

Faculdade de Engenharia da Universidade do Porto



FEUP

Módulo Pedagógico sobre Problemas de Optimização Combinatório com base no TSP

Nuno Miguel Teixeira Pinto

Projecto realizado no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Telecomunicações

Orientador: Prof. Dr. Eurico Manuel Elias Morais Carrapatoso

Janeiro de 2009

Resumo

Este trabalho diz respeito a um dos temas mais importantes no ensino informatizado de hoje em dia. Este tema designa-se como Módulo Pedagógico e tem como objectivo ter a capacidade de poder ser integrado em qualquer plataforma desejada. No caso deste trabalho o Módulo Pedagógico terá como conteúdo o problema do Caixeiro Viajante.

Começar-se-á por referir um breve historial sobre a evolução do ensino informatizado, serão apresentadas as várias formas de ensino utilizadas via internet bem como a definição de Módulo, passando pelos seus diferentes tipos de programação e referindo também algumas das suas regras de construção segundo o ambiente de visualização gráfica.

Posteriormente serão explicados os conceitos inerentes aos problemas de optimização combinatória, será exemplificada a sua aplicação ao problema do Caixeiro Viajante.

Após o estudo e análise do que já foi referido é feita uma avaliação e escolha sobre as alternativas de implementação consideradas, nomeadamente, a escolha de PHP como linguagem de programação.

Este documento contém ainda uma descrição da implementação efectuada sobre o trabalho proposto, incluindo um manual de uso para programadores que tencionem utilizar o Módulo noutra plataforma e outro de funcionamento do mesmo do ponto de vista de utilizador.

Pode-se concluir que a integração do Módulo noutra plataforma que não aquela onde este foi desenvolvido, foi um sucesso. Sendo ainda referenciadas algumas sugestões para pesquisas e melhoramentos futuros.

Abstract

This document aims one of the most important topics in computer education nowadays. This topic is called Pedagogic Module and it's designed to be incorporated in any desired platform. In this case, the content of this Pedagogic Module is the travelling salesman problem.

It starts by referring a brief history on the evolution of education system, then the definition of module, through its various types of programming and also noting some of its rules of construction according to the graphical user interface.

After that it will be explained the concepts related to the combinatorial optimization problems, and it will be exemplified its application to the travelling salesman problem.

After the study and analysis of what already has been referred, an evaluation and a choice is made about the implementation alternatives considered, including the choice of PHP as the programming language to be used.

This document also contains a description of the implementation done on the proposed work including the tutorial for developers who intend to use the module on another platform and another about the way of operating it from the user point of view.

It's confirmed that the integration of the module on a platform beside the one where it was developed was a success. Being also listed some suggestions for further searches and upgrades.

Agradecimentos

Em primeiro lugar quero agradecer ao meu orientador, Prof. Dr. Eurico Manuel Elias Morais Carrapatoso, sem a ajuda do qual, a realização e apresentação deste trabalho teria sido impossível, deixo aqui os meus mais sinceros agradecimentos pela sua disponibilidade e atenção.

A ajuda do Prof. Dr. António Miguel Gomes foi extremamente preciosa no âmbito de investigação operacional e por isso o meu muito obrigado por tudo.

Quero agradecer a Sílvia Oliveira, pelo apoio inigualável e inesgotável que me deu, mesmo quando a finalização desta tese parecia impossível. Obrigado pela esperança e confiança que me deste.

Obrigado avó, por toda a confiança em mim depositada e apoio financeiro fornecido, sem os quais eu não teria conseguido.

À minha mãe que sempre se preocupou com o meu bem-estar, com o meu progresso na vida académica e por estar sempre presente quando mais precisei de ajuda, deixo os meus mais sinceros agradecimentos.

Estendo a minha consideração e respeito ao meu padrasto por todo o apoio moral que me deu e pela oportunidade de continuar os meus estudos no Porto.

Um bom ambiente envolvente é sempre um ponto positivo a acrescentar. Os meus irmãos foram os que mais contribuíram para esse ambiente, fazendo-me sorrir quando não existiam motivos para tal.

Quero agradecer também a todos os meus amigos por todo o apoio fornecido e por me aturarem nestas alturas de grande agitação e nervosismo.

Resta-me dizer que estou profundamente agradecido a todos os que contribuíram directa ou indirectamente para a realização deste trabalho.

Índice

Resumo	iii
Abstract	iv
Agradecimentos	v
Índice	vi
Lista de figuras	viii
Lista de tabelas	ix
Abreviaturas.....	x
Capítulo 1 - Introdução	1
1.1 Contexto.....	1
1.2 Proposta	3
1.3 Estrutura do documento.....	4
Capítulo 2 - Estado da arte.....	5
2.1 Ensino informatizado	5
2.1.1 Como surgiu	5
2.1.2 Ensino via internet	10
2.2 Módulo Pedagógico	11
2.2.1 Definição de Módulo Pedagógico	11
2.2.2 Análise de metodologia de programação	11
2.3 Problemas Combinatórios.....	13
2.3.1 Conceitos.....	13
2.3.2 Caso de estudo - TSP	15
2.4 Implementações existentes	19
2.4.1 Plataformas	19
2.4.2 Módulos do Moodle.....	19

2.4.3	Algoritmos	20
2.5	Alternativas de implementação	20
Capítulo 3	- Implementação	23
3.1	Plataforma	23
3.2	Módulo	25
3.2.1	Geral	25
3.2.2	Partes comuns	28
3.2.3	Algoritmos	31
Capítulo 4	- Tutorial de integração do módulo	35
4.1	Geral	35
4.2	Integração no Moodle	37
Capítulo 5	- Funcionamento do módulo	40
Capítulo 6	- Conclusão	45
6.1	Comentários finais.....	45
6.2	Recomendações para trabalho futuro	46
	Bibliografia	47
	Anexos.....	49

Lista de figuras

Figura 1 - Electronic Numerical Integrator and Computer.....	6
Figura 2 - Primeiro PC da IBM.	6
Figura 3 - "Máquina de ensinar"	7
Figura 4 - Burrhus Frederic Skinner	7
Figura 5 - Vizinho mais próximo [24].....	17
Figura 6 - Inserção mais próxima [24].	17
Figura 7 - 2opt [24].	18
Figura 8 - Configurações da página.....	24
Figura 9 - Estrutura do módulo	25
Figura 10 - Estrutura de bloco	25
Figura 11 - Gestão de ficheiros	28
Figura 12 - Comparação de distâncias	33
Figura 13 - Menu principal	40
Figura 14 - Inserção de dados	41
Figura 15 - Geração de cidades	42
Figura 16 - Visualização de resultados.....	43
Figura 17 - Cálculo	44

Lista de tabelas

Tabela 1 - Algumas instâncias da biblioteca TSPLIB [23].	15
Tabela 2 - Validação	26
Tabela 3 - Menu principal	26
Tabela 4 - Menu secundário	27
Tabela 5 - Classe	27
Tabela 6 - DTD.	29
Tabela 7 - Exemplo.xml.	29
Tabela 8 - Pré-cálculo.	30
Tabela 9 - Passo-a passo.	31
Tabela 10 - Vizinho mais próximo.	31
Tabela 11 - Inserção mais próxima.	32
Tabela 12 - 2opt	33
Tabela 13 - config.php.	35
Tabela 14 - conf.php.	36
Tabela 15 - Parte de "index.php" do módulo no Moodle.	37
Tabela 16 - lib.php	38

Abreviaturas

Lista de abreviaturas

ASP	Active Server Pages
CAI	Computer Assisted Instruction
CD	Compact Disc
CSS	Cascading Style Sheets
DTD	Document Type Definition
EIAC	Ensino Inteligente Assistido por Computador
GUI	Graphical User Interface
HTML	HyperText Markup Language
IA	Inteligência Artificial
IBM	International Business Machines
ICAI	Intelligent Computer Assisted Instruction
NP	Nondeterministic polynomial time
PC	Personal Computer
PCV	Problema do Caixeiro Viajante
PHP	Hypertext Preprocessor
POC	Problemas de Optimização Combinatória
SMIL	Synchronized Multimedia Integration Language
SQL	Structured Query Language
STI	Sistemas Tutoriais Inteligentes
SVG	Scalable Vectorial Graphics
TSP	Travelling Salesman Problem
WEB	World Wide Web
XML	eXtensible Markup Language

Capítulo 1 - Introdução

1.1 Contexto

Este trabalho tem como finalidade a criação de um módulo pedagógico sobre problemas de optimização combinatória incidido no TSP (Travelling Salesman Problem). Um módulo pedagógico consiste num suporte de conteúdo virado para o ensino passível de poder ser integrado numa plataforma. Uma plataforma não é mais que uma ferramenta com a finalidade de gerir vários módulos integrados na própria. O TSP trata-se de um caso de estudo já bastante convencional para exemplificar as várias formas de resolução num problema de optimização combinatória. De uma forma simplista, pode dizer-se, no caso concreto deste trabalho, que o módulo pedagógico não é mais que uma ferramenta de apoio ao ensino e que tem como conteúdo a simulação de vários algoritmos sobre problemas de optimização combinatória aplicados ao TSP.

A construção de um módulo pedagógico é bastante complexa, uma vez que, existem uma série de vertentes a abordar e cada uma delas tem as suas regras. Nesse sentido, a implementação de um módulo, a plataforma onde este é inserido, a técnica de ensino usada e o conteúdo a que esta é aplicada são algumas dessas vertentes, que interagem todas entre si. Encarando o módulo apenas do ponto de vista de uma peça de um *puzzle*, é imprescindível a observação das suas ranhuras para que se saiba onde estas coincidem com as ranhuras de um espaço vazio no *puzzle*. Por outro lado, se considerarmos o módulo como ferramenta de ensino, é necessário ter em conta a técnica de ensino a aplicar e o conteúdo a que esta é aplicada entre muitos outros aspectos. É fácil de perceber que estas duas referências mencionadas reflectir-se-ão na implementação do módulo e dependem uma da outra, mas existem mais vertentes a abordar para além das anteriormente referidas, como, por exemplo, o próprio conteúdo que, por sua vez, também possui as suas próprias regras.

Uma boa construção de qualquer módulo consiste em torná-lo o mais generalista possível, ou seja, ter uma interface muito simples e de fácil configuração entre este e a plataforma. Se isto se verificar, a compatibilidade do módulo tornar-se-á extremamente ampla permitindo que este se adapte a qualquer plataforma que se pretenda usar.

Da mesma forma que o discurso de um emissor tem de ser adaptada ao tipo de plateia que esse discurso se destina, também a visualização do módulo necessita de ser adaptada ao tipo de público-alvo que pretendemos atingir uma vez que é extremamente relevante a forma como o módulo é interpretado pelo utilizador.

Dado tratar-se de um instrumento de ensino, deve-se ter em atenção, o modo como é realizado este processo, quais as suas bases e as técnicas de ensino existentes. A análise da evolução ao longo das últimas décadas sobre os vários métodos de ensino é a melhor forma de compreender, assimilar e utilizar os conceitos e regras na criação do módulo em questão.

No caso concreto deste trabalho, o conteúdo do módulo consiste nos problemas de optimização combinatória e terá como objecto de estudo alguns algoritmos aplicados ao clássico problema TSP.

Os problemas de optimização combinatória surgem em várias áreas práticas, tais como, desenvolvimento de circuitos digitais, diagnóstico técnico, cartografia, reconhecimento de padrões, planeamento de rotas, gestão e manutenção de linhas de produção, entre outros. Estes problemas têm características distintivas, como a sua resolução, que está ligada ao exame de elementos de um conjunto finito especificado pelos dados iniciais. Os problemas de optimização possuem um conjunto de soluções $S = \{s_1, \dots, s_n\}$ no qual é definida uma função de custo $c(s_i)$, sendo sempre necessária à sua resolução encontrar uma solução com a função de custo mínimo. Existem dois critérios para a avaliação dos algoritmos desenvolvidos e utilizados para a resolução destes problemas: a qualidade da solução e a sua complexidade. De acordo com o critério da complexidade os algoritmos são divididos normalmente em duas classes: polinomiais e exponenciais. Frequentemente, devido à sua complexidade ser muito elevada, os algoritmos exactos, que garantem a obtenção da solução exacta do problema, estão habitualmente impedidos no que diz respeito à sua aplicação prática. Por outro lado os algoritmos aproximados nem sempre levam à solução exacta mas asseguram uma aproximação bastante boa a esta.

A resolução de problemas combinatórios é bastante difícil especialmente quando é necessário encontrar uma solução óptima. Contudo, não é necessário desenvolver algoritmos eficientes teoricamente, uma vez que a complexidade dos dados iniciais em muitos problemas práticos é limitada, nesse sentido, os algoritmos apenas necessitam de ser eficientes na prática. Na resolução de muitos problemas combinatórios os requisitos de qualidade da solução podem ser diminuídos, frequentemente a obtenção da solução óptima não é necessária, portanto os métodos exactos podem ser substituídos pelos aproximados que são normalmente bastante mais rápidos. Em muitas situações práticas é possível aplicar métodos de redução que,

diminuindo a dimensão do problema em questão, reduzem essencialmente o número de variantes a rever. O desenvolvimento de métodos deste tipo com base na investigação de características de problemas concretos é a área essencial da aplicação da análise combinatória. Destaca-se assim conjuntos, grafos, matrizes e funções lógicas de vários modelos matemáticos utilizados no desenvolvimento de algoritmos combinatórios. É de salientar ainda que um modelo pode ser transformado noutra.

1.2 Proposta

Este projecto tem como objectivo o desenvolvimento e implementação metódica de um módulo pedagógico sobre problemas de Optimização Combinatória, problemas esses que surgem quando é necessário, seleccionar o melhor subconjunto que satisfaça determinados critérios a partir de um conjunto discreto e finito de dados.

Tendo como principal objectivo o auxílio do ensino dos principais conceitos e estratégias de resolução deste tipo de problemas, o módulo pedagógico poderá vir a ser utilizado no apoio a aulas de exposição ou em regime de auto-estudo. Será feita uma ilustração destes conceitos e estratégias de resolução com base no problema do Caixeiro Viajante, também conhecido por TSP. Este Módulo Pedagógico terá a particularidade de poder ser acedido localmente ou via Web, facilitando assim a sua utilização e tornando-o acessível a qualquer pessoa interessada. Terá que ser intuitivo e será dada particular atenção a questões de interactividade. Desta forma elimina-se a probabilidade de não ser utilizado devido à sua complexidade.

A introdução dos principais conceitos e estratégias de resolução para este problema (Solução Admissível, Vizinhança, Solução Óptima, Ótimo Local, Técnicas Exactas, Bounds, Heurística Construtiva, Estrutura de Vizinhança, Pesquisa Local, Meta-heurísticas) será ilustrada recorrendo a pequenos exemplos didácticos, adicionando-se sempre que necessário uma lista de referências a textos de apoio e a hiperligações externas.

A selecção dos algoritmos a implementar, bem como a respectiva implementação deverá privilegiar os aspectos pedagógicos em detrimento da eficiência e da qualidade das soluções. Deverão no entanto também ser apresentadas referências e hiperligações para as implementações mais eficientes destes algoritmos.

Este Módulo deverá, apesar da sua complexidade interna, ser bastante intuitivo e simples aos olhos do utilizador comum, para que este possa tirar o maior partido das suas potencialidades.

1.3 Estrutura do documento

Neste capítulo podemos encontrar o contexto do trabalho, a proposta efectuada e a estrutura deste documento.

O contexto apresenta de uma forma sucinta os temas abordados ao longo dos vários capítulos. A proposta consiste nos requisitos e objectivos para este trabalho

O capítulo dois encontra-se organizado em quatro secções. Encontra-se um breve historial sobre o tema em questão na primeira secção deste capítulo. A segunda secção refere-se à investigação efectuada para a construção do módulo. Posteriormente, a pesquisa sobre o conteúdo do mesmo compõe a terceira secção. A última secção reporta as mais relevantes implementações já existentes no mercado.

O trabalho prático realizado pode ser diferenciado em três pontos em que os dois primeiros são os constituintes do capítulo três e o terceiro corresponde ao quarto capítulo. No primeiro ponto temos a implementação da plataforma de suporte ao desenvolvimento do módulo e no segundo ponto o desenvolvimento e implementação do módulo em si e os seus algoritmos constituintes. No terceiro ponto temos a adaptação da interface entre o módulo e a plataforma onde este foi desenvolvido, bem como, a adaptação a uma plataforma diferente a fim de validar a integração do mesmo numa plataforma generalista.

O funcionamento do módulo do ponto de vista do utilizador é algo semelhante a um texto de ajuda ao qual se possa recorrer para tirar qualquer dúvida no uso do mesmo. Podemos encontrar este texto no capítulo cinco.

Por fim, o capítulo seis referencia alguns comentários sobre a análise, investigação e implementação efectuada ao longo deste projecto, assim como, algumas sugestões a considerar para futuras pesquisas e melhoramentos.

Capítulo 2 - Estado da arte

2.1 Ensino informatizado

O ensino informatizado já é tratado como objecto de estudo há alguns anos. A melhor forma de assimilar e aplicar conceitos é percebendo como esses conceitos surgiram. A sequência de tópicos que se segue ajuda a perceber esses conceitos em que alguns deles ainda hoje são aplicados e que é algo já generalizado por variadíssimos autores, como por exemplo o professor Henrique de Oliveira [1] e António Silva [2].

2.1.1 Como surgiu

Os computadores com a arquitectura tal como a conhecemos são algo recentes. Muito anterior a isso já existiam “máquinas de computar” ou, em melhor português, ferramentas de cálculo. Umas analógicas, como por exemplo o ábaco, outras mecânicas tal como a famosa máquina de pascal que utilizava engrenagens para fazer somas e multiplicações. [3]

Os primeiros computadores digitais electrónicos surgiram após a segunda Guerra Mundial com a finalidade de efectuar cálculos de balística. O Electronic Numerical Integrator and Computer (ENIAC) (Figura 1) que foi um desses computadores conseguia a proeza de efectuar cálculos em 30 segundos que até à data demoravam cerca de 12 horas com as calculadoras manuais. [4]

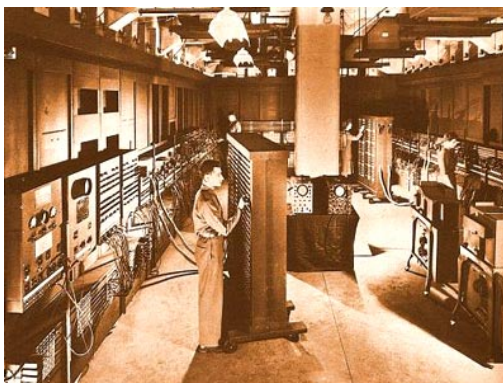


Figura 1 - Electronic Numerical Integrator and Computer.

Em 1950 ninguém pensava em ter um computador em casa, pois este ocupava várias salas, o seu propósito era inteiramente virado para problemas militares e não tinha uso útil para o cidadão comum.

Com a evolução da tecnologia, surge nos anos 50 a criação do “chip”. Este permitiu que as grandes dimensões dos computadores fossem reduzidas em grande escala. Em 1971 aparecia o primeiro micro processador produzido pela Intel a pedido da Busicon. [5]

Com estes dois avanços tecnológicos, a IBM viria a alterar para sempre a forma como olhávamos para o computador em 1981, com a introdução do conceito de PC (Figura 2 Erro! A origem da referência não foi encontrada.) e a transformação de uma enorme máquina de guerra numa pequena e compacta máquina de trabalho e diversão para toda a família. [4][6]



Figura 2 - Primeiro PC da IBM.

O ensino da informática teve a sua origem com as máquinas de ensinar, ideia introduzida nos anos 20 para corrigir testes de escolha múltipla.

Em 1954 foi introduzido o conceito de instrução programada utilizando a “máquina de ensinar”.

A instrução programada baseia-se na ideia de dividir para conquistar. Esta consiste na divisão do conteúdo a ensinar em pequenas porções de informação, chamadas de módulos,

sequencialmente estruturadas para que o aluno possa adquirir esse conhecimento em pequenos passos facilitando assim a aprendizagem do mesmo. [7]

A “máquina de ensinar” (Figura 3) foi uma invenção de Burrhus Frederic Skinner (Figura 4) que aplica o conceito de instrução programada. Esta máquina obriga o aluno a responder a uma pergunta no final de cada módulo obtendo um *feedback* imediato positivo ou negativo sobre a sua resposta. O aluno não conseguirá avançar para o módulo seguinte enquanto não responder correctamente à pergunta que lhe é colocada. Esta nova forma de ensino trouxe duas grandes vantagens. Uma dessas vantagens foi a motivação que esta dá ao aluno uma vez que este não tem que esperar, na melhor das hipóteses, um dia para saber se as suas respostas estão certas. Não só o facto de o aluno saber que a sua resposta está correcta se traduz sempre num factor de motivação mas também ao eliminar essa ansiedade faz com que este tenha uma melhor capacidade de concentração. [8]



Figura 3 - “Máquina de ensinar”



Figura 4 - Burrhus Frederic Skinner

A instrução programada era apresentada no formato impresso e foi muito usada no final da década de 50 e início dos anos 60, contudo, esta metodologia não adquiriu muita popularidade devido à dificuldade apresentada à produção de material didáctico e ao facto dos materiais existentes não possuírem padronização dificultando assim a sua disseminação.

Durante os anos 60, diversos programas de instrução programada foram implementados no computador ao invés da “máquina de ensinar” que era totalmente mecânica e assim surgiu então a instrução auxiliada por computador, também conhecida como CAI (Computer Assisted Instruction).

A criação do chip originou a produção em massa, a grande propagação nas escolas e a diversificação de tipos de CAI tais como tutoriais, programas de demonstração, exercício e prática, avaliação do aluno, jogos educacionais e simulação.

A utilização do computador no ensino abriu também outros horizontes em relação à sua utilização. Passou a ser uma ferramenta de apoio a quase tudo, não só no ensino directo ao aluno mas também na preparação desse ensino entre muitos outros exemplos tais como a manipulação de dados ou controlo de processos em tempo real.

Analisando a utilização do computador no ensino directo ao aluno, este passa então a ser visto como uma ferramenta de apoio que é caracterizado apenas como uma versão computadorizada dos métodos tradicionais já existentes no ensino. No entanto existem algumas diferentes formas do seu uso sendo as categorias mais comuns os tutoriais, exercícios e prática, jogos e simulação.

Estes programas tutoriais não passam de uma nova versão da já conhecida instrução programada, se bem que apresentam algumas vantagens. Pelo facto de ser uma versão informatizada vai tirar partido do computador utilizando alguns recursos que não se encontram numa folha de papel, tais como som e animação tornado o seu conteúdo mais apelativo perante o seu utilizador [9] [10] [11]. No entanto continua a ter as suas desvantagens. No caso do aluno, este não recebe a mesma atenção e apoio como seria de esperar que recebesse de um professor. Do ponto de vista tecnológico, estes programas são bastante grandes e consomem grandes recursos computacionais. Foi feito um esforço para eliminar outra desvantagem deste tipo de programas. A sua evolução foi no sentido de criar tutoriais que utilizassem as técnicas de Inteligência Artificial (IA) uma vez que a grande maioria destes programas eram desprovidos de qualquer técnica pedagógica e não tinham interacção com o aluno além de este responder a perguntas sobre textos que tinham lido previamente. As técnicas de IA eliminavam essa desvantagem ao conseguirem analisar padrões de erro, avaliar o estilo e a capacidade de aprendizagem do aluno além de que ofereciam uma instrução especial sobre o conceito em que o aluno estivesse a apresentar dificuldade [12].

Programas de exercício e prática consistiam em *softwares* que eram utilizados para a revisão da matéria dada através de memorização e repetição. Estes eram apresentados em formas de jogos educacionais tentando tirar o máximo partido do computador tanto em características gráficas como sonoras e em que a resposta do aluno tinha imediatamente o seu aval ou não [11]. Esta forma de ensino apresenta vantagens não só para o professor como também para o aluno. Desta forma o professor passa a dispor de uma grande variedade de exercícios para apresentar ao aluno de acordo com o nível de dificuldade mais adequado ao mesmo. Por outro lado, o aluno motiva-se mais a aprender olhando para o ensino como um jogo, facilitando esse processo de forma agradável e divertida.

Outro tipo de programas de ensino é os simuladores. O objectivo deles é tratar de situações semelhantes à realidade mas de forma simplista para melhor compreensão por parte do

utilizador [11]. Esta modalidade permite desenvolver hipóteses, testá-las, analisar resultados e refinar os conceitos.

Com o evoluir da tecnologia e desenvolvimento dos recursos computacionais, passou a ser possível integrar textos, imagens, som, animação assim como a interligação da informação de forma não linear, implementando o conceito de multimédia e hipermédia. Com todos os recursos apresentados, o computador era então considerado uma ferramenta educacional mais eficiente do que a “máquina de ensinar” utilizada para promover a aprendizagem.

Desde os anos 60 que se utiliza o computador para fins educacionais, no entanto, só na década de 70 é que surge a IA. Surge então os Intelligent Computer- Assisted Instruction (ICAI) que utilizam a técnica já conhecida dos CAI com a IA. Este tipo de programas viera dar o apoio ao aluno de forma mais personalizada uma vez que o programa se adapta às necessidades do aluno criando o grau de dificuldade adequado. O ICAI foi um dos primeiros *softwares* inteligentes a aparecer dentro de um grupo de programas denominado Sistemas Tutoriais Inteligentes (STI). [13]

É importante perceber que a forma de ensino tem que ser modelado consoante algumas características. Como por exemplo, a informação ou conteúdo que se pretende transmitir ou ensinar a um público-alvo ou uma turma que se pretende atingir vai ter um enorme impacto na técnica ou pedagogia de ensino de forma a alcançar uma maior eficiência.

Hoje em dia existem muitas e diferentes abordagens com o objectivo de aumentar a eficiência do ensino. Porém quase todas assentam em duas grandes perspectivas, o que se pretende do aluno e a forma de uso computador como ferramenta.

Dado à grande quantidade de informação que existe e que vai crescendo de dia para dia, o ensino desta informação também teve que evoluir. Nesse sentido, é fácil de perceber o porquê de, a meio do processo de ensino, este ser dividido e subdividido em áreas de especialização. Por outro lado, a tendência do ensino começa a ser cada vez mais a prática de busca e uso da informação ao invés do método tradicional da memorização da mesma.

Existem duas grandes formas de abordar o uso do computador no ensino que influencia muito o *software* a utilizar. Por um lado, temos a forma simplista de ver a utilidade do computador de forma a auxiliar e simplificar a tarefa de ensinar realizada pelo professor, por outro lado temos a hipótese de dar ao aluno o controlo de poder construir de sua livre vontade o conhecimento que pretende adquirir para o seu futuro profissional ou apenas por sua simples curiosidade.

Um dos melhores exemplos que temos na actualidade é as plataformas de *e-learning*. Estas conseguem ser tão versáteis ao ponto de abranger quase todas as abordagens que existem no ensino.

2.1.2 Ensino via internet

O ensino via internet pressupõe à partida um ensino à distância. No método tradicional temos um professor a ensinar o aluno durante uma aula. Se o ensino for à distância este método torna-se um pouco mais complicado de acontecer, pois na grande maioria dos casos não existe um professor ou orientador durante o ensino propriamente dito. Pretende-se então atingir os alunos autodidactas que recorrem ao método tradicional, falar directamente com um professor, apenas para preencher uma ou outra lacuna que ficou.

Comparando com o método tradicional, facilmente se percebe que esta forma traz grandes benefícios tanto para o aluno como para o professor. O aluno não tem horário fixo para ir às aulas, pode ir avançando ao seu ritmo à medida que vai assimilando a informação, entre outros. Por outro lado, o professor fica com o tempo muito mais livre e poderá aproveitá-lo de forma a poder acompanhar as lacunas de cada aluno individualmente. Para além destas vantagens, ainda se pode realçar o facto dos conteúdos disponibilizados poderem ser actualizados em qualquer altura. Porém existe algumas desvantagens, como por exemplo, este método requer uma maior força de vontade por parte do aluno e um maior esforço por parte do professor na preparação dos conteúdos disponibilizados. [14]

Outro aspecto muito importante do ensino via internet é a sua apresentação, a forma como este é estruturado e a sua visualização gráfica. Após alguma investigação, verifica-se que as regras para a criação da parte gráfica não são propriamente rígidas. A grande maioria das regras utilizadas para a criação de uma página de internet é apenas de senso comum. Regras como “a página tem que ser de fácil compreensão e ter uma boa navegabilidade” e “a página tem que ser de fácil acesso e rápida no carregamento” [15] são algumas das quais se pode encontrar em diversos sites onde é o próprio utilizador a falar do que gosta, pois se a página não for apelativa ao utilizador, este não voltará [16]. Como exemplo, no caso de um menu, rapidamente se percebe que a fácil compreensão e boa navegabilidade passa por ter um menu relativamente curto mas com submenus.

2.2 Módulo Pedagógico

2.2.1 Definição de Módulo Pedagógico

O módulo pedagógico destina-se ao ensino do aluno podendo este ser avaliado através do mesmo.

O seu ambiente gráfico deve ser elaborado tendo em conta a informação que se pretende transmitir e o público-alvo que se pretende atingir. No caso concreto deste trabalho, pretende-se fazer simulações de heurísticas aplicadas ao TSP e passar essa informação a alunos já com alguma experiência em informática do ponto de vista do utilizador e com alguns conhecimentos base em investigação operacional.

Este módulo como o próprio nome indica, deve de ser portátil de forma a poder ser integrado em qualquer plataforma de “e-learning”

2.2.2 Análise de metodologia de programação

O módulo em si, abstraído do seu conteúdo e da plataforma onde se encontra, consiste numa interface configurável que tem como objectivo a integração de qualquer conteúdo em qualquer plataforma. Para a implementação do módulo é necessário analisar o tipo de conteúdo e a sua finalidade para adequar a visualização e o tipo de programação a implementar.

A visualização ou GUI é um dos aspectos mais importantes em qualquer programa que se pretenda desenvolver. A criação de bons GUIs é uma habilidade crítica que surgiu na década de 90. Um bom GUI requerer a compressão e aplicação de alguns princípios básicos por parte do criador, incluindo fazer do aspecto gráfico algo com que o utilizador gostará de trabalhar todos os dias. É mais fácil de perceber isto se fizermos a analogia com o videogravador. Verifica-se que o modelo construído em 1985 tem uma enorme quantidade de botões na face do aparelho, muitos dos quais continuam ainda nos dias de hoje um mistério uma vez que o manual de instruções se perdeu algures ao longo do tempo. O modelo de 1995 contrariamente ao de 1985 apenas terá na face os botões essenciais ao desempenho natural do aparelho mas continua a ter uma grande quantidade deles escondidos atrás de um painel retráctil. Apesar de o videogravador ter caído em desuso, os que ainda se encontram no mercado substituíram a grande quantidade de botões por um menu virtual intuitivo. Assim como no caso do

videogravador, um programa só deve de ter na parte principal os botões essenciais à sua navegação.

Qualquer utilizador comum acaba por ficar a olhar distraidamente para o ecrã à espera da conclusão de uma determinada operação e sabe-se a frustração que é olhar para uma página que tenha um pobre aspecto visual. Os utilizadores gostam bastante de saber o tempo de duração de uma determinada operação antes de poderem desfrutar dos frutos da sua paciência. Em geral, a maioria dos utilizadores apreciam ter uma janela que lhes indique o progresso de uma determinada operação quando esta vai demorar mais de sete a dez segundos.

Outra importante característica de um bom GUI é a velocidade, isto é, velocidade de resposta. Dependendo do tipo de aplicação, a velocidade pode ser um factor determinante na aceitação por parte da comunidade. Por exemplo, se a aplicação é voltada para processos de transacção online, uma performance lenta levará rapidamente ao abandono do sistema por parte dos utilizadores. Muitos aspectos respeitantes à velocidade são determinados a nível de programação do GUI e não do hardware.

São vários os tipos e técnicas de programação que existem hoje em dia. No caso concreto do ensino, essas técnicas centram-se em programação estruturada e programação modular. Na maioria dos casos pode conter uma programação orientada a objectos em simultâneo a fim de tornar a implementação mais eficiente.

A programação estruturada consiste num conjunto de técnicas que foram evoluindo ao longo do tempo visando o aumento da produtividade do programa e reduzindo o tempo do processo de identificação e eliminação de erros. Este tipo de técnica tem a grande vantagem de ser simplista, uma vez utiliza um conjunto limitado de regras. Esta programação tem como regras básicas a implementação das várias etapas do programa segundo um modelo hierárquico, a decomposição de acções complexas em conjuntos de acções mais simples e os três tipos de estruturas básicas em que se baseia. Estas estruturas são:

Estrutura sequencial - as acções são encadeadas umas nas outras em que a saída de uma coincide com a entrada da seguinte.

Estrutura selectiva - utiliza funções lógicas para determinar qual a acção que se segue mediante o resultado de saída da anterior.

Estrutura repetitiva - trata-se de um ciclo composto por uma serie de acções que é repetido um número pré-definido de vezes.

Na programação modular é feito um programa em secções em que apenas uma é principal e as outras todas secundárias. A secção principal é constituída por poucas instruções que fazem a chamada a procedimentos passando os dados necessários em forma de parâmetros. As secções secundárias são esses procedimentos que podem receber parâmetros externos e ter os seus próprios parâmetros produzindo um retorno após a sua execução.

A programação orientada a objectos serve para o manuseamento de estruturas de dados. A sua grande vantagem é a velocidade de execução do programa devido à reutilização dos

objectos. Um objecto é um conjunto de modelizações, atributos e restrições recorrendo a pequenas funções que são aplicadas a uma estrutura de dados pré-definida de forma a facilitar a interpretação dos mesmos.

2.3 Problemas Combinatórios

2.3.1 Conceitos

Os Problemas de Optimização Combinatória, devido à grande dificuldade de solução e sua presença em várias situações do quotidiano, têm atraído cada vez mais a atenção de pesquisadores de diversas áreas que têm envidado esforços para desenvolver algoritmos cada vez mais eficientes para serem aplicados a tais problemas. Os problemas de alocação, de roteamento e programação de horários são exemplos de situações onde POCs estão presentes.

Nestes problemas, o objectivo é atribuir valores a um conjunto de variáveis de decisão, de tal modo que uma determinada função objectivo seja optimizada (minimizada ou maximizada, dependendo do objectivo) atendendo um determinado conjunto de restrições. O problema do Caixeiro Viajante, da Mochila, da Cobertura Mínima por Conjuntos, da Árvore Geradora Mínima, da Árvore de Steiner e do roteamento de veículos são exemplos de problemas de Optimização Combinatória.

Uma forma de resolver tais problemas seria simplesmente enumerar todas as soluções admissíveis e guardar aquela de menor custo, considerando-se solução admissível, toda aquela que verifica o conjunto de restrições. Entretanto, essa é uma ideia ingénua, na medida em que, para a maioria dos problemas, esse método torna-se impraticável, uma vez que existe um elevado número de soluções admissíveis. Assim, mesmo que se utilize um super computador para resolver o problema de forma a obter uma solução óptima, sendo esta a solução de custo mínimo do conjunto das soluções admissíveis, o tempo de processamento pode ser de várias horas, dias ou até anos, dependendo do número de variáveis constituintes do problema. Portanto, técnicas computacionais mais apuradas são necessárias para resolver esses problemas.

Os Problemas de Optimização Combinatória têm sido utilizados como modelos para diversas situações reais, tanto nas suas versões com um único objectivo a ser optimizado como nas versões com múltiplos critérios. Grande parte desses problemas pertence à classe dos

problemas NP-árduos, o que significa que dificilmente, algum dia, serão apresentados algoritmos polinomiais que os solucionem.

Ao longo dos anos, diversas técnicas foram desenvolvidas para a construção de algoritmos exactos que resolvessem tais problemas como: Programação Dinâmica [19].

Entretanto, como os tempos para solucionar instâncias de grande porte por algoritmos exactos são inviáveis, a opção, nesses casos, seria o uso de algoritmos heurísticos. As heurísticas são algoritmos que não têm garantia de encontrar a solução óptima, ou seja, a melhor solução existente para o problema. No entanto, são capazes de retornar uma solução num tempo adequado para as necessidades da aplicação.

Existem diversas abordagens para a construção de algoritmos heurísticos.

Inicialmente, começou-se por utilizar heurísticas construtivas no desenvolvimento de tais algoritmos. Esta heurística consiste em ir criando a solução com o desenrolar do algoritmo com base no custo do próximo passo a dar sem ter em conta o custo dos passos seguintes e a sua influência para o custo total da solução. Estas heurísticas são conhecidas por técnicas ditas gulosas ou míopes devido à forma de como o algoritmo aborda o problema e seu desenrolar no processo de achar uma solução.

A heurística de melhoria, como o próprio nome indica, tem como propósito melhorar uma solução já existente. O algoritmo parte de uma solução previamente encontrada e consiste na análise combinatória de todos os elementos da solução inicial com a finalidade de encontrar uma de menor custo. Com esta heurística aparece o conceito de vizinhança e pesquisa local que, tal como a heurística construtiva, estes conceitos e designações advêm da abordagem do algoritmo ao problema.

Com o passar dos anos, técnicas mais gerais, conhecidas como meta-heurísticas, foram apresentadas. Estas técnicas, por meio de adaptações, podem ser usadas para vários problemas [20].

As meta-heurísticas são paradigmas de desenvolvimento de algoritmos heurísticos.

Uma meta-heurística é definida por uma estrutura genérica que deve ser seguida, adaptando suas partes de acordo com o problema que se pretende resolver. Diversas propostas de meta-heurísticas surgiram nos últimos anos.

Nas últimas décadas, diversos softwares vêm sendo desenvolvidos para um grande número de aplicações onde são utilizados algoritmos de optimização. Em muitos casos, existe um tempo limitado para que o algoritmo embutido no sistema ou software encontre uma solução de “boa” qualidade. Desse modo, é importante que se possa avaliar o equilíbrio entre os factores qualidade versus tempo na geração de uma solução para o problema em questão.

2.3.2 Caso de estudo - TSP

O PCV ou TSP tem servido de plataforma de teste para a investigação de diversas ideias algorítmicas, porque, além de ser um problema de larga aplicabilidade no mundo real, é de fácil compreensão e descrição, mas de difícil solução, pois pertence à classe de problemas NP-Árduos. Por esses motivos, vários métodos têm sido desenvolvidos com o propósito de se resolver instâncias cada vez maiores desse problema e em menos tempo. Este problema começou por ser tratado, mas com algumas divergências no início do século XIX pelo matemático irlandês Sir William Rowan Hamilton e pelo britânico Thomas Kirkman. Podendo ser encontrada em Graph Theory 1736-1936 [21] uma discussão dos primeiros trabalhos de Hamilton e Kirkman.

Apenas em 1920 viria a ser estudado o problema do caixeiro viajante como o conhecemos, por Karl Menger em Vienna e Harvard, reaparecendo passados dez anos em grupos de estudo da Universidade de Princeton. Na década de 1940, foi estudado por estatísticos que ligaram este problema com aplicações na agricultura, tendo sido popularizado pelo matemático Merrill Flood entre os seus colegas da corporação RAND na Califórnia.

Os algoritmos heurísticos em detrimento dos algoritmos exactos visam obter uma resposta aceitável num menor intervalo de tempo. Nem sempre sendo óptima, esta resposta é porém aceite devido ao ganho de tempo de CPU que normalmente compensa esta diferença de apenas alguns pontos percentuais na média.

O TSP consiste habitualmente num grafo $G = (N,E)$ onde $N = \{1, \dots, n\}$ é o conjunto de vértices e $E = \{1, \dots, m\}$ é o conjunto de arestas de G .

Os custos c_{ij} estão associados com cada aresta ligando os vértices i e j .

O problema consiste em localizar o menor ciclo Hamiltoniano do grafo G , sendo o tamanho do ciclo calculado pelo somatório dos custos das arestas que formam o ciclo. Os vértices do grafo são, habitualmente, referidos como "cidades" sendo o objectivo visitar todas as cidades passando apenas uma vez por cada uma, retornando ao ponto de origem. Este percurso deve ser feito de forma a minimizar a distância total percorrida.

O TSP pode ainda ser classificado como simétrico, se para todos os pares de vértices $\{i,j\}$, os custos c_{ij} e c_{ji} forem iguais, considerando-se assimétrico caso isto não se verifique. Pode ser encontrada uma revisão histórica mais completa sobre o TSP em Hoffman e Wolfe [22]. Em 1990, Reinelt criou uma biblioteca contendo várias instâncias do TSP que têm vindo a ser testadas e discutidas na literatura. Conhecida como TSPLIB, essa biblioteca contém mais de 100 exemplos com tamanhos que variam de 14 até 85.900 cidades.

Tabela 1 - Algumas instâncias da biblioteca TSPLIB [23].

Ano	Instância	Tamanho	Pesquisadores
-----	-----------	---------	---------------

1954	dantzig42	49	Dantzig, Fulkerson e Johnson
1962	aleatória	69	Held e Karp
1974	aleatória	67	Camerini, Fratta e Maffioli
1980	gr120	120	Grötschel
1980	lin318	318	Crowder e Padberg
1987	att532	532	Padberg e Rinaldi
1991	gr666	666	Grötschel e Holland
1991	pr2392	2392	Padberg e Rinaldi
1995	pla7397	7.397	Applegate, Bixby, Chvátal e Cook
1998	usa13509	13.509	Applegate, Bixby, Chvátal e Cook
2001	d15112	15.112	Applegate, Bixby, Chvátal e Cook
2004	sw24798	24.978	Applegate, Bixby, Chvátal, Cook e Helsgaun

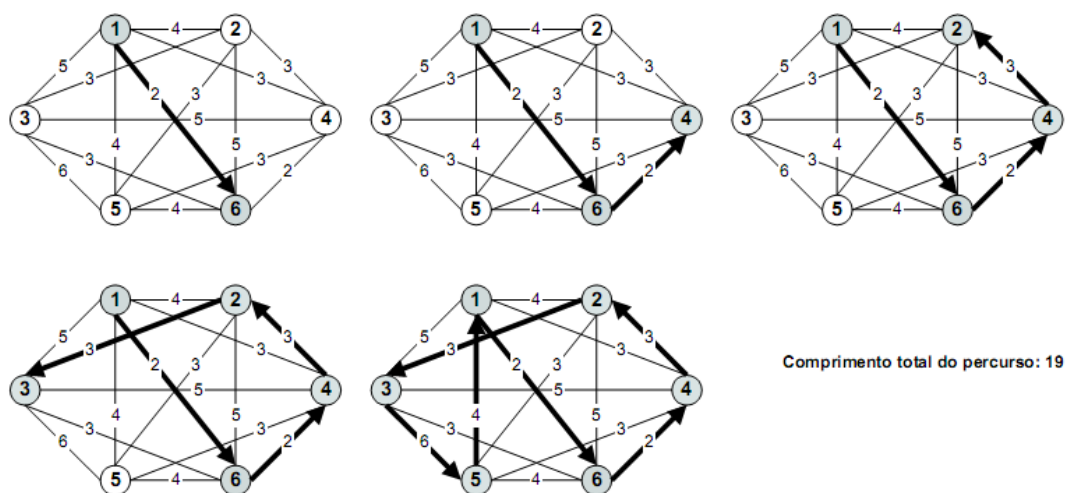
Apesar das instâncias ainda não solucionadas, é notável o avanço que se tem alcançado na solução do TSP nos últimos anos. A tabela 1 mostra a evolução na solução de instâncias do PCV a partir de 1954 quando Dantzig, Fulkerson e Johnson resolveram uma instância com 49 cidades até 2004 quando Applegate, Bixby, Chvátal, Cook e Helsgaun solucionaram uma instância com 24.978 e provaram que não existe um percurso menor. Essa instância é, actualmente, a maior instância do Problema do Caixeiro Viajante já resolvida.

Necessário à resolução deste problema é o uso de determinados algoritmos que não só a tornam possível como também a tornam rápida e eficaz.

Em baixo são descritos alguns dos algoritmos mais importantes à resolução do TSP:

Vizinho mais próximo

Partindo de uma cidade inicial, consiste na procurada da cidade que se encontra mais próxima, uma vez nesta faz-se o mesmo processo até se ter passado por todas as cidades e voltado à inicial sem nunca se ter passado duas vezes pela mesma cidade. Este algoritmo baseia-se numa heurística construtiva e temos um exemplo simples na figura 5.

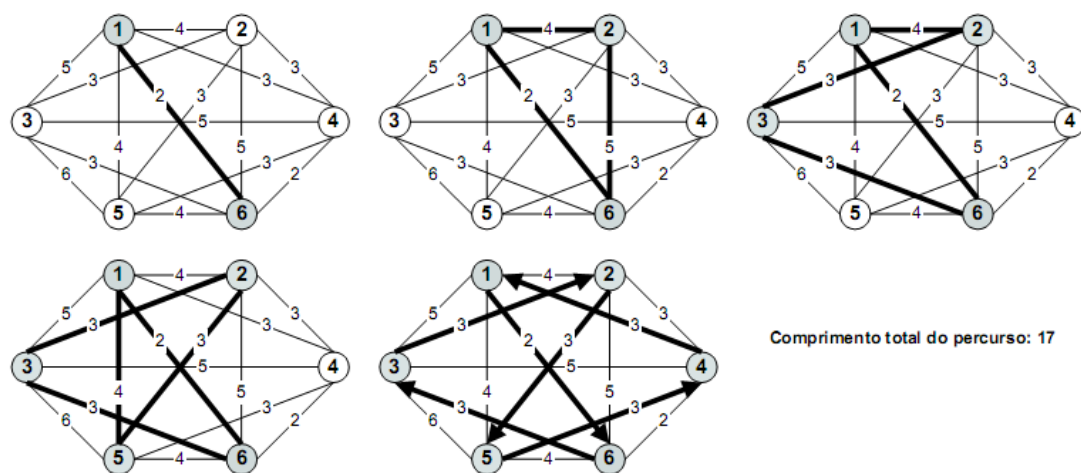


Comprimento total do percurso: 19

Figura 5 - Vizinho mais próximo [24].

Inserção mais próxima

Começa com um circuito parcial de três cidades aleatórias e vai inserindo as cidades que faltam, uma a uma escolhida aleatoriamente, na aresta mais próxima do circuito parcial já existente como é mostrado na figura 6. Trata-se, tal como o vizinho mais próximo, de uma heurística construtiva.



Comprimento total do percurso: 17

Figura 6 - Inserção mais próxima [24].

2opt

Este algoritmo parte de um circuito inicial já existente e trata-se de uma heurística de melhoria. Se nesse circuito retirarmos duas arestas, apenas duas soluções restam para completar o circuito, a solução que já se tinha e a hipótese de se trocar as arestas mas tendo em atenção para que não se formem dois circuitos isolados, tal como ilustra a figura 7. O processo fica completo após terem sido feitas todas as trocas possíveis entre todos os pares de arestas não adjacentes constituintes do circuito.

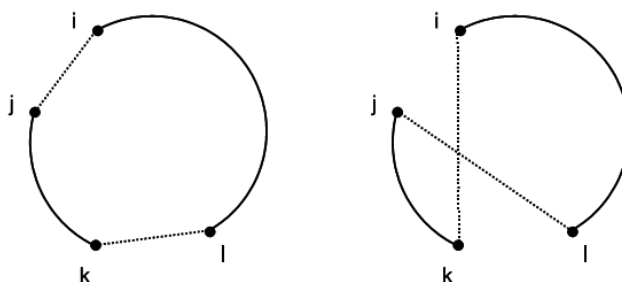


Figura 7 - 2opt [24].

Grasp

Esta meta-heurística junta uma heurística construtiva com uma de melhoramento sendo feitas várias simulações. Começa por utilizar uma pequena variante do vizinho mais próximo aplicado em seguida o 2opt, o algoritmo consiste nestes dois passos executados várias vezes. A variante do vizinho mais próximo trata-se de escolher uma solução aleatoriamente dentro das “n” melhores soluções, este “n” tipicamente assume o valor de 5% do número total de cidades, podendo ser ajustado para um caso em particular a fim de se obter uma melhor performance do algoritmo.

Para além destes algoritmos acima referido existem ainda outros mais complexos e por sinal mais fiáveis, ainda que mais lentos. Fazem parte deste grupo, de entre muitos outros, os seguintes algoritmos:

Helsgaun

Este algoritmo consiste em modificar as funções de distância baseadas na lista de vizinhos substituindo todas as sequências de movimentos 2opt por sequências de movimentos de 5opt, tornado possível pelo uso de listas de pequenas vizinhanças.

Geni

Este algoritmo pode ser visto como uma heurística construtiva, com a diferença de que após cada inserção é feito um movimento de 4 ou 5opt. O parâmetro p representa o tamanho da lista de vizinhança utilizada. O valor $P=10$ foi escolhido como melhor relação entre o tempo de execução do algoritmo e a qualidade do percurso obtido.

Tabu

Esta meta-heurística aumenta a performance de um outro algoritmo de procura local através do uso estruturas de memória. Após uma potencial solução ter sido encontrada esta é marcada como um tabu para que o algoritmo não repita essa possibilidade.

2.4 Implementações existentes

2.4.1 Plataformas

Uma plataforma é algo que fornece suporte e ferramentas de apoio ao que se pretende utilizar para a disponibilização de informação. No caso concreto, falamos de um gestor de conteúdos informático, que incorpora ferramentas de apoio aos módulos que se pretendem utilizar em que cada módulo disponibiliza um conteúdo.

A plataforma que melhor explica este conceito é o “php-nuke”. Trata-se da construção de uma página de internet dividida em módulos com as mais diversas finalidades. Cada módulo constituinte da página corresponde a um grupo de funcionalidades da mesma. A escolha e integração de cada módulo numa determinada página dependem apenas da finalidade com que esta é construída.

Para este trabalho, será necessária uma plataforma que contenha ferramentas de gestão de conteúdos, ou seja, uma plataforma de “e-learnig”. Hoje em dia, este tipo de plataformas já é usada na grande maioria das escolas e faculdades. Apesar da existência de algumas plataformas deste tipo no mercado, existe uma que teve maior sucesso. Essa plataforma chama-se Moodle **Erro! A origem da referência não foi encontrada.** e o seu sucesso deve-se, não só ao facto de ter sido uma das primeiras a ser criada mas também, à sua simplicidade e fácil manuseamento.

2.4.2 Módulos do Moodle

O Moodle apresenta dois grandes tipos de módulos na sua constituição, os específicos e os genéricos.

Os específicos são aqueles módulos que fazem parte do Moodle e que podem ser substituídos ou alterados mas nunca eliminados. Estes normalmente alteram pequenos detalhes de visualização, como por exemplo, o tema ou a linguagem.

Os módulos genéricos são os que são integrados com um conjunto de funcionalidades adicionais às já existentes e que podem ser encontrados na internet para uso gratuito com as mais diversas finalidades.

2.4.3 Algoritmos

Os algoritmos baseados no TSP são tipicamente pesados, ou seja, é efectuada uma grande quantidade de cálculos. Devido a este problema, a grande maioria dos algoritmos que existem já implementados são programas que correm localmente apesar de se conseguir encontrar alguns que correm directamente numa página de internet.

Facilmente se encontra alguns algoritmos desenvolvidos em Java disponíveis na internet para uso directo ou para guardar o programa. Apesar de estes poderem ser integrados numa página de internet para uso directo, têm a desvantagem de depender fortemente da máquina que estamos a utilizar [26][26].

Um exemplo de utilização directamente numa página de internet com os algoritmos “vizinho mais próximo” e “2-opt” é uma implementação em “java applet” mas que continua a ter a desvantagem de depender da máquina em que está a ser visualizado e ainda contém algumas restrições dentro do próprio algoritmo a fim de tornar este mais rápido [27].

Para algoritmos mais complexos, a integração numa página de internet torna-se impraticável e resta mesmo a solução de guardar localmente para depois o poder utilizar. Temos como exemplo um algoritmo mais complexo (algoritmo genético) que também é implementado em “java applet” mas para correr à parte da página [28].

2.5 Alternativas de implementação

Com a grande evolução da informática que se deu nas últimas décadas do ponto de vista do ensino podemos dar alguns exemplos de sucesso de ferramentas e seus suportes informáticos.

A famosa ferramenta FLIP que incorpora um dicionário de língua portuguesa, surgiu no mercado no início dos anos 90 ainda em disquetes e é um desses exemplos embora estas já estejam um pouco desactualizadas. Pouco tempo depois surgiu o Encarta, em CD multimédia, um produto da Microsoft destinado a albergar uma base de dados de informação sobre os mais variados temas como história ou música. Hoje em dia, os CDs multimédia são muito utilizados e são relativamente fáceis de fazer, tipicamente com a linguagem baseada em “flash”, com o programa proprietário “Director” da Macromédia.

Com o aparecimento da internet, apareceram páginas de internet dos mais variados temas que foram evoluindo com o passar do tempo. Do ponto de vista do ensino, evoluiu imenso desde as primeiras páginas que apareceram que se resumiam apenas a uma forma fácil de passar

informação em texto até às plataformas de “e-learning”, já bastante utilizadas em muitas escolas actualmente.

Olhando um pouco para o passado e avaliando o presente de forma a prever um pouco o futuro, dá para perceber que as plataformas de “e-learning” ainda estão em desenvolvimento e que precisam de uma maior divulgação. A plataforma mais utilizada e desenvolvida que existe actualmente é o Moodle.

Apesar de já existir uma grande variedade de módulos que possam ser integrados neste tipo de plataforma, ainda não se encontra nenhum facilmente com o tema do conteúdo proposto para este projecto.

A escolha de utilização de uma linguagem de programação em vez de outra quase sempre depende muito do gosto do programador e pouco do que se pretende realizar. Para não cair na tentação, foi feita uma análise das linguagens de programação mais utilizadas com o intuito de escolher a mais adequada.

Analisando apenas o conteúdo do módulo que é proposto abordar é fácil de perceber que, de entre as várias linguagens de programação existentes, a que se torna mais adequada seria a programação em “C” devido à sua rapidez de execução de simulações sendo esta uma das questões fundamentais dos problemas de optimização combinatória. Por outro lado, a linguagem mais adequada à criação de páginas de internet é HTML. Então o ideal seria escolher uma linguagem que tenha algumas propriedades da linguagem “C” mas que possa ser apresentada em HTML. A única limitação que existe na escolha da linguagem de programação é que esta tem de ser uma linguagem WEB, o que não ajuda muito, dada a sua diversidade. Assim sendo, enumero apenas as mais utilizadas: HTML, ASP, PHP, Perl, C#, Flash, Java Script e Java Applet.

O HTML é a forma básica de apresentar qualquer página de internet. Todo e qualquer programa de visualização de páginas de internet, mais conhecido por “browser”, consegue interpretar a linguagem HTML. Um assunto polémico é que nem todos pensam que as outras linguagens de programação necessitam de fazer a sua representação WEB através do HTML. Como por exemplo o “flash” que pode ser apresentado sem o apoio de codificação em HTML do ponto de vista do programador. Na realidade o “browser” tem que ter a capacidade de interpretar o “flash” e apresentá-lo na página como um objecto de HTML mas este procedimento é interno ao “browser” e independente do programa “flash”.

A linguagem “flash” está virada para a animação podendo também conter interacção com o utilizador mas muito pobre no que toca à criação de programas de cálculo.

O “Java Applet” é usado para a construção de pequenos programas para a WEB. É baseado na codificação “Java” mas peca devido às suas limitações, nomeadamente a obrigatoriedade de um interpretador integrado no “browser” do utilizador. No entanto, encontram-se alguns programas nesta linguagem que demonstram o funcionamento de algoritmos de problemas combinatórios mas com uma grande limitação na introdução de dados na demonstração.

O "Java Script" também é baseado em linguagem "Java" e foi criada essencialmente para interagir com a linguagem HTML para fazer a validação de formulários e uso de expressões regulares. Hoje em dia, permite muito mais que isso mas sempre na perspectiva de interagir com a página a ser apresentada e não com a implementação de programas de cálculos [17][18].

O "Perl" é uma linguagem idêntica ao PHP do ponto de vista de programação. É uma linguagem extremamente poderosa no manuseamento de texto. A sua grande desvantagem é que necessita de compilação, o que implica ter o conhecimento prévio da máquina onde vai estar a correr o programa.

O "C#" é uma linguagem de programação orientada a objectos. Visa a simplicidade da linguagem e a generalização de objectivos. A sua desvantagem, tal como o "Perl", é a necessidade de compilação do código.

Ficando apenas com duas hipóteses, temos o PHP e o ASP como linguagens idênticas a nível de resultados práticos. O ASP é uma linguagem desenvolvida pela Microsoft em que uma grande parte do seu desempenho se baseia em componentes que só funcionam no "Internet Explorer", ou seja, no "browser" da Microsoft.

A escolha da linguagem PHP advém do facto da linguagem conseguir reunir algumas das características essenciais à programação que se pretende executar. Esta linguagem tem como vantagens: a sua rapidez de execução, é interpretada ao invés de compilada, considera a opção de programação orientada a objectos e tem como propósito a construção de páginas WEB dinâmicas. A sua maior desvantagem é a facilidade de cometer erros na inicialização de variáveis e imposição de um tipo nas mesmas.

A apresentação de resultados das simulações é algo que faz a diferença do ponto de vista do utilizador. Existem várias linguagens dedicadas à representação gráfica, como por exemplo o PHP ("phpplot"), SMIL, SVG, "flash" entre outras. Grande parte destas linguagens fora excluída devido a serem incompatíveis ou de difícil integração em HTML. A decisão de escolha residiu essencialmente entre SVG e uma biblioteca chamada "phpplot" que usa a linguagem PHP e é dedicada à construção de gráficos. Foi escolhida a linguagem SVG pela forma como é composta, trata-se quase de uma descrição textual da imagem a apresentar, é facilmente gerada pelo PHP com os parâmetros que se pretende passar e é de fácil integração em HTML podendo ser guardado como uma imagem.

Capítulo 3 - Implementação

3.1 Plataforma

Para a criação de um módulo é necessário uma plataforma e essa plataforma é suportada por uma linguagem de programação. Após a escolha do PHP como linguagem de programação, há que escolher uma plataforma. No caso concreto, o PHP é usado tipicamente para a apresentação de páginas de internet dinâmicas. Assim sendo, foi projectada e criada uma plataforma de propósito para este trabalho.

Tendo uma plataforma conhecida é mais fácil o desenvolvimento de um módulo para a mesma, uma vez que se trata de uma ambiente controlado. Esta trata-se de uma página de internet que foi criada de raiz seguindo algumas regras básicas comuns às páginas de estruturação modular.

Uma das coisas fundamentais à implementação de uma página é a sua versatilidade, ou seja, que permita rapidamente mudar certos aspectos, tais como, a linguagem, aspectos de visualização e inserir novas funcionalidades.

A fácil inserção de novas funcionalidades só depende da estrutura da programação utilizada. Neste trabalho, foi usada uma estrutura modular de forma a privilegiar a dinâmica da página. Pode-se dizer que a página é constituída por blocos com uma moldura base que suporta os mesmos, sendo cada bloco considerado um módulo independente como por exemplo um calendário.

Esta página contém uma secção de definições da própria que permite, entre outras coisas, a mudança da linguagem. A escolha desta é feita através de uma caixa de opções que apresenta todas as linguagens disponíveis para a página. Para que seja fácil a inserção de novas línguas, a implementação consiste numa pasta que contém os ficheiros correspondentes a cada língua disponível para a página. Este ficheiro tem um nome intuitivo como por exemplo "lang_pt.php" e é constituído por uma série de variáveis especiais e globais que correspondem a pedaços de

texto que são utilizados em toda a página. Dentro de cada módulo constituinte da página deve de existir um ficheiro de linguagem correspondente ao módulo em questão.

Foi implementado também o conceito de tema. Pretende-se com isto a fácil e rápida mudança do aspecto e posição relativa dos vários constituintes da página. Isto é conseguido através de um conjunto de ficheiros CSS com uma pasta adicional de imagens. Os ficheiros CSS mudam os aspectos relativos à formatação bem como a posição de cada espaço utilizado na página e a pasta de imagens corresponde aos contornos de tabelas, logótipo, etc.

Na figura 8 temos o quadro de definições da página do ponto de vista do administrador. Aqui podemos ver algumas das definições a cima descritas, salientando a diferença entre “pessoal” e “público”, em que as definições são aplicadas após e antes respectivamente ao utilizador se registar.

The image shows a web interface for page configuration. It is titled 'Configurações' and is divided into two sections: 'Pessoal' and 'Público'. Both sections have a 'Guardar' (Save) button. The 'Público' section includes additional settings for content blocks and a news display period.

Section	Idioma	Tema	Css	Bloco 1	Bloco 2	Bloco 3	Bloco 4	Bloco 5	Bloco 6	News Period
Pessoal	pt	smooth	lite							
Público	pt	smooth	lite	calendar	news	user	Nenhum	Nenhum	Nenhum	30 dias

Figura 8 - Configurações da página

3.2 Módulo

3.2.1 Geral

A aplicação das várias técnicas de programação tem de ser de acordo com o tipo de informação que se pretende transmitir.

A base deste módulo foi construída segundo tipo de programação estruturada. Os menus que se apresentam seguem uma lógica hierárquica, como se vê claramente no menu principal que está representado na figura seguinte.

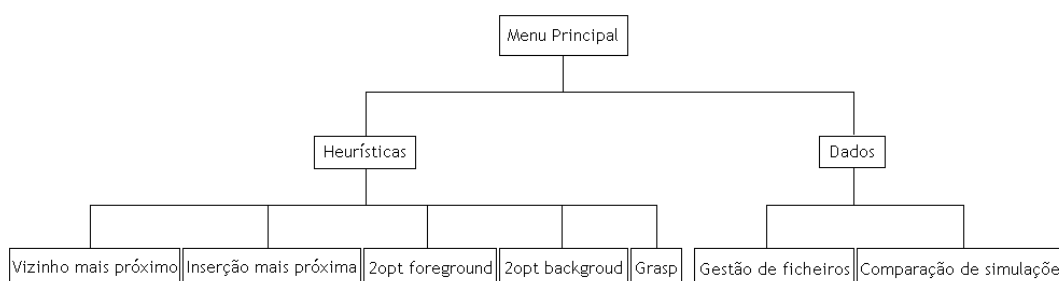


Figura 9 - Estrutura do módulo

Cada um dos blocos representados nas extremidades desta árvore segue a técnica sequencial de forma a conduzir o utilizador no desenrolar dos processos desses blocos, tal como está exemplificado na figura 10 para o “vizinho mais próximo”. Pretende-se com isto uma melhor compreensão por parte do utilizador da informação que se pretende transmitir.

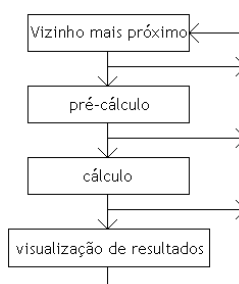


Figura 10 - Estrutura de bloco

Do ponto de vista de implementação, a página a visualizar em cada momento é constituída por várias secções. Estas secções estão descritas em baixo de forma sequencial, tal como são chamadas no processo de construção da página a visualizar.

Validação

Esta validação é necessária para que o utilizador não consiga aceder directamente a um ficheiro de código. Isto permite evitar erros de execução do programa e acessos indevidos a informação. Repara-se que o código da tabela seguinte aparece em todos os ficheiros.

Tabela 2 - Validação

```
if (!preg_match("modules.php", $_SERVER["PHP_SELF"])) {
    die (header("location: ../../access_denied.php"));
}
```

Configuração

A configuração é dividida em dois grupos. O primeiro é respeitante à passagem de todas as variáveis comuns a toda a página, tais como: linguagem, formatação, ligação à base de dados, etc. O segundo trata-se de uma classe que tem como objectivo a criação e manuseamento de objectos.

Cabeçalho

O cabeçalho, neste caso concreto, é o que diz respeito à plataforma. Encontra-se na função "head" que está descrita no ficheiro "mainfile.php" dentro da pasta "includes". Esta função pode e deve ser alterada de forma a fazer a integração do módulo na plataforma.

Menu principal

É o menu principal do módulo e pode-se ver o seu código na tabela seguinte.

Tabela 3 - Menu principal

```
function index() {
    echo "<table width='100%'><tr><td class='leftcontent'><a
href='modules.php?name=data'>._ALG.</a></td></tr></table>"
    . "<table width='100%' class='content'><tr>"
    . "<td><ul><li><a
href='modules.php?name=data&op=alg1'>._ALG1.</a></li></ul></td>"
    . "<td><ul><li><a
href='modules.php?name=data&op=alg3'>._ALG3.</a></li></ul></td>"
    . "<td><ul><li><a
href='modules.php?name=data&op=alg4'>._ALG4.</a></li></ul></td>"
    . "</tr><tr>"
    . "<td><ul><li><a
href='modules.php?name=data&op=alg2'>._ALG2.</a></li></ul></td>"
    . "<td><ul><li><a
href='modules.php?name=data&op=alg3i'>._ALG3I.</a></li></ul></td>"
    . "<td></td>"
    . "</tr></table>"
    . "<table width='100%'><tr><td class='leftcontent'><a
href='modules.php?name=data'>._DATA.</a></td></tr></table>"
    . "<table width='100%' class='content'><tr>"
    . "<td><ul><li><a
href='modules.php?name=data&op=file_manage'>._FILE_MANAGE.</a></li></ul></td>"
    . "<td><ul><li><a
href='modules.php?name=data&op=file_compare'>._FILE_COMPARE.</a></li></ul></td>"
    . "</tr></table>";
}
```

Menu secundário

É o menu da secção onde o utilizador se encontra. No caso de se encontrar no menu principal este não existe. A tabela 4 mostra um exemplo de código deste menu usando o menu do “vizinho mais próximo”.

Tabela 4 - Menu secundário

```
function alg1(){
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._ALG1."</td></tr></table>"
        . "<table width=\"100%\" class=\"content\"><tr>"
        . "<td><ul><li><a"
        href=\"modules.php?name=data&amp;op=alg1&amp;opx=data_type&amp;ncidades=10\">._DATA."<
        /a></li></ul></td>"
        . "<td><ul><li><a"
        href=\"modules.php?name=data&amp;op=alg1&amp;opx=data_result\">._RESULT."</a></li></ul></
        td>"
        . "<td><ul><li><a"
        href=\"modules.php?name=data&amp;op=alg1&amp;opx=export_data\">._EXPORT_DATA."</a></li>
        </ul></td>"
        . "</tr></table>";
}

```

Conteúdo

Espaço da página em questão destinada à informação que se pretende transmitir.

Rodapé

Tal como o cabeçalho, o rodapé também diz respeito à plataforma e convém ser mudado para a integração do módulo na plataforma. É a função “foot” e encontra-se no mesmo ficheiro que a função “head”.

Na ligação que existe entre o módulo e a base de dados foi utilizado o tipo de programação orientada a objectos. A sua implementação consiste na criação de uma classe onde é descrito o objecto e os seus procedimentos de manuseamento. O objecto consiste na criação de uma estrutura de dados que virá a ser preenchida com os dados fornecidos ao mesmo. Durante a execução de qualquer alteração dos dados temos a grande vantagem de nunca se perder a estrutura onde estes estão inseridos. Utilizado o caso do “vizinho mais próximo” podemos ver uma parte do código respeitante à sua classe na tabela seguinte.

Tabela 5 - Classe

```
<?
class a_cidades{
    var $nome;
    var $x;
    var $y;

    function a_cidades($nome=null, $x=null, $y=null){
        $this->nome = $nome;
        $this->x = $x;
        $this->y = $y;
    }
}

```

```
}  
  
function dropTables(){  
    global $dbt;  
  
    $result=$dbt->query("DROP TABLE 1_cidades, 1_distancias, 1_cidades_order");  
    if (DB::isError($result)) die ($result->getDebugInfo());  
  
    return 1;  
}  
  
function createTable1(){  
    global $dbt;  
  
    $result=$dbt->query("CREATE TABLE `1_cidades` (  
        `nome` varchar(11) NOT NULL default '0',  
        `x` int(11) NOT NULL default '0',  
        `y` int(11) NOT NULL default '0',  
        PRIMARY KEY (`nome`)  
    ) ENGINE=MyISAM DEFAULT CHARSET=latin1;");  
    if (DB::isError($result)) die ($result->getDebugInfo());  
  
    return 1;  
}  
  
// foi omitido o resto do ficheiro  
  
}  
?>
```

3.2.2 Partes comuns

O módulo também se apresenta com uma estrutura modular, em que cada algoritmo é apresentado como um módulo independente. Independente de todos os algoritmos implementados serem apresentados como módulos independentes, a gestão de ficheiros de dados e a visualização gráfica dos mesmos são algumas das particularidades que todos eles têm em comum.

A gestão de ficheiros consiste num repositório onde são guardadas as simulações, em que cada ficheiro corresponde a uma simulação. Este repositório apenas permite importar, exportar e apagar ficheiros como é mostrado na figura 11, pois a utilização destes será feita dentro de cada algoritmo.

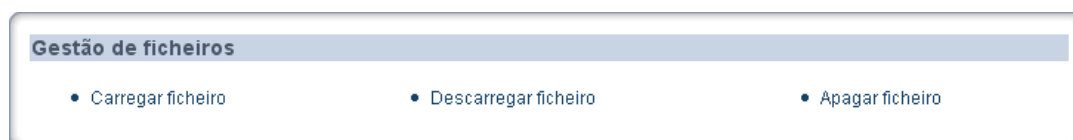


Figura 11 - Gestão de ficheiros

Os ficheiros guardam a informação em formato XML e cada ficheiro contém os dados iniciais, o algoritmo usado, a solução mediante esse algoritmo e mais algumas informações relevantes à simulação. Para a criação do ficheiro XML foi usada uma DTD própria, como consta na tabela 6, a fim de otimizar a forma como são guardados os dados. Na tabela 7 pode-se observar um exemplo de um XML válido para essa DTD.

Tabela 6 - DTD.

```

<!DOCTYPE SIMULATION [

  <!ELEMENT SIMULATION (CIDADES, CIDADES_O, DADOS)>
  <!ELEMENT CIDADES (CIDADE+)>
  <!ELEMENT CIDADES_O (CIDADE+)>
  <!ELEMENT DADOS (DATA)>
  <!ELEMENT CIDADE (NOME, X, Y)>
  <!ELEMENT DATA (TYPE, TIME1, TIME2, SEED, PERCENT)>
  <!ELEMENT NOME (#PCDATA)>
  <!ELEMENT X (#PCDATA)>
  <!ELEMENT Y (#PCDATA)>
  <!ELEMENT TYPE (#PCDATA)>
  <!ELEMENT TIME1 (#PCDATA)>
  <!ELEMENT TIME2 (#PCDATA)>
  <!ELEMENT SEED (#PCDATA)>
  <!ELEMENT PERCENT (#PCDATA)>

]>

```

Tabela 7 - Exemplo.xml.

```

<?xml version="1.0"?>
<SIMULATION>
  <CIDADES>
    <CIDADE>
      <NOME>p0</NOME>
      <X>26</X>
      <Y>29</Y>
    </CIDADE>
    <CIDADE>
      <NOME>p1</NOME>
      <X>78</X>
      <Y>37</Y>
    </CIDADE>
    <CIDADE>
      <NOME>p2</NOME>
      <X>65</X>
      <Y>43</Y>
    </CIDADE>
  </CIDADES>
  <CIDADES_O>
    <CIDADE>
      <NOME>p0</NOME>
      <X>26</X>
      <Y>29</Y>
    </CIDADE>
    <CIDADE>
      <NOME>p2</NOME>
      <X>65</X>
      <Y>43</Y>
    </CIDADE>
    <CIDADE>
      <NOME>p1</NOME>

```

```

<X>78</X>
<Y>37</Y>
</CIDADE>
</CIDADES_O>
<DADOS>
<DATA>
<TYPE>_ALG1</TYPE>
<TIME1>0.02</TIME1>
<TIME2>0.02</TIME2>
</DATA>
</DADOS>
</SIMULATION>

```

Nas várias simulações, os dados fornecidos à simulação podem ser inseridos manualmente ou através de um ficheiro XML existente no repositório.

Em todos eles, após ser fornecido os dados iniciais, é feito um pré-cálculo que consiste na construção de uma matriz quadrada simétrica em relação à diagonal que por sua vez é preenchida com zeros e diz respeito às distâncias entre cada duas cidades. Os passos deste algoritmo estão descritos na tabela 8.

Tabela 8 - Pré-cálculo.

```

"n" é igual ao número de cidades
é inicializado uma variável "i" a zero
desde "i" até "n" {
    é inicializado uma variável "j" a zero
    desde "j" até "n" {
        coordenada "x" igual a ("x" de "i" menos "x" de "j") ao quadrado
        coordenada "y" igual a ("y" de "i" menos "y" de "j") ao quadrado
        guardar na matriz a distância da cidade "i" à cidade "j" o valor da raiz de
        (coordenada "x" mais coordenada "y")
        guardar na matriz a distância da cidade "j" à cidade "i" o valor da raiz de
        (coordenada "x" mais coordenada "y")
    }
}

```

A simulação do algoritmo é feita após este pré-cálculo e pode ser feita de uma vez só ou passo a passo. Na simulação passo a passo existe ainda a opção do intervalo do passo, correspondendo este ao número de ciclos a saltar no algoritmo em questão. A cada passo é possível ver o estado corrente da solução.

A visualização gráfica dos circuitos é feita através de ficheiros SVG. É apresentado dois gráficos por ficheiro, os dados iniciais e o resultado da simulação, lado a lado e mais algumas informações referentes à simulação, tais como o tipo de simulação e o tempo que demorou. É possível visualizar mais do que uma simulação de cada vez, a fim de fazer a comparação entre as mesmas.

3.2.3 Algoritmos

Nesta secção do documento é descrito o desenvolvimento e alguma implementação dos vários algoritmos constituintes do trabalho como blocos independentes. Na realidade, todos seguem uma sequência idêntica diferindo apenas na parte do cálculo. Falta salientar que a execução do cálculo de cada algoritmo permite ser feita toda de uma vez ou passo-a-passo de forma a privilegiar a interacção com o utilizador.

O cálculo passo-a-passo consiste na utilização de uma variável para fazer paragem do ciclo principal de cada algoritmo, a fim de mostrar o seu estado actual. Na tabela seguinte, podemos encontrar a descrição do código que é utilizado para este fim.

Tabela 9 - Passo-a passo.

```

é inicializado o "n" e "i" com o valor do tamanho do salto passado pelo utilizador
ciclo principal do algoritmo {
    se "i" igual a zero {
        põe "i" igual a "n"
        mostra estado actual e espera pelo utilizador
    }
    se "i" diferente de zero {
        decrementa "i"
        cálculo de um ciclo do algoritmo ...
    }
}

```

3.2.3.1 Vizinho mais próximo

O cálculo do "vizinho mais próximo" começa com a escolha de uma cidade que neste caso é a primeira fornecida da lista de cidades. A escolha da cidade seguinte é feita com base nas distâncias da actual às que ainda faltam visitar. A cidade que tiver a distância mais curta é retirada da lista de cidades a visitar e acrescentada à lista de cidades visitadas. Este procedimento é repetido até a lista de cidades a visitar estar vazia. Na tabela seguinte encontra-se a parte de código implementado relevante aos passos a cima descritos.

Tabela 10 - Vizinho mais próximo.

```

$cidades = a_cidades::getCidades();
$i=count($cidades)-1;
$cidades_o[] = $cidades[0];
$n=0;
while ($i!=0){
    $i--;
    $dist = a_cidades::getDist($cidades_o[$n]->nome);
    unset($dist[nome]);
    foreach ($cidades_o as $cidade_o)
        unset($dist[$cidade_o->nome]);
    foreach ($dist as $key => $val)
        if ($val == min($dist)) $dist_min_key=$key;
    $cidades_o[] = a_cidades::getCidade($dist_min_key);
}

```

```

    $n++;
}

```

3.2.3.2 Inserção mais próxima

Para a codificação da “inserção mais próxima” também foram utilizadas duas listas. A de cidades visitadas que é inicializada com três cidades escolhidas aleatoriamente e a lista de cidades a visitar que contém as restantes. O objectivo é ir inserindo uma cidade escolhida aleatoriamente das cidades a visitar no circuito pré-existente. Avaliando a distância total de todas as hipóteses possíveis entre duas que já tenham sido visitadas escolhe-se a que apresente a distância total menor.

Tabela 11 - Inserção mais próxima.

```

$x=array_rand($cidades);
$cidades_o[] = $cidades[$x];
unset($cidades[$x]);
$x=array_rand($cidades);
$cidades_o[] = $cidades[$x];
unset($cidades[$x]);
$x=array_rand($cidades);
$cidades_o[] = $cidades[$x];
unset($cidades[$x]);

$i=count($cidades_o)-1;

while ($i<$n){
    $x=array_rand($cidades);
    $cidade=$cidades[$x];
    unset($cidades[$x]);
    unset ($d);
    $j=-1;
    while($j++<$i-1)
        $d[]=$dists[$cidades_o[$j]->nome][$cidade->nome]+$dists[$cidades_o[$j+1]-
>nome][$cidade->nome]-$dists[$cidades_o[$j]->nome][$cidades_o[$j+1]->nome];
        $d[]=$dists[$cidades_o[$j]->nome][$cidade->nome]+$dists[$cidades_o[0]-
>nome][$cidade->nome]-$dists[$cidades_o[$j]->nome][$cidades_o[0]->nome];
        $value=min($d);
        $key = array_search($value, $d);
        array_splice($cidades_o, $key+1, 0, 0);
        $cidades_o[$key+1]=$cidade;
        $i=count($cidades_o)-1;
}

```

Na realidade, o código a cima descrito não se baseia no cálculo da distância total do circuito para tomar uma decisão de onde vai inserir a próxima cidade, baseia-se no cálculo de uma distância relativa tal como é mostrado na figura seguinte de forma a tornar a sua execução mais rápida.

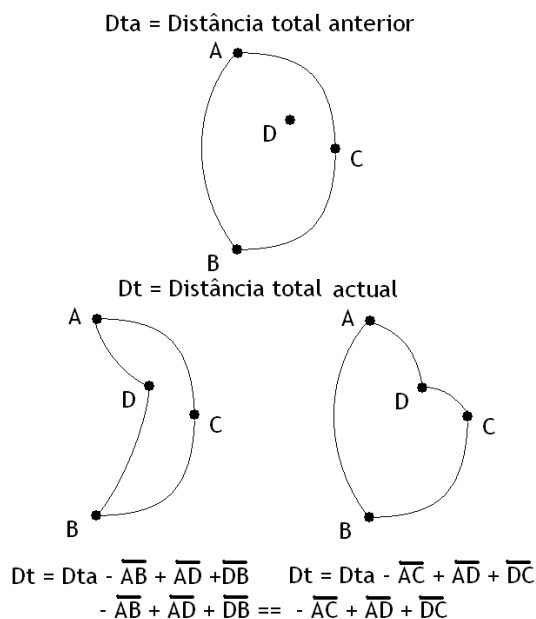


Figura 12 - Comparação de distâncias

3.2.3.3 2opt

Foram implementados dois algoritmos "2opt". Ambos têm o mesmo tipo de codificação diferindo essencialmente no tipo de restrições impostas. A tabela seguinte mostra dois ciclos, um dentro do outro, que representam todas as trocas possíveis sem repetição de todos os pares de percursos entre duas cidades não consecutivas.

Tabela 12 - 2opt

```

$ciudades = a_ciudades::getCiudades();
$i=count($ciudades)-1;
$ciudades_o[] = $ciudades[0];
$n=0;
while ($i!=0){
    $i--;
    $dist = a_ciudades::getDist($ciudades_o[$n]->nome);
    unset($dist[nome]);
    foreach ($ciudades_o as $ciudad_o)
        unset($dist[$ciudad_o->nome]);
    foreach ($dist as $key => $val)
        if ($val == min($dist)) $dist_min_key=$key;
    $ciudades_o[] = a_ciudades::getCiudad($dist_min_key);
    $n++;
}

```

"2opt foreground"

Aqui é feita a restrição do número de trocas efectuadas podendo ser no máximo de 500. Esta restrição tem como propósito reduzir o tempo da simulação para que este não ultrapasse os 30 segundos. A restrição dos 30 segundos pode ser alterada no servidor em questão mas é o

valor típico imposto pelo PHP para a execução de qualquer conjunto de operações até à apresentação da página seguinte, podendo ser alterado no servidor em questão.

“2opt background”

Nesta implementação é usada uma restrição de percentagem de avaliação. Em todas as trocas possíveis é determinada a possibilidade de fazer a avaliação da troca mediante essa percentagem. Não existe a restrição do tempo de execução. Esta é contornada, interrompendo o algoritmo e fazendo gerar uma página ao fim de cada “n” trocas efectuadas e retomando no ponto em que ficou. No entanto, esta solução tem um custo adicional de tempo que corresponde à geração de páginas desnecessárias mais a transmissão das mesmas.

3.2.3.4 Grasp

A codificação desta simulação traduz-se na repetição da codificação do “vizinho mais próximo” com uma restrição adicional seguida da codificação do 2opt sem qualquer restrição. A restrição imposta na primeira parte da simulação consiste na escolha aleatória de uma das “n” melhores hipóteses.

Capítulo 4 - Tutorial de integração do módulo

4.1 Geral

Este tutorial explica, de uma forma sucinta, os passos a dar para integrar o módulo, em questão neste trabalho, em qualquer plataforma. Toda a descrição de integração que se segue, usa como exemplo a plataforma onde o módulo foi desenvolvido.

Todas as inicializações de páginas de internet, que tenham uma base de dados de suporte, requerem a inicialização prévia dessa base de dados. Essa inicialização é feita de forma automática ou manual. Diz-se que esta inicialização é manual quando é fornecido um ficheiro SQL para fazer esse procedimento. Para facilitar essa inicialização da base de dados, esta é feita com a utilização normal do módulo sem que o utilizador se aperceba.

Para integrar concretamente o módulo na plataforma que se pretende, basta copiar a pasta do mesmo para o sítio pré designado pela plataforma em questão. Para finalizar, é necessária a alteração de pequenas variáveis para que o módulo funcione em concordância com a plataforma. Essas variáveis e sua modificação podem ser encontradas nos seguintes ficheiros descritos em baixo:

config.php

Este ficheiro encontra-se na raiz do módulo e contém apenas as variáveis que dizem respeito à ligação à base de dados. O conteúdo deste ficheiro é apresentado na tabela seguinte.

Tabela 13 - config.php.

```
<?
//dir
$_DIR_ = "lib/";
```

```
//db connect
$_type_ = "mysql"; // (mysql/pgsql)
$_user_ = "root";
$_pass_ = "";
$_host_ = "localhost";
$_database_ = "tese";
?>
```

conf.php

Na raiz do módulo existe uma pasta chama “includes” que inclui os ficheiros comuns a todo o módulo. O “conf.php” é um deles e tem como objectivo a atribuição e gestão de todas as variáveis de sessão. As variáveis que constam neste ficheiro determinam uma série de aspectos de visualização do módulo, entre outras coisas. A linguagem, a formatação de texto e o tema em que o módulo é apresentado são alguns dos exemplos que se pode ver na tabela a seguir que mostra o conteúdo do ficheiro.

Tabela 14 - conf.php.

```
<?
    session_start();
    include_once ("./config.php");
    ini_set("include_path", $_DIR_);

    include_once ("DB.php");
    include_once ("./includes/db.php");
    include_once ("./classes/Settings.php");

//site var
$prefix = "tese_";
$theme_file = "theme";
$style = "style.css";
$index = 0;

//ini var
if ($_SESSION["user_id_"])
    $usr_id = $_SESSION["user_id_"];
else
    if (!$usr_id) $usr_id = 0;

if (!$_SESSION["user_level_"])
    $_SESSION["user_level_"]=1;

if ($_SESSION["user_nickname_"])
    $usr_nickname=$_SESSION["user_nickname_"];
if (!$usr_nick=$usr_nickname)
    $usr_nick=_ANONYMUS;

$usr_level=$_SESSION["user_level_"];

if (!$module=$_REQUEST["name"]) $module = "home";

if ($_SESSION["lang"])
    $lang=$_SESSION["lang"];

if ($_SESSION["theme"])
    $theme=$_SESSION["theme"];

if ($_SESSION["css"])
    $css=$_SESSION["css"];

```

```

$settings=Settings::getSettings();

$i=0;
while ($settings[$i]->setting != "site")
    $i++;

if (!$lang) $lang = $settings[$i]->_1;
if (!$theme) $theme = $settings[$i]->_2;
if (!$css) $css = $settings[$i]->_3;

?>

```

mainfile.php

Este ficheiro está dentro da pasta “includes” que se encontra na raiz do módulo. Este consiste essencialmente em duas funções, a função “head” e a função “foot”. Os nomes destas funções são intuitivos do ponto de vista de programação e servem para configurar a visualização do cabeçalho e rodapé do módulo.

4.2 Integração no Moodle

Existem várias regras para a integração de um módulo no Moodle. Algumas dessas regras são obrigatórias, outras apenas meras sugestões de conveniência.

Como regras mais importantes temos as seguintes:

Todos os ficheiros do módulo devem estar dentro da pasta do módulo que por sua vez tem que estar dentro da pasta “mod” do Moodle.

As variáveis “modulename” e “modulenames” devem ter o nome do módulo no singular e plural respectivamente como é mostrado na tabela seguinte que consiste numa parte do ficheiro “index.php” do módulo.

Tabela 15 - Parte de “index.php” do módulo no Moodle.

```

<?
require_once("../../config.php");
...
$stobject = get_string("modulename", "object");
$stobjects = get_string("modulenameplural", "objects");
...

```

Todos os ficheiros devem incluir o ficheiro "config.php" do Moodle, tal como se mostra na tabela anterior. Este trata de uma série de funções de segurança.

Os seguintes ficheiros também são de carácter obrigatório:

index.php

Ficheiro que é chamado pela plataforma para interagir com o módulo e deve conter todas as funções que o módulo permite executar.

view.php

Este ficheiro destina-se a ver uma designada instância em particular.

lib.php

É obrigatório a inicialização das instâncias respeitantes ao módulo como o exemplo da tabela seguinte.

Tabela 16 - lib.php

```

<?

function object_add_instance($object) {
    /// Given an object containing all the necessary data,
    /// (defined by the form in mod.html) this function
    /// will create a new instance and return the id number
    /// of the new instance.
    //print_r($object);
    if (empty($object->material_root) || empty($object->imsmanifest)
    || empty($object->start_url)) { error("No Course Selected"); return
false; }
    if ($object->material_root == " || $object->imsmanifest == " ||
$object->start_url == ") { error("No Course Selected"); return false; }

    return insert_record("object", $object);
}
...
function object_delete_instance($id) {
    /// Given an ID of an instance of this module,
    /// this function will permanently delete the instance
    /// and any data that depends on it.

    if (! $object = get_record("object", "id", "$id")) {
        return false;
    }

    $result = true;

    if (! delete_records("object", "id", "$object->id")) {
        $result = false;
    }

    return $result;
}
...

```

mod.html

Trata-se de um formulário que se destina a criar e modificar um objecto como instância do módulo.

Para além destas regras e ficheiros obrigatórios, temos também alguns ficheiros adicionais que são bastante úteis ao módulo, como por exemplo:

version.php

Serve para identificar a versão do módulo.

icon.gif

Imagem utilizada pela plataforma para identificar o módulo.

Capítulo 5 - Funcionamento do módulo

Este módulo apresenta uma estrutura hierárquica como é mostrado na figura 13 de forma a facilitar a sua compreensão. No menu inicial podemos encontrar duas grandes secções: "Heurísticas" e "Dados".

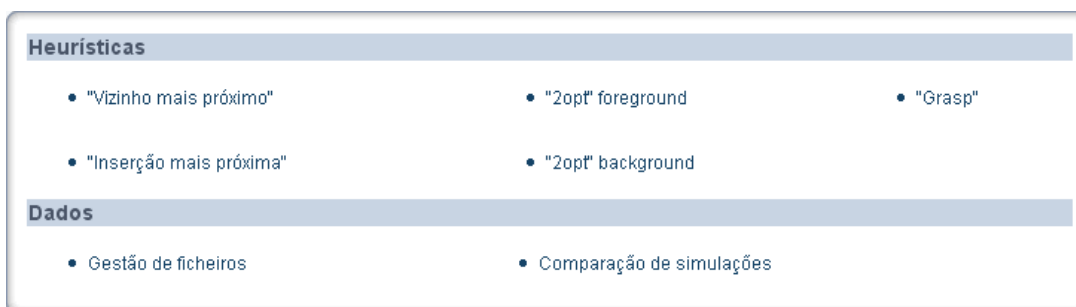


Figura 13 - Menu principal

A primeira consiste na lista dos vários algoritmos disponíveis para simulação sendo estes: "Vizinho mais próximo", "Inserção mais próxima", "2opt foreground", "2opt background" e "Grasp". A segunda trata de um repositório de ficheiros comum às várias simulações, que por sua vez, se encontra dividido em duas partes: "Gestão de ficheiros" e "Comparação de simulações".

Relativamente à segunda opção, "Dados", é dada a escolha ao utilizador para gerir ficheiros ou comparar simulações através de "Gestão de ficheiros" e "Comparação de simulações", respectivamente. Na primeira opção, são apresentadas três escolhas:

"Carregar ficheiro"

Trata-se da operação de guardar um ficheiro que se encontre na máquina local no repositório do módulo a fim de ser usado mais tarde.

"Descarregar ficheiro"

É a operação inversa à anterior em que consiste na cópia de um ficheiro do repositório a ser gravado na máquina local.

“Apagar ficheiro”

Serve para eliminar os ficheiros que não serão mais usados no futuro.

A opção “Comparação de simulações”, serve para o utilizador verificar as diferenças entre dois ou mais ficheiros que se encontrem no repositório. Esta comparação pode ser gráfica em que são mostrados os dados de entrada e resultados por meio de imagens ou de valores que são fornecidos como informações adicionais às imagens.

As “Heurísticas” consistem na simulação dos algoritmos acima mencionados.

Em todas as simulações as opções dadas ao utilizador são praticamente as mesmas, bem como os passos pelos quais estas são constituídas. As simulações dividem-se essencialmente em quatro passos: inserção de dados, pré-cálculo, cálculo e resultados.

A inserção de dados como se vê na figura 14 pode ser feita manualmente inserindo o número de cidades que se pretende para a simulação ou através de um ficheiro existente no repositório do módulo.

A imagem mostra uma interface de usuário com o título "Dados". Ela é dividida em duas seções principais:

- Inserir dados:** Possui um campo de entrada rotulado "Nº de cidades:" com o valor "10" inserido. Abaixo dele há um botão "Seguinte".
- Importar dados:** Possui um menu suspenso rotulado "Ficheiro:" e um botão "Seguinte" abaixo dele.

Figura 14 - Inserção de dados

Se for escolhida a inserção manual, em que o utilizador já escolheu o número de cidades que vai utilizar, pode inserir as coordenadas dessas cidades de forma manual ou automaticamente gerando as coordenadas aleatoriamente entre zero e o número pedido ao utilizador, preenchendo assim a tabela de cidades que é mostrada como na figura 15. Na geração aleatória, pode-se ainda alterar qualquer coordenada antes de seguir para o próximo passo tendo em conta que os números gerados aleatoriamente têm todos a mesma probabilidade.

Inserir

Gerir cidades aleatórias:

Cidade: <input type="text" value="p0"/>	Coordenadas: (<input type="text" value="76"/> ; <input type="text" value="9"/>)
Cidade: <input type="text" value="p1"/>	Coordenadas: (<input type="text" value="63"/> ; <input type="text" value="16"/>)
Cidade: <input type="text" value="p2"/>	Coordenadas: (<input type="text" value="98"/> ; <input type="text" value="43"/>)
Cidade: <input type="text" value="p3"/>	Coordenadas: (<input type="text" value="61"/> ; <input type="text" value="73"/>)
Cidade: <input type="text" value="p4"/>	Coordenadas: (<input type="text" value="27"/> ; <input type="text" value="79"/>)
Cidade: <input type="text" value="p5"/>	Coordenadas: (<input type="text" value="62"/> ; <input type="text" value="92"/>)
Cidade: <input type="text" value="p6"/>	Coordenadas: (<input type="text" value="54"/> ; <input type="text" value="21"/>)
Cidade: <input type="text" value="p7"/>	Coordenadas: (<input type="text" value="0"/> ; <input type="text" value="47"/>)
Cidade: <input type="text" value="p8"/>	Coordenadas: (<input type="text" value="54"/> ; <input type="text" value="55"/>)
Cidade: <input type="text" value="p9"/>	Coordenadas: (<input type="text" value="20"/> ; <input type="text" value="88"/>)

Figura 15 - Geração de cidades

O pré-cálculo também é comum a todos os algoritmos e é um passo necessário para fazer o cálculo das distâncias entre todas as cidades.

Ainda comum aos vários algoritmos, temos a visualização de resultados e exportação dos mesmos.

A visualização de resultados, tal como é mostrado na figura 16, consiste num pequeno menu com opções de visualização seguido de uma representação gráfica dos dados de entrada e saída.

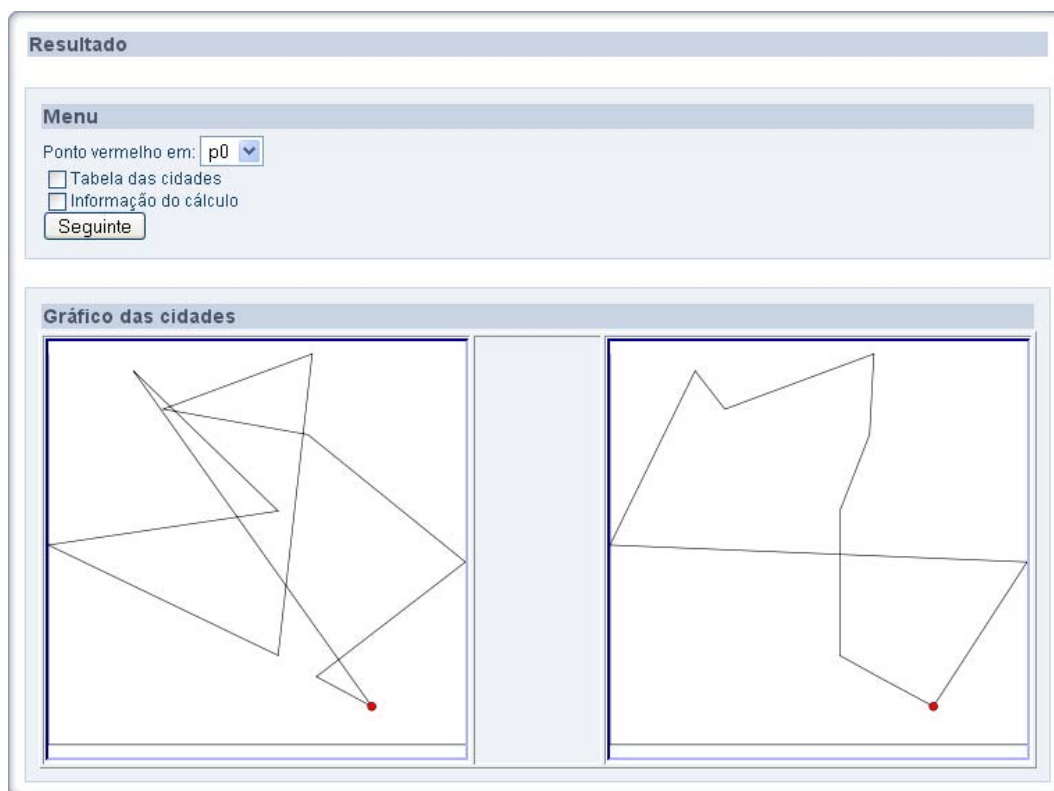


Figura 16 - Visualização de resultados

Nesse pequeno menu pode-se escolher qual a cidade a ser representada por um ponto vermelho com o objectivo de uma melhor interpretação da imagem apresentada, bem como a visualização ou não das tabelas de dados correspondentes aos gráficos e algumas informações adicionais à simulação.

A opção de exportar dados pode ser acedida em qualquer altura, seja na inserção de dados ou na visualização de resultados, podendo guardar os dados já existentes num ficheiro que ficará no repositório. O nome do ficheiro é à escolha do utilizador não podendo ser igual a um que já exista.

Todas as simulações diferem na parte do cálculo, onde cada algoritmo requer diferentes entradas. No entanto, mesmo nesta parte, todos têm em comum a possibilidade de fazer a execução do cálculo do algoritmo passo-a-passo. Neste modo de execução é pedido ao utilizador o tamanho do salto a dar em cada passo, tal como é mostrado na figura 17. A cada passo é sempre apresentada a evolução mediante a representação gráfica. Esta opção privilegia a interacção com o utilizador para uma melhor compreensão por parte deste de cada algoritmo.



The image shows a software window titled "Cálculo". Inside the window, there are two input fields: "Semente:" followed by an empty text box, and "Quantidade de saltos:" followed by a text box containing the number "1". Below these fields are two buttons: "Seguinte" and "Passo a Passo".

Figura 17 - Cálculo

“Vizinho mais próximo”

Não requer qualquer entrada para o cálculo.

“Inserção mais próxima”

É apenas pedida a semente, que se não fornecida será gerada a partir do relógio da máquina, para que se possa realizar no futuro a mesma simulação obtendo o mesmo resultado.

“2opt foreground”

São pedidos a semente e o número máximo de melhoramentos efectuados. O número máximo permitido é 500 devido ao factor tempo.

“2opt background”

São pedidos a semente e a percentagem de avaliação de trocas. A avaliação de trocas consiste em cada ciclo do algoritmo avaliar ou não se a troca é vantajosa.

Grasp

É pedido a semente e um número “n” de melhores hipóteses. Este valor corresponde à escolha aleatória de uma entre as “n” melhores hipóteses da heurística construtiva para usar na pesquisa local.

Capítulo 6 - Conclusão

6.1 Comentários finais

Este trabalho tinha como objectivo a criação de um módulo pedagógico com base no TSP. Foi cumprido, tanto o objectivo teórico, conceitos e regras utilizados na criação de um módulo pedagógico e os diversos papéis do ensino, como o prático, criação do módulo e respectiva implementação.

Desta forma, comprovasse que é possível a criação de um módulo capaz de integrar qualquer plataforma e, simultaneamente, capaz de corresponder às expectativas básicas do utilizador. Neste caso específico o Módulo foi integrado com sucesso no Moodle, embora não tenha sido utilizada a sua versão mais recente.

Com o intuito de chegar à resolução óptima do problema do *Caixeiro Viajante*, foram implementados no Módulo alguns algoritmos de diferentes tipos.

Espera-se que com este trabalho se tenha aberto as portas para a continuação da exposição deste tipo de problemas combinatorios sob a forma de módulo pedagógico, facilitando assim a sua aprendizagem e respectiva utilização por parte dos utilizadores.

6.2 Recomendações para trabalho futuro

Ao longo dos tempos, as técnicas de ensino têm vindo a aperfeiçoarem-se, contudo, estas ainda podem ser melhoradas facilitando assim a aprendizagem e, conseqüentemente, aumentando a qualidade do ensino.

Deixam-se aqui algumas sugestões para o melhoramento do ensino, como a criação de novos módulos pedagógicos, assim como o melhoramento dos métodos de ensino mais antiquados. O melhoramento do ambiente gráfico dos vários métodos de ensino, digitais e não digitais e por fim o melhor acompanhamento dos alunos a fim de evitar o abandono escolar.

Ficam também aqui algumas sugestões futuras no que diz respeito ao melhoramento deste Módulo Pedagógico.

A nível de conteúdo seria interessante a utilização de novos e mais complexos algoritmos capazes de resolver este problema de optimização combinatória com a menor percentagem de erro possível.

Pode-se ainda alargar este módulo à resolução de outros problemas de optimização combinatória para além do TSP como o SAT (Boolean satisfiability problem) e o problema da mochila (knapsack problem) tornando assim este módulo numa plataforma geral de resolução de problemas combinatórios NP-árduos.

Do ponto de vista de programação, uma grande limitação do PHP é a utilização de trinta segundos para o carregamento de toda a página a ser visualizada. Pode vir a ser viável a implementação de algoritmos mais pesados se a página for apenas carregada parcialmente, ou seja, a utilização de outro método de transferência de informação entre o servidor (máquina que corre a simulação) e o cliente (máquina que recebe os resultados) para que seja apenas passado o fruto da simulação e não toda a página.

Bibliografia

- [1] Computador e Ensino, Henrique J. C. de Oliveira. Disponível em <http://www.prof2000.pt/users/secjeste/ensinopc/index.htm>. Acesso em 09/Junho/2008.
- [2] A Internet no Ensino das Ciências Físico-químicas, António Silva. Disponível em http://nautilus.fis.uc.pt/softc/Read_c/destaque/teseind.htm Acesso em 09/Junho/2008.
- [3] A Evolução dos Computadores, Julia Gadelha. Disponível em http://w3.ualg.pt/~fcar/evolucao_computadores.pdf. Acesso em 11/Outubro/2008.
- [4] InforQuali. Disponível em <http://www.inforquali.pt/iq/pt/tutoriais/informativos/historia-dos-computadores.php>. Acesso em 11/Outubro/2008.
- [5] Ghisi na Área, Simonei Ghisi. Disponível em <http://sghisi.blogspot.com/2008/03/evolucao-do-chip-8080.html>. Acesso em 11/Outubro/2008.
- [6] Guia do Hardware, Carlos E. Morimoto. Disponível em <http://www.guiadohardware.net/artigos/origem-pc/>. Acesso em 11/Outubro/2008.
- [7] Wikipedia. Disponível em http://pt.wikipedia.org/wiki/Instru%C3%A7%C3%A3o_Programada. Acesso em 15/Outubro/2008.
- [8] Wikipedia. Disponível em http://pt.wikipedia.org/wiki/M%C3%A1quina_de_ensinar. Acesso em 15/Outubro/2008.
- [9] Wikipedia. Disponível em <http://pt.wikipedia.org/wiki/Tutorial>. Acesso em 15/Outubro/2008.
- [10] Iwebtool. Disponível em http://www.iwebtool.com/what_is_tutorial.html. Acesso em 15/Outubro/2008.
- [11] Blogspot, Flarranyelly Kaymmi. Disponível em http://flarranyelly.blogspot.com/2009_09_01_archive.html. Acesso em 15/Outubro/2008.
- [12] Baixaki, Allan Valin Ribeiro da Fonseca. Disponível em <http://www.baixaki.com.br/info/3014-inteligencia-artificial-reconhecimento-de-padroes.htm>. Acesso em 15/Outubro/2008.
- [13] COPPE - UFRJ, Rosa Maria Esteves Moreira da Costa e Vera Maria Benjamin Werneck . Disponível em <http://www.cos.ufrj.br/uploadfiles/es42797.pdf>. Acesso em 15/Outubro/2008.
- [14] Wikipedia. Disponível em <http://pt.wikipedia.org/wiki/E-learning>. Acesso em 19/Outubro/2008.
- [15] Grantastic Designs. Disponível em <http://www.grantasticdesigns.com/5rules.html#consistent>. Acesso em 23/Outubro/2008.

- [16] Josiah Cole dot com. Disponível em <http://www.josiahcole.com/2007/02/14/a-webmasters-19-commandments/>. Acesso em 23/Outubro/2008.
- [17] Introduction to Genetic Algorithms. Disponível em <http://www.obitko.com/tutorials/genetic-algorithms/tsp-example.php>. Acesso em 28/Outubro /2008.
- [18] The GA Playground: TSP Demo. Disponível em <http://www.aridolan.com/ga/gaa/TspDemo.html>. Acesso em 28/ Outubro /2008.
- [19] Introduction to Operations Research, Frederick S. Hillier & Gerald J. Lieberman.
- [20] How to solve it: modern heuristics, Zbigniew Michalewicz & David B. Fogel, 2000.
- [21] Graph Theory 1736-1936, Norman L. Biggs, E. Keith Lloyd and Robin J. Wilson, 1986.
- [22] "History" in The Traveling Salesman Problem, A. J. Hoffman and P. Wolfe, 1985.
- [23] TSPLIB. Disponível em <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>. Acesso em 12/Novembro/2008.
- [24] "Optimização Combinatória: Modelos e Algoritmos", José Fernando Oliveira, Maria Antónia Carravilla - FEUP, 2001.
- [25] Moodle. Disponível em <http://moodle.org>. Acesso em 25/Novembro/2008.
- [26] Google Maps Fastest Roundtrip Solver. Disponível em <http://www.gebweb.net/optimap>. Acesso em 25/Novembro/2008.
- [27] Problema do Caixeiro Viajante. Disponível em <http://to-campos.planetaclix.pt/neural/hop.html>. Acesso em 25/Novembro/2008.
- [28] TSP solving with using 3 different algorithms: Nearest Neighbour, 2opt and Genetic Algorithm. Disponível em <http://www.dieslunae.net/tsp>. Acesso em 25/Novembro/2008.
- [29] Traveling Salesman Problem Using Genetic Algorithms. Disponível em <http://www.lalena.com/AI/Tsp>. Acesso em 25/Novembro/2008.

Anexos

root

index.php

config.php

modules.php

root/includes

conf.php

mainfile.php

root/language

lang_pt.php

root/themes/smooth

theme.php

root/modules/data

alg1.php

alg2.php

alg3.php

alg3i.php

alg4.php


```
<?
if ( $_SERVER[HTTP_REFERER] == "http://www.sadmp.web.pt" || $_SERVER[HTTP_REFERER] ==
"http://www.sadmp.web.pt/")
echo "<script language=\"JavaScript\" type=\"text/JavaScript\">
  <!--
  function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+\".location=\"+args[i+1]+\"\\");
  }
  //-->
</script>
  <body onLoad=\"MM_goToURL('parent','http://\".$_SERVER[HTTP_HOST].$_SERVER[PHP_SELF].\"');return
document.MM_returnValue\"> ";
else
{
  require_once("includes/mainfile.php");

  $_SERVER['PHP_SELF'] = "modules.php";

  if (file_exists("modules.php"))
    include("modules.php");
  else
    die (header("location: access_denied.php"));
}
?>
```



```
<?
//dir
//    $_DIR_ = "./lib/";
    $_DIR_ = "lib/";

//db connect
//    $_type_ = "pgsql";
    $_type_ = "mysql";
    $_user_ = "Admin";
    $_pass_ = "*****";
//    $_pass_ = "sif";
    $_host_ = "localhost";
    $_database_ = "tese";

//email register
    $reg_active=1; //insert user 1-active 2-inactiv
?>
```



```
<?
require_once("../includes/mainfile.php");

$module = trim($module);
if (isset($module)) {
    if (eregi("http:\\\\", $module)) {
        die("Hi&nbsp;and&nbsp;Bye");
    }else{
        if (file_exists("language/lang_". $lang. ".php"))
            include ("language/lang_". $lang. ".php");
        if (file_exists("themes/$theme/" . $theme_file. ".php"))
            include ("themes/$theme/" . $theme_file. ".php");
//         else
//             die (header("location: access_denied.php"));
        if (file_exists("modules/$module/index.php"))
            include ("modules/$module/index.php");
        else
            die (header("location: access_denied.php"));
    }
}
}else
    die (header("location: access_denied.php"));
?>
```



```
<?
    session_start();
    include_once ("./config.php");
    ini_set("include_path",$_DIR_);

    include_once ("DB.php");
    include_once ("./includes/db.php");
    include_once ("./classes/Settings.php");

//site var
    $prefix = "tese_";
    $theme_file = "theme";
    $style = "style.css";
    $index = 0;

//ini var
    if ($_SESSION["user_id_"])
        $usr_id = $_SESSION["user_id_"];
    else
        if (!$usr_id) $usr_id = 0;

    if (!$_SESSION["user_level_"])
        $_SESSION["user_level_"]=1;

    if ($_SESSION["user_nickname_"])
        $usr_nickname=$_SESSION["user_nickname_"];
    if (!$usr_nick=$usr_nickname)
        $usr_nick=_ANONYMUS;

    $usr_level=$_SESSION["user_level_"];

    if (!$module=$_REQUEST["name"]) $module = "home";

    if ($_SESSION["lang"])
        $lang=$_SESSION["lang"];

    if ($_SESSION["theme"])
        $theme=$_SESSION["theme"];
```

```
if ($_SESSION["css"])
    $css=$_SESSION["css"];

$settings=Settings::getSettings();

$i=0;
while ($settings[$i]->setting != "site")
    $i++;

if (!$lang) $lang = $settings[$i]->_1;
if (!$theme) $theme = $settings[$i]->_2;
if (!$css) $css = $settings[$i]->_3;
```

?>

```

<?
$mtime = microtime();
$mtime = explode(" ", $mtime);
$mtime = $mtime[1] + $mtime[0];
$start_time = $mtime;

if (!ereg("modules.php", $_SERVER['PHP_SELF']) && !ereg("index.php", $_SERVER['PHP_SELF'])) {
    die (header("location: access_denied.php"));
}

require_once("conf.php");

function head() {
    global $module, $css, $theme, $style, $usr_id, $usr_nickname, $usr_level, $settings;

    echo "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN\"
        . "\"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">"
        . "<html xmlns=\"http://www.w3.org/1999/xhtml\" lang=\"en\" xml:lang=\"en\">"
        . "<head>"
        . "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\"/>"
        . "<title>._SITENAME.\" :: \"ucwords($module).\"</title>"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/\".$css.\"/fonts.css\"
title=\"default\" />"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/\".$css.\"/colors.css\"
title=\"default\" />"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/\".$css.\"/layout.css\"
title=\"default\" />"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/\".$css.\"/forms.css\"
title=\"default\" />";
    if ($style)
        echo "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\"
href=\"themes/\".$theme.\"/\".$style.\"\" title=\"default\" />";
    echo "<link rel=\"icon\" type=\"image/jpg\" href=\"images/logo.jpg\" />"
        . "<link rel=\"shortcut icon\" type=\"image/jpg\" href=\"images/logo.jpg\" />"
        . "</head>"
        . "<body>"
        . "<div id=\"header\">";
    OpenHeader();
}

```

```

echo "&nbsp;&nbsp;&nbsp;<a href=\"index.php\">\"._HOME.\"</a>&nbsp;&nbsp;&nbsp;";
if ($usr_id){
    echo "&nbsp;&nbsp;&nbsp;<a href=\"modules.php?name=user\">";
    if ($usr_level==3) echo _USERS;
    else echo _USER;
    echo "</a>&nbsp;&nbsp;&nbsp;";
}
echo "&nbsp;&nbsp;&nbsp;<a href=\"modules.php?name=news\">\"._NEWS.\"</a>&nbsp;&nbsp;&nbsp;";
    . "&nbsp;&nbsp;&nbsp;<a href=\"modules.php?name=calendar\">\"._CALENDAR.\"</a>&nbsp;&nbsp;&nbsp;";
if ($usr_id)
    echo "&nbsp;&nbsp;&nbsp;<a href=\"modules.php?name=data\">\"._ALG.\"</a>&nbsp;&nbsp;&nbsp;";
        . "&nbsp;&nbsp;&nbsp;<a href=\"modules.php?name=data\">\"._DECISION.\"</a>&nbsp;&nbsp;&nbsp;";
        . "&nbsp;&nbsp;&nbsp;<a href=\"modules.php?name=settings\">\"._SETTINGS.\"</a>&nbsp;&nbsp;&nbsp;";
echo "&nbsp;&nbsp;&nbsp;<a href=\"modules.php?name=about\">\"._ABOUT.\"</a>";
CloseHeader();
echo "</div>";
echo "<div id=\"main\"> ";
OpenMain();
echo "<div id=\"column1\">";

$n=0;
while ($settings[$n]->setting != "blocks")
    $n++;

$block[]=$settings[$n]->_1;
$block[]=$settings[$n]->_2;
$block[]=$settings[$n]->_3;
$block[]=$settings[$n]->_4;
$block[]=$settings[$n]->_5;
$block[]=$settings[$n]->_6;
$block[]=$settings[$n]->_7;
$block[]=$settings[$n]->_8;

$n=0;
while ($n<9){
    if (is_file("./modules/".$block[$n]."/block.php")){
        echo "<div id=\"".$block[$n]."\>";
        OpenBlock();
    }
}

```

```

        include ("./modules/".$block[$n]."/block.php");
        CloseBlock();
        echo "</div>";
    }
    $n++;
}

echo "</div>"
    . "<div id=\"main0\">";
    OpenTable();
}

function head_simple() {
    global $module, $css, $theme, $style;

    echo "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\"
        . "\"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">"
        . "<html xmlns=\"http://www.w3.org/1999/xhtml\" lang=\"en\" xml:lang=\"en\">"
        . "<head>"
        . "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\"/>"
        . "<title>".$_SITENAME." :: ".ucwords($module)."</title>"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/.$css./fonts.css\"
title=\"default\" />"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/.$css./colors.css\"
title=\"default\" />"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/.$css./layout.css\"
title=\"default\" />"
        . "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\" href=\"css/.$css./forms.css\"
title=\"default\" />";
    if ($style)
        echo "<link rel=\"stylesheet\" media=\"screen, projection\" type=\"text/css\"
href=\"themes/.$theme./\".$style.\"\" title=\"default\" />";
    echo "<link rel=\"icon\" type=\"image/jpg\" href=\"images/logo.jpg\" />"
        . "<link rel=\"shortcut icon\" type=\"image/jpg\" href=\"images/logo.jpg\" />"
        . "</head>"
        . "<body>";
}

```

```
function foot() {
    CloseTable();
    echo "</div>";
    CloseMain();
    echo "<div id=\"footer\">";
    OpenFooter();
    footmsg();
    CloseFooter();
    echo "</div></div></body></html>";
}

function date_control($dd,$mm,$yyyy) {
    $date_[1] = time();
    $date_[2] = mktime(0,0,0,date("m")-$mm,date("d")-$dd,date("Y")-$yyyy);
    $date_[3] = date("Y-m-d",$date_[1] );
    $date_[4] = date("Y-m-d",$date_[2] );

    return $date_;
}

function footmsg() {
    global $start_time;
    $mtime = microtime();
    $mtime = explode(" ", $mtime);
    $mtime = $mtime[1] + $mtime[0];
    $end_time = $mtime;
    $total_time = $end_time - $start_time;
    $total_time = _PAGEGENERATION." ".substr($total_time,0,4)."._SECONDS;
    echo $total_time;
}

?>
```



```
<?
//site
define("_SITENAME", "eTSP (Módulo Pedagógico sobre Problemas de Optimização Combinatória com base no TSP)");
define("_OFSITENAME", "do eTSP");

//geral
define("_WELCOMETOWEB", "Bem vindo à plataforma de teste ");
define("_SLOGAN", _WELCOMETOWEB._OFSITENAME);
define("_HOME", "Página Principal");
define("_USER", "Utilizador");
define("_ANONYMUS", "Anónimo");
define("_VISIT", "Visitante");
define("_ABOUT", "Acerca de ...");
define("_NEWS", "Notícias");
define("_CALENDAR", "Calendário");
define("_USERS", "Utilizadores");
define("_SETTINGS", "Configurações");
define("_VALIDATE", "Validação");
define("_USERNAME", "Nome de acesso");
define("_PASSWORD", "Senha:");
define("_USERORPASSINCORRECT", "Dados incorrectos.");
define("_LOGIN", "Entrar");
define("_LOGOUT", "Sair");
define("_WELCOMEUSER", "Bem vindo utilizador:");
define("_REGISTER", "Novo Registo");
define("_LOSTPASS", "Recuperação da senha");
define("_PAGEGENERATION", "Geração da página:");
define("_SECONDS", "segundos");
define("_GOBACK", "Retornar");
define("_REDFIELD", "Por favor, preencha os campos marcados a vermelho correctamente.");
define("_LOGINAS", "Entrou como: ");
define("_NEWNEWS", "Criar notícia");
define("_CLEARNEWS", "Apagar antigas");
define("_CONFIRMDELETE", "Tem a certeza que quer apagar...");

//news
define("_VIEWALLNEWS", "Ver todas as notícias");
define("_CREATENEW", "Criar nova");
```

```
define("_DELETENEWSOLDERTHEN", "Apagar notícias com mais de:");
define("_DAYS", "dias");
define("_DELETE", "Apagar");
define("_REDFIELDS", "(Por favor, verifique os campos marcados com * vermelho.)");
define("_POSTED", "Colocado em");
define("_HEADLINE", "Cabeçalho:");
define("_NEWS_", "Notícia:");
define("_SUBMIT", "Guardar");
define("_PREVIEW", "Pré-visualizar");
define("_ALL", "Todos");
define("_ADMINS", "Administradores");
define("_NEWSFOR", "Notícia para:");

//user
define("_VIEWME", "Ver a minha ficha");
define("_VIEWALL", "Todos os utilizadores");
define("_SEARCHRESULT", "Resultados da pesquisa");
define("_NORESULT", "Não obtive resultados");
define("_SEARCHUSER", "Pesquisar utilizadores");
define("_SEARCH", "Pesquisar");
define("_USERID", "ID");
define("_USERBI", "Bilhete de identidade");
define("_USERNAME_", "Nome");
define("_USERTEL", "Telefone");
define("_USERTLM", "Telemóvel");
define("_USERPASS", "Password");
define("_USERMAIL", "Correio electrónico");
define("_USERLEVEL", "Tipo de utilizador");
define("_USERSTATE", "Estado do utilizador");
define("_CHANGE", "Alterar");
define("_IGNORE", "Ignorar");
define("_INSERT", "Inserir");
define("_ADMIN", "Administrador");
define("_ACTIVE", "Activo");
define("_INACTIVE", "Inactivo");
define("_FIND", "Procurar");
define("_ORDERBY", "Ordernar por:");
```

```
//reg
define("_REGFIRST","Registar - Dados pessoais");
define("_REGSECOND","Registar - Confirmar");
define("_REGTHIRD","Registar - Registo completo");
define("_REGABORT","Registar - Registo incompleto");
define("_REGFOUR","Registar - Registo activado");
define("_REGACTIVATE","A partir de agora, já pode entrar e navegar na nossa página sem restrições.<br>Não se esqueça de preencher os seus restantes dados pessoais.<br>Note que estes dados só serão usados para uso interno.");
define("_ENJOY","Boa navegação na página...");
define("_PASSFIRST","Recuperação da senha - Dados pessoais");
define("_PASSSECOND","Recuperação da senha - Completo");
define("_NEXT","Seguinte");
define("_INUSE","já em uso");
define("_ERRORPASS","A senha tem que ter no mínimo 6 caracteres");
define("_CONFIRMTEXT","Por favor, confirme os seus dados.
<br/>Note que será enviado para o seu correio electrónico
<br/>o \"link\" de activação do seu registo e que sem este não poderá entrar.");
define("_CONFIRMSENDPASS","Receberá na sua caixa de correio electrónico nos próximos minutos uma senha gerada aleatoriamente<br>que poderá alterar posteriormente na própria página.");
define("_BACK","Anterior");
define("_MAISENT","Registo efectuado com êxito.");
define("_MAILNOTSENT","Registo abortado.");
define("_MAILWILLBESENT","Receberá na sua caixa de correio electrónico o \"link\" de activação de registo nos próximos minutos.<br>Obrigado pelo seu registo.");
define("_MAILWILLNOTBESENT","A ligação com o servidor de correio electrónico foi recusada, desta forma, o registo não foi efetuado.<br>Por favor, tente novamente.");
define("_SUBJECT1","Registo no site da SADMP");
define("_SUBJECT2","Recuperação da palavra chave do site da SADMP");
define("_EMAILMSG1","Para activar o registo efectuado\ndeve de aceder ao \"link\" que lhe é enviado abaixo.");
define("_EMAILMSG2","Atendendo ao seu pedido, foi-lhe enviada uma senha gerada aleatoriamente\ne poderá alterar posteriormente na própria página.");
define("_DEARUSER","Caro utilizador");

//calendar
define("_OF","de");
define("_JANUARY","Janeiro");
define("_FEBRUARY","Fevereiro");
define("_MARCH","Março");
```

```
define("_APRIL", "Abril");
define("_MAY", "Maio");
define("_JUNE", "Junho");
define("_JULY", "Julho");
define("_AUGUST", "Agosto");
define("_SEPTEMBER", "Setembro");
define("_OCTOBER", "Outubro");
define("_NOVEMBER", "Novembro");
define("_DECEMBER", "Dezembro");
define("_VIEW_MONTH_SUNDAY_SHORT", "Dom");
define("_VIEW_MONTH_MONDAY_SHORT", "Seg");
define("_VIEW_MONTH_TUESDAY_SHORT", "Ter");
define("_VIEW_MONTH_WEDNESDAY_SHORT", "Qua");
define("_VIEW_MONTH_THURSDAY_SHORT", "Qui");
define("_VIEW_MONTH_FRIDAY_SHORT", "Sex");
define("_VIEW_MONTH_SATURDAY_SHORT", "Sab");
define("_VIEW_MONTH_TITLE", "Calendário de Eventos");
define("_VIEW_MONTH_TODAYS_DATE", "Data de Hoje:");
define("_MANAGE_CATEGORIES", "Gerir categorias");
define("_SUBMIT_EVENT_INFO", "Adicionar um evento");
define("_PRINTER_FRIENDLY_VERSION", "Versão para Impressão");
define("_VIEW_MONTH_SUNDAY", "Domingo");
define("_VIEW_MONTH_MONDAY", "Segunda");
define("_VIEW_MONTH_TUESDAY", "Terça");
define("_VIEW_MONTH_WEDNESDAY", "Quarta");
define("_VIEW_MONTH_THURSDAY", "Quinta");
define("_VIEW_MONTH_FRIDAY", "Sexta");
define("_VIEW_MONTH_SATURDAY", "Sábado");
define("_VIEW_MONTH_GOTO_MONTH", "Ir para o mês");
define("_VIEW_DAY_GOTO_DAY", "Ir para o dia");
define("_EVENTS_CAT", "Categorias de eventos");
define("_CAT_EXIST", "Categorias existentes");
define("_VIEW_CAT", "Ver/Editar");
define("_ADD_NEW_CAT", "Adicionar nova categoria");
define("_EVENT_CAT", "Categoria");
define("_TITLE", "Título");
define("_DESCRIPTION", "Descrição");
define("_NO_TITLE", "(Tem que atribuir um título)");
```

```
define("_TITLE_EXIST", "(Esse título já existe)");
define("_DELETE_CAT", "Apagar");
define("_VIEW_MONTH_DAY", "Dia:");
define("_VIEW_MONTH_MONTH", "Mês:");
define("_VIEW_MONTH_YEAR", "Ano:");
define("_RETURN_TO_CALENDAR", "Voltar ao calendário");
define("_LOCATION", "Localização");
define("_STARTTIME", "Hora de início");
define("_DURATION", "Duração");
define("_DATE", "Data");
define("_RECUR", "Recorrência");
define("_NO_RECUR", "sem recorrência");
define("_RECUR_TYPE", "Tipo de recorrência");
define("_RECUR_WEEK", "Dia da semana da recorrência");
define("_WEEKLY", "semanalmente");
define("_MONTHLY", "mensalmente");
define("_YEARLY", "anualmente");
define("_YES", "Sim");
define("_NO", "Não");
define("_INCORRECT_DATE", "(Data incorreta)");
define("_EVENTS_OF_CALENDAR", "Eventos do calendário");
define("_VIEWALLEVENTS", "Ver todos os eventos");
define("_SEARCHEVENT", "Pesquisar eventos");
define("_ALL_EVENTS", "Todos os eventos");
define("_INI_DATE", "Depois da data");
define("_END_DATE", "Antes da data");
define("_APP", "Aprovado");
define("_EVENT_INFO", "Dados do evento");
define("_VIEW_DAY_NO_EVENTS", "Não há eventos marcados para este dia.");
define("_ICAL", "Exportar eventos");

//data
define("_ALG", "Heurísticas");
define("_ALG1", "\"Vizinho mais próximo\"");
define("_ALG2", "\"Inserção mais próxima\"");
define("_ALG3", "\"2opt\" foreground");
define("_ALG3I", "\"2opt\" background");
define("_ALG4", "\"Grasp\"");
```

```
define("_ALG5", "Método exacto");

define("_FILE_MANAGE", "Gestão de ficheiros");
define("_FILE_COMPARE", "Comparação de simulações");

define("_DATA", "Dados");
define("_CALC_DIST", "Pré-cálculo");
define("_CALC_ORDER", "Cálculo");
define("_RESULT", "Resultado");
define("_SHOW_RESULT", "Resultado");
define("_IMPORT_DATA", "Importar dados");
define("_EXPORT_DATA", "Exportar dados");
define("_INSERT_DATA", "Inserir dados");
define("_NCIDADES", "Nº de cidades");
define("_GO", "Seguinte");
define("_OK", "Sucesso");
define("_FILE", "Ficheiro");
define("_FILES", "Ficheiros");
define("_GEN_RAND_COORDINATES", "Gerir cidades aleatórias");
define("_CIDADE", "Cidade");
define("_CIDADES", "Cidades");
define("_CIDADES_ORDER", "Cidades ordenadas");
define("_COORDINATES", "Coordenadas");
define("_DATA_SAVE", "Gravação dos dados");
define("_DROPTABLES", "Apagar tabelas");
define("_CREATETABLE1", "Criar tabela de dados");
define("_CREATETABLE2", "Criar tabela de distâncias");
define("_CREATETABLE3", "Criar tabela de solução");
define("_INSERTDATATABLE1", "Inserir dados na tabela de dados");
define("_CALC_TIME", "Tempo de cálculo");
define("_DB_ACCESS_TIME", "Tempo de acesso à base de dados");
define("_CIDADES_TABLE", "Tabela de cidades");
define("_DIST_TOTAL", "Distância total");
define("_CIDADES_GRAF", "Gráfico das cidades");
define("_MENU", "Menu");
define("_RED_DOT_ON", "Ponto vermelho em");
define("_SHOW_TABLE_RESULT", "Tabela das cidades");
define("_SHOW_INFO_RESULT", "Informação do cálculo");
```

```
define("_INFO_RESULT", "Informação do cálculo");
define("_SEED", "Semente");
define("_FILE_NAME", "Nome do ficheiro");
define("_FILE_SAVED", "Ficheiro guardado");
define("_FILE_EXIST", "Ficheiro já existente");
define("_MAX_NUMBER_OF_TRADES", "Nº máximo de melhoramentos");
define("_TRADES", "Melhoramentos");
define("_IMPORT_FAILED", "Importação dos dados não efectuada");
define("_VALUE_CIDADES_BETWEEN", "Nº de cidades entre");
define("_AND", "e");
define("_VALUE_RAND_BETWEEN", "Valor de geração aleatória entre");
define("_PERCENT_BETWEEN", "Percentagem entre");
define("_TRADE_EVALUATION_PERCENT", "Percentagem de avaliações de trocas");
define("_KILL_PREVIOUS_SIMULATION", "Abortar a simulação corrente");
define("_UPLOAD_FILE", "Carregar ficheiro");
define("_DOWNLOAD_FILE", "Descarregar ficheiro");
define("_DELETE_FILE", "Apagar ficheiro");
define("_FILE_MANAGEMENT", "Gestão de ficheiros");
define("_ALREADY_SIMULATING", "Já existe uma simulação a correr");
define("_SIMULATION_RUNNING_IN_BACKGROUND", "Simulação a correr em background");
define("_PERCENT", "Percentagem");
define("_IN_PROGRESS", "Em execução");
define("_DONE", "Terminado");
define("_STATE", "Estado");
define("_CLOSE_WINDOW", "Fechar janela");

//decision
define("_DECISION", "Decisões");

//settings
define("_SETTINGS_ERROR", "Os dados que introduziu não foram aceites,<br>"
    ."por favor verifique-os e tente novamente.");
define("_SETTINGS_DAYS", "Só serão mostradas as notícias dos últimos");
define("_CSS", "Css");
define("_THEME", "Tema");
define("_LANG", "Idioma");
define("_PERSONAL", "Pessoal");
define("_PUBLIC", "Público");
```

```

define("_NONE", "Nenhum");
define("_BLOCK", "Bloco");

//about
define("_WEBSITE", "Plataforma de teste do");
define("_SECURITY", "Aspectos de segurança");
define("_REPORT", "Tese");
define("_AUTHOR", "Autor");

define("_ABOUT_TEXT", "<br><center>._WEBSITE." "_OFSITENAME
    .<h5>Esta página foi criada no âmbito da tese do mestrado integrado e<br/>"
    ."tem como função principal o apoio à tese e suporte ao desenvolvimento da mesma.<br>"
    ."É disponibilizado também um espaço dedicado a notícias e um calendário de eventos.<br><br>"
    ."Boa navegação na página..."
    ."<br></h5>._SECURITY
    .<h5>Para segurança dos nossos utilizadores, o acesso aos serviços disponibilizados<br>"
    ."serão limitados enquanto o utilizador não efectuar um registo e o respectivo \"login\".<br>"
    ."Só será necessário efectuar o registo uma vez e este consiste em criar uma ficha<br>"
    ."de utilizador com um nome de acesso, uma senha de acesso e um correio electrónico.<br>"
    ."Uma vez efectuado o registo, receberá no seu correio electrónico um endereço de activação,<br>"
    ."após o qual poderá efectuar o \"login\" com o nome de acesso e senha de acesso que escolheu.<br>"
    ."Os restantes dados pessoais são optativos não esquecendo que estes servem apenas para uso interno."
    ."<br></h5>._REPORT.</center>");

define("_ABOUT_DENIED", "<center><h5>(Dados só visíveis após \"login\".)</h5></center>");
define("_ABOUT_REPORT", "<center><span class=\"red\"><h5>link do relatório</h5></span></center>");

define("_ABOUT_AUTHOR", "<br><center>._AUTHOR
    .<h5><table width=\"50%\"><tr><td>"
    ."_USERNAME_": Nuno Miguel Teixeira Pinto<br/>"
    ."</td></tr><tr><td>"
    ."Número de aluno: 030503043<br/>"
    ."</td></tr></table></h5></center>");
?>

```



```

<?
function OpenBlock() {
echo "<table border=\0\ cellpadding=\0\ cellspacing=\0\ width=\100%\>"
    ."<tr>"
    ."<td class=\TableImg1\><img src=\themes/smooth/images/spacer.gif\ width=\12\ height=\13\ alt=\\"/></td>"
    ."<td class=\TableImg2\></td>"
    ."<td class=\TableImg3\></td>"
    ."</tr><tr>"
    ."<td class=\TableImg4\></td>"
    ."<td valign=\top\>";
}

function CloseBlock() {
    CloseTable();
}

function OpenTable() {
echo "<table border=\0\ cellpadding=\0\ cellspacing=\0\ width=\100%\>"
    ."<tr>"
    ."<td class=\TableImg1\><img src=\themes/smooth/images/spacer.gif\ width=\12\ height=\13\ alt=\\"/></td>"
    ."<td class=\TableImg2\></td>"
    ."<td class=\TableImg3\></td>"
    ."</tr><tr>"
    ."<td class=\TableImg4\></td>"
    ."<td valign=\top\>";
}

function CloseTable() {
echo "</td><td class=\TableImg6\></td>"
    ."</tr><tr>"
    ."<td class=\TableImg7\></td>"
    ."<td class=\TableImg8\></td>"
    ."<td class=\TableImg9\><img src=\themes/smooth/images/spacer.gif\ width=\12\ height=\13\ alt=\\"/></td>"
    ."</tr></table>";
}

function OpenTable1() {
echo "<table border=\0\ cellpadding=\0\ cellspacing=\0\ width=\100%\>"

```

```

        . "<tr>"
        . "<td><img src=\"themes/smooth/images/box_r1_c1.jpg\" width=\"12\" height=\"13\" alt=\"\"/></td> <td"
width=\"100%\" background=\"themes/smooth/images/box_r1_c2.jpg\"><img src=\"themes/smooth/images/spacer.gif\"
width=\"1\" height=\"1\" alt=\"\"/></td>"
        . "<td><img src=\"themes/smooth/images/box_r1_c3.jpg\" width=\"12\" height=\"13\" alt=\"\"/></td>"
        . "</tr><tr>"
        . "<td background=\"themes/smooth/images/box_r2_c1.jpg\"><img src=\"themes/smooth/images/spacer.gif\"
width=\"1\" height=\"1\" alt=\"\"/></td>"
        . "<td valign=\"top\" bgcolor=\"#EEF2F7\">";
}

function CloseTable1() {
echo "</td>"
        . "<td background=\"themes/smooth/images/box_r2_c3.jpg\"><img src=\"themes/smooth/images/spacer.gif\"
width=\"1\" height=\"1\" alt=\"\"/></td>"
        . "</tr><tr>"
        . "<td><img src=\"themes/smooth/images/box_r3_c1.jpg\" width=\"12\" height=\"13\" alt=\"\"/></td>"
        . "<td background=\"themes/smooth/images/box_r3_c2.jpg\"><img src=\"themes/smooth/images/spacer.gif\"
width=\"1\" height=\"1\" alt=\"\"/></td>"
        . "<td><img src=\"themes/smooth/images/box_r3_c3.jpg\" width=\"12\" height=\"13\" alt=\"\"/></td>"
        . "</tr></table>";
}

function OpenTable2() {
echo "<br/><table width=\"100%\" border=\"0\" cellspacing=\"1\" cellpadding=\"0\" class=\"Table2bg1\"><tr><td>\n";
echo "<table width=\"100%\" border=\"0\" cellspacing=\"1\" cellpadding=\"8\" class=\"Table2bg2\"><tr><td>\n";
}

function CloseTable2() {
echo "</td></tr></table></td></tr></table>\n";
}

function OpenHeader(){
    global $slogan;
echo "<table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" width=\"100%\" class=\"OpenHeader1\">"
        . "<tr>"
        . "<td><img src=\"themes/smooth/images/spacer.gif\" width=\"8\" height=\"18\" alt=\"\"/></td>"
        . "<td><span class=\"slogan\">\"._SLOGAN.\"</span></td>"

```



```
}
```

```
function CloseFooter() {
```

```
echo "</span></td></tr></table>";
```

```
}
```

```
?>
```

```

<?
if (!ereg("modules.php", $_SERVER['PHP_SELF'])) {
    die (header("location: ../../access_denied.php"));
}

include_once("../classes/a_cidades.php");

function export_data($aux){
    if ($aux=="submit" && $file_s=$_REQUEST[file_s]){
        $rep=opendir('./xml');
        $flag=0;
        while ($file = readdir($rep)){
            if (is_file("./xml/".$file)){
                if ($file_s.".xml"==$file)
                    $flag=1;
            }
        }
    }
    closedir($rep);
    clearstatcache();
    if ($flag)
        header("location: modules.php?name=data&op=alg1&opx=export_data&aux=erro&file_s=$file_s");
    else{
        $cidades = a_cidades::getCidades();
        $_time=a_cidades::getTime();
        $content="<?xml version='1.0'?'>\r\n"
            . "<SIMULATION>\r\n"
            . " <CIDADES>\r\n";
        foreach($cidades as $cidade){
            $content.=" <CIDADE>\r\n"
                . " <NOME>". $cidade->nome. "</NOME>\r\n"
                . " <X>". $cidade->x. "</X>\r\n"
                . " <Y>". $cidade->y. "</Y>\r\n"
                . " </CIDADE>\r\n";
        }
        $content.=" </CIDADES>\r\n";
        $cidades_o=a_cidades::getCidades_o();
        $diff = array_diff($cidades, $cidades_o);
        if (!$diff){

```

```

$content.=" <CIDADES_O>\r\n";
foreach($cidades_o as $cidade){
    $content.=" <CIDADE>\r\n"
        ." <NOME>". $cidade->nome."</NOME>\r\n"
        ." <X>". $cidade->x."</X>\r\n"
        ." <Y>". $cidade->y."</Y>\r\n"
        ." </CIDADE>\r\n";
}
$content.=" </CIDADES_O>\r\n";
}
$content.=" <DADOS>\r\n"
    ." <DATA>\r\n"
    ." <TYPE>_ALG1</TYPE>\r\n"
    ." <TIME1>".substr($_time['time2'],0,4)."</TIME1>\r\n"
    ." <TIME2>".substr(($_time['time1']-$time['time2']),0,4)."</TIME2>\r\n"
    ." </DATA>\r\n"
    ." </DADOS>\r\n"
    ."</SIMULATION>\r\n";
$fd = fopen("./xml/".$file_s.".xml", "w");
fwrite($fd, $content);
fclose($fd);
header("location: modules.php?name=data&op=alg1&opx=export_data&aux=saved&file_s=$file_s");
}
}else{
    index();
    alg1();
    CloseTable();
    OpenTable();
    echo "<table width=\%100%\><tr><td class=\%leftcontent%\>._EXPORT_DATA.</td></tr></table>";
    echo "<form class=\%content%"
action=\%modules.php?name=data&op=alg1&opx=export_data&aux=submit" method=\%post">
    ."<p><label for=\%file_s%\>._FILE_NAME.</label> "
    ."<input type=\%text" name=\%file_s" id=\%file_s" value=\%".$_REQUEST[file_s]."%/></p>"
    ."<input type=\%submit" value=\%".$_GO."% />";
    if ($aux=="erro")
        echo "<p><span class=\%tiny_red%\>._FILE_EXIST.</span></p>";
    if ($aux=="saved")
        echo "<p><span class=\%content%\>._FILE_SAVED.</span></p>";
}

```

```

    foot();
}
}

function data_type(){
    $rep=opendir('./xml');
    $flag=0;
    while ($file = readdir($rep)){
        if (is_file("./xml/".$file)){
            $file_=split("[.]", $file);
            if($file != '.' && $file != '..' && $file != "" && $file_[1]=="xml")
                $files[]=$file_[0];
        }
    }
    closedir($rep);
    clearstatcache();

    CloseTable();
    OpenTable();
    echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>._DATA."</td></tr></table>";
    if ($files){
        OpenTable2();
        echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>._INSERT_DATA."</td></tr></table>";
    }

    echo "<form class=\`content\`"
action=\`modules.php?name=data&op=alg1&opx=data_insert&rand=100\` method=\`post\`"
    . "<p><label for=\`ncidades\`>._NCIDADES.": </label> "
    . "<input type=\`text\` name=\`ncidades\` id=\`ncidades\` value=\`".$_REQUEST[ncidades]."\`/></p>"
    . "<input type=\`submit\` value=\`".$_GO." \` />"
    . "</form>";

    if ($_REQUEST[ncidades]<=0 || $_REQUEST[ncidades]>200)
        echo "<p><span class=\`tiny_red\`>._VALUE_CIDADES_BETWEEN." 1 "._AND." 200.</span></p>";

    if ($files){
        CloseTable2();
        OpenTable2();
        echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>._IMPORT_DATA."</td></tr></table>";
        echo "<form class=\`content\`" action=\`modules.php?name=data&op=alg1&opx=import_data\`"
method=\`post\`>"

```

```

        . "<p><label for=\"ncidades\">\"._FILE.\": </label>\"
        . "<select name=\"file_s\" id=\"file_s\">";
    foreach($files as $file_)
        echo "<option value=\"\".$file_.\">\".$file_.\"</option>\";
    echo "</select></p>\"
        . "<input type=\"submit\" value=\"\"._GO.\" />\"
        . "</form>\";
    CloseTable2();
}
}

function import_data(){
    require(\"./includes/xml_parser.php\");
    $xml=getXML(\"./xml/\".$_REQUEST[file_s].\".xml\");
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._INSERT.\"</td></tr></table>\";
    if ($xml){
        $cidades=$xml[cidades];
        echo "<form action=\"modules.php?name=data&op=alg1&opx=data_insert_result\" method=\"post\">\";
        $c=-1;
        foreach ($cidades as $cidade){$c++;
            echo "<p><label for=\"cidade[\".$c.\"][nome]\">\"._CIDADE.\": </label> \"
                . "<input type=\"text\" name=\"cidade[\".$c.\"][nome]\" id=\"cidade[\".$c.\"][nome]\"
value=\"\".$cidade[nome].\" /> \"
                . \"_COORDINATES
                . \" ( <input type=\"text\" name=\"cidade[\".$c.\"][x]\" id=\"cidade[\".$c.\"][x]\" size=\"3\"
value=\"\".$cidade[x].\" /> \"
                . \"; <input type=\"text\" name=\"cidade[\".$c.\"][y]\" id=\"cidade[\".$c.\"][y]\" size=\"3\" value=\"\".$cidade[y].\" />
        )</p>\";
        }
        echo "<input type=\"submit\" value=\"\"._GO.\" />\"
            . "</form>\";

    }else
        echo "<p><span class=\"tiny_red\">\"._IMPORT_FAILED.\"</span></p>\";
}
}

```



```

function data_insert(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._INSERT.</td></tr></table>"
        . "<table class=\"content\" width=\"100%\"><tr><td>";
    echo "<form action=\"modules.php?name=data&amp;op=alg1&amp;opx=data_insert\" method=\"post\">"
        . "<label for=\"rand\">._GEN_RAND_COORDINATES.</label> "
        . "<input type=\"text\" name=\"rand\" id=\"rand\" value=\"".$_REQUEST[rand]."\"/>"
        . "<input type=\"hidden\" name=\"ncidades\" id=\"ncidades\" value=\"".$_REQUEST[ncidades]."\"/>"
        . "<input type=\"submit\" value=\"._GO.\" />"
        . "</form>";
    if ($_REQUEST[rand]<=0 || $_REQUEST[rand]>4000)
        echo "<p><span class=\"tiny_red\">._VALUE_RAND_BETWEEN." 1 "._AND." 4000.</span></p>";
    echo "<form action=\"modules.php?name=data&amp;op=alg1&amp;opx=data_insert_result\" method=\"post\">";
    $c=-1;
    while ($c++<$_REQUEST[ncidades]-1){
        echo "<p><label for=\"cidade[".$c."][nome]\">._CIDADE.</label> "
            . "<input type=\"text\" name=\"cidade[".$c."][nome]\" id=\"cidade[".$c."][nome]\" value=\"";
        if ($_REQUEST[rand])
            echo "p". $c;
        echo "\"/> "
            . "_COORDINATES
            .": ( <input type=\"text\" name=\"cidade[".$c."][x]\" id=\"cidade[".$c."][x]\" size=\"3\" value=\"";
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\"/> "
            . "; <input type=\"text\" name=\"cidade[".$c."][y]\" id=\"cidade[".$c."][y]\" size=\"3\" value=\"";
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\"/> )</p>";
    }
    echo "<input type=\"submit\" value=\"._GO.\" />"
        . "</form></td></tr></table>";
}

function data_insert_result(){
    CloseTable();
    OpenTable();

```

```

echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>"._DATA_SAVE."</td></tr></table>";
OpenTable2();
echo "<span class=\`content\`>";
echo "_DROPTABLES.": "; if (a_cidades::dropTables()) echo _OK;
echo "<br>"._CREATETABLE1.": "; if (a_cidades::createTable1()) echo _OK;
echo "<br>"._CREATETABLE2.": "; if (a_cidades::createTable2($_REQUEST[cidade])) echo _OK;
echo "<br>"._CREATETABLE3.": "; if (a_cidades::createTable3()) echo _OK;
echo "<br>"._INSERTDATATABLE1.": "; if (a_cidades::insertdataTable1($_REQUEST[cidade])) echo _OK;
echo "</span>";
CloseTable2();
echo "<br><form action=\`modules.php?name=data&amp;op=alg1&amp;opx=data_calc_dist\` method=\`post\`>"
    . "<input type=\`submit\` value=\`"._CALC_DIST." \` />"
    . "</form>";
}

function data_calc_dist(){
    CloseTable();
    OpenTable();
    echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>"._CALC_DIST."</td></tr></table>";
    $stime1 = microtime();
    $cidades = a_cidades::getCidades();
    $n=count($cidades)-1;
    //----- calculo das distancias
    $stime2 = microtime();
    $i=-1;
    foreach ($cidades as $cidade) {
        $i++;
        $j=$i;
        $cidades_dist[$cidades[$i]->nome][$cidades[$i]->nome]=0;
        while ($j++<$n){
            $x=pow($cidades[$i]->x-$cidades[$j]->x,2);
            $y=pow($cidades[$i]->y-$cidades[$j]->y,2);
            $cidades_dist[$cidades[$i]->nome][$cidades[$j]->nome]=$x+$y;
            $cidades_dist[$cidades[$j]->nome][$cidades[$i]->nome]=$x+$y;
        }
        //          $cidade->a_distancia_insert($cidades[$i]->nome,$cidades[$j]->nome,$x+$y);
    }
}
$etime2 = microtime();

```

```

$cidade->a_distancia_insert($cidades_dist);
$time1 = microtime();
//-----calculo do tempo
$time = explode(" ", $time1);
$start_time1 = $time[1] + $time[0];
$time = explode(" ", $time1);
$end_time1 = $time[1] + $time[0];
$time = explode(" ", $time2);
$start_time2 = $time[1] + $time[0];
$time = explode(" ", $time2);
$end_time2 = $time[1] + $time[0];

        OpenTable2();
        echo "<span class=\"content\">";
        echo _CALC_DIST.: "._OK."<br>";
        echo _CALC_TIME.: ".substr(($end_time2 - $start_time2),0,4)."._SECONDS."<br>";
        echo _DB_ACCESS_TIME.: ".substr((( $end_time1 - $start_time1)-($end_time2 - $start_time2)),0,4)."
        ._SECONDS;
        echo "</span>";
        CloseTable2();
        echo "<br><form action=\"modules.php?name=data&op=alg1&opx=data_calc_order\"
method=\"post\">
        .<input type=\"submit\" value=\"\"._CALC_ORDER.\" />
        .</form>";
    }

function data_calc_order(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._CALC_ORDER.</td></tr></table>";
$cidade=new a_cidades();
$time1 = microtime();
$cidades = a_cidades::getCidades();
$i=count($cidades)-1;
//----- ordenação pelo + prox a começar no 1º da tabela anterior
$cidades_o[] = $cidades[0];
$n=0;
$time2 = microtime();

```

```

while ($i!=0){
    $i--;
    $dist = a_cidades::getDist($cidades_o[$n]->nome);
    unset($dist[nome]);
    foreach ($cidades_o as $cidade_o)
        unset($dist[$cidade_o->nome]);
    foreach ($dist as $key => $val)
        if ($val == min($dist)) $dist_min_key=$key;
    $cidades_o[] = a_cidades::getCidade($dist_min_key);
    $n++;
}
$time2 = microtime();
$cidade_o->insertdataTable3($cidades_o);
$time1 = microtime();
//-----calculo do tempo
$time = explode(" ", $time1);
$start_time1 = $mtime[1] + $mtime[0];
$time = explode(" ", $time1);
$end_time1 = $mtime[1] + $mtime[0];
$time = explode(" ", $time2);
$start_time2 = $mtime[1] + $mtime[0];
$time = explode(" ", $time2);
$end_time2 = $mtime[1] + $mtime[0];
$cidade->updateTime($time1=(($end_time1 - $start_time1), $time2=(($end_time2 - $start_time2)));
    OpenTable2();
    echo "<span class=\"content\">";
    echo _CALC_ORDER.: "._OK."<br>";
    echo _CALC_TIME.: ".substr((($end_time2 - $start_time2),0,4)."._SECONDS."<br>";
    echo _DB_ACCESS_TIME.: ".substr(((($end_time1 - $start_time1)-($end_time2 - $start_time2)),0,4).".
    "_SECONDS";
    echo "</span>";
    CloseTable2();
    echo "<br><form action=\"modules.php?name=data&amp;op=alg1&amp;opx=data_result\" method=\"post\">"
        . "<input type=\"submit\" value=\"\"._SHOW_RESULT.\" \" />"
        . "</form>";
}

function data_result(){

```

```

    CloseTable();
    OpenTable();
        echo "<table width='100%'><tr><td class='leftcontent'>._RESULT.</td></tr></table>";
$scale=a_cidades::getMax();
//$scale['max(x)'];
//$scale['max(y)'];
$pts=" ";
$pts_o=" ";
$cidades[1] = a_cidades::getCidades();
$n=count($cidades[1]);
$cidades[2] = a_cidades::getCidades_o();
if (!$p=$_REQUEST['p'])
    $p=0;
$o_x=$cidades[1][$p]->x;
$o_y=$scale['max(y)']-$cidades[1][$p]->y;

    OpenTable2();
    echo "<table width='100%'><tr><td class='leftcontent'>._MENU.</td></tr></table>"
        ."<table width='100%' class='content'><tr><td>"
        ."<form class='content' action='modules.php?name=data&op=alg1&opx=data_result'"
method="post">"
        ."<label for='p'>._RED_DOT_ON.</label> "
        ."<select id='p' name='p'>";
$_p=0;
foreach($cidades[1] as $cidade){
    echo "<option value='$_p.'$p.'";
    if ($p==$_p)
        echo " selected";
    echo ">".$cidade->nome."</option>";
    $_p++;
}

    echo "</select><br>"
        ."<input type='checkbox' name='table' id='table'";
if ($table=$_REQUEST['table'])
    echo " checked";
echo "><label for='table'>._SHOW_TABLE_RESULT.</label><br>"

```

```

        ."<input type=\"checkbox\" name=\"info\" id=\"info\"";
if ($info=$_REQUEST['info'])
    echo " checked";
echo "><label for=\"info\">\"._SHOW_INFO_RESULT.\"</label><br>"
        ."<input type=\"submit\" value=\"\"._GO.\" />"
        ."</form></td></tr></table>";
        CloseTable2();

if ($table){
    OpenTable2();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._CIDADES_TABLE.\"</td></tr></table>"
//    ."<table width=\"100%\" class=\"content\"><tr><td>"
        ."<table width=\"100%\" border=1 class=\"content\">"
            ."<tr align=\"center\"><td width=\"33%\" colspan=\"3\">\"._CIDADES.\"</td><td width=\"33%\"
colspan=\"3\">\"._CIDADES_ORDER.\"</td></tr>";
        }
    $i=0;
    $dist_1=0;
    $dist_2=0;
    while ($i<$n-1){
        $pts_=$cidades[1][$i]->x.", " . ($scale['max(y)']-$cidades[1][$i]->y)." ";
        $pts_o_=$cidades[2][$i]->x.", " . ($scale['max(y)']-$cidades[2][$i]->y)." ";

        $x=pow($cidades[1][$i]->x-$cidades[1][$i+1]->x,2);
        $y=pow($cidades[1][$i]->y-$cidades[1][$i+1]->y,2);
        $dist_1+=sqrt($x+$y);

        $x=pow($cidades[2][$i]->x-$cidades[2][$i+1]->x,2);
        $y=pow($cidades[2][$i]->y-$cidades[2][$i+1]->y,2);
        $dist_2+=sqrt($x+$y);
    }
    if ($table)
        echo "<tr><td>\".$cidades[1][$i]->nome.\"</td>"
            ."<td align=\"right\">\".$cidades[1][$i]->x.\"</td>"
            ."<td align=\"right\">\".$cidades[1][$i]->y.\"</td><td></td>"
            ."<td>\".$cidades[2][$i]->nome.\"</td>"
            ."<td align=\"right\">\".$cidades[2][$i]->x.\"</td>"
            ."<td align=\"right\">\".$cidades[2][$i]->y.\"</td></tr>";
    $i++;
}

```

```

}

    $pts.= $cidades[1][$i]->x.", " . ($scale['max(y)']-$cidades[1][$i]->y). " ";
    $pts_o.= $cidades[2][$i]->x.", " . ($scale['max(y)']-$cidades[2][$i]->y). " ";
if ($table)
echo "<tr><td>". $cidades[1][$i]->nome. "</td>"
    . "<td align='right'>". $cidades[1][$i]->x. "</td>"
    . "<td align='right'>". $cidades[1][$i]->y. "</td><td></td>"
    . "<td>". $cidades[2][$i]->nome. "</td>"
    . "<td align='right'>". $cidades[2][$i]->x. "</td>"
    . "<td align='right'>". $cidades[2][$i]->y. "</td></tr>";

    $x=pow($cidades[1][$i]->x-$cidades[1][0]->x,2);
    $y=pow($cidades[1][$i]->y-$cidades[1][0]->y,2);
    $dist_1+=sqrt($x+$y);

    $x=pow($cidades[2][$i]->x-$cidades[2][0]->x,2);
    $y=pow($cidades[2][$i]->y-$cidades[2][0]->y,2);
    $dist_2+=sqrt($x+$y);
if ($table){
echo "<tr><td>". _DIST_TOTAL. "</td>"
    . "<td colspan='2' align='right'>". $dist_1. "</td><td></td>"
    . "<td>". _DIST_TOTAL. "</td>"
    . "<td colspan='2' align='right'>". $dist_2. "</td></tr>"
    . "</table>";
    CloseTable2();
}

    OpenTable2();
$args1="scale_x=". $scale['max(x)']. "&&scale_y=". $scale['max(y)']. "&&pts=". $pts. "&&o_x=". $o_x. "&&o_y=". $o_y;
$args2="scale_x=". $scale['max(x)']. "&&scale_y=". $scale['max(y)']. "&&pts=". $pts_o. "&&o_x=". $o_x. "&&o_y=". $o_y;
echo "<table width='100%'><tr><td class='leftcontent'>". _CIDADES_GRAF. "</td></tr></table>"
    . "<table width='100%' border=1>"
    . "<tr align='center'><td width='33%' colspan='3' bgcolor='white'>"
    . "<a href='modules.php?name=data&&op=svg&&'. $arg1. "</td>" target='_blank'>"
    . "<iframe src='modules/data/svg.php?'. $arg1. "</td>" width='300' height='300'></iframe>"
    . "</a></td><td width='33%'></td><td width='33%' colspan='3' bgcolor='white'>"
    . "<a href='modules.php?name=data&&op=svg&&'. $arg2. "</td>" target='_blank'>"
    . "<iframe src='modules/data/svg.php?'. $arg2. "</td>" width='300' height='300'></iframe>"
    . "</a></td></tr></table>";

```

```

        CloseTable2();
    if ($info){
        $_time=a_cidades::getTime();
        OpenTable2();
        echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._INFO_RESULT.</td></tr></table>"
            . "<table width=\"100%\" class=\"content\"><tr><td>"
            . "_CALC_TIME.": ".substr($_time['time2'],0,4)." . "_SECONDS." <br>"
            . "_DB_ACCESS_TIME.": ".substr(($_time['time1']-$_time['time2']),0,4)." . "_SECONDS"
            . "</td></tr></table>";
        CloseTable2();
    }
}

switch($_REQUEST["opx"]) {

    case "export_data":
        export_data($_REQUEST[aux]);
        break;

    case "import_data":
        index();
        alg1();
        import_data();
        foot();
        break;

    case "data_type":
        index();
        alg1();
        data_type();
        foot();
        break;

    case "data_insert":
        index();
        alg1();
        if ($_REQUEST[ncidades]>0&&$_REQUEST[ncidades]<=200)
            data_insert();

```



```
else
    data_type();
foot();
break;

case "data_insert_result":
    index();
    alg1();
    data_insert_result();
    foot();
    break;

case "data_calc_dist":
    index();
    alg1();
    data_calc_dist();
    foot();
    break;

case "data_calc_order":
    index();
    alg1();
    data_calc_order();
    foot();
    break;

case "data_result":
    index();
    alg1();
    data_result();
    foot();
    break;

default:
    index();
    alg1();
    foot();
    break;
```

}

?>

```

<?
if (!ereg("modules.php", $_SERVER['PHP_SELF'])) {
    die (header("location: ../../access_denied.php"));
}

include_once("../classes/b_cidades.php");

function export_data($aux){
    if ($aux=="submit" && $file_s=$_REQUEST[file_s]){
        $rep=opendir('./xml');
        $flag=0;
        while ($file = readdir($rep)){
            if (is_file("./xml/".$file)){
                if ($file_s.".xml"==$file)
                    $flag=1;
            }
        }
    }
    closedir($rep);
    clearstatcache();
    if ($flag)
        header("location: modules.php?name=data&op=alg2&opx=export_data&aux=erro&file_s=$file_s");
    else{
        $cidades = b_cidades::getCidades();
        $seedruntime=b_cidades::getSeedRunTime();
        $content="<?xml version='1.0'?'>\r\n"
            . "<SIMULATION>\r\n"
            . " <CIDADES>\r\n";
        foreach($cidades as $cidade){
            $content.=" <CIDADE>\r\n"
                . " <NOME>". $cidade->nome."</NOME>\r\n"
                . " <X>". $cidade->x."</X>\r\n"
                . " <Y>". $cidade->y."</Y>\r\n"
                . " </CIDADE>\r\n";
        }
        $content.=" </CIDADES>\r\n";
        $cidades_o=b_cidades::getCidades_o();
        $diff = array_diff($cidades, $cidades_o);
        if (!$diff){

```

```

$content.=" <CIDADES_O>\r\n";
foreach($cidades_o as $cidade){
    $content.=" <CIDADE>\r\n"
        ." <NOME>". $cidade->nome."</NOME>\r\n"
        ." <X>". $cidade->x."</X>\r\n"
        ." <Y>". $cidade->y."</Y>\r\n"
        ." </CIDADE>\r\n";
}
$content.=" </CIDADES_O>\r\n";
}
$content.=" <DADOS>\r\n"
    ." <DATA>\r\n"
    ." <TYPE>_ALG2</TYPE>\r\n"
    ." <TIME1>".substr($seedruntime['time2'],0,4)."</TIME1>\r\n"
    ." <TIME2>".substr(($seedruntime['time1']-$seedruntime['time2']),0,4)."</TIME2>\r\n"
    ." <SEED>". $seedruntime['seed']."</SEED>\r\n"
    ." </DATA>\r\n"
    ." </DADOS>\r\n"
    ."</SIMULATION>\r\n";
$fd = fopen("./xml/".$file_s.".xml", "w");
fwrite($fd, $content);
fclose($fd);
header("location: modules.php?name=data&op=alg2&opx=export_data&aux=saved&file_s=$file_s");
}
}else{
    index();
    alg2();
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._EXPORT_DATA.</td></tr></table>";
    echo "<form action=\"modules.php?name=data&op=alg2&opx=export_data&aux=submit\"
method=\"post\">"
        ."<p><label for=\"file_s\">._FILE_NAME.</label> "
        ."<input type=\"text\" name=\"file_s\" id=\"file_s\" value=\"".$REQUEST[file_s]."\"/></p>"
        ."<input type=\"submit\" value=\"._GO.\" />";
    if ($aux=="erro")
        echo "<p><span class=\"tiny_red\">._FILE_EXIST.</span></p>";
    if ($aux=="saved")

```

```

        echo "<p><span class=\"content\">\"._FILE_SAVED.\"</span></p>";
    foot();
}
}

function data_type(){
    $rep=opendir('./xml');
    $flag=0;
    while ($file = readdir($rep)){
        if (is_file("./xml/".$file)){
            $file_=split(".", $file);
            if($file != '.' && $file != '..' && $file != " " && $file_[1]=="xml")
                $files[]=$file_[0];
        }
    }
    closedir($rep);
    clearstatcache();

    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._DATA.\"</td></tr></table>";
    if ($files){
        OpenTable2();
        echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._INSERT_DATA.\"</td></tr></table>";
    }
    echo "<form class=\"content\"
action=\"modules.php?name=data&op=alg2&opx=data_insert&rand=100\" method=\"post\">
    .<p><label for=\"ncidades\">\"._NCIDADES.\": </label> \"
    .<input type=\"text\" name=\"ncidades\" id=\"ncidades\" value=\"\".$_REQUEST[ncidades].\"/></p>\"
    .<input type=\"submit\" value=\"\"._GO.\" />\"
    .</form>";
    if ($_REQUEST[ncidades]<=0 || $_REQUEST[ncidades]>200)
        echo "<p><span class=\"tiny_red\">\"._VALUE_CIDADES_BETWEEN.\" 1 \"._AND.\" 200.</span></p>";
    if ($files){
        CloseTable2();
        OpenTable2();
        echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._IMPORT_DATA.\"</td></tr></table>";
    }
}

```

```

        echo "<form class=\"content\" action=\"modules.php?name=data&op=alg2&opx=import_data\"
method=\"post\">"
        . "<p><label for=\"ncidades\">\"_FILE.\": </label>"
        . "<select name=\"file_s\" id=\"file_s\">";
        foreach($files as $file_)
            echo "<option value=\"\".$file_.\".\".$file_.\"</option>";
        echo "</select></p>"
        . "<input type=\"submit\" value=\"\"._GO.\" />"
        . "</form>";
    CloseTable2();
}
}

function import_data(){
    require("../includes/xml_parser.php");
    $xml=getXML("../xml/\".$_REQUEST[file_s].xml");
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"_INSERT.\"</td></tr></table>";
    if ($xml){
        $cidades=$xml[cidades];
        echo "<form action=\"modules.php?name=data&op=alg2&opx=data_insert_result\" method=\"post\">";
        $c=-1;
        foreach ($cidades as $cidade){$c++;
            echo "<p><label for=\"cidade[\".$c.\"][nome]\">\"_CIDADE.\": </label> "
                . "<input type=\"text\" name=\"cidade[\".$c.\"][nome]\" id=\"cidade[\".$c.\"][nome]\"
value=\"\".$cidade[nome].\" /> "
                . "_COORDINATES
                .": ( <input type=\"text\" name=\"cidade[\".$c.\"][x]\" id=\"cidade[\".$c.\"][x]\" size=\"3\"
value=\"\".$cidade[x].\" /> "
                . "; <input type=\"text\" name=\"cidade[\".$c.\"][y]\" id=\"cidade[\".$c.\"][y]\" size=\"3\" value=\"\".$cidade[y].\" />
        )</p>";
        }
        if ($xml[dados][0][seed])
            echo "<label for=\"seed\">\"_SEED.\": </label> "
                . "<input type=\"text\" name=\"seed\" id=\"seed\" value=\"\".$xml[dados][0][seed].\" /><br>";
    }
}

```

```

        echo "<input type='submit' value='\"._GO.\"' />"
        . "</form>";

    }else
        echo "<p><span class='tiny_red'>\"._IMPORT_FAILED.\"</span></p>";
    }

function data_insert(){
    CloseTable();
    OpenTable();
    echo "<table width='100%'><tr><td class='leftcontent'>\"._INSERT.\"</td></tr></table>"
        . "<table class='content' width='100%'><tr><td>";
    echo "<form action='modules.php?name=data&op=alg2&opx=data_insert' method='post'>"
        . "<label for='rand'>\"._GEN_RAND_COORDINATES.\"</label> "
        . "<input type='text' name='rand' id='rand' value='\".$_REQUEST[rand].\"' />"
        . "<input type='hidden' name='ncidades' id='ncidades' value='\".$_REQUEST[ncidades].\"' />"
        . "<input type='submit' value='\"._GO.\"' />"
        . "</form>";

    if ($_REQUEST[rand]<=0 || $_REQUEST[rand]>4000)
        echo "<p><span class='tinyred'>\"._VALUE_RAND_BETWEEN.\" 1 \"._AND.\" 4000.</span></p>";

    echo "<form action='modules.php?name=data&op=alg2&opx=data_insert_result' method='post'>";
    $c=-1;
    while ($c++<$_REQUEST[ncidades]-1){
        echo "<p><label for='cidade[\".$c.\"][nome]'>\"._CIDADE.\"</label> "
            . "<input type='text' name='cidade[\".$c.\"][nome]' id='cidade[\".$c.\"][nome]' value='\"';
        if ($_REQUEST[rand])
            echo "p\".$c;
        echo "\"/> "
            . "_COORDINATES
            .": ( <input type='text' name='cidade[\".$c.\"][x]' id='cidade[\".$c.\"][x]' size='3' value='\"';
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\"/> "
            . "; <input type='text' name='cidade[\".$c.\"][y]' id='cidade[\".$c.\"][y]' size='3' value='\"';
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\"/> )</p>";
    }
}

```

```

        echo "<input type='submit' value='\"".$_GO."' />"
        . "</form></td></tr></table>";
    }

function data_insert_result(){
    CloseTable();
    OpenTable();
    if ($_REQUEST[seed]){
        $cidade=new b_cidades();
        $cidade->updateSeed($_REQUEST[seed],null,null);
    }

    echo "<table width='100%'><tr><td class='leftcontent'>\"._DATA_SAVE.\"</td></tr></table>";
    OpenTable2();
    echo "<span class='content'>";
    echo _DROPTABLES." "; if (b_cidades::dropTables()) echo _OK;
    echo "<br>\"._CREATETABLE1.\" "; if (b_cidades::createTable1()) echo _OK;
    echo "<br>\"._CREATETABLE2.\" "; if (b_cidades::createTable2($_REQUEST[cidade])) echo _OK;
    echo "<br>\"._CREATETABLE3.\" "; if (b_cidades::createTable3()) echo _OK;
    echo "<br>\"._INSERTDATATABLE1.\" "; if (b_cidades::insertdataTable1($_REQUEST[cidade])) echo _OK;
    echo "</span>";
    CloseTable2();
    echo "<br><form action='modules.php?name=data&op=alg2&opx=data_calc_dist' method='post'>"
        . "<input type='submit' value='\"".$_CALC_DIST.\"' />"
        . "</form>";
    }

function data_calc_dist(){
    CloseTable();
    OpenTable();
    echo "<table width='100%'><tr><td class='leftcontent'>\"._CALC_DIST.\"</td></tr></table>";
    $stime1 = microtime();
    $cidades = b_cidades::getCidades();
    $n=count($cidades)-1;
    //----- calculo das distancias
    $stime2 = microtime();
    $i=-1;
    foreach ($cidades as $cidade) {
        $i++;
    }
}

```



```

        $j=$i;
        $cidades_dist[$cidades[$i]->nome][$cidades[$i]->nome]=0;
        while ($j++<$n){
            $x=pow($cidades[$i]->x-$cidades[$j]->x,2);
            $y=pow($cidades[$i]->y-$cidades[$j]->y,2);
            $cidades_dist[$cidades[$i]->nome][$cidades[$j]->nome]=$x+$y;
            $cidades_dist[$cidades[$j]->nome][$cidades[$i]->nome]=$x+$y;
        //            $cidade->a_distancia_insert($cidades[$i]->nome,$cidades[$j]->nome,$x+$y);
        }
    }
}

$etime2 = microtime();
$cidade->a_distancia_insert($cidades_dist);
$etime1 = microtime();
//-----calculo do tempo
$mtime = explode(" ",$etime1);
$start_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ",$etime1);
$end_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ",$etime2);
$start_time2 = $mtime[1] + $mtime[0];
$mtime = explode(" ",$etime2);
$end_time2 = $mtime[1] + $mtime[0];

    OpenTable2();
    echo "<span class=\"content\">";
    echo "_CALC_DIST.": ". _OK. "<br>";
    echo "_CALC_TIME.": ".substr(($end_time2 - $start_time2),0,4).". _SECONDS. "<br>";
    echo "_DB_ACCESS_TIME.": ".substr(((($end_time1 - $start_time1)-($end_time2 - $start_time2)),0,4).".
    "_SECONDS;
    echo "</span>";
    CloseTable2();
    echo "<br><form action=\"modules.php?name=data&op=alg2&opx=data_calc_order_menu\"
method=\"post\">"
        . "<input type=\"submit\" value=\"\"._CALC_ORDER.\" \" />"
        . "</form>";
}

function data_calc_order_menu(){

```

```

CloseTable();
OpenTable();
    echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>\"._CALC_ORDER.\"</td></tr></table>";
OpenTable2();
    echo "<form class=\`content\` action=\`modules.php?name=data&amp;op=alg2&amp;opx=data_calc_order\`
method=\`post\`>"
        . "<label for=\`seed\`>\"._SEED.\"</label> "
        . "<input type=\`text\` name=\`seed\` id=\`seed\` value=\`\"";
    if ($seed=b_cidades::getSeed())
        echo $seed;
    echo "\`/><br>"
        . "<label for=\`seed\`>Quantidade de saltos:</label> "
        . "<input type=\`text\` name=\`seed\` id=\`seed\` value=\`1\`/><br>"
        . "<input type=\`submit\` value=\`\"._GO.\" \` />"
        . "<input type=\`submit\` value=\`Passo a Passo\` />"
        . "</form>";
CloseTable2();
}

function data_calc_order(){
    CloseTable();
    OpenTable();
        echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>\"._CALC_ORDER.\"</td></tr></table>";
    $stime1 = microtime();
    $cidades = b_cidades::getCidades();
    $dists = b_cidades::getDists($cidades);
    $n=count($cidades)-1;
    //----- ordenação pelo + prox a começar no 1º da tabela anterior
    if(!$seed=$_REQUEST[seed]){
        list($usec, $sec) = explode(' ', microtime());
        $seed=$sec + ($usec * 100000);
    }
    srand($seed);
    $cidade=new b_cidades;

    $x=array_rand($cidades);
    $cidades_o[] = $cidades[$x];
    unset($cidades[$x]);

```

```

$x=array_rand($cidades);
$cidades_o[] = $cidades[$x];
unset($cidades[$x]);
$x=array_rand($cidades);
$cidades_o[] = $cidades[$x];
unset($cidades[$x]);

$i=count($cidades_o)-1;
$time2 = microtime();

while ($i<$n){
    $x=array_rand($cidades);
    $cidade=$cidades[$x];
    unset($cidades[$x]);
    unset ($d);
    $j=-1;
    while($j++<$i-1)
        $d[]=$dists[$cidades_o[$j]->nome][$cidade->nome]+$dists[$cidades_o[$j+1]->nome][$cidade->nome]-
$dists[$cidades_o[$j]->nome][$cidades_o[$j+1]->nome];
        $d[]=$dists[$cidades_o[$j]->nome][$cidade->nome]+$dists[$cidades_o[0]->nome][$cidade->nome]-
$dists[$cidades_o[$j]->nome][$cidades_o[0]->nome];
        $value=min($d);
        $key = array_search($value, $d);
        array_splice($cidades_o, $key+1, 0, 0);
        $cidades_o[$key+1]=$cidade;
        $i=count($cidades_o)-1;
    }

$time2 = microtime();
$cidade->cleanTable3();
$cidade->insertdataTable3($cidades_o);
$time1 = microtime();
//-----calculo do tempo
$time = explode(" ",$time1);
$start_time1 = $mtime[1] + $mtime[0];
$time = explode(" ",$time1);
$end_time1 = $mtime[1] + $mtime[0];
$time = explode(" ",$time2);

```

```

$start_time2 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $time2);
$end_time2 = $mtime[1] + $mtime[0];
$cidade->updateSeed($seed, $time1=($end_time1 - $start_time1), $time2=($end_time2 - $start_time2));

    OpenTable2();
    echo "<span class=\"content\">";
    echo _SEED.: " . $seed. "<br>";
    echo _CALC_ORDER.: " . _OK. "<br>";
    echo _CALC_TIME.: " . substr($time2, 0, 4). " . _SECONDS. "<br>";
    echo _DB_ACCESS_TIME.: " . substr(($time1-$time2), 0, 4). " . _SECONDS;
    echo "</span>";
    CloseTable2();

    echo "<br><form action=\"modules.php?name=data&op=alg2&opx=data_result\" method=\"post\">"
        . "<input type=\"submit\" value=\"".$_SHOW_RESULT."\" />"
        . "</form>";
}

```

```

function data_result(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">".$_RESULT."</td></tr></table>";
    $scale=b_cidades::getMax();
    // $scale['max(x)'];
    // $scale['max(y)'];
    $pts=" ";
    $pts_o=" ";
    $cidades[1] = b_cidades::getCidades();
    $n=count($cidades[1]);
    $cidades[2] = b_cidades::getCidades_o();
    if (!$p=$_REQUEST['p'])
        $p=0;
    $o_x=$cidades[1][$p]->x;
    $o_y=$scale['max(y)']-$cidades[1][$p]->y;

    OpenTable2();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">".$_MENU."</td></tr></table>"
        . "<table width=\"100%\" class=\"content\"><tr><td>"

```

```

        . "<form class=\"content\" action=\"modules.php?name=data&amp;op=alg2&amp;opx=data_result\"
method=\"post\">"
        . "<label for=\"p\">._RED_DOT_ON.</label> "
        . "<select id=\"p\" name=\"p\">";
$_p=0;
foreach($cidades[1] as $cidade){
    echo "<option value=\"\$_p.\"\"";
    if ($p==$_p)
        echo " selected";
    echo ">".$cidade->nome."</option>";
    $_p++;
}

echo "</select><br>"
    . "<input type=\"checkbox\" name=\"table\" id=\"table\"\"";
if ($table=$_REQUEST['table'])
    echo " checked";
echo "><label for=\"table\">._SHOW_TABLE_RESULT.</label><br>"
    . "<input type=\"checkbox\" name=\"info\" id=\"info\"\"";
if ($info=$_REQUEST['info'])
    echo " checked";
echo "><label for=\"info\">._SHOW_INFO_RESULT.</label><br>"
    . "<input type=\"submit\" value=\"._GO.\" />"
    . "</form></td></tr></table>";

    CloseTable2();

if ($table){
    OpenTable2();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._CIDADES_TABLE.</td></tr></table>"
        . "<table width=\"100%\"border=1 class=\"content\">"
        . "<tr align=\"center\"><td width=\"33%\" colspan=\"3\">._CIDADES.</td><td width=\"33%\"></td><td width=\"33%\"
colspan=\"3\">._CIDADES_ORDER.</td></tr>";
    }
$i=0;
$dist_1=0;
$dist_2=0;
while ($i<$n-1){
    $pts.= $cidades[1][$i]->x. ", ". ($scale['max(y)']-$cidades[1][$i]->y). " ";
}

```

```

    $pts_o.=$cidades[2][$i]->x.",".($scale['max(y)']-$cidades[2][$i]->y)." ";

    $x=pow($cidades[1][$i]->x-$cidades[1][$i+1]->x,2);
    $y=pow($cidades[1][$i]->y-$cidades[1][$i+1]->y,2);
    $dist_1+=sqrt($x+$y);

    $x=pow($cidades[2][$i]->x-$cidades[2][$i+1]->x,2);
    $y=pow($cidades[2][$i]->y-$cidades[2][$i+1]->y,2);
    $dist_2+=sqrt($x+$y);
if ($table)
echo "<tr><td>". $cidades[1][$i]->nome."</td>"
    ."<td align='right'\>". $cidades[1][$i]->x."</td>"
    ."<td align='right'\>". $cidades[1][$i]->y."</td><td></td>"
    ."<td>". $cidades[2][$i]->nome."</td>"
    ."<td align='right'\>". $cidades[2][$i]->x."</td>"
    ."<td align='right'\>". $cidades[2][$i]->y."</td></tr>";
    $i++;
}

    $pts.=$cidades[1][$i]->x.",".($scale['max(y)']-$cidades[1][$i]->y)." ";
    $pts_o.=$cidades[2][$i]->x.",".($scale['max(y)']-$cidades[2][$i]->y)." ";
if ($table)
echo "<tr><td>". $cidades[1][$i]->nome."</td>"
    ."<td align='right'\>". $cidades[1][$i]->x."</td>"
    ."<td align='right'\>". $cidades[1][$i]->y."</td><td></td>"
    ."<td>". $cidades[2][$i]->nome."</td>"
    ."<td align='right'\>". $cidades[2][$i]->x."</td>"
    ."<td align='right'\>". $cidades[2][$i]->y."</td></tr>";

    $x=pow($cidades[1][$i]->x-$cidades[1][0]->x,2);
    $y=pow($cidades[1][$i]->y-$cidades[1][0]->y,2);
    $dist_1+=sqrt($x+$y);

    $x=pow($cidades[2][$i]->x-$cidades[2][0]->x,2);
    $y=pow($cidades[2][$i]->y-$cidades[2][0]->y,2);
    $dist_2+=sqrt($x+$y);
if ($table){
echo "<tr><td>"._DIST_TOTAL."</td>"
    ."<td colspan='2'\ align='right'\>". $dist_1."</td><td></td>"

```

```

        . "<td>._DIST_TOTAL." </td>"
        . "<td colspan='2' align='right'>.$dist_2." </td></tr>"
//      . "<tr><td colspan='3' align='right'>._SEED." </td><td></td>"
//      . "<td colspan='3'>.b_cidades::getSeed()." </td></tr>"
        . "</table>";
        CloseTable2();
    }
    OpenTable2();
    $arg1="scale_x=".$scale['max(x)']."&&scale_y=".$scale['max(y)']."&&pts=".$pts."&&o_x=".$o_x."&&o_y=".$o_y;
    $arg2="scale_x=".$scale['max(x)']."&&scale_y=".$scale['max(y)']."&&pts=".$pts_o."&&o_x=".$o_x."&&o_y=".$o_y;
    echo "<table width='100%'><tr><td class='leftcontent'>._CIDADES_GRAF." </td></tr></table>"
        . "<table width='100%'border=1>"
        . "<tr align='center'><td width='33%' bgcolor='white'>"
        . "<a href='modules.php?name=data&&op=svg&&'. $arg1.'" target='_blank'>"
        . "<iframe src='modules/data/svg.php?'. $arg1.'" width='300' height='300'></iframe>"
        . "</a></td><td width='33%'></td><td width='33%' colspan='3' bgcolor='white'>"
        . "<a href='modules.php?name=data&&op=svg&&'. $arg2.'" target='_blank'>"
        . "<iframe src='modules/data/svg.php?'. $arg2.'" width='300' height='300'></iframe>"
        . "</a></td></tr></table>";
        CloseTable2();
    if ($info){
        $seedruntime=b_cidades::getSeedRunTime();
        OpenTable2();
        echo "<table width='100%'><tr><td class='leftcontent'>._INFO_RESULT." </td></tr></table>"
            . "<table width='100%' class='content'><tr><td>"
            . "_SEED.": ". $seedruntime['seed']. "<br>"
            . _CALC_TIME.": ". substr($seedruntime['time2'],0,4). " "._SECONDS." <br>"
            . _DB_ACCESS_TIME.": ". substr(($seedruntime['time1']-$seedruntime['time2']),0,4). " "._SECONDS
            . "</td></tr></table>";
        CloseTable2();
    }
}

switch($_REQUEST["opx"]) {

    case "export_data":
        export_data($_REQUEST[aux]);
        break;

```

```
case "import_data":
    index();
    alg2();
    import_data();
    foot();
    break;

case "data_type":
    index();
    alg2();
    data_type();
    foot();
    break;

case "data_insert":
    index();
    alg2();
    if ($_REQUEST[ncidades]>0&&$_REQUEST[ncidades]<=200)
        data_insert();
    else
        data_type();
    foot();
    break;

case "data_insert_result":
    index();
    alg2();
    // if ($_REQUEST[rand]>0&&$_REQUEST[rand]<=4000)
        data_insert_result();
    // else
    //     data_insert();
    foot();
    break;

case "data_calc_dist":
    index();
    alg2();
```



```
        data_calc_dist();
        foot();
        break;

    case "data_calc_order_menu":
        index();
        alg2();
        data_calc_order_menu();
        foot();
        break;

    case "data_calc_order":
        index();
        alg2();
        data_calc_order();
        foot();
        break;

    case "data_result":
        index();
        alg2();
        data_result();
        foot();
        break;

    default:
        index();
        alg2();
        foot();
        break;
}
/* -----old
function data_calc_order(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._CALC_ORDER.</td></tr></table>";
    $stime1 = microtime();
    $cidades = b_cidades::getCidades();
```

```

$dists = b_cidades::getDists($cidades);
$n=count($cidades)-1;
//----- ordenação pelo + prox a começar no 1º da tabela anterior

$cidades_o[] = $cidades[0];
$cidades_o[] = $cidades[1];
$cidades_o[] = $cidades[2];
$i=count($cidades_o)-1;
$cidades = array_slice($cidades, 3);
$time2 = microtime();

while ($i<$n){
    $cidade = array_shift($cidades);
    unset ($d);
    $j=-1;
    while($j++<$i-1)
        $d[]=$dists[$cidades_o[$j]->nome][$cidade->nome]+$dists[$cidades_o[$j+1]->nome][$cidade->nome]-
$dists[$cidades_o[$j]->nome][$cidades_o[$j+1]->nome];
        $d[]=$dists[$cidades_o[$j]->nome][$cidade->nome]+$dists[$cidades_o[0]->nome][$cidade->nome]-
$dists[$cidades_o[$j]->nome][$cidades_o[0]->nome];
        $value=min($d);
        $key = array_search($value, $d);
        array_splice($cidades_o, $key+1, 0, 0);
        $cidades_o[$key+1]=$cidade;
        $i=count($cidades_o)-1;
    }

$time2 = microtime();
$cidade->insertdataTable3($cidades_o);
$time1 = microtime();
//-----calcula do tempo
$time = explode(" ",$time1);
$start_time1 = $mtime[1] + $mtime[0];
$time = explode(" ",$time1);
$end_time1 = $mtime[1] + $mtime[0];
$time = explode(" ",$time2);
$start_time2 = $mtime[1] + $mtime[0];
$time = explode(" ",$time2);

```

```
$end_time2 = $mtime[1] + $mtime[0];

    OpenTable2();
    echo "<span class=\"content\">";
    echo _CALC_ORDER.: "._OK."<br>";
    echo _CALC_TIME.: ".substr(($end_time2 - $start_time2),0,4)."._SECONDS."<br>";
    echo _DB_ACCESS_TIME.: ".substr(((($end_time1 - $start_time1)-($end_time2 - $start_time2)),0,4)."
    ._SECONDS;
    echo "</span>";
    CloseTable2();
    echo "<br><form action=\"modules.php?name=data&op=alg2&opx=data_result\" method=\"post\">"
        . "<input type=\"submit\" value=\"".$_SHOW_RESULT." />"
        . "</form>";
}
*/

?>
```



```

<?
if (!ereg("modules.php", $_SERVER['PHP_SELF'])) {
    die (header("location: ../../access_denied.php"));
}

include_once("../classes/c_cidades.php");

function export_data($aux){
    if ($aux=="submit" && $file_s=$_REQUEST[file_s]){
        $rep=opendir('./xml');
        $flag=0;
        while ($file = readdir($rep)){
            if (is_file("./xml/".$file)){
                if ($file_s.".xml"==$file)
                    $flag=1;
            }
        }
    }
    closedir($rep);
    clearstatcache();
    if ($flag)
        header("location: modules.php?name=data&op=alg3&opx=export_data&aux=erro&file_s=$file_s");
    else{
        $cidades = c_cidades::getCidades();
        $seedruntime=c_cidades::getSeedRunTime();
        $content="<?xml version='1.0'?'>\r\n"
            . "<SIMULATION>\r\n"
            . " <CIDADES>\r\n";
        foreach($cidades as $cidade){
            $content.=" <CIDADE>\r\n"
                . " <NOME>". $cidade->nome. "</NOME>\r\n"
                . " <X>". $cidade->x. "</X>\r\n"
                . " <Y>". $cidade->y. "</Y>\r\n"
                . " </CIDADE>\r\n";
        }
        $content.=" </CIDADES>\r\n";
        $cidades_o=c_cidades::getCidades_o();
        $diff = array_diff($cidades, $cidades_o);
        if (!$diff){

```



```

        if ($aux=="saved")
            echo "<p><span class=\"content\">\"._FILE_SAVED.\"</span></p>";
    foot();
}
}

function data_type(){
    $rep=opendir('./xml');
    $flag=0;
    while ($file = readdir($rep)){
        if (is_file("./xml/".$file)){
            $file_=split("[.]", $file);
            if($file != '.' && $file != '..' && $file != '' && $file_[1]=="xml")
                $files[]=$file_[0];
        }
    }
    closedir($rep);
    clearstatcache();

    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._DATA.\"</td></tr></table>";
    if ($files){
        OpenTable2();
        echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._INSERT_DATA.\"</td></tr></table>";
    }

    echo "<form class=\"content\"
action=\"modules.php?name=data&op=alg3&opx=data_insert&rand=100\" method=\"post\">
    .<p><label for=\"ncidades\">\"._NCIDADES.\": </label> "
    .<input type=\"text\" name=\"ncidades\" id=\"ncidades\" value=\"\".$_REQUEST[ncidades].\"</input></p>\"
    .<input type=\"submit\" value=\"\"._GO.\" />\"
    .</form>";

    if ($_REQUEST[ncidades]<=0 || $_REQUEST[ncidades]>200)
        echo "<p><span class=\"tiny_red\">\"._VALUE_CIDADES_BETWEEN.\" 1 \"._AND.\" 200.</span></p>";
    if ($files){
        CloseTable2();
        OpenTable2();
        echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._IMPORT_DATA.\"</td></tr></table>";
    }
}

```

```

        echo "<form class=\"content\" action=\"modules.php?name=data&op=alg3&opx=import_data\"
method=\"post\">"
        . "<p><label for=\"ncidades\">\"._FILE.\": </label>"
        . "<select name=\"file_s\" id=\"file_s\">";
        foreach($files as $file_)
            echo "<option value=\"\".$file_.\".\".$file_.\"</option>";
        echo "</select></p>"
        . "<input type=\"submit\" value=\"\"._GO.\"\" />"
        . "</form>";
    CloseTable2();
}
}

function import_data(){
    require("../includes/xml_parser.php");
    $xml=getXML("../xml/\".$_REQUEST[file_s].xml");
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._INSERT.\"</td></tr></table>";
    if ($xml){
        $cidades=$xml[cidades];
        echo "<form action=\"modules.php?name=data&op=alg3&opx=data_insert_result\" method=\"post\">";
        $c=-1;
        foreach ($cidades as $cidade){$c++;
            echo "<p><label for=\"cidade[\".$c.\"][nome]\">\"._CIDADE.\": </label> "
                . "<input type=\"text\" name=\"cidade[\".$c.\"][nome]\" id=\"cidade[\".$c.\"][nome]\"
value=\"\".$cidade[nome].\"\" /> "
                . "_COORDINATES
                .": ( <input type=\"text\" name=\"cidade[\".$c.\"][x]\" id=\"cidade[\".$c.\"][x]\" size=\"3\"
value=\"\".$cidade[x].\"\" /> "
                . "; <input type=\"text\" name=\"cidade[\".$c.\"][y]\" id=\"cidade[\".$c.\"][y]\" size=\"3\" value=\"\".$cidade[y].\"\" />
        )</p>";
        }
        if ($xml[dados][0][seed])
            echo "<label for=\"seed\">\"._SEED.\": </label> "
                . "<input type=\"text\" name=\"seed\" id=\"seed\" value=\"\".$xml[dados][0][seed].\"\" /><br>";
    }
}

```



```

        echo "<input type='submit' value='\". _GO. \"' />"
        . "</form>";

    }else
        echo "<p><span class='tiny_red'>\". _IMPORT_FAILED. \"</span></p>";
    }

function data_insert(){
    CloseTable();
    OpenTable();
    echo "<table width='100%'><tr><td class='leftcontent'>\". _INSERT. \"</td></tr></table>"
        . "<table class='content' width='100%'><tr><td>";
    echo "<form action='modules.php?name=data&op=alg3&opx=data_insert' method='post'>"
        . "<label for='rand'>\". _GEN_RAND_COORDINATES. \"</label> "
        . "<input type='text' name='rand' id='rand' value='\". $_REQUEST[rand]. \"' />"
        . "<input type='hidden' name='ncidades' id='ncidades' value='\". $_REQUEST[ncidades]. \"' />"
        . "<input type='submit' value='\". _GO. \"' />"
        . "</form>";

    if ($_REQUEST[rand]<=0 || $_REQUEST[rand]>4000)
        echo "<p><span class='tinyred'>\". _VALUE_RAND_BETWEEN. \" 1 \"._AND.\" 4000.</span></p>";

    echo "<form action='modules.php?name=data&op=alg3&opx=data_insert_result' method='post'>";
    $c=-1;
    while ($c++<$_REQUEST[ncidades]-1){
        echo "<p><label for='cidade[\". $c. \"][nome]'>\". _CIDADE. \"</label> "
            . "<input type='text' name='cidade[\". $c. \"][nome]' id='cidade[\". $c. \"][nome]' value='\"";
        if ($_REQUEST[rand])
            echo "p". $c;
        echo "\"/> "
            . _COORDINATES
            . ": ( <input type='text' name='cidade[\". $c. \"][x]' id='cidade[\". $c. \"][x]' size='3' value='\"";
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\"/> "
            . "; <input type='text' name='cidade[\". $c. \"][y]' id='cidade[\". $c. \"][y]' size='3' value='\"";
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\"/> )</p>";
    }
}

```

```

        echo "<input type='submit' value='\". _GO.\"' />"
        . "</form></td></tr></table>";
    }

function data_insert_result(){
    CloseTable();
    OpenTable();
    echo "<table width='100%'><tr><td class='leftcontent'>\". _DATA_SAVE.\"</td></tr></table>";
    OpenTable2();
    echo "<span class='content'>";
    echo _DROPTABLES." "; if (c_cidades::dropTables()) echo _OK;
    echo "<br>\". _CREATETABLE1.\" "; if (c_cidades::createTable1()) echo _OK;
    echo "<br>\". _CREATETABLE2.\" "; if (c_cidades::createTable2($_REQUEST[idade])) echo _OK;
    echo "<br>\". _CREATETABLE3.\" "; if (c_cidades::createTable3()) echo _OK;
    echo "<br>\". _INSERTDATATABLE1.\" "; if (c_cidades::insertdataTable1($_REQUEST[idade])) echo _OK;
    echo "</span>";
    CloseTable2();
    echo "<br><form action='modules.php?name=data&amp;op=alg3&amp;opx=data_calc_dist' method='post'>"
        . "<input type='submit' value='\". _CALC_DIST.\"' />"
        . "</form>";
    }

function data_calc_dist(){
    CloseTable();
    OpenTable();
    echo "<table width='100%'><tr><td class='leftcontent'>\". _CALC_DIST.\"</td></tr></table>";
    $stime1 = microtime();
    $cidades = c_cidades::getCidades();
    $n=count($cidades)-1;
    //----- calculo das distancias
    $stime2 = microtime();
    $i=-1;
    foreach ($cidades as $cidade) {
        $i++;
        $j=$i;
        $cidades_dist[$cidades[$i]->nome][$cidades[$i]->nome]=0;
        while ($j++<$n){
            $x=pow($cidades[$i]->x-$cidades[$j]->x,2);

```

```

        $y=pow($cidades[$i]->y-$cidades[$j]->y,2);
        $cidades_dist[$cidades[$i]->nome][$cidades[$j]->nome]=sqrt($x+$y);
        $cidades_dist[$cidades[$j]->nome][$cidades[$i]->nome]=sqrt($x+$y);
//        $cidade->a_distancia_insert($cidades[$i]->nome,$cidades[$j]->nome,$x+$y);
    }
}

$etime2 = microtime();
$cidade->a_distancia_insert($cidades_dist);
$etime1 = microtime();
//-----calculo do tempo
$mtime = explode(" ",$etime1);
$start_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ",$etime1);
$end_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ",$etime2);
$start_time2 = $mtime[1] + $mtime[0];
$mtime = explode(" ",$etime2);
$end_time2 = $mtime[1] + $mtime[0];

    OpenTable2();
    echo "<span class=\"content\">";
    echo "_CALC_DIST.": "._OK."<br>";
    echo "_CALC_TIME.": ".substr(($end_time2 - $start_time2),0,4)."._SECONDS."<br>";
    echo "_DB_ACCESS_TIME.": ".substr(((($end_time1 - $start_time1)-($end_time2 - $start_time2)),0,4)."
    "_SECONDS;
    echo "</span>";
    CloseTable2();
    echo "<br><form action=\"modules.php?name=data&op=alg3&opx=data_calc_order_menu\"
method=\"post\">"
        . "<input type=\"submit\" value=\"\"._CALC_ORDER.\" \" />"
        . "</form>";
}

function data_calc_order_menu(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._CALC_ORDER.\"</td></tr></table>";
    OpenTable2();
}

```

```

        echo "<form class=\"content\" action=\"modules.php?name=data&op=alg3&opx=data_calc_order\"
method=\"post\">"

        . "<label for=\"trades\">\"._MAX_NUMBER_OF_TRADES.\"</label> "
        . "<input type=\"text\" name=\"trades\" id=\"trades\" value=\"500\"/><br>"
        . "<label for=\"seed\">\"._SEED.\"</label> "
        . "<input type=\"text\" name=\"seed\" id=\"seed\" value=\"\"";
    if ($seed=c_cidades::getSeed())
        echo $seed;
    echo "\"/><br>"
        . "<input type=\"submit\" value=\"\"._GO.\"\" />"
        . "</form>";
    CloseTable2();
}

function data_calc_order(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._CALC_ORDER.\"</td></tr></table>";
    $stime1 = microtime();
    $cidades_o = c_cidades::getCidades();
    $dists = c_cidades::getDists($cidades_o);
    $n=count($cidades_o);
    //----- ordenação pelo + prox a começar no 1º da tabela anterior
    if(!$seed=$_REQUEST[seed]){
        list($usec, $sec) = explode(' ', microtime());
        $seed=$sec + ($usec * 100000);
    }
    srand($seed);
    $cidade=new c_cidades;

    $ar=rand(0,$n-1);

    if ($ar!=0){
        $cidades_=array_splice($cidades_o,$ar);
        array_splice($cidades_o,0,0,$cidades_);
    }

    $t=0;

```

```

$flag=1;
$time2 = microtime();
while ($flag){
    $flag=0;
    $i=0;
    while($i<$n-2){
        $j=$i+2;
        while (($j<$n-1&&$i==0) || ($j<$n&&$i!=0)){
            if ($j+1==$n) $_j=0; else $_j=$j+1;
            $a=$dists[$cidades_o[$i]->nome][$cidades_o[$i+1]->nome]+$dists[$cidades_o[$j]->nome][$cidades_o[$_j]->nome];
            $b=$dists[$cidades_o[$i]->nome][$cidades_o[$j]->nome]+$dists[$cidades_o[$i+1]->nome][$cidades_o[$_j]->nome];
            if ($a>$b){
                $cidades_o_=$cidades_o;
                $tmp=array_reverse(array_splice($cidades_o_,$i+1,$j-$i));
                array_splice($cidades_o,$i+1,$j-$i,$tmp);
                $flag=1;
                break;
            }
            $j++;
        }
        if ($flag){
            if ($t<$_REQUEST[trades]) $t++; else $flag=0;
            break; }
        $i++;
    }
}

//-----

$time2 = microtime();
$cidade->cleanTable3();
$cidade->insertdataTable3($cidades_o);
$time1 = microtime();
//-----calculo do tempo
$mtime = explode(" ",$time1);
$start_time1 = $mtime[1] + $mtime[0];

```

```

$time = explode(" ", $time1);
$send_time1 = $mtime[1] + $mtime[0];
$time = explode(" ", $time2);
$start_time2 = $mtime[1] + $mtime[0];
$time = explode(" ", $time2);
$send_time2 = $mtime[1] + $mtime[0];
$cidade->updateSeed($seed, $time1=($send_time1 - $start_time1), $time2=($send_time2 - $start_time2), $t);

    OpenTable2();
    echo "<span class=\"content\">";
    echo _TRADES. ": ". $t. "<br>";
    echo _SEED. ": ". $seed. "<br>";
    echo _CALC_ORDER. ": ". _OK. "<br>";
    echo _CALC_TIME. ": ". substr($time2, 0, 4). " ". _SECONDS. "<br>";
    echo _DB_ACCESS_TIME. ": ". substr(($time1-$time2), 0, 4). " ". _SECONDS;
    echo "</span>";
    CloseTable2();

    echo "<br><form action=\"modules.php?name=data&op=alg3&opx=data_result\" method=\"post\">"
        . "<input type=\"submit\" value=\"". _SHOW_RESULT. "\" />"
        . "</form>";
}

function data_result(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">". _RESULT. "</td></tr></table>";
    $scale=c_cidades::getMax();
    // $scale['max(x)'];
    // $scale['max(y)'];
    $pts=" ";
    $pts_o=" ";
    $cidades[1] = c_cidades::getCidades();
    $n=count($cidades[1]);
    $cidades[2] = c_cidades::getCidades_o();
    if (!$p=$_REQUEST['p'])
        $p=0;
    $o_x=$cidades[1][$p]->x;
    $o_y=$scale['max(y)']-$cidades[1][$p]->y;

```

```

        OpenTable2();
    echo "<table width='100%'><tr><td class='leftcontent'>"._MENU."</td></tr></table>"
    ."<table width='100%' class='content'><tr><td>"
    ."<form class='content' action='modules.php?name=data&op=alg3&opx=data_result'"
method="post">"
    ."<label for='p'>"._RED_DOT_ON."</label> "
    ."<select id='p' name='p'>";
    $_p=0;
    foreach($cidades[1] as $cidade){
        echo "<option value='".$_p."'";
        if ($p==$_p)
            echo " selected";
        echo ">".$cidade->nome."</option>";
        $_p++;
    }

    echo "</select><br>"
    ."<input type='checkbox' name='table' id='table'";
    if ($table=$_REQUEST['table'])
        echo " checked";
    echo "><label for='table'>"._SHOW_TABLE_RESULT."</label><br>"
    ."<input type='checkbox' name='info' id='info'";
    if ($info=$_REQUEST['info'])
        echo " checked";
    echo "><label for='info'>"._SHOW_INFO_RESULT."</label><br>"
    ."<input type='submit' value='"._GO."' />"
    ."</form></td></tr></table>";
    CloseTable2();

    if ($table){
        OpenTable2();
        echo "<table width='100%'><tr><td class='leftcontent'>"._CIDADES_TABLE."</td></tr></table>"
        ."<table width='100%' border=1 class='content'>"
        ."<tr align='center'><td width='33%' colspan='3'>"._CIDADES."</td><td width='33%'></td><td width='33%'>"
        colspan="3">"._CIDADES_ORDER."</td></tr>";
    }
    $i=0;
    $dist_1=0;

```

```

$dist_2=0;
while ($i<$n-1){
    $pts.=$cidades[1][$i]->x.",".($scale['max(y)']-$cidades[1][$i]->y)." ";
    $pts_o.=$cidades[2][$i]->x.",".($scale['max(y)']-$cidades[2][$i]->y)." ";

    $x=pow($cidades[1][$i]->x-$cidades[1][$i+1]->x,2);
    $y=pow($cidades[1][$i]->y-$cidades[1][$i+1]->y,2);
    $dist_1+=sqrt($x+$y);

    $x=pow($cidades[2][$i]->x-$cidades[2][$i+1]->x,2);
    $y=pow($cidades[2][$i]->y-$cidades[2][$i+1]->y,2);
    $dist_2+=sqrt($x+$y);
if ($table)
echo "<tr><td>". $cidades[1][$i]->nome."</td>"
    . "<td align='right'>". $cidades[1][$i]->x."</td>"
    . "<td align='right'>". $cidades[1][$i]->y."</td><td></td>"
    . "<td>". $cidades[2][$i]->nome."</td>"
    . "<td align='right'>". $cidades[2][$i]->x."</td>"
    . "<td align='right'>". $cidades[2][$i]->y."</td></tr>";
    $i++;
}

    $pts.=$cidades[1][$i]->x.",".($scale['max(y)']-$cidades[1][$i]->y)." ";
    $pts_o.=$cidades[2][$i]->x.",".($scale['max(y)']-$cidades[2][$i]->y)." ";
if ($table)
echo "<tr><td>". $cidades[1][$i]->nome."</td>"
    . "<td align='right'>". $cidades[1][$i]->x."</td>"
    . "<td align='right'>". $cidades[1][$i]->y."</td><td></td>"
    . "<td>". $cidades[2][$i]->nome."</td>"
    . "<td align='right'>". $cidades[2][$i]->x."</td>"
    . "<td align='right'>". $cidades[2][$i]->y."</td></tr>";

    $x=pow($cidades[1][$i]->x-$cidades[1][0]->x,2);
    $y=pow($cidades[1][$i]->y-$cidades[1][0]->y,2);
    $dist_1+=sqrt($x+$y);

    $x=pow($cidades[2][$i]->x-$cidades[2][0]->x,2);
    $y=pow($cidades[2][$i]->y-$cidades[2][0]->y,2);
    $dist_2+=sqrt($x+$y);

```



```

if ($table){
    echo "<tr><td>"._DIST_TOTAL."</td>"
        ."<td colspan='2' align='right'>".$dist_1."</td><td></td>"
        ."<td>"._DIST_TOTAL."</td>"
        ."<td colspan='2' align='right'>".$dist_2."</td></tr>"
        ."</table>";
        CloseTable2();
}

    OpenTable2();
    $arg1="scale_x=".$scale['max(x)']."&&scale_y=".$scale['max(y)']."&&pts=".$pts."&&o_x=".$o_x."&&o_y=".$o_y;
    $arg2="scale_x=".$scale['max(x)']."&&scale_y=".$scale['max(y)']."&&pts=".$pts_o."&&o_x=".$o_x."&&o_y=".$o_y;
    echo "<table width='100%'><tr><td class='leftcontent'>"._CIDADES_GRAF."</td></tr></table>"
        ."<table width='100%'border=1>"
        ."<tr align='center'><td width='33%' bgcolor='white'>"
        ."<a href='modules.php?name=data&&op=svg&&".$arg1."&' target='_blank'>"
        ."<iframe src='modules/data/svg.php?'.$arg1."&' width='300' height='300'></iframe>"
        ."</a></td><td width='33%'></td><td width='33%' colspan='3' bgcolor='white'>"
        ."<a href='modules.php?name=data&&op=svg&&".$arg2."&' target='_blank'>"
        ."<iframe src='modules/data/svg.php?'.$arg2."&' width='300' height='300'></iframe>"
        ."</a></td></tr></table>";
        CloseTable2();
if ($info){
    $seedruntime=c_cidades::getSeedRunTime();
        OpenTable2();
    echo "<table width='100%'><tr><td class='leftcontent'>"._INFO_RESULT."</td></tr></table>"
        ."<table width='100%' class='content'><tr><td>"
        ."_SEED.": ".$seedruntime['seed']."<br>"
        ."_TRADES.": ".$seedruntime['trades']."<br>"
        ."_CALC_TIME.": ".substr($seedruntime['time2'],0,4)." "_SECONDS."<br>"
        ."_DB_ACCESS_TIME.": ".substr(($seedruntime['time1']-$seedruntime['time2']),0,4)." "_SECONDS
        ."</td></tr></table>";
        CloseTable2();
}
}

switch($_REQUEST["opx"]) {

    case "export_data":

```

```
    export_data($_REQUEST[aux]);  
    break;  
  
case "import_data":  
    index();  
    alg3();  
    import_data();  
    foot();  
    break;  
  
case "data_type":  
    index();  
    alg3();  
    data_type();  
    foot();  
    break;  
  
case "data_insert":  
    index();  
    alg3();  
    if ($_REQUEST[ncidades]>0&&$_REQUEST[ncidades]<=200)  
        data_insert();  
    else  
        data_type();  
    foot();  
    break;  
  
case "data_insert_result":  
    index();  
    alg3();  
    data_insert_result();  
    foot();  
    break;  
  
case "data_calc_dist":  
    index();  
    alg3();  
    data_calc_dist();
```

```
        foot();
        break;

    case "data_calc_order_menu":
        index();
        alg3();
        data_calc_order_menu();
        foot();
        break;

    case "data_calc_order":
        index();
        alg3();
        data_calc_order();
        foot();
        break;

    case "data_result":
        index();
        alg3();
        data_result();
        foot();
        break;

    default:
        index();
        alg3();
        foot();
        break;
}

?>
```



```

<?
if (!ereg("modules.php", $_SERVER['PHP_SELF'])) {
    die (header("location: ../../access_denied.php"));
}

include_once("../classes/ci_cidades.php");

function export_data($aux){
    if ($aux=="submit" && $file_s=$_REQUEST[file_s]){
        $rep=opendir('./xml');
        $flag=0;
        while ($file = readdir($rep)){
            if (is_file("./xml/".$file)){
                if ($file_s.".xml"==$file)
                    $flag=1;
            }
        }
    }
    closedir($rep);
    clearstatcache();
    if ($flag)
        header("location: modules.php?name=data&op=alg3i&opx=export_data&aux=erro&file_s=$file_s");
    else{
        $cidades = ci_cidades::getCidades();
        $seedruntime=ci_cidades::getSeedRunTime();
        $content="<?xml version='1.0'>\r\n"
            . "<SIMULATION>\r\n"
            . " <CIDADES>\r\n";
        foreach($cidades as $cidade){
            $content.=" <CIDADE>\r\n"
                . " <NOME>". $cidade->nome."</NOME>\r\n"
                . " <X>". $cidade->x."</X>\r\n"
                . " <Y>". $cidade->y."</Y>\r\n"
                . " </CIDADE>\r\n";
        }
        $content.=" </CIDADES>\r\n";
        $cidades_o=ci_cidades::getCidades_o();
        $diff = array_diff($cidades, $cidades_o);
        if (!$diff){

```

```

$content.=" <CIDADES_O>\r\n";
foreach($cidades_o as $cidade){
    $content.=" <CIDADE>\r\n"
        ." <NOME>". $cidade->nome." </NOME>\r\n"
        ." <X>". $cidade->x." </X>\r\n"
        ." <Y>". $cidade->y." </Y>\r\n"
        ." </CIDADE>\r\n";
}
$content.=" </CIDADES_O>\r\n";
}
$content.=" <DADOS>\r\n"
    ." <DATA>\r\n"
    ." <TYPE>_ALG3I</TYPE>\r\n"
    ." <TRADES>". $seedruntime['trades']." </TRADES>\r\n"
    ." <PERCENT>". $seedruntime['percent']." </PERCENT>\r\n"
    ." <TIME1>". substr($seedruntime['time2'],0,4)." </TIME1>\r\n"
    ." <TIME2>". substr(($seedruntime['time1']-$seedruntime['time2']),0,4)." </TIME2>\r\n"
    ." <SEED>". $seedruntime['seed']." </SEED>\r\n"
    ." </DATA>\r\n"
    ." </DADOS>\r\n"
    ." </SIMULATION>\r\n";
$fd = fopen("./xml/".$file_s.".xml", "w");
fwrite($fd, $content);
fclose($fd);
header("location: modules.php?name=data&op=alg3i&opx=export_data&aux=saved&file_s=$file_s");
}
}else{
    index();
    alg3i();
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._EXPORT_DATA.</td></tr></table>";
    echo "<form action=\"modules.php?name=data&amp;op=alg3i&amp;opx=export_data&amp;aux=submit\"
method=\"post\">"
        ."<p><label for=\"file_s\">._FILE_NAME.</label> "
        ."<input type=\"text\" name=\"file_s\" id=\"file_s\" value=\"\". $_REQUEST[file_s].\"/\></p>"
        ."<input type=\"submit\" value=\"._GO.\" />";
    if ($aux=="erro")

```

```

        echo "<p><span class=\"tiny_red\">._FILE_EXIST.</span></p>";
    if ($aux=="saved")
        echo "<p><span class=\"content\">._FILE_SAVED.</span></p>";
    foot();
}
}

function data_type(){
    $rep=opendir('./xml');
    $flag=0;
    while ($file = readdir($rep)){
        if (is_file("./xml/".$file)){
            $file_=split(".", $file);
            if($file != '..' && $file != '.' && $file != '' && $file_[1]=="xml")
                $files[]=$file_[0];
        }
    }
    closedir($rep);
    clearstatcache();

    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._DATA.</td></tr></table>";
    if ($files){
        OpenTable2();
        echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._INSERT_DATA.</td></tr></table>";
    }
    echo "<form class=\"content\"
action=\"modules.php?name=data&op=alg3i&opx=data_insert&rand=100\" method=\"post\">
    .<p><label for=\"ncidades\">._NCIDADES.: </label> "
    .<input type=\"text\" name=\"ncidades\" id=\"ncidades\" value=\"".$_REQUEST[ncidades]."\"/></p>"
    .<input type=\"submit\" value=\"._GO.\" />"
    .</form>";
    if ($_REQUEST[ncidades]<=0 || $_REQUEST[ncidades]>200)
        echo "<p><span class=\"tiny_red\">._VALUE_CIDADES_BETWEEN." 1 "._AND." 200.</span></p>";
    if ($files){
        CloseTable2();
        OpenTable2();
    }
}

```

```

echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>._IMPORT_DATA."</td></tr></table>";
echo "<form class=\`content\`" action=\`modules.php?name=data&op=alg3i&opx=import_data\`"
method=\`post\`>"
    . "<p><label for=\`ncidades\`>._FILE.": </label>"
    . "<select name=\`file_s\`" id=\`file_s\`>";
    foreach($files as $file_)
        echo "<option value=\`".$file_."`. $file_."</option>";
    echo "</select></p>"
    . "<input type=\`submit\`" value=\`._GO.\`" />"
    . "</form>";
CloseTable2();
}
}

function import_data(){
    require("../includes/xml_parser.php");
    $xml=getXML("../xml/".$_REQUEST[file_s].".xml");
    CloseTable();
    OpenTable();
    echo "<table width=\`100%\`><tr><td class=\`leftcontent\`>._INSERT."</td></tr></table>";
    if ($xml){
        $cidades=$xml[cidades];
        echo "<form action=\`modules.php?name=data&op=alg3i&opx=data_insert_result\`" method=\`post\`>";
        $c=-1;
        foreach ($cidades as $cidade){$c++;
            echo "<p><label for=\`cidade[".$c."][nome]\`>._CIDADE.":</label> "
                . "<input type=\`text\`" name=\`cidade[".$c."][nome]\`" id=\`cidade[".$c."][nome]\`"
value=\`".$cidade[nome]."\`"/> "
                . "_COORDINATES
                .": ( <input type=\`text\`" name=\`cidade[".$c."][x]\`" id=\`cidade[".$c."][x]\`" size=\`3\`"
value=\`".$cidade[x]."\`"/> "
                . "; <input type=\`text\`" name=\`cidade[".$c."][y]\`" id=\`cidade[".$c."][y]\`" size=\`3\`" value=\`".$cidade[y]."\`"/>
            )</p>";
        }
        if ($xml[dados][0][seed])
            echo "<label for=\`seed\`>._SEED.":</label> "
                . "<input type=\`text\`" name=\`seed\`" id=\`seed\`" value=\`".$xml[dados][0][seed]."\`"/><br>";
    }
}

```



```

        echo "<input type=\"submit\" value=\"._GO.\" />"
        . "</form>";

    }else
        echo "<p><span class=\"tiny_red\">._IMPORT_FAILED.</span></p>";
}

function data_insert(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">._INSERT.</td></tr></table>"
        . "<table class=\"content\" width=\"100%\"><tr><td>";
    echo "<form action=\"modules.php?name=data&amp;op=alg3i&amp;opx=data_insert\" method=\"post\">"
        . "<label for=\"rand\">._GEN_RAND_COORDINATES.</label> "
        . "<input type=\"text\" name=\"rand\" id=\"rand\" value=\"".$_REQUEST[rand]."\" />"
        . "<input type=\"hidden\" name=\"ncidades\" id=\"ncidades\" value=\"".$_REQUEST[ncidades]."\" />"
        . "<input type=\"submit\" value=\"._GO.\" />"
        . "</form>";

    if ($_REQUEST[rand]<=0 || $_REQUEST[rand]>4000)
        echo "<p><span class=\"tinyred\">._VALUE_RAND_BETWEEN." 1 ". _AND." 4000.</span></p>";
    echo "<form action=\"modules.php?name=data&amp;op=alg3i&amp;opx=data_insert_result\" method=\"post\">";
    $c=-1;
    while ($c++<$_REQUEST[ncidades]-1){
        echo "<p><label for=\"cidade[".$c."][nome]\">._CIDADE.</label> "
            . "<input type=\"text\" name=\"cidade[".$c."][nome]\" id=\"cidade[".$c."][nome]\" value=\"";
        if ($_REQUEST[rand])
            echo "p".$c;
        echo "\" /> "
            . "_COORDINATES
            .": ( <input type=\"text\" name=\"cidade[".$c."][x]\" id=\"cidade[".$c."][x]\" size=\"3\" value=\"";
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\" /> "
            . "; <input type=\"text\" name=\"cidade[".$c."][y]\" id=\"cidade[".$c."][y]\" size=\"3\" value=\"";
        if ($_REQUEST[rand])
            echo mt_rand(0,$_REQUEST[rand]);
        echo "\" /> )</p>";
    }
}

```

```

    }
    echo "<input type='submit' value='\"._GO.\"' />"
        . "</form></td></tr></table>";
}

function data_insert_result(){
    CloseTable();
    OpenTable();
    echo "<table width='100%'><tr><td class='leftcontent'>\"._DATA_SAVE.\"</td></tr></table>";
    OpenTable2();
    echo "<span class='content'>";
    echo "_DROPTABLES.": "; if (ci_cidades::dropTables()) echo _OK;
    echo "<br>\"._CREATETABLE1.": "; if (ci_cidades::createTable1()) echo _OK;
    echo "<br>\"._CREATETABLE2.": "; if (ci_cidades::createTable2($_REQUEST[cidade])) echo _OK;
    echo "<br>\"._CREATETABLE3.": "; if (ci_cidades::createTable3()) echo _OK;
    echo "<br>\"._INSERTDATATABLE1.": "; if (ci_cidades::insertdataTable1($_REQUEST[cidade])) echo _OK;
    echo "</span>";
    CloseTable2();
    echo "<br><form action='modules.php?name=data&amp;op=alg3i&amp;opx=data_calc_dist' method='post'>"
        . "<input type='submit' value='\"._CALC_DIST.\"' />"
        . "</form>";
}

function data_calc_dist(){
    CloseTable();
    OpenTable();
    echo "<table width='100%'><tr><td class='leftcontent'>\"._CALC_DIST.\"</td></tr></table>";
    $stime1 = microtime();
    $cidades = ci_cidades::getCidades();
    $n=count($cidades)-1;
    //----- calculo das distancias
    $stime2 = microtime();
    $i=-1;
    foreach ($cidades as $cidade) {
        $i++;
        $j=$i;
        $cidades_dist[$cidades[$i]->nome][$cidades[$j]->nome]=0;
        while ($j++<$n){

```

```

        $x=pow($cidades[$i]->x-$cidades[$j]->x,2);
        $y=pow($cidades[$i]->y-$cidades[$j]->y,2);
        $cidades_dist[$cidades[$i]->nome][$cidades[$j]->nome]=sqrt($x+$y);
        $cidades_dist[$cidades[$j]->nome][$cidades[$i]->nome]=sqrt($x+$y);
//        $cidade->a_distancia_insert($cidades[$i]->nome,$cidades[$j]->nome,$x+$y);
    }
}

$etime2 = microtime();
$cidade->a_distancia_insert($cidades_dist);
$etime1 = microtime();
//-----calculo do tempo
$mtime = explode(" ", $etime1);
$start_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $etime1);
$end_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $etime2);
$start_time2 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $etime2);
$end_time2 = $mtime[1] + $mtime[0];

    OpenTable2();
    echo "<span class=\"content\">";
    echo _CALC_DIST.".": ". _OK. "<br>";
    echo _CALC_TIME.".": ".substr(($end_time2 - $start_time2),0,4).". _SECONDS. "<br>";
    echo _DB_ACCESS_TIME.".": ".substr(((($end_time1 - $start_time1)-($end_time2 - $start_time2)),0,4).".
    _SECONDS;
    echo "</span>";
    CloseTable2();
    echo "<br><form action=\"modules.php?name=data&op=alg3i&opx=data_calc_order_menu\"
method=\"post\">"
        . "<input type=\"submit\" value=\"\". _CALC_ORDER. \"\" />"
        . "</form>";
}

function data_calc_order_menu(){
    CloseTable();
    if(!$percent=$_REQUEST[percent])
        $percent=100;

```

```

    $err=$_REQUEST['err'];
    OpenTable();
        echo "<table width=\\"100%\"><tr><td class=\\"leftcontent\">\"._CALC_ORDER.\"</td></tr></table>";
    OpenTable2();
        echo "<form class=\\"content\" action=\\"modules.php?name=data&op=alg3i&opx=data_calc_order_pre\"
method=\\"post\">"
        . "<label for=\\"seed\">\"._SEED.\"</label> "
        . "<input type=\\"text\" name=\\"seed\" id=\\"seed\" value=\\"".$_REQUEST['seed'].\"\"/><br>"
        . "<label for=\\"percent\">\"._TRADE_EVALUATION_PERCENT.\"</label> "
        . "<input type=\\"text\" name=\\"percent\" id=\\"percent\" value=\\"".$_percent.\"\"/ size=\\"1\">%<br>";
    if($err==2)
        echo "<span class=\\"tiny_red\">\"._PERCENT_BETWEEN.\" 1 \"._AND.\" 100</span><br>";
    if($err==1)
        echo "<input type=\\"checkbox\" name=\\"kill\" id=\\"kill\">"
        . "<label for=\\"kill\">\"._KILL_PREVIOUS_SIMULATION.\"</label><br>";
        echo "<input type=\\"submit\" value=\\"\"._GO.\"\" />"
        . "</form>";
    if($err==1){
        echo "<span class=\\"tiny_red\">\"._ALREADY_SIMULATING.\"</span><br>";
    /*
        echo "<form action=\\"javascript:window.close();\" method=\\"post\">"
        . "<input type=\\"submit\" value=\\"\"._CLOSE_WINDOW.\"\" />"
        . "</form>";
    */
    }
    CloseTable2();
}

function data_calc_order_pre($simul){
    if ($_REQUEST[simul]){
        index();
        alg3i();
        CloseTable();
        OpenTable();
        echo "<table width=\\"100%\"><tr><td class=\\"leftcontent\">\"._CALC_ORDER.\"</td></tr></table>"
        . "<br><span class=\\"content\">\"._SIMULATION_RUNNING_IN_BACKGROUND.\"</span><br>";
        foot();
    }else{
        $cidade=new ci_cidades;
    }
}

```

```

    $seedruntime=ci_cidades::getSeedRunTime();
if ($seedruntime['run']&&!$_REQUEST['kill'])
    header("location: modules.php?name=data&op=alg3i&opx=data_calc_order_menu&err=1");
else{
    $cidade->updateRun('run');
    $percent=$_REQUEST['percent'];
    if ($percent<=0 || $percent>100)
        header("location: modules.php?name=data&op=alg3i&opx=data_calc_order_menu&err=2");
    else{
        $stime1 = microtime();
//    ci_cidades::dropTables();
//    ci_cidades::copyTable1();
        $cidades_o = ci_cidades::getCidades();
//    ci_cidades::copyTable2($cidades_o);
        ci_cidades::cleanTable3();

        $n=count($cidades_o);
//-----
        if(!$seed=$_REQUEST[seed]){
            list($usec, $sec) = explode(' ', microtime());
            $seed=$sec + ($usec * 100000);
        }
        srand($seed);
        $cidade=new ci_cidades;
        $cidade->updateSeed($seed,$percent);
        $ar=rand(0,$n-1);

        if ($ar!=0){
            $cidades_=array_splice($cidades_o,$ar);
            array_splice($cidades_o,0,0,$cidades_);
        }

        $cidade->insertdataTable3($cidades_o);
        $setime1 = microtime();

        $mtime = explode(" ",$stime1);
        $start_time1 = $mtime[1] + $mtime[0];
        $mtime = explode(" ",$setime1);

```

```

$end_time1 = $mtime[1] + $mtime[0];
$cidade->updateTime($end_time1-$start_time1,0,0);

header("location: modules.php?name=data&op=alg3i&opx=data_calc_order_pre&simul=1");
/*      echo "<script language='JavaScript' type='text/JavaScript'>
<!--
setTimeout(\"MM_goToURL('parent','modules.php?name=data&op=alg3i&opx=data_calc_order')\",10);
function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+'\".location='\"+args[i+1]+'\"");
}
//-->
</script>";
echo "runnig...";
*/      }
}
}
}

function data_calc_order(){

    $seedruntime=ci_cidades::getSeedRunTime();
if ($seedruntime['run']){
    $stime1 = microtime();
    if (!$_SESSION['alg3i']['cidades']){
        $cidades_o = ci_cidades::getCidades_0();
        $dists = $_SESSION['alg3i']['dists'] = ci_cidades::getDists_($cidades_o);
        $_SESSION['alg3i']['time1']=$seedruntime['time1'];
        $_SESSION['alg3i']['time2']=$seedruntime['time2'];
    }else{
        $cidades_o=$_SESSION['alg3i']['cidades'];
        $dists = $_SESSION['alg3i']['dists'];
    }

    $n=count($cidades_o);
    //---- ordenação pelo + prox a começar no 1º da tabela anterior
    srand($seedruntime['seed']);
    $cidade=new ci_cidades;

```

```

$t=$_REQUEST['t'];
$time2 = microtime();

$flag=0;
$i=0;
while($i<$n-2){
    $j=$i+2;
    while (($j<$n-1&&$i==0) || ($j<$n&&$i!=0)){
        if ($j+1==$n) $_j=0; else $_j=$j+1;
        $a=$dists[$cidades_o[$i][nome]][$cidades_o[$i+1][nome]]+$dists[$cidades_o[$j][nome]][$cidades_o[$_j][nome]];
        $b=$dists[$cidades_o[$i][nome]][$cidades_o[$_j][nome]]+$dists[$cidades_o[$i+1][nome]][$cidades_o[$_j][nome]];
        if ($a>$b){
            $cidades_o_=$cidades_o;
            $tmp=array_reverse(array_splice($cidades_o_,$i+1,$j-$i));
            array_splice($cidades_o,$i+1,$j-$i,$tmp);
            $flag=1;
            break;
        }
        $j++;
    }
    if ($flag){
        $t++;
        break;
    }
    $i++;
}

//-----

$time2 = microtime();

if (($t && is_int($t/10)) || !$flag){
    $cidade->cleanTable3();
    $cidade->insertdataTable3_($cidades_o);
}

$time1 = microtime();

```

```

//-----calculo do tempo
$mtime = explode(" ", $stime1);
$start_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $etime1);
$end_time1 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $stime2);
$start_time2 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $etime2);
$end_time2 = $mtime[1] + $mtime[0];

$_SESSION['alg3i']['cidades']=$cidades_o;
$_SESSION['alg3i']['time1']+=$end_time1-$start_time1;
$_SESSION['alg3i']['time2']+=$end_time2-$start_time2;

if (is_int($t/10) || !$flag)
    $cidade->updateTime($_SESSION['alg3i']['time1'],$_SESSION['alg3i']['time2'],$t);

if ($flag)
//    header("location: modules.php?name=data&op=alg3i&opx=data_calc_order&t=$t");
    echo "<script language='JavaScript' type='text/JavaScript'>
<!--
setTimeout(\"MM_goToURL('parent','modules.php?name=data&op=alg3i&opx=data_calc_order&t=$t')\",2000);
function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+'\".location='\"+args[i+1]+'\"");
}
//-->
</script>running... \"$t;
else{
    $cidade->updateRun(null);
    unset($_SESSION['alg3i']);
//    header("location: modules.php?name=data&op=alg3i&opx=data_calc_order_done");
    echo "<script language='JavaScript' type='text/JavaScript'>
<!--
setTimeout(\"MM_goToURL('parent','modules.php?name=data&op=alg3i&opx=data_calc_order')\",2000);
function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+'\".location='\"+args[i+1]+'\"");
}
-->
</script>";
}

```



```

    }
    //-->
</script>;
}
}else
//    header("location: modules.php?name=data&op=alg3i&opx=data_calc_order&t=$t");
    echo "<script language=\"JavaScript\" type=\"text/JavaScript\">
<!--
setTimeout(\"MM_goToURL('parent','modules.php?name=data&op=alg3i&opx=data_calc_order')\",10000);
function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+\".location=\"+args[i+1]+\"'\");
}
//-->
</script>waiting...";
}

function data_calc_order_done(){
    $seedruntime=ci_cidades::getSeedRunTime();
    head_simple();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">\"._CALC_ORDER.\"</td></tr></table>";
    OpenTable2();
    echo "<span class=\"content\">";
    echo _SEED.": ". $seedruntime['seed']. "<br>";
    echo _CALC_ORDER.": ". _OK. "<br>";
    echo _TRADES.": ". $seedruntime['trades']. "<br>";
    echo _PERCENT.": ". $seedruntime['percent']. "<br>";
    echo _CALC_TIME.": ". substr($seedruntime['time2'],0,4). " ". _SECONDS. "<br>";
    echo _DB_ACCESS_TIME.": ". substr(($seedruntime['time1']-$seedruntime['time2']),0,4). " ". _SECONDS. "<br>";
    echo "</span>";
    echo "<form action=\"javascript:window.close();\" method=\"post\">\"
        . "<input type=\"submit\" value=\"\"._CLOSE_WINDOW.\" />\"
        . "</form>";
    CloseTable2();
    CloseTable();
}

```

```

function data_result(){
    CloseTable();
    OpenTable();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">".$_RESULT."</td></tr></table>";
$scale=ci_cidades::getMax();
//$scale['max(x)'];
//$scale['max(y)'];
$pts=" ";
$pts_o=" ";
$ciudades[1] = ci_cidades::getCidades();
$n=count($ciudades[1]);
$ciudades[2] = ci_cidades::getCidades_o();
if (!$p=$_REQUEST['p'])
    $p=0;
$o_x=$ciudades[1][$p]->x;
$o_y=$scale['max(y)']-$ciudades[1][$p]->y;

    OpenTable2();
    echo "<table width=\"100%\"><tr><td class=\"leftcontent\">".$_MENU."</td></tr></table>"
    ."<table width=\"100%\" class=\"content\"><tr><td>"
    ."<form class=\"content\" action=\"modules.php?name=data&op=alg3i&opx=data_result\"
method=\"post\">"
    ."<label for=\"p\">".$_RED_DOT_ON."</label> "
    ."<select id=\"p\" name=\"p\">";
$_p=0;
foreach($ciudades[1] as $cidade){
    echo "<option value=\"".$_p.""";
    if ($p==$_p)
        echo " selected";
    echo ">".$cidade->nome."</option>";
    $_p++;
}

echo "</select><br>"
    ."<input type=\"checkbox\" name=\"table\" id=\"table\"";
if ($table=$_REQUEST['table'])
    echo " checked";

```

```

echo "><label for='table'>".$_SHOW_TABLE_RESULT."</label><br>"
    . "<input type='checkbox' name='info' id='info' />";
if ($info=$_REQUEST['info'])
    echo " checked";
echo "><label for='info'>".$_SHOW_INFO_RESULT."</label><br>"
    . "<input type='submit' value='".$_GO.'" />"
    . "</form></td></tr></table>";

    CloseTable2();

if ($table){
    OpenTable2();
    echo "<table width='100%'><tr><td class='leftcontent'>".$_CIDADES_TABLE."</td></tr></table>"
        . "<table width='100%' border=1 class='content'>"
        . "<tr align='center'><td width='33%' colspan='3'>".$_CIDADES."</td><td width='33%'>"
        . "<tr align='center'><td colspan='3'>".$_CIDADES_ORDER."</td></tr>";
    }
    $i=0;
    $dist_1=0;
    $dist_2=0;
    while ($i<$n-1){
        $pts_1=$cidades[1][$i]->x.", " . ($scale['max(y)']-$cidades[1][$i]->y). " ";
        $pts_o_1=$cidades[2][$i]->x.", " . ($scale['max(y)']-$cidades[2][$i]->y). " ";

        $x=pow($cidades[1][$i]->x-$cidades[1][$i+1]->x,2);
        $y=pow($cidades[1][$i]->y-$cidades[1][$i+1]->y,2);
        $dist_1+=sqrt($x+$y);

        $x=pow($cidades[2][$i]->x-$cidades[2][$i+1]->x,2);
        $y=pow($cidades[2][$i]->y-$cidades[2][$i+1]->y,2);
        $dist_2+=sqrt($x+$y);
    }
    if ($table)
        echo "<tr><td>".$_CIDADES[1][$i]->nome."</td>"
            . "<td align='right'>".$_CIDADES[1][$i]->x."</td>"
            . "<td align='right'>".$_CIDADES[1][$i]->y."</td><td></td>"
            . "<td>".$_CIDADES[2][$i]->nome."</td>"
            . "<td align='right'>".$_CIDADES[2][$i]->x."</td>"
            . "<td align='right'>".$_CIDADES[2][$i]->y."</td></tr>";
    $i++;
}

```

```

}

    $pts.= $cidades[1][$i]->x.", " . ($scale['max(y)']-$cidades[1][$i]->y). " ";
    $pts_o.= $cidades[2][$i]->x.", " . ($scale['max(y)']-$cidades[2][$i]->y). " ";

if ($table)
echo "<tr><td>". $cidades[1][$i]->nome. "</td>"
    . "<td align='right'\>". $cidades[1][$i]->x. "</td>"
    . "<td align='right'\>". $cidades[1][$i]->y. "</td><td></td>"
    . "<td>". $cidades[2][$i]->nome. "</td>"
    . "<td align='right'\>". $cidades[2][$i]->x. "</td>"
    . "<td align='right'\>". $cidades[2][$i]->y. "</td></tr>";

    $x=pow($cidades[1][$i]->x-$cidades[1][0]->x,2);
    $y=pow($cidades[1][$i]->y-$cidades[1][0]->y,2);
    $dist_1+=sqrt($x+$y);

    $x=pow($cidades[2][$i]->x-$cidades[2][0]->x,2);
    $y=pow($cidades[2][$i]->y-$cidades[2][0]->y,2);
    $dist_2+=sqrt($x+$y);

if ($table){
echo "<tr><td>". _DIST_TOTAL. "</td>"
    . "<td colspan='2' align='right'\>". $dist_1. "</td><td></td>"
    . "<td>". _DIST_TOTAL. "</td>"
    . "<td colspan='2' align='right'\>". $dist_2. "</td></tr>"
    . "</table>";

    CloseTable2();
}

    OpenTable2();

$arg1="scale_x=". $scale['max(x)']. "&&scale_y=". $scale['max(y)']. "&&pts=". $pts. "&&o_x=". $o_x. "&&o_y=". $o_y;
$arg2="scale_x=". $scale['max(x)']. "&&scale_y=". $scale['max(y)']. "&&pts_o=". $pts_o. "&&o_x=". $o_x. "&&o_y=". $o_y;
echo "<table width='100%'\><tr><td class='leftcontent'\>". _CIDADES_GRAF. "</td></tr></table>"
    . "<table width='100%' border=1>"
    . "<tr align='center'\><td width='33%' bgcolor='white'\>"
    . "<a href='modules.php?name=data&&op=svg&&'. $arg1. "\" target='_blank'\>"
    . "<iframe src='modules/data/svg.php?'. $arg1. "\" width='300' height='300'\></iframe>"
    . "</a></td><td width='33%'\></td><td width='33%' colspan='3' bgcolor='white'\>"
    . "<a href='modules.php?name=data&&op=svg&&'. $arg2. "\" target='_blank'\>"
    . "<iframe src='modules/data/svg.php?'. $arg2. "\" width='300' height='300'\></iframe>"
    . "</a></td></tr></table>";

```

```

        CloseTable2();
    if ($info){
        $seedruntime=ci_cidades::getSeedRunTime();

        OpenTable2();
        echo "<table width='100%'><tr><td class='leftcontent'>._INFO_RESULT.</td></tr></table>"
            . "<table width='100%' class='content'><tr><td>"
            . "_SEED.: " . $seedruntime['seed'] . "<br>"
            . "_STATE.: ";
        if ($seedruntime['run'])
            echo _IN_PROGRESS;
        else
            echo _DONE;
        echo "<br>"
            . "_TRADES.: " . $seedruntime['trades'] . "<br>"
            . "_PERCENT.: " . $seedruntime['percent'] . "<br>"
            . "_CALC_TIME.: " . substr($seedruntime['time2'],0,4) . " " . "_SECONDS." . "<br>"
            . "_DB_ACCESS_TIME.: " . substr(($seedruntime['time1']-$seedruntime['time2']),0,4) . " " . "_SECONDS"
            . "</td></tr></table>";
        CloseTable2();
    }
}

switch($_REQUEST["opx"]) {

    case "export_data":
        export_data($_REQUEST[aux]);
        break;

    case "import_data":
        index();
        alg3i();
        import_data();
        foot();
        break;

    case "data_type":
        index();
        alg3i();

```

```
data_type();
foot();
break;

case "data_insert":
    index();
    alg3i();
    if ($_REQUEST[ncidades]>0&&$_REQUEST[ncidades]<=200)
        data_insert();
    else
        data_type();
    foot();
    break;

case "data_insert_result":
    index();
    alg3i();
    data_insert_result();
    foot();
    break;

case "data_calc_dist":
    index();
    alg3i();
    data_calc_dist();
    foot();
    break;

case "data_calc_order_menu":
    index();
    alg3i();
    data_calc_order_menu();
    foot();
    break;

case "data_calc_order_pre":
    data_calc_order_pre($_REQUEST[simul]);
    break;
```

```
case "data_calc_order":
    data_calc_order();
    break;

case "data_calc_order_done":
    data_calc_order_done();
    break;

case "data_result":
    index();
    alg3i();
    data_result();
    foot();
    break;

default:
    index();
    alg3i();
    foot();
    break;
}

/*
function data_calc_order(){
    $seedruntime=ci_cidades::getSeedRunTime();
    if ($seedruntime[run]){
        $stime1 = microtime();
        $cidades_o = ci_cidades::getCidades_o();
        $dists = ci_cidades::getDists($cidades_o);

        $n=count($cidades_o);
        //---- ordenação pelo + prox a começar no 1º da tabela anterior
        srand($seedruntime['seed']);
        $cidade=new ci_cidades;

        $stime2 = microtime();
        $t=$_REQUEST['t'];
```

```

$flag=0;
$i=0;
while($i<$n-2){
    $j=$i+2;
    while (($j<$n-1&&$i==0) || ($j<$n&&$i!=0)){
        if ($j+1==$n) $_j=0; else $_j=$j+1;
        $a=$dists[$cidades_o[$i]->nome][$cidades_o[$i+1]->nome]+$dists[$cidades_o[$j]->nome][$cidades_o[$_j]->nome];
        $b=$dists[$cidades_o[$i]->nome][$cidades_o[$_j]->nome]+$dists[$cidades_o[$i+1]->nome][$cidades_o[$_j]->nome];
        if ($a>$b){
            $cidades_o_=$cidades_o;
            $tmp=array_reverse(array_splice($cidades_o_,$i+1,$j-$i));
            array_splice($cidades_o,$i+1,$j-$i,$tmp);
            $flag=1;
            break;
        }
        $j++;
    }
    if ($flag){
        $t++;
        break;
    }
    $i++;
}

```

```
//-----
```

```

$time2 = microtime();
$cidade->cleanTable3();
$cidade->insertdataTable3($cidades_o);
$time1 = microtime();
//-----calculo do tempo
$time = explode(" ",$time1);
$start_time1 = $mtime[1] + $mtime[0];
$time = explode(" ",$time1);
$end_time1 = $mtime[1] + $mtime[0];
$time = explode(" ",$time2);

```



```

$start_time2 = $mtime[1] + $mtime[0];
$mtime = explode(" ", $time2);
$end_time2 = $mtime[1] + $mtime[0];

$cidade->updateTime($end_time1-$start_time1+$seedruntime['time1'], $end_time2-
$start_time2+$seedruntime['time2'], $t);

if ($flag)
//   header("location: modules.php?name=data&op=alg3i&opx=data_calc_order&t=$t");
    echo "<script language=\"JavaScript\" type=\"text/JavaScript\">
<!--
setTimeout(\"MM_goToURL('parent','modules.php?name=data&op=alg3i&opx=data_calc_order&t=$t')\",1000);
function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+\".location=\"+args[i+1]+\"'\");
}
//-->
</script>running... \".$t;
else{
    $cidade->updateRun(null);
//   header("location: modules.php?name=data&op=alg3i&opx=data_calc_order_done");
    echo "<script language=\"JavaScript\" type=\"text/JavaScript\">
<!--
setTimeout(\"MM_goToURL('parent','modules.php?name=data&op=alg3i&opx=data_calc_order')\",1000);
function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+\".location=\"+args[i+1]+\"'\");
}
//-->
</script>";
}
}
//   header("location: modules.php?name=data&op=alg3i&opx=data_calc_order&t=$t");
    echo "<script language=\"JavaScript\" type=\"text/JavaScript\">
<!--
setTimeout(\"MM_goToURL('parent','modules.php?name=data&op=alg3i&opx=data_calc_order')\",1000);
function MM_goToURL() { //v3.0
    var i, args=MM_goToURL.arguments; document.MM_returnValue = false;

```

```
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+\".location=\"+args[i+1]+\"\\");  
  }  
  //-->  
  </script>waiting...";  
}  
*/  
?>
```