com o domínio de trabalho do sistema e com os computadores em geral.

Algumas técnicas tornam a interface mais amigável, por exemplo: o uso de janelas, menus e gráficos.

Shell Kappa

4. Shell Kappa

O sistema de desenvolvimento de aplicações, KAPPA, é uma concha para construir *softwares* que simulam sistemas complexos, além de propiciar facilidades para representar o conhecimento sobre estes sistemas.

4.1. Uso da *Shell* KAPPA

- Construir Sistemas Especialistas baseados no conhecimento necessário para entender a complexidade de certos campos de ação.

Ao estruturar um Sistema Especialista, pretende-se simular um modelo, que necessita de um especialista, o mais real possível.

Com estes sistemas, os computadores podem incorporar:

- O conhecimento de *expert* humanos para resolver problemas específicos.
- Quantidade significativa de conhecimento que pode ser armazenada diretamente nestes sistemas, a máquina então adquire habilidade para raciocinar além do conhecimento diretamente armazenado.

Um bom candidato à base de conhecimento no Kappa é um sistema onde sabe-se quais são seus componentes, como comportam-se e interagem.

4.2. Estrutura da *Shell* Kappa

A *Shell* KAPPA proporciona muitas ferramentas para construir e também usar um Sistema Especialista. Diferente de muitas ferramentas chamadas “espertos” (*expert systems tool*), KAPPA, proporciona muito mais que regras. Nela as componentes do domínio do Sistema são apresentados por estruturas chamadas objetos.

Os objetos podem ser:

- Classes ou
- Instâncias.

Eles podem representar conceitos ou algo concreto. Uma classe é um objeto mais geral, enquanto uma instância é um item mais específico. Uma subclasse é um conjunto de uma classe, sendo que as instâncias não possuem subclasses.

As relações entre os objetos podem ser representadas encadeando-os em uma estrutura chamada hierarquia. Parente é o nome da classe que está diretamente acima da subclasse na hierarquia. Antecessor e descendente descrevem indiretamente as relações entre classes, subclasses e instâncias.

Cada objeto, classe ou instância, possui *slots*, que são entidades que descrevem atributos deste objeto, bem como uma coleção de objetos. Eles adicionam detalhes, estruturas, listas de atributos e propriedades. Cada uma destas descrições é representada por um *slot* e pelo valor de um *slot* que pode ser um número, um texto, um valor booleano (verdadeiro ou falso) ou até mesmo um objeto.

Pode-se assim especificar as propriedades gerais apenas uma vez em uma classe apropriada, já que os *slots* podem ser herdados. Isso proporciona vantagens, pois a base de conhecimento torna-se mais fácil de ser criada e manipulada, além de que o conhecimento é mais facilmente armazenado, pois as propriedades gerais são declaradas em apenas uma classe e aplicadas a todos os seus descendentes.

Usando as opções de *slot(slot option)*, pode-se proporcionar informações mais detalhadas sobre os objetos e, portanto, criar uma aplicação mais precisa. Pode-se designar limites para os tipos de valores que o usuário poderá entrar. Pode-se especificar o valor de um objeto, como sendo o valor de outro objeto.

Usando a hereditariedade, pode-se assumir os valores dos objetos abaixo, na hierarquia, mais rapidamente e facilmente. Fazer mudanças nesses valores, também torna-se muito mais simples com o esquema de hierarquia que a KAPPA proporciona.

4.3. Tipos de raciocínios

O raciocínio usado para resolver os problemas propostos pode ser:

- para frente (*forward*) ou
- para trás (*backward*).

No raciocínio para frente o Sistema vai testando as regras selecionadas e conforme forem satisfeitas, vai produzindo os resultados. Neste caso não é necessário o estabelecimento de uma meta(*goal*). Este tipo de raciocínio é usado para determinar conseqüências de um novo fato. O raciocínio para trás, por sua vez, exige o estabelecimento de uma meta, e ele é bastante usado para responder questões propostas.

Na *Shell* KAPPA, a estrutura do domínio é representada criando objetos que se correspondem com os objetos (conceituais ou concretos) do domínio.

4.3.1. Processos de domínio

Os processos do domínio são representados em três diferentes maneiras:

- métodos;
- funções e
- regras de produção.

Todas elas escritas em linguagem *KAL*, que é a linguagem própria de programação da *Shell* KAPPA.

Os métodos especificam como um objeto pode se comportar. Esta técnica de armazenar o comportamento de um objeto, é um dos atributos que fazem parte da técnica de programação chamada Programação orientada ao objeto.

A segunda aproximação da representação dos processos em KAPPA envolve funções. KAPPA fornece uma biblioteca com 240 funções que ajudam a manipular a base de conhecimento. As funções estendem um simples operador numérico a uma função lógica. Usando *Kal*, pode-se construir o que se necessita.

A terceira aproximação envolve regras para representar as etapas do processo. Uma regra especifica as condições que devem ser satisfeitas para que a inferência seja aplicada. Elas podem especificar interações complexas entre os componentes do sistema.

O raciocínio baseado em regras é usado para resolver muitos problemas que possuem afirmações condicionadas. As regras possuem a forma *if-then*(se-então), onde na parte *if* estão contidas as premissas e na parte *then* as conclusões. Elas são manipuladas por uma máquina de inferência que liga premissas de uma regra com conclusões de outra. São usadas variáveis, para as regras mais geralmente aplicadas. Estas variáveis podem ser criadas juntamente com as regras, ou podem vir de objetos e *slots* que estão sendo aplicados. A sintaxe para métodos, funções e regras é idêntica.

Além destas ferramentas, KAPPA proporciona uma rica e poderosa interface para o desenvolvimento das aplicações. Esta interface consiste em ferramentas para a visualização e modificação de vários elementos em KAPPA. O sistema deve dar todos os recursos necessários para que o usuário, ou programador, entre com os dados, formule questões e dê respostas. A interface com o usuário é a parte do sistema que administra esta troca de informações.

Há duas razões diferentes para usar a interface com o usuário:

- A primeira é para o programador que constrói a aplicação. As ferramentas são proporcionadas em KAPPA para o desenvolvimento das aplicações. Nessas interfaces, a linguagem KAL proporciona um modo conveniente de modificar a base de conhecimento e fazer perguntas sobre ela.
- A segunda razão para usar a interface KAPPA é que ela auxilia ao usuário do sistema, no alcance de suas metas. KAPPA torna possível que o programador construa estas interfaces que dão ao usuário fácil acesso a aplicação completa, ou mostrando somente os detalhes que interessam ao usuário.

Como a *shell* KAPPA trabalha em ambiente *Windows*, a janela é componente básica da interface KAPPA. Essas janelas são áreas de separação onde aparecem textos ou gravuras. O menu é uma janela especial que contém a lista de opções que podem ser escolhidas. Estas opções aparecem na forma de textos ou gravuras chamadas *ícones*. É possível criar gráficos usando Imagens na janela *Session*. Há quinze diferentes tipos de imagens.

Como falou-se anteriormente, a linguagem KAL é um caminho rápido para acessar a capacidade da *shell*, usando a interface. Ela ajuda a adicionar novas informações na base de conhecimento. Pode-se usá-la para criar ou remover um objeto ou *slot*, criar ou remover uma ligação entre dois objetos, etc. Pode ajudar também a fazer perguntas sobre o armazenamento de informações na base de conhecimento.

KAPPA proporciona um mecanismo para definir expressões similares ao da programação com linguagem clássica. Estas expressões devem seguir a forma da linguagem KAL. Pode-se escrever e entrar diretamente no KAL *interpreter* com uma expressão escrita na linguagem KAL.

O tipo de programação que KAPPA utiliza (Programação orientada ao objeto), permite a associação de comportamentos com objetos, através do armazenamento de métodos no próprio objeto. Ela também prevê uma interface uniforme que será disparada a diferentes objetos, com diferentes comportamentos. Esta programação é mais eficiente do que regras para expressar procedimentos definidos.

Os métodos, também podem ser herdados, como podem também ser adicionados métodos locais em qualquer etapa da programação. Os métodos são ativados pelo envio de mensagens aos objetos, usando funções com *SendMessage*.

Estas são algumas das características e propriedades da *shell* KAPPA.

**Matrix 97- Um sistema para o
estudo de Matrizes**

5. Matrix 97- Um sistema para o estudo de Matrizes

5.1 Introdução

O problema fundamental do ensino da matemática hoje, não se reduz à simples questão : “*Como Ensinar técnicas ao nosso aluno*”. Queremos que o aluno adquira *conhecimento* subsidiado pela *compreensão e construção* do conhecimento matemático.

O conhecimento é resultado da *compreensão-vivência-resolução* de situações problemas.

Com a proposta de dar apoio à prática de sala de aula , com um roteiro, ao nosso ver, *adequado* `a realidade do conteúdo com uma **linguagem** simples e acessível, foi desenvolvido o sistema “ **Matrix 97**”, um material didático que pretende contribuir no processo de construção do conhecimento matemático referente ao ensino do conteúdo de Matrizes.

Segundo Piaget o pensamento formal desabrocha na adolescência. Matematicamente, isto significa que o adolescente está apto a raciocinar logicamente. No entanto observa-se que uma das grandes dificuldades apresentadas pelos alunos, refere-se à falta de raciocínio lógico-dedutivo que lhes permita identificar uma situação matemática e o que é necessário para solucioná-la. Assim, o desenvolvimento e utilização de pequenos sistemas utilizando técnicas de Inteligência Artificial pode constituir-se em importante apoio para a aprendizagem dos conteúdos de Matemática.

Para justificar este estudo apresenta-se aqui o Matrix 97 - um sistema para o estudo de Matrizes. Especifica-se sua estrutura, as fases de montagem da base de conhecimento e regras de produção do sistema , bem como o enfoque matemático e toda a sua interface.

5.2. Objetivos

5.2.1. Objetivos Gerais

Aplicar técnicas de Inteligência Artificial no ensino de Matemática;

5.2.2 Objetivos Específicos

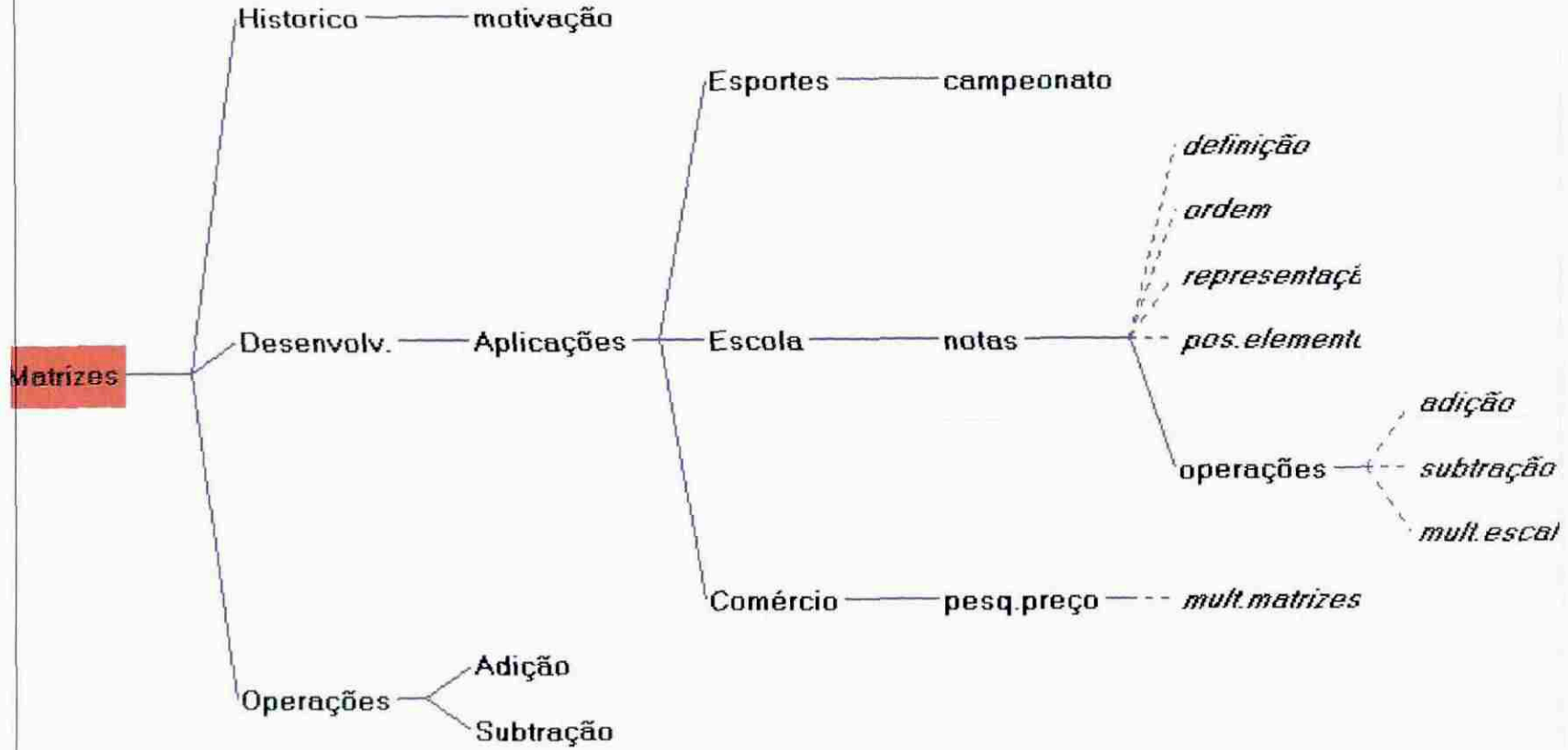
- Proporcionar um ambiente computacional para o ensino de Matrizes;
- Promover a aprendizagem deste conteúdo ;
- Motivar o aluno na compreensão deste conteúdo.

5.3. Estrutura

Apresenta-se, a seguir, uma estrutura arborescente do sistema.

Estrutura do Sistema Matrix 97

Matrix 97- Um sistema para o estudo de Matrizes



5.4. Desenvolvimento

Utilizando tecnologia como mais um meio eficiente e motivador pretende-se fazer com que o aluno comece a construir a base de seu conhecimento relacionado a um conteúdo de Matemática, especificamente sobre Matrizes.

O *Matrix 97* é um sistema Especialista que proporciona ao usuário conviver na prática com exemplos concretos de aplicações de matrizes e a partir destes pretende-se que o aluno desenvolva seu conhecimento sobre este conteúdo.

Desta forma, **O Matrix 97** foi estruturado da seguinte maneira:

- Histórico
- Definições
- Operações

A seguir apresenta-se descrições destas opções com as telas referentes do sistema (as telas são apresentadas em anexo)

O sistema apresenta ao usuário as opções:

- Histórico
- Definições
- Operações

Aqui sugere-se ao usuário que navegue por todo o sistema. [ver tela 2.1]



Na parte histórico o sistema apresenta alguns dados históricos referente a Matrizes. [ver tela 3].



Ao optar por definições, o **Matrix 97** apresenta duas situações problemas:

- Campeonato Brasileiro
- Médias escolares

Para o campeonato brasileiro o usuário é convidado a montar seu próprio campeonato, ou seja ele escolhe os times finalistas [ver tela 4]

Caso sejam selecionados times iguais o sistema alerta que sejam diferentes.

O usuário confirmará ou poderá mudar seus times [ver tela 5]

Caso o usuário opte por corrigir o sistema volta à tela anterior reiniciando. Caso contrário, apresentará os dados selecionados.

Então o usuário checará todos os possíveis jogos dos dois turnos, e o sistema solicita o placar de cada jogo. [ver tela 6]

Depois de ter decidido o placar do primeiro e do segundo turno o sistema proporciona ao usuário, ver na tela a tabela do placar dos dois turnos. [ver tela 7].

Então o sistema solicita ao usuário que forneça os dados referentes a:

- gols marcados (GM)
- gols sofridos (GS)
- saldo de gols (SG)
- a pontuação de cada time (PG)

Estes dados são solicitados para os quatros times finalistas. [ver tela 8].

Em 6.1 [1] Observa-se o que acontece se estes dados estiverem errados.

Uma tabela é então apresentada, com estes resultados. [ver tela 9].

A partir deste instante pretende-se que o aluno comece a visualizar a tabela numérica, bem como o número de linhas, de colunas e posição de elementos. [ver tela 10].

Observe-se em 6.1 [2] o que acontece se a entrada de dados estiver errada!

Para fixar a posição de cada elemento por linha e coluna, o sistema possibilita ao usuário checar na tabela os elementos correspondentes, e estes simultaneamente vão aparecendo na linha e coluna referente. [ver tela 11].

Após ter exercitado a posição de cada elemento é então, generalizado este exemplo. [ver tela 12].

A representação dos dados é realizada em uma tabela numérica e depois generalizada através de uma matriz a_{ij} . [ver tela 13].

Na situação médias escolares, o sistema propõe ao usuário selecionar as matérias que ele deseja analisar: [ver tela 14].

O aluno após a escolha pode confirmar ou corrigir.

Se ele optar por corrigir voltará a tela anterior, caso contrário o sistema solicita ao usuário que forneça os dados referente a cada matéria. [ver tela 15].

Esta entrada de dados é solicitada para todas as matérias selecionadas pelo usuário. O usuário determinará as notas para cada matéria. [ver tela 16].

Os dados referentes as matérias, são dispostos em uma tabela. Observe que neste exemplo pretende-se que o aluno visualize que o número de linhas é diferente do número de colunas. [ver tela 17].

Aqui, o aluno reforça a localização de cada elemento, agora com o usuário conhecendo a idéia de elemento genérico a_{ij} . [ver tela 18].

Em caso de erro o sistema aciona 6.1 [3] .

A cada entrada de dados os elementos são dispostos em um quadro na sua respectiva localização. [ver tela 19].

Neste momento são exploradas situações que evidenciam a diferença entre o número de linhas e colunas. Após estes exercícios é então apresentado um conceito de Matriz. [ver tela 20].

E também algumas notações contextuais. [ver tela 21].

À partir deste momento o sistema começa a explorar operações com matrizes. [ver tela 22].

Assim, o sistema solicita ao usuário que calcule o total de pontos obtidos (TP), a nota necessária no 4^o bimestre (FP) e a média já atingida (média). [ver tela 23].

Esta entrada de dados é solicitada para cada matéria selecionada. [ver tela 24].

Depois do aluno ter calculado as médias escolares, então uma tabela de referência é visualizada.

Neste contexto pretende-se que o usuário comece a estruturar operações entre matrizes à partir de exemplos concretos, neste caso, notas escolares. [ver tela 25].

Através do total de pontos obtidos nos três bimestres, explora-se a idéia da soma de matrizes. Com a determinação da nota necessária no quarto bimestre, para o aluno ser aprovado sem recuperação, explora-se a idéia de subtração de matrizes.

Através da média obtida nos três bimestres, explora-se a idéia de multiplicação por escalar. [ver tela 26].

O sistema interage, de forma similar analisando o campeonato brasileiro selecionado pelo usuário.

Após estudar cada exemplo, o sistema então, apresenta uma definição para operar matrizes através das tabelas construídas. São então desenvolvidas definições para cada operação com o usuário podendo retornar aos exemplos desenvolvidos com o objetivo de uma melhor visualização do conteúdo em desenvolvimento.

São então apresentadas uma definição de: soma, diferença e multiplicação por escalar. [ver tela 27].



Na parte de operações o usuário pode realizar operações como adição e subtração entre duas matrizes. Sendo que a ordem destas está num contexto limitado. É explicado neste contexto, a soma, a diferença e o produto por escalar. [ver tela 28].

A limitação da ordem da matriz é devido a limitações da *shell* Kappa.

O sistema sempre solicita a ordem das matrizes A e B. São exploradas mais uma vez soma, diferença e multiplicação por escalar. [ver tela 29].

Observe-se em 6.1 [4] o que acontece se a ordem das matrizes forem distintas.

A entrada de elementos das matrizes A e B é feita por colunas. [ver tela 30].

Após a entrada dos elementos são então visualizadas as matrizes A, B e A+B. O mesmo processo acontece com a subtração de matrizes.

Uma Análise do Matrix 97

6. Uma Análise do desenvolvimento do Matrix 97.

A seguir, faz-se uma breve análise do desenvolvimento do Matrix 97 identificando neste, algumas características de Inteligência Artificial e de Sistemas Especialistas; e também apresenta-se alguns dos recursos da *Shell Kappa* nele utilizados.

6.1. Desenvolvimento X Inteligência Artificial

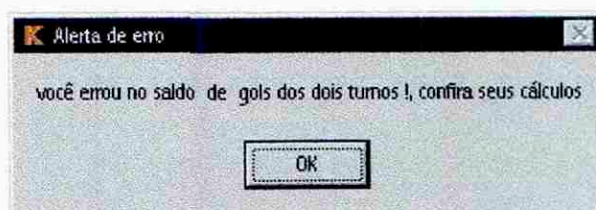
Na entrada de dados referentes a: gols sofridos, gols marcados, saldo de gols e pontos obtidos pelos times finalistas, o sistema Matrix 97 apresenta conhecimento, por exemplo, para controle de erros. Isto é, quando o usuário entrar com dados não corretos o sistema checa cada dado e alerta o usuário. Esta situação é observada em:

[1]

The screenshot shows a data entry form for the team 'Inter'. It contains four input fields with the following values: 'gols sofridos' (goals conceded) is 8, 'gols marcados' (goals scored) is 4, 'saldo de gols' (goal difference) is -3, and 'pontos ganhos' (points earned) is 3. At the bottom of the form are two buttons: 'OK' and 'Reset'.

Caso ocorra erro o sistema retorna a tela onde o mesmo ocorreu, deixando os dados corretos e apagando neste caso, o saldo de gols do Inter que está calculado de maneira errada. E assim o sistema checa todos os outros times, item por item, enviando alerta de erro caso houver.

Aqui o aluno cometeu um erro no saldo de gols. O sistema alerta o erro e volta novamente a entrada de dados para que o usuário possa digitar o saldo de gols correto.



[2] A posição dos elementos também é checada uma a uma.

Size labels				
GS	GM	SG	PG	Times
8	7	1	6	Palmeiras
8	4	-4	3	Inter
2	6	4	12	Grêmio
5	6	1	9	Cruzeira

col:1	col:2	col:3	col:4	
8	7		6	linha:1
	4			linha:2
		4		linha:3
5			9	linha:4

Posição do elemento

Qual é o elemento que se encontra na lin:4 e col:3

Responda!



Caso o usuário apresente algum elemento errado o sistema ativará a seguinte mensagem: “confira seus dados”. Este raciocínio é usado para todos os elementos.

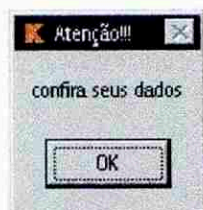
[3] Referente a posição das notas escolares. Neste caso o mesmo raciocínio é utilizado.

8 9
5 6
 6
9 10

Notas Bimestrais			
Bim: 1	Bim: 2	Bim: 3	matérias
8	7	9	Sociologia
5	6	8	Química
5	7	6	Matemática
9	8	10	Português

Responda

Qual é o valor do a₃₂?



Enquanto o elemento não estiver correto o sistema continua enviando a mensagem de alerta.

[4] Se a ordem da matriz A for diferente da ordem da matriz B.

Responda:

Número de linhas da matriz A:

Número de colunas da matriz A:

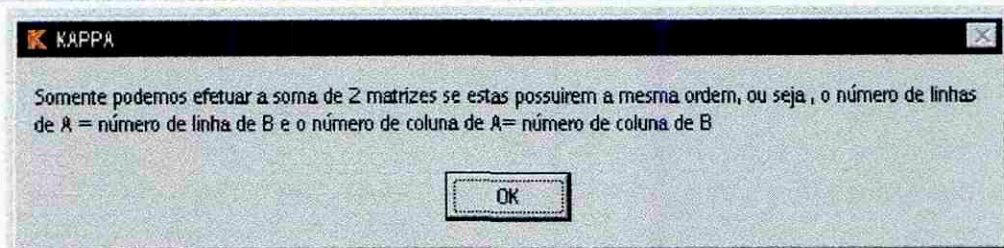
O usuário dá entrada da ordem da matriz A.

Responda:

Número de linhas da matriz B:

Número de colunas da matriz B:

Entrada de dados da ordem da matriz B.



6.2. Desenvolvimento X Sistemas Especialistas:

O Matrix 97, possui algumas características de um sistema especialista. Uma base de regras foi estruturada para o sistema comunicar-se com o usuário. Seguem algumas destas:

Para o sistema apresentar 6.1 [1] a regra ativada é a seguinte:

“Se a entrada de dados feita pelo usuário for diferente do saldo de gols correto do time 2 então envie mensagem: “você errou no saldo de gols, confira seus cálculos”.

Observe-se que a linguagem utilizada para descrever as regras não é a que foi utilizada na estrutura de regras dentro do sistema, para ver estas regras dentro da linguagem *Kal* pode se verificar (1) em 6.3.

Para o sistema apresentar 6.1 [4] a regra ativada é a seguinte:

“Se o número de linhas da matriz A for diferente do número de linhas da matriz B ou Se o número de colunas da matriz A for diferente do número de colunas da matriz B então envie mensagem: “Somente podemos efetuar a soma de 2 matrizes se estas possuírem a mesma ordem, ou seja , o número de linhas de A = número de linha de B e o número de coluna de A= número de coluna de B”.

“ Se a entrada de dados feita pelo usuário for diferente da média correta da matéria 1 então envie mensagem: “confira seus cálculos”.

Para ver esta mesma regra no contexto da linguagem *Kal* pode-se verificar (2) em 6.3.

Detalhes sobre a implementação dos itens 6.1 e 6.2 são dados a seguir.

6.3. Desenvolvimento X Shell Kappa

6.3.1. Apresenta-se algumas regras na linguagem *Kal*, conforme estão programadas no Matrix 97

```

*****
(1)      ****  RULE: GSC1
*****
MakeRule( GSC1, [],
  Matrizes:GS1 != Matrizes:GSC1,
  {
  SetPostMessageTitle( "Alerta de erro!!" );
  PostMessage( "você errou nos gols sofridos!, confira seus cálculos" );
  ResetValue( Matrizes, GS1 );
  PostInputForm( Matrizes:time1, Matrizes, GS1, "gols sofridos:",
    Matrizes, GM1, "gols marcados:", Matrizes, SG1, "saldo de gols",
    Matrizes, PM1, "pontos ganhos" );
  } );

/*****
(2)      ****  RULE: GMC1
*****/
MakeRule( GMC1, [],
  Matrizes:GM1 != Matrizes:GMC1,
  {
  SetPostMessageTitle( "Alerta de erro!!" );
  PostMessage( "você errou no total de gols marcados nos dois turnos !,
  refaça seus cálculos" );
  ResetValue( Matrizes, GM1 );
  PostInputForm( Matrizes:time1, Matrizes, GS1, "gols sofridos:",
    Matrizes, GM1, "gols marcados:", Matrizes, SG1, "saldo de gols",
    Matrizes, PM1, "pontos ganhos" );
  } );

/*****
(3)      ****  RULE: SGC1
*****/
MakeRule( SGC1, [],
  Matrizes:SG1 != Matrizes:SGC1,
  {
  SetPostMessageTitle( "Alerta de erro" );
  PostMessage( "você errou no saldo de gols dos dois turnos !, confira
  seus cálculos" );
  ResetValue( Matrizes, SG1 );
  PostInputForm( Matrizes:time1, Matrizes, GS1, "gols sofridos:",
    Matrizes, GM1, "gols marcados:", Matrizes, SG1, "saldo de gols",
    Matrizes, PM1, "pontos ganhos" );
  } );

/*****
(4)      ****  RULE: PMC1
*****/
MakeRule( PMC1, [],
  Matrizes:PM1 != Matrizes:PMC1,

```

```

Matrizes:PM1 != Matrizes:PMC1,
{
SetPostMessageTitle( "Alerta de erro" );
PostMessage( "você errou na pontuação do time !, confira seus cálculos"
);
ResetValue( Matrizes, PM1 );
PostInputForm( Matrizes:timel, Matrizes, GS1, "gols sofridos:",
Matrizes, GM1, "gols marcados:", Matrizes, SG1, "saldo de gols",
Matrizes, PM1, "pontos ganhos" );

} );

/*****
(5)      **** RULE: Ref.soma
*****/
MakeRule( Ref.soma, [],
Matrizes:Numerodelinhas1 != Matrizes:Numerodelinhas2 Or
Matrizes:Numerodecolunas1
!= Matrizes:Numerodecolunas2,
{
PostMessage( "Somente podemos efetuar a soma de 2 matrizes se estas
possuirmos a mesma ordem, ou seja , o número de linhas de A = número de
linha de B e o número de coluna de A= número de coluna de B" );
ShowWindow( Session3 );
Esconder( );
Operacoes( );

/*****
(6)      **** RULE: média
*****/
MakeRule( média, [],
Matrizes:média1 != Matrizes:médiaC1 Or Matrizes:média2 !=
Matrizes:médiaC2 Or Matrizes:média3 != Matrizes:médiaC3
Or Matrizes:média4 != Matrizes:médiaC4,
PostMessage( "você errou na média" ) );

/*****
(7)      **** RULE: faltapontos
*****/
MakeRule( faltapontos, [],
Matrizes:pontosfal != Matrizes:pontosfaC1 Or Matrizes:pontosfa2
!= Matrizes:pontosfaC2 Or Matrizes:pontosfa3 != Matrizes:pontosfaC3
Or Matrizes:pontosfa4 != Matrizes:pontosfaC4,
PostMessage( "você errou na pontuação que necessita" ) );

/*****
(8)      **** RULE: PA1
*****/
MakeRule( PA1, [],
Matrizes:pontosatl != Matrizes:pontosatC1 Or Null?( Matrizes:pontosatl
),
{
PostMessage( "confira seus pontos atingidos" );
ResetValue( Matrizes:pontosatl );
PostInputForm( Matrizes:matérial, Matrizes, pontosatl, "pontuação dos
três bimestres:",
Matrizes, pontosfal, "necessito no quarto bimestre de:" );

```



```

} );

/*****
(9)    ****  RULE: PA2
        *****/
MakeRule( PA2, [],
  Matrizes:pontosat2 != Matrizes:pontosatC2 Or Null?( Matrizes:pontosat1
),
  {
  PostMessage( "confira seus pontos atingidos" );
  ResetValue( Matrizes:pontosat2 );
  PostInputForm( Matrizes:matéria2, Matrizes, pontosat2, "pontuação dos
três bimestres:",
    Matrizes, pontosfa2, "necessito no quarto bimestre de:" );
  } );

/*****
(10)   ****  RULE: PA3
        *****/
MakeRule( PA3, [],
  Matrizes:pontosat3 != Matrizes:pontosatC3 Or Null?( Matrizes:pontosat3
),
  {
  PostMessage( "confira seus pontos atingidos" );
  ResetValue( Matrizes:pontosat3 );
  PostInputForm( Matrizes:matéria3, Matrizes, pontosat3, "pontuação dos
três bimestres:",
    Matrizes, pontosfa3, "necessito no quarto bimestre de:" );
  } );

/*****
(11)   ****  RULE: PA4
        *****/
MakeRule( PA4, [],
  Matrizes:pontosat4 != Matrizes:pontosatC4 Or Null?( Matrizes:pontosat4
),
  {
  PostMessage( "confira seus pontos atingidos" );
  ResetValue( Matrizes:pontosat4 );
  PostInputForm( Matrizes:matéria4, Matrizes, pontosat4, "pontuação dos
três bimestres:",
    Matrizes, pontosfa4, "necessito no quarto bimestre de:" );
  } );

/*****
(12)   ****  RULE: FP1
        *****/
MakeRule( FP1, [],
  Matrizes:pontosfal != Matrizes:pontosfaC1 Or Null?( Matrizes:pontosfal
),
  {
  PostMessage( "você errou na nota do quarto bimestre" );
  ResetValue( Matrizes, pontosfal );
  PostInputForm( Matrizes:matéria1, Matrizes, pontosat1, "pontuação dos
três bimestres:",
    Matrizes, pontosfal, "necessito no quarto bimestre de:" );
  } );

```

```

PostInputForm( Matrizes:matéria1, Matrizes, pontosatl, "pontuação dos
três bimestres:",
Matrizes, pontosfal, "necessito no quarto bimestre de:" );
} )

```

```

/*****
(13)  **** RULE: M1
*****/

```

```

MakeRule( M1, [],
Matrizes:média1 != Matrizes:médiaCC1 Or Null?( Matrizes:média1 ),
{
PostMessage( "você errou na média" );
ResetValue( Matrizes, média1 );
PostInputForm( Matrizes:matéria1, Matrizes, média1, "média dos 3
bimestres" );
} );

```

```

/*****
(14)  **** RULE: médiabimestrais
*****/

```

```

MakeRule( médiabimestrais, [],
Matrizes:média1 != NULL Or Matrizes:média2 != NULL Or Matrizes:média3
!= NULL Or Matrizes:média4 != NULL,
{
PostMessage( "você deve calcular a média das matéria" );
PostInputForm( Matrizes:matéria1, Matrizes, média1, "média dos 3
bimestres" );
PostInputForm( Matrizes:matéria2, Matrizes, média2, "média dos 3
bimestres" );
PostInputForm( Matrizes:matéria3, Matrizes, média3, "média dos 3
bimestres" );
PostInputForm( Matrizes:matéria4, Matrizes, média4, "média dos 3
bimestres" );
} );

```

6.3.2. Descreve-se algumas funções utilizadas na estrutura do Matrix 97

```

/*****
(a)  **** FUNCTION: Tela1
*****/

```

```

MakeFunction( Tela1, [],
{
HideWindow( SESSION );
ShowWindow( Session1 );
ShowImage( Bitmap3 );
ShowImage( Button2 );
ShowImage( Button3 );
ShowImage( SingleListBox1 );
} );

```

```

/*****
(b)  **** FUNCTION: V_voltartela0
*****/

```



```

MakeFunction( V_voltartela0, [],
{
  HideWindow( Session1 );
  ShowWindow( SESSION );
  ShowImage( Bitmap1 );
  ShowImage( Bitmap2 );
  ShowImage( Button1 );
} );

/*****
(c) **** FUNCTION: Tela3
*****/
MakeFunction( Tela3, [],
{
  Esconder( );
  HideImage( Text2 );
  HideImage( Text3 );
  HideImage( Text130 );
  HideImage( Text4 );
  Ident.usuario( );
  ClearTranscriptImage( Transcript1 );
  ShowImage( Transcript1 );
  DisplayText( Transcript1, Dados:nome, ", Seja bem-vindo ao Matrix 97!"
);
  ShowImage( Transcript2 );
  ClearTranscriptImage( Transcript2 );
  DisplayText( Transcript2, Date( ) );
  HideWindow( Session1 );
  ShowWindow( Session3 );
  ShowImage( Button5 );
  ShowImage( Button69 );
  SetPostMessageTitle( "Informações sobre o sistema" );
  PostMessage( "Este sistema tem como objetivo ajudá-lo na aprendizagem
do conteúdo de Matriz" );
  PostMessage( "Ele está estruturado em 3 partes:" );
  PostMessage( "1) Histórico: alguns dados históricos de matrizes" );
  PostMessage( "2) Definições: a partir de aplicações" );
  PostMessage( "3) Operações: exercícios com adição e subtração de
matrizes" );
  PostMessage( "É importante que você navegue por todo o sistema" );
  ShowImage( Text130 );
} );

/*****
(d) **** FUNCTION: Ident.usuario
*****/
MakeFunction( Ident.usuario, [],
{
  ResetValue( Dados:nome );
  PostInputForm( "Qual é o seu nome?", Dados:nome, " " );
  If Null?( Dados:nome )
  Then SetValue( Dados:nome, Anônimo );
} );

/*****
(e) **** FUNCTION: timel
*****/

```

```
MakeFunction( timel, [],
  SetValue( Matrizes, timel, Dados:times ) );
```

```

  /*****
/*****
(f)      **** FUNCTION: Horas
  *****/
```

```
MakeFunction( Horas, [],
  {
  ClearTranscriptImage( Transcript3 );
  ShowImage( Transcript3 );
  DisplayText( Transcript3, Time( ) );
  Wait( 2 );
  HideImage( Transcript3 );
  } );
```

```

  /*****
(g)      **** FUNCTION: Histórico
  *****/
```

```
MakeFunction( Histórico, [],
  {
  HideImage( Button8 );
  HideImage( Button9 );
  HideImage( Button10 );
  HideImage( Button11 );
  HideImage( Bitmap4 );
  HideImage( Bitmap5 );
  } );
```

```

  /*****
(h)      **** FUNCTION: Desenvolvimento
  *****/
```

```
MakeFunction( Desenvolvimento, [],
  {
  HideImage( Button7 );
  HideImage( Button8 );
  HideImage( Button9 );
  HideImage( Button11 );
  HideImage( Bitmap4 );
  HideImage( Bitmap5 );
  HideWindow( Session3 );
  ShowWindow( Session5 );
  } );
```

```

  /*****
(i)      **** FUNCTION: Operacoes
  *****/
```

```
MakeFunction( Operacoes, [],
  {
  HideImage( Button7 );
  HideImage( Button10 );
  HideImage( Button11 );
  HideImage( Bitmap4 );
  HideImage( Button5 );
  HideImage( Button6 );
  HideImage( Bitmap5 );
  ShowImage( Button15 );
  } );
```

```

ShowImage( Button13 );
ShowImage( Button14 );
ShowImage( Button12 );
} );

/*****
(j)      **** FUNCTION: Aplicacoes
*****/
MakeFunction( Aplicacoes, [],
{
HideImage( Button7 );
HideImage( Button8 );
HideImage( Button9 );
HideImage( Button10 );
HideImage( Bitmap4 );
HideImage( Bitmap5 );
} );

/*****
(1)     **** FUNCTION: soma
*****/
MakeFunction( soma, [],
{
ResetValue( Matrizes:Numerodelinhas1 );
ResetValue( Matrizes:Numerodelinhas2 );
ResetValue( Matrizes:Numerodecolunas1 );
ResetValue( Matrizes:Numerodecolunas2 );
PostInputForm( "Responda: ", Matrizes, Numerodelinhas1, "Número de
linhas da matriz A:",
Matrizes, Numerodecolunas1, "Número de colunas da matriz A:" );
PostInputForm( "Responda: ", Matrizes, Numerodelinhas2, "Número de
linhas da matriz B:",
Matrizes, Numerodecolunas2, "Número de colunas da matriz B:" );
If ( Matrizes:Numerodelinhas1 #= Matrizes:Numerodelinhas2
And Matrizes:Numerodecolunas1 #= Matrizes:Numerodecolunas2 )
Then {
Fazendoslots( );
For linha From 1 To Matrizes:Numerodecolunas1
Do ResetValue( Matrizes, A # linha );
Entradadeelementos( );
MostraMatriz( );
ShowImage( Text5 );
Wait( 2 );
MatrizB( );
HideImage( Text5 );
ShowImage( Text6 );
Operacaosoma( );
Wait( 3 );
Mostrasoma( );
MatrizAB( );
HideImage( Text6 );
ShowImage( Text7 );
}
Else {
Assert( Matrizes:Numerodelinhas1 );
ForwardChain( NULL, Ref.soma );
HideWindow( Session4 );
};
} );

```

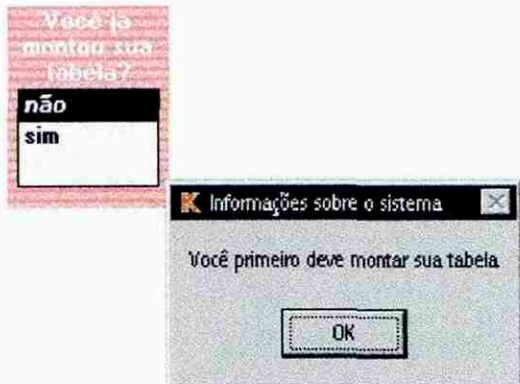
```

      /*****
(m)      ****  FUNCTION: Fazendoslots
      *****/
MakeFunction( Fazendoslots, [],
{
  For linha From 1 To Matrizes:Numerodelinhas1
    Do MakeSlot( Matrizes, A # linha );
  For linha From 1 To Matrizes:Numerodelinhas2
    Do MakeSlot( Matrizes, B # linha );
  For linha From 1 To Matrizes:Numerodelinhas1
    Do SetSlotOption( Matrizes, A # linha, MULTIPLE );
  For linha From 1 To Matrizes:Numerodelinhas2
    Do SetSlotOption( Matrizes, B # linha, MULTIPLE );
} );

      /*****
(n)      ****  FUNCTION: Entradadeelementos
      *****/
MakeFunction( Entradadeelementos, [],
{
  For linha From 1 To Matrizes:Numerodecolunas1
    Do ResetValue( Matrizes, A # linha );
  For linha From 1 To Matrizes:Numerodecolunas1
    Do PostInputForm( "Matriz A: elementos da coluna:" # linha,
      Matrizes, A # linha, "valores " );
} );
```


6.3.3. Descreve-se aqui alguns métodos :

(I)



Dentro do KAPPA, esta é uma imagem. Toda imagem é uma instância. Esta instância está associada a um slot que pode ter valor: numérico, texto, booleano ou objeto. Neste caso, texto, observa-se que a resposta é sim ou não. Este slot (Resposta) está associado a um método(Ref.Resposta), ou seja dependendo da resposta ele efetuará um determinado contexto. Este método está descrito abaixo em linguagem Kal.

```
(II) /***** METHOD: Ref.Resposta *****/
MakeMethod( Matrizes, Ref.Resposta, [],
{
  HideImage( Button29 );
  HideImage( Button63 );
  somaptos( );
  pontostime1( );
  pontostime2( );
  pontostime3( );
  pontostime4( );
  If ( Matrizes:Resposta #= não )
    Then {
      PostMessage( "Você primeiro deve montar sua tabela" );
      HideWindow( Session14 );
      ShowWindow( Session13 );
    };
  If ( Matrizes:Resposta #= sim )
    Then {
      HideImage( SingleListBox16 );
      For x From 1 To 4
        Do ResetValue( Matrizes, GS # x );
      For x From 1 To 4
        Do ResetValue( Matrizes, SG # x );
      For x From 1 To 4
        Do ResetValue( Matrizes, PM # x );
      For x From 1 To 4
        Do ResetValue( Matrizes, GM # x );
      PostInputForm( Matrizes:time1, Matrizes, GS1, "gols sofridos:",
        Matrizes, GML, "gols marcados:", Matrizes, SG1, "saldo de
gols",
        Matrizes, PML, "pontos ganhos" );
      {
        pontosjogo( );
        Assert( Matrizes:GS1 );
        ForwardChain( NULL, GSC1 );

```



```

ForwardChain( NULL, GSC1 );
Assert( Matrizes:GM1 );
ForwardChain( NULL, GMC1 );
Assert( Matrizes:SG1 );
ForwardChain( NULL, SGC1 );
Assert( Matrizes:PM1 );
ForwardChain( NULL, PMC1 );
};
PostInputForm( Matrizes:time2, Matrizes, GS2, "gols sofridos:",
    Matrizes, GM2, "gols marcados:", Matrizes, SG2, "saldo de
gols",
    Matrizes, PM2, "pontos ganhos" );
{
Assert( Matrizes:GS2 );
ForwardChain( NULL, GSC2 );
Assert( Matrizes:GM2 );
ForwardChain( NULL, GMC2 );
Assert( Matrizes:SG2 );
ForwardChain( NULL, SGC2 );
Assert( Matrizes:PM2 );
ForwardChain( NULL, PMC2 );
};
PostInputForm( Matrizes:time3, Matrizes, GS3, "gols sofridos:",
    Matrizes, GM3, "gols marcados:", Matrizes, SG3, "saldo de
gols",
    Matrizes, PM3, "pontos ganhos" );
{
Assert( Matrizes:GS3 );
ForwardChain( NULL, GSC3 );
Assert( Matrizes:GM3 );
ForwardChain( NULL, GMC3 );
Assert( Matrizes:SG3 );
ForwardChain( NULL, SGC3 );
Assert( Matrizes:PM3 );
ForwardChain( NULL, PMC3 );
};
PostInputForm( Matrizes:time4, Matrizes, GS4, "gols sofridos:",
    Matrizes, GM4, "gols marcados:", Matrizes, SG4, "saldo de
gols",
    Matrizes, PM4, "pontos ganhos" );
Assert( Matrizes:GS4 );
{
ForwardChain( NULL, GSC4 );
Assert( Matrizes:GM4 );
ForwardChain( NULL, GMC4 );
Assert( Matrizes:SG4 );
ForwardChain( NULL, SGC4 );
Assert( Matrizes:PM4 );
ForwardChain( NULL, PMC4 );
};
ShowImage( Transcript20 );
ClearTranscriptImage( Transcript20 );
DisplayText( Transcript20, FormatValue( "%s\t ", GS ) );
DisplayText( Transcript20, FormatValue( "%s\t ", GM ) );
DisplayText( Transcript20, FormatValue( "%s\t ", SG ) );
DisplayText( Transcript20, FormatValue( "%s\t ", PG ) );
DisplayText( Transcript20, FormatValue( "%s\n ", times ) );
DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GS1 )
);
DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GM1 )
);

```

```
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:SG1 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:PM1 )
);
    DisplayText( Transcript20, FormatValue( "%s\n ", Matrizes:time1
) );
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GS2 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GM2 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:SG2 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:PM2 )
);
    DisplayText( Transcript20, FormatValue( "%s\n ", Matrizes:time2
) );
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GS3 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GM3 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:SG3 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:PM3 )
);
    DisplayText( Transcript20, FormatValue( "%s\n ", Matrizes:time3
) );
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GS4 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:GM4 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:SG4 )
);
    DisplayText( Transcript20, FormatValue( "%s\t ", Matrizes:PM4 )
);
    DisplayText( Transcript20, FormatValue( "%s\n ", Matrizes:time4
) );

    ShowImage( Button43 );
    ShowImage( Button79 );
    For x From 197 To 199
        Do ShowImage( Text # x );
    );
```

6.4. Slots

São características de um determinado objeto, em nosso contexto, **Matrizes**.

Descreve-se abaixo alguns slots definidos no **Matrix 97**.

6.4.1 Slots com valores numéricos:

Single(apenas um valor numérico)

- GM1- Gols marcados do primeiro time;
- GS2-Gols sofridos do segundo time;
- PM3- Pontos atingidos do terceiro time;
- PA1- Pontos atingidos da primeira matéria;
- M1-Média da primeira matéria;
- Numerodecolunas1-Número de colunas da primeira matriz;
- Numerodelinhas2- Número de linhas da segunda matriz;

MULTIPLE (mais de um valor numérico) como nos *slots* a seguir:

- A1- elementos da primeira coluna da matriz A;
- A2- elementos da segunda coluna da matriz A;
- B1- elementos da primeira coluna da matriz B;
- AB1- elementos da primeira coluna da matriz soma(A+B);

6.4.2.slots com valores: texto:

Single:

- matéria1: referente ao nome da primeira matéria a ser analisada;
- matéria2: referente ao nome da segunda matéria a ser utilizada;
- time1: nome do primeiro time finalista;

Multiple:

- times: referente ao nome dos quatro times finalistas;
- matérias: referente ao nome das quatro matérias selecionadas.

6.5. Interface da shell Kappa

Foram usados os seguintes recursos no desenvolvimento do Matrix 97.

Textos: os textos escritos dentro do sistema;

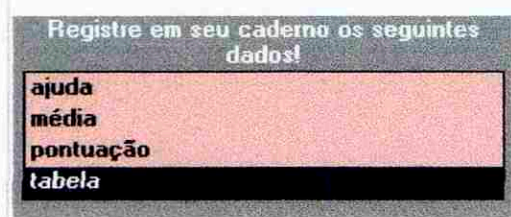


Bitmap: recurso disponível para importar imagens;



Button: botões apresentados no sistema.

3 **Edit**: Esta imagem permite o usuário alterar o valor do slots(placar do jogo, por exemplo).



SingleListBox: Esta imagem permite o usuário selecionar certos itens(ex. analisar: média, pontuação e a tabela);

Podemos analisar :

- Ajuda
- gols marcados
- gols sofridos
- saldo de gols
- pontuação de cada time

RadioButton: praticamente mesmo objetivo que o SingleListBox

Notas bimestrais			
Bim:1	Bim:2	Bim:3	matérias
6	8	9	Sociologia
10	8	9	Química
6	7	3	Matemática
8	7	9	Português

TranscriptImage: Esta image permite visualizar as informações contidas em slots.

Selecione a matéria a ser analisada!

Matemática	▼
Geografia	▲
História	
Inglês	
Matemática	
Português	
Química	
Religião	
Sociologia	▼

ComboBox: Uma imagem que não sobrecarrega a tela.(ex. os times participantes do campeonato brasileiro e as matérias escolares).

Conclusão

7. Conclusões

Ao término do presente trabalho constata-se que o desenvolvimento de um Software Educacional é resultado da busca de novas metodologias a serem aplicadas no ensino de Matemática visando a aprendizagem. Desta forma estes sistemas devem apresentar meios que despertem nos alunos suas capacidades de: raciocínio, criação, interesse e motivação.

Dispõe-se atualmente da tecnologia, e quer-se que esta contribua de maneira significativa para a educação.

Os resultados desta pesquisa mostram que os recursos de Inteligência Artificial aplicados à Matemática poderão tornar-se um meio eficiente para que o aluno construa seu conhecimento em conteúdos específicos.

O **Matrix 97** é um sistema que pode ser considerado como um embrião para o estudo de Matrizes. Acredita-se que sua utilização efetiva em sala de aula proporcionará condições favoráveis para o usuário inferir novos conceitos matriciais bem como utilizar este conteúdo em novas situações problemas.

O objetivo do **Matrix 97** é criar um ambiente computacional para estudar matrizes, isto é, é uma ferramenta que pretende auxiliar no processo de ensino-aprendizagem deste conteúdo. Para tanto, sua aplicação deverá ser feita em paralelo às aulas ministradas em sala de aula.

O **Matrix 97** deverá ser ampliado, pretende-se ainda desenvolver o item multiplicação de matrizes. Esta é uma atividade que a autora propõe-se a desenvolver para integralizá-lo.

Após a integralização do **Matrix 97**, este será submetido a uma fase preliminar de avaliação. Deverá então ser aplicado em turmas piloto, onde será avaliado, revisado e então validado.

Ressalta-se que a experiência adquirida com o desenvolvimento de ferramentas que possam contribuir para o ensino de Matemática, em particular o desenvolvimento de pequenos sistemas geram grande expectativa e motivação para a continuidade do presente trabalho.

O **Matrix 97** validado estará disponível no departamento de Matemática da UFSC, e poderá ser aplicado a públicos-alvo de interesse.

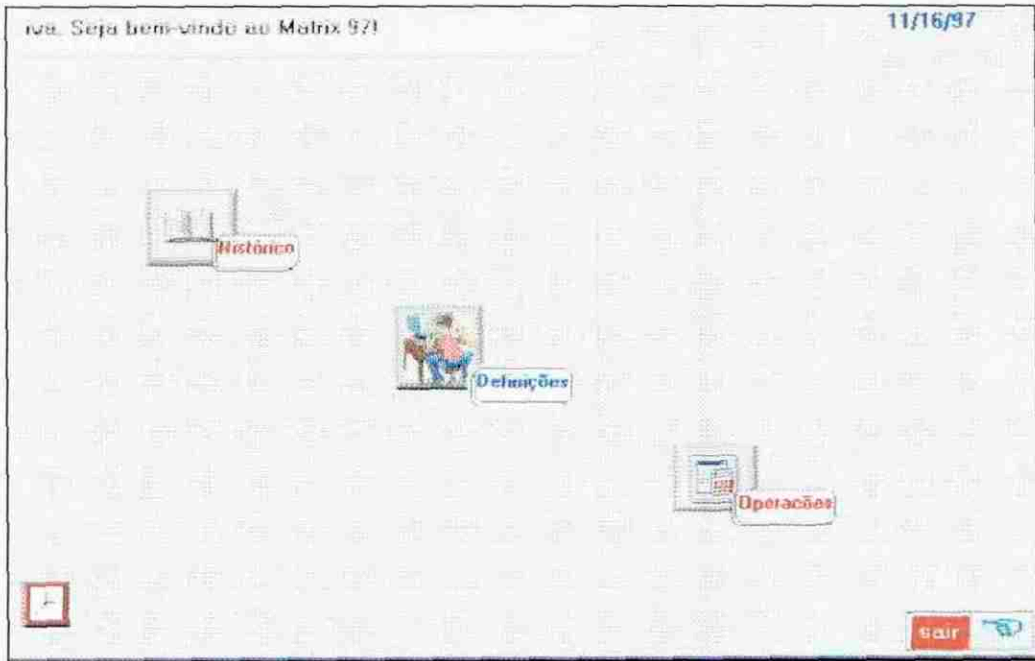
Anexos




tela 1



tela 2




tela 2.1.



Das Tabelas às Matrizes

Ao final do século XVIII, Leibnitz, na Alemanha, e Seki Kowa, no Japão, desenvolveram métodos de resolução de sistemas lineares baseados em tabelas numéricas formadas por coeficientes das que compunham estes sistemas.

Estas tabelas numéricas deram origem ao que hoje chamamos de matrizes, que além de serem aplicadas na resolução de sistemas lineares, possibilitaram o desenvolvimento de novos ramos da matemática.



tela 3



Definições

Campeonato Brasileiro

Na tabela abaixo você selecionará os 4 times finalistas.

Veja quais são os times participantes!

Criciúma
Criciúma
Cruzeiro
Flamengo
Fluminense
Goiás
Grêmio
Guarani/SP
Inter



1 Seleccione o primeiro finalista e clique no número 1 com o botão direito do mouse.

2 Seleccione o segundo finalista e clique no número 2 com o botão direito do mouse.

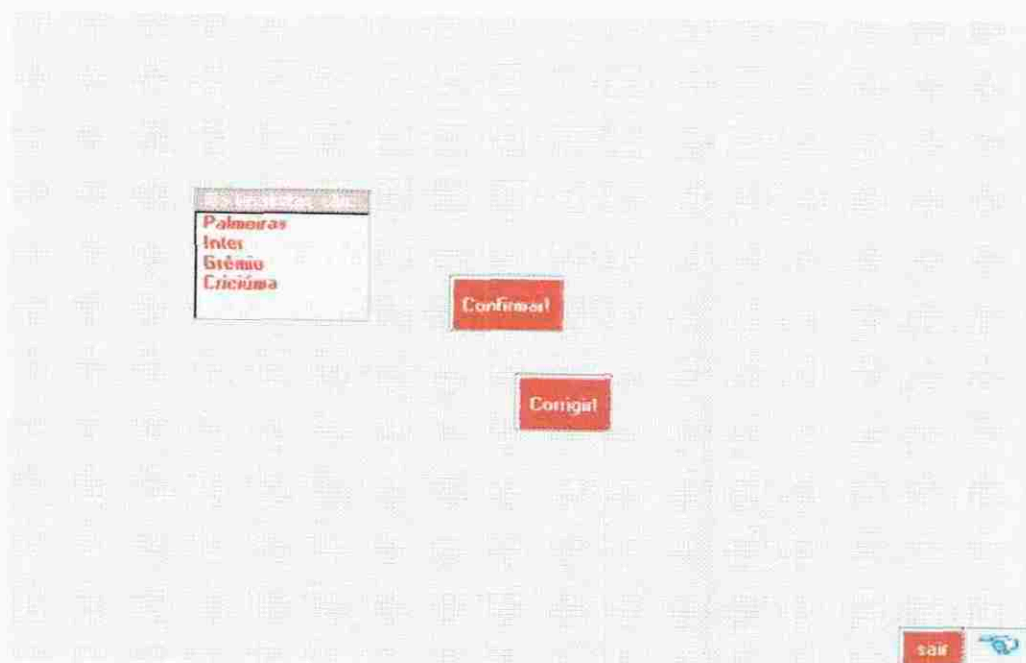
3 Seleccione o terceiro finalista e clique no número 3 com o botão direito do mouse.

4 E por último seleccione o quarto finalista e clique no número 4 com o botão direito do mouse.

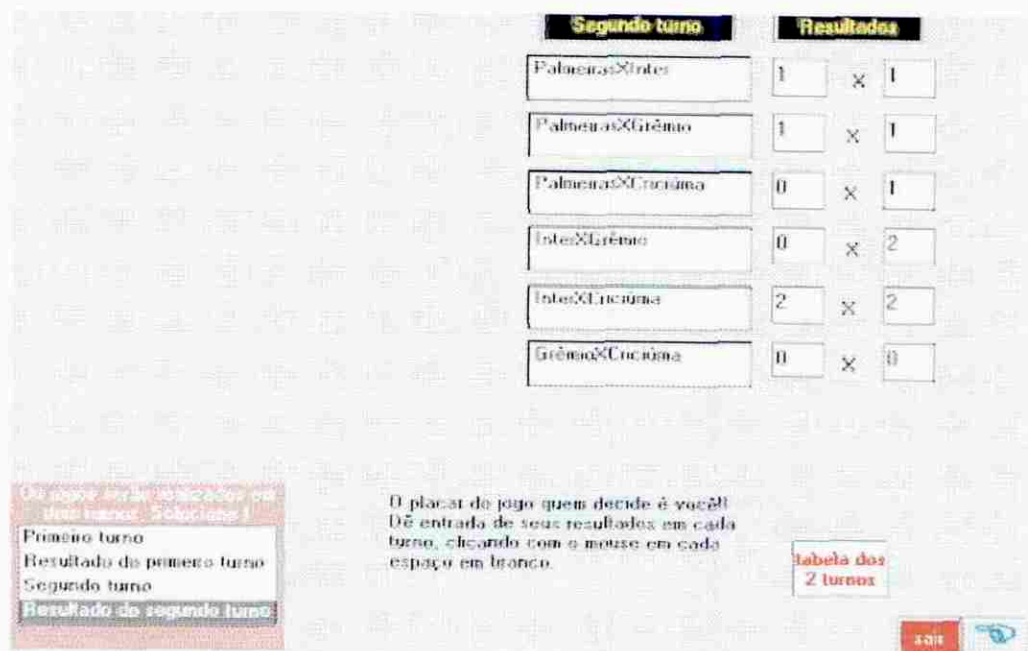
1 Palmeiras 2 Inter 3 Grêmio 4 Criciúma

Sair  

tela 4



tela5




tela 6

Primeiro turno	Resultados	Segundo turno	Resultados
Palmeiras x Inter	2 x 1	Palmeiras x Inter	1 x 1
Palmeiras x Grêmio	1 x 2	Palmeiras x Grêmio	1 x 1
Palmeiras x Cruzeiro	2 x 2	Palmeiras x Cruzeiro	0 x 1
Inter x Grêmio	0 x 0	Inter x Grêmio	0 x 2
Inter x Cruzeiro	0 x 1	Inter x Cruzeiro	2 x 2
Grêmio x Cruzeiro	1 x 0	Grêmio x Cruzeiro	0 x 0

Podemos analisar:

- Apada
- gols marcados
- gols sofridos
- saldo de gols
- pontuação de cada time

OK 

tela 7

Palmeiras

gols sofridos:

gols marcados:

saldo de gols:

pontos ganhos:

tela 8

Observe que esta tabela pode ser apresentada apenas por seus valores numéricos.

Sua tabela				
GS	GM	SG	PG	Times
8	7	-1	6	Palmeiras
8	4	4	3	Inter
2	6	4	12	Grêmio
5	6	1	9	Criciúma

Os elementos desta tabela estão dispostos ordenadamente em linhas e colunas. Sendo que esta tabela apresenta 4 linhas e 4 colunas.



tela 9

Sua tabela				
GS	GM	SG	PG	Times
8	7	-1	6	Palmeiras
8	4	4	3	Inter
2	6	4	12	Grêmio
5	6	1	9	Criciúma

col.1 col.2 col.3 col.4

linha.1
linha.2
linha.3
linha.4

Posição do elemento

Qual é o elemento que se encontra 8 na lin.1 e col.1

OK Reset

Responda!

tela 10

Sua tabela				
GS	GM	SG	PG	Inter
8	7	1	6	Palmeiras
8	4	-4	3	Inter
2	6	4	12	Guarani
5	6	1	9	Encanto

col.1	col.2	col.3	col.4	
8	7	-1	6	linha.1
8	4	-4	3	linha.2
2	6	4	12	linha.3
5	3	1	9	linha.4

Observe que a localização de cada elemento da tabela é determinada pela linha e coluna, as quais ele pertence.

Podemos agora generalizar este exemplo!

3.00

tela 11

Para tanto denominaremos cada elemento por a_{ij} onde

i → a linha onde ele se encontra e

j → a coluna onde ele se encontra.

col.1	col.2	col.3	col.4	
8	7	-1	6	linha.1
8	4	-4	3	linha.2
2	6	4	12	linha.3
5	3	1	9	linha.4

Desta Forma a tabela de nosso exemplo pode ser assim representada

representação

3.00

tela 12


col.1	col.2	col.3	col.4	
8	7	-1	6	linha.1
8	4	-4	3	linha.2
2	6	4	12	linha.3
5	3	1	9	linha.4

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Observe que:

i -> varia de 1 a 4.

j -> varia de 1 a 4.



tela13

Outra aplicação de Matrizes é nas médias Escolares

Vamos supor que estamos no final de um ano letivo; portanto, você já possui notas do:

Primeiro bimestre
Segundo bimestre e
Terceiro bimestro.

Selecione a matéria a ser analisada!

- Matemática
- Geografia
- História
- Inglês
- Matemática
- Português
- Química
- Religião
- Sociologia

Selecione a primeira matéria e clique com botão direito do mouse no número um.

Repita o mesmo processo para as matérias 2, 3 e 4.

dica: selecione 4 matérias distintas.

1

Sociologia

2


Química

3

Matemática

4

Português



tela14

Matérias selecionadas

- 1) Sociologia
- 2) Química
- 3) Matemática
- 4) Português

confirmar!!

Corrigir!!

tela 15

Sociologia

nota do primeiro bimestre:

nota do segundo bimestre:

nota do terceiro bimestre:

OK Reset

tela 16

Observe que nesta tabela o número de linhas é 4 e o número de colunas é 3.

Notas bimestrais			
Bim:1	Bim:2	Bim:3	matérias
8	7	9	Sociologia
5	6	8	Química
5	7	6	Matemática
9	8	10	Português

A localização de cada elemento desta tabela também fica bem determinada pela sua posição na linha i e coluna j .

Responda!

OK

tela 17

8			
5	6		
		6	
9		10	

Notas bimestrais			
Bim:1	Bim:2	Bim:3	matérias
8	7	9	Sociologia
5	6	8	Química
5	7	6	Matemática
9	8	10	Português

Responda

Qual é o valor do a_{13} ?

OK Reset

tela 18




8	7	9
5	6	8
5	7	6
9	8	10

Notas bimestrais			
Bim:1	Bim:2	Bim:3	matérias
8	7	9	Sociologia
5	6	8	Química
5	7	6	Matemática
9	8	10	Português

Na tabela acima, os números colocados nas disposições horizontais formam o que denominamos **linha**

e os colocados nas disposições verticais formam o que denominamos **coluna**

Vimos neste exemplo que o número de linhas é diferente do número de coluna.

tela 19



8	7	9
5	6	8
5	7	6
9	8	10

Notas bimestrais			
Bim:1	Bim:2	Bim:3	matérias
8	7	9	Sociologia
5	6	8	Química
5	7	6	Matemática
9	8	10	Português

Denominamos "Matriz" de ordem $m \times n$ a toda tabela constituída por $m \cdot n$ elementos dispostos em m linhas e n colunas.

O número de linhas m e de colunas n determinam a ordem $m \times n$ da Matriz.

Neste exemplo, temos uma matriz de ordem 4×3 , isto é, uma matriz formada por 4 linhas e 3 colunas.

tela 20

Algumas notações usadas!

1) Para nomear matrizes, utilizamos letras maiúsculas, indexadas ou não pela ordem da matriz.

$$A_{2 \times 3} = \begin{bmatrix} 2 & 3 & 5 \\ 3 & -2 & 9 \end{bmatrix} \quad \text{ou } A = \begin{bmatrix} 2 & 3 & 5 \\ 3 & -2 & 9 \end{bmatrix}$$

A representação de matrizes podem ser entre colchetes ou parênteses.

Observe que cada elemento da matriz tem uma posição bem definida, isto é, uma linha i e uma coluna j .

Neste exemplo a matriz tem 2 linhas e 3 colunas.

[Botões de navegação]

tela 21

Analisando a tabela das notas podemos nos questionar sobre:

Qual o total de pontos obtidos nos três bimestres em cada matéria?

Qual a média obtida nos três bimestres?

Quantos pontos serão necessários no quarto bimestre, em cada matéria, para atingir média 7?

Registre em seu caderno os seguintes dados!

ajuda
média
pontuação
tabela

Notas bimestrais			
Bim:1	Bim:2	Bim:3	matérias
8	7	9	Sociologia
5	6	8	Química
5	7	6	Matemática
9	8	10	Português

Dica: Use seus cálculos e registre a resposta!

[Botão de navegação]

tela 22

Bim.1	Bim.2	Bim.3	matéria
8	7	9	Sociologia
5	6	8	História
5	7	6	Matemática
9	8	10	Português

Sociologia

pontuação dos três bimestres:

necessário no quarto bimestre de:

ajuda

média

pontuação

tabela

tela 23

Matemática

média dos 3 bimestres:

ajuda

média

pontuação

tabela

tela 24

Veja como foram trabalhados os dados de sua tabela!

Observe!

média
nota necessária no quarto bimestre
pontos atingidos

tela 25

Observe!

média
nota necessária no quarto bimestre
pontos atingidos

Notas bimestrais

Bim:1	Bim:2	Bim:3	matérias
8	7	9	Sociologia
5	6	8	Química
5	7	6	Matemática
9	8	10	Português

Para obter as médias você dividiu todos os elementos da matriz [pontos atingidos] por 3 ou seja multiplicou por 1/3.

$$\frac{1}{3} \begin{bmatrix} 24 \\ 19 \\ 18 \\ 27 \end{bmatrix} = \begin{bmatrix} 8 \\ 6,34 \\ 6 \\ 9 \end{bmatrix}$$

Atente para a operação!

Sua Tabela

TP	FP	média	matérias
24	4	8	Sociologia
19	9	6,34	Química
18	10	6	Matemática
27	1	9	Português

3/3

tela 26

Operações Elementares

adição

subtração

multiplicação por escalar

Dada uma matriz $A = (a_{ij})_{m \times n}$ e um número real K , denomina-se produto de K por A a matriz obtida multiplicando-se cada elemento de A por K .

$$K.A = (k.a_{ij})_{m \times n} \quad \text{onde } 1 \leq i \leq m \text{ e } 1 \leq j \leq n.$$

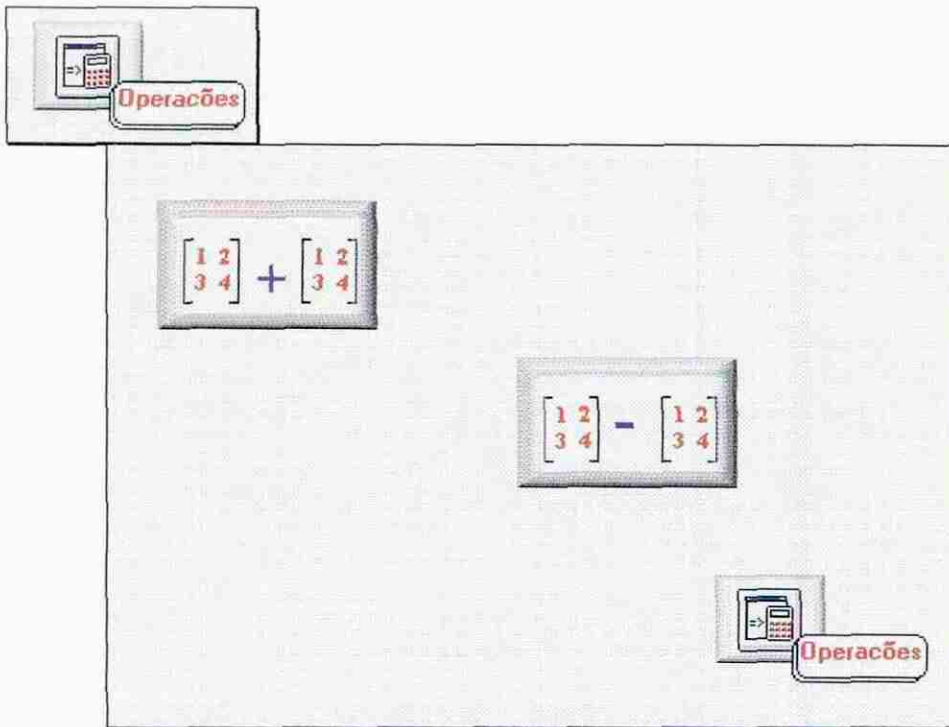
A multiplicação por escalar foi utilizada para determinar as médias escolares.

Rever médias

mais exemplos

1/32

tela 27



tela 28

Responda:

Número de linhas da matriz A:

Número de colunas da matriz A:

tela 29

Matriz B: elementos da
coluna:1

valores

5	▲
6	▼

OK Reset

tela 30

$A+B=$

7	3
8	2

tela 31

**Referências
Bibliográficas**

8. Referências Bibliográficas

- 1 DIAS FILHO, Astor Guimarães, 1951-**Matemática: 2^o** grau- São Paulo: Scipione Autores Editores, 1985.
- 2 GENARO, Sérgio. **Sistemas Especialistas: o conhecimento artificial** - Rio de Janeiro; São Paulo:LTC - Livros Técnicos e Científicos Editora S.A.; 1986.
- 3 GIOVANNI, José Ruy. **Matemática 2: 2^o** grau: trigonometria, matrizes, análise combinatória, geometria- São Paulo: FTD, 1992.
- 4 G. BITTENCOURT. **Inteligência Artificial: ferramentas e teorias** . UFSC-LCMI- Laboratório de Controle e Microinformática. Florianópolis, 1996.
- 5 **INTELIGÊNCIA ARTIFICIAL APLICADA AO ENSINO DE MATEMÁTICA**
Relatório do GEIAAM- dezembro de 1996.
- 6 **KAPPA User's Guide**. IntelliCorp, Inc. May, 1991- Publication Number: KAP1.2-UG-MSW2-1.
- 7 **KAPPA Reference Manual**. IntelliCorp, Inc. Publication Number: KAP1.2-RM-MSW-2; April 1992.
- 8 SETZER, V. W. **Computadores na educação: Porquê, Quando e como**. Anais do V SBIE, Sociedade Brasileira de Computação.
- 9 YOUSSEF, Antonio Nicolau. **Matemática e Fundamentos**: vol 2., 2^o grau, Editora Scipione, São Paulo-SP.