

Aplicações Pedagógicas para Terminais Móveis

DISSERTAÇÃO DE MESTRADO

José Valentim Aguiar de Freitas

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

julho | 2012

UMA

ApI

Aplicações Pedagógicas para Terminais Móveis

DISSERTAÇÃO DE MESTRADO

José Valentim Aguiar de Freitas

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTAÇÃO

Fernando Manuel Rosmaninho Morgado Ferrão Dias

RESUMO

Esta tese descreve o desenvolvimento de duas aplicações de *software* cujo objetivo consiste em demonstrar o funcionamento de duas ferramentas base de Sistemas Digitais.

A primeira aplicação, denominada KarnUMa, pretende demonstrar o funcionamento dos Mapas de Karnaugh, que são utilizados na simplificação de expressões algébricas Booleanas. Esta aplicação está disponível em duas versões com destino a duas plataformas distintas: a primeira KarnUMa, disponível para computador, e a segunda Pocket KarnUMa, disponível para terminais móveis sob a forma de *Midlet* ou *Android Package*.

A segunda aplicação desenvolvida, denominada ParTec, terá como objetivo demonstrar o funcionamento da Técnica das Partições, que é utilizada na eliminação de estados redundantes nas máquinas de estados. Esta segunda aplicação tem apenas o computador como plataforma de destino.

Este documento conta com um levantamento das aplicações atualmente existentes nas áreas de interesse, uma descrição das tecnologias utilizadas no desenvolvimento do *software*, uma apresentação desse mesmo *software* descrevendo o que este tem de inovador e por fim uma descrição da forma como as aplicações foram publicadas.

Palavras-chave: Sistemas Digitais, *software*, complementos educacionais, *E-Learning*, *M-Learning*,

ABSTRACT

This thesis describes the development of two software applications whose purpose is to demonstrate the operation of two common Digital System's tools.

The first application, called KarnUMa, aims to demonstrate the operation of Karnaugh maps, which are used to simplify Boolean algebra expressions. This application is available in two variants with different target platforms, the first, KarnUMa available for computer and the second Pocket KarnUMa available for mobile terminals in either *Midlet* or *Android Package*.

The second application, called ParTec, aims to demonstrate the operation of the Partition's Technique, which is used in the elimination of redundant states in state machines. This second application has only the computer as target platform.

This document includes a survey of the currently existing applications in the fields of interest, a description of the chosen technologies to develop the software, a presentation of this same software identifying its innovations and finally a description of how the applications were published.

Keywords: Digital systems, computer software, educational aids, E-Learning, M-Learning.

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer ao meu orientador pelo apoio e prontidão demonstrados ao longo do desenvolvimento deste projeto. Sem a sua orientação e partilha de conhecimentos não me teria sido possível completar este projeto.

Gostava também de agradecer a todos os meus familiares e amigos e oferecer as minhas saudações a todos aqueles que de uma maneira ou outra me apoiaram durante a execução deste projeto.

Por fim, vai o meu agradecimento a todos aqueles que dispensaram parte do seu tempo nos testes realizados às aplicações aqui desenvolvidas.

ÍNDICE

| | |
|--|----|
| Resumo | 3 |
| Abstract | 4 |
| Agradecimentos..... | 5 |
| Índice | 7 |
| Introdução | 11 |
| Glossário | 13 |
| Acrónimos..... | 14 |
| 1. Fundamentos Teóricos de Sistemas Digitais | 15 |
| 1.1. Mapas de Karnaugh..... | 15 |
| 1.1.1. Organização do Mapa de Karnaugh | 16 |
| 1.1.2. Escolha de Implicantes | 18 |
| 1.1.3. Extraíndo a fórmula simplificada dos Implicantes..... | 19 |
| 1.1.4. Simplificação de funções com Mapa de Karnaugh..... | 19 |
| 1.1.5. Resolver um mapa de Karnaugh..... | 20 |
| 1.1.5.1. Mapas de Karnaugh sem implicantes Primos Essenciais..... | 21 |
| 1.1.5.2. Resolução de Mapas de Karnaugh pela forma do Produto de Somas | 22 |
| 1.1.6. Mapas de Karnaugh de cinco e seis variáveis..... | 23 |
| 1.2. Algoritmo de Quine-McCluskey..... | 25 |
| 1.3. Técnica das Partições..... | 28 |
| 1.3.1. Máquinas de Estados..... | 28 |
| 1.3.2. Eliminação de Estados redundantes pela Técnica das Partições | 31 |
| 2. Estado de Arte | 35 |
| 2.1. <i>E-Learning</i> | 35 |
| 2.1.1. Tipos de <i>E-Learning</i> | 36 |
| 2.1.2. Composição do <i>E-Learning</i> | 36 |
| 2.1.3. Possíveis formas de <i>E-Learning</i> | 37 |
| 2.1.4. <i>E-Learning 2.0</i> | 38 |
| 2.1.5. Vantagens e desvantagens do <i>E-Learning</i> | 38 |
| 2.2. <i>M-Learning</i> | 39 |
| 2.2.1. Vantagens e Desvantagens do <i>M-Learning</i> :..... | 40 |
| 2.3. Trabalho Relacionado | 41 |

| | | |
|-----------|---|----|
| 2.3.1. | Aplicações já desenvolvidas na área de Mapas de Karnaugh..... | 41 |
| 2.3.1.1. | 4X4 Karnaugh Map Minimizer (4x4 KMM) | 41 |
| 2.3.1.2. | Karnaugh Applet (KA)..... | 41 |
| 2.3.1.3. | KarnaughMap (KM)..... | 42 |
| 2.3.1.4. | Karnaugh Map Minimizer (KMM) | 42 |
| 2.3.1.5. | Karnaugh Minimizer (KM2)..... | 42 |
| 2.3.1.6. | EasyKarnaugh (EK) | 43 |
| 2.3.1.7. | Karma | 43 |
| 2.3.1.8. | Boole-Deusto | 44 |
| 2.3.1.9. | Karnaugh Map Optimizer (KMO) | 44 |
| 2.3.1.10. | Pocket Karnaugh Map (PKM)..... | 44 |
| 2.3.1.11. | Karnaugh Map Advanced (KMA)..... | 45 |
| 2.3.1.12. | Visão geral das ferramentas de Mapas de Karnaugh | 45 |
| 2.3.2. | Desenvolvimento de <i>software</i> na área da Técnica das Partições | 47 |
| 2.4. | Conclusão..... | 47 |
| 3. | Desenvolvimento do <i>software</i> | 49 |
| 3.1. | Tecnologias utilizadas..... | 49 |
| 3.1.1. | Tecnologias utilizadas na implementação | 49 |
| 3.2. | Requisitos das Aplicações | 50 |
| 3.2.1. | Requisitos - KarnUMa | 50 |
| 3.2.1.1. | Requisitos Funcionais Gerais (RFG): | 50 |
| 3.2.1.2. | Requisitos Funcionais da versão PC (RFPC) | 51 |
| 3.2.1.3. | Requisitos Funcionais da versão Telemóvel (RFT) | 52 |
| 3.2.1.4. | Requisitos Não Funcionais Gerais (RNFG): | 52 |
| 3.2.1.5. | Requisitos Não Funcionais da versão PC (RNFP) | 52 |
| 3.2.1.6. | Requisitos Não Funcionais da versão Telemóvel (RNFT) | 52 |
| 3.2.2. | Requisitos – ParTek..... | 52 |
| 3.2.2.1. | Requisitos Funcionais | 52 |
| 3.2.2.2. | Requisitos Não Funcionais | 53 |
| 3.3. | Modelação e Implementação | 53 |
| 3.3.1. | KarnUMa versão PC | 54 |
| 3.3.2. | Pocket KarnUMa versão <i>Midlet (Java ME)</i> | 57 |
| 3.3.3. | Pocket KarnUMa versão <i>Android (Android Package)</i> | 61 |
| 3.3.4. | ParTec | 61 |

| | | |
|--------|--|-----|
| 3.4. | Análise de Tarefas..... | 64 |
| 3.5. | Apresentação das aplicações..... | 67 |
| 3.5.1. | KarnUMa (PC) | 67 |
| 3.5.2. | Pocket KarnUMa (<i>Midlet e Android Package</i>) | 74 |
| 3.5.3. | ParTec | 76 |
| 3.6. | Análise às aplicações desenvolvidas..... | 79 |
| 3.6.1. | Aplicações KarnUMa..... | 79 |
| 3.6.2. | Aplicação ParTek..... | 80 |
| 4. | Avaliação de Usabilidade..... | 83 |
| 4.1. | Falhas de Usabilidade | 84 |
| 4.1.1. | Falhas de Usabilidade da aplicação KarnUMa (ambas as versões) | 84 |
| 4.1.2. | Falhas específicas da versão PC do KarnUMa | 85 |
| 4.1.3. | Falhas específicas da versão Pocket KarnUMa (<i>Midlet e Android Package</i>)..... | 87 |
| 4.1.4. | Falhas de usabilidade da aplicação ParTec..... | 88 |
| 4.2. | Conclusão | 89 |
| 5. | Publicação das aplicações numa página <i>web</i> | 91 |
| 5.1. | Conceção da página <i>web</i> | 92 |
| 5.2. | Alojamento da página <i>web</i> | 93 |
| | Conclusão | 95 |
| | Referências | 97 |
| | Anexos | 101 |
| | Anexo A..... | 101 |
| | Anexo B..... | 109 |
| | Anexo C..... | 117 |

INTRODUÇÃO

Na última década tem-se verificado uma constante evolução no que aos computadores e terminais móveis diz respeito. Como tal tornou-se vantajoso utilizar estas novas tecnologias, que atualmente fazem parte do nosso dia-a-dia, como fontes de informação na área do ensino. Estas tecnologias têm demonstrado grande potencial quando aproveitadas para fins pedagógicos.

O trabalho desenvolvido nesta tese tem como objetivo desenvolver aplicações capazes de auxiliar na tarefa de ensinar a metodologia dos mapas de Karnaugh e da Técnica das Partições. Ao desenvolver uma aplicação com a qual se pretende facilitar a compreensão dos mapas de Karnaugh, esta não se deve limitar à apresentação de resultados, mas sim, ser capaz de demonstrar como são alcançados tais resultados. Levando isto em consideração, esta aplicação deve ser capaz de exibir todo o processo da resolução de um mapa de Karnaugh, bem como fornecer informação detalhada sobre cada passo deste processo. Um outro aspeto importante no ensino é a componente prática, como tal, aplicação deve permitir que o próprio utilizador possa executar o processo de resolução do Mapa de Karnaugh.

No que à aplicação sobre a Técnica das Partições diz respeito, também se pretende uma aplicação capaz de algo mais do que simplesmente apresentar o resultado final. O objetivo para esta aplicação passa igualmente por demonstrar o processo de resolução de forma detalhada.

Com a finalidade de aumentar exposição destas aplicações foi elaborado um artigo para a 10ª conferência Portuguesa de Controlo Automático – Controlo'12, que se realiza no Funchal entre os dias 16 e 18 de Julho com a organização da Universidade da Madeira. O artigo já foi revisto e aceite e será apresentado durante a conferência.

A motivação para este trabalho reside em desenvolver *software* a partir do qual o utilizador possa aprender a utilizar as ferramentas de Sistemas Digitais já referidas bem como testar os conhecimentos adquiridos.

Ao longo dos capítulos seguintes serão apresentados as várias etapas do desenvolvimento destas aplicações.

GLOSSÁRIO

Android – Sistema Operativo desenvolvido pela *Google* destinado a terminais móveis.

Classe – Estrutura de dados com a capacidade de abstrair um conjunto de objetos de características idênticas. Esta classe será responsável por definir os comportamentos dos seus objetos através dos métodos que nela estão definidos.

Download – Refere-se ao ato de descarregar ficheiros de um computador remoto para um computador local.

Framework – Conjunto de conceitos que são comuns a vários projetos de *software* que são fornecidos numa funcionalidade genérica. Como acontece numa *API*, a *framework* também contém um conjunto de métodos a serem utilizados na programação, mas distingue-se desta por ser responsável por definir o fluxo de controlo da aplicação.

Hipertexto – Refere-se ao texto em formato digital acedido via internet, do qual podem fazer parte blocos de texto, imagens ou sons.

Java – Linguagem de programação orientada a objetos.

Midlet – Aplicação *JAVA* para terminais móveis. Estas aplicações normalmente destinam-se a telemóveis e funcionam em máquina virtual *Java ME*.

Site – Consiste num sítio eletrónico, que não é mais que um conjunto de páginas *web*, ou seja, hipertextos que obedecem ao protocolo *http* da Internet.

Software – Produto da Engenharia de *software*, constituído não só pelo programa de computador mas também por toda a sua documentação (manuais, especificações,...).

Storyboard – Consiste numa sequência de ilustrações com a finalidade de pré-visualizar o funcionamento de uma aplicação.

Superclasse – Classe abstrata que define um modelo. Uma superclasse contém uma implementação incompleta sendo da responsabilidade das classes que dela derivam, completar a implementação e adicionar o seu comportamento específico.

Template – Em *web design* um *template* consiste na disposição em que são apresentados os conteúdos de uma página *web*.

Verilog – Linguagem de Descrição de *Hardware* utilizada para modelar circuitos eletrónicos.

Web – *Word Wide Web* ou *WWW* é um sistema de documentos de hipertexto acedidos via Internet.

ACRÓNIMOS

API – *Application Programming Interface* (ou Interface de Programação de Aplicativos). Consiste num conjunto de rotinas e padrões na forma de funções a serem utilizadas na programação. Na sua utilização não existe preocupação com os detalhes da sua implementação, apenas com os seus serviços.

CMS – *Content Management System*. É um sistema informático no qual é possível editar, publicar e modificar os conteúdos de uma página *web*. A gestão de conteúdos é realizada numa página central paralela ao conteúdo a ser publicado.

IDE – Do inglês *Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento. Consiste num programa de computador com o objetivo de aumentar a eficiência no processo de desenvolvimento de *software*.

ISS – Do inglês *Internet Information Services* é um servidor *web* desenvolvido pela *Microsoft*.

LDH – Linguagem Descritiva de *Hardware*. Esta é a denominação utilizada para as linguagens utilizadas na descrição formal e *design* de circuitos eletrónicos.

PDF – Do inglês *Portable Document Format*. Desenvolvido pela *Adobe Systems* em 1993, este é um formato para a escrita de documentos que tem com objetivo a visualização de documentos independentemente da aplicação, Sistema Operativo e *Hardware* utilizados.

VHDL – Do inglês *VHSIC Hardware Description Language*, é uma LDH utilizada para modelar circuitos eletrónicos.

VHSIC – Do inglês *Very-High-Speed Integrated Circuits*, foi um programa do governo Norte-americano que teve uma forte contribuição no surgimento do **VHDL**.

1. FUNDAMENTOS TEÓRICOS DE SISTEMAS DIGITAIS

1.1. MAPAS DE KARNAUGH

Os Mapas de Karnaugh são uma ferramenta de Sistemas Digitais que têm como objetivo tornar a simplificação de funções algébricas booleanas mais acessível e intuitiva ao ser humano. Tendo sido inventados em 1953 por Maurice Karnaugh, um engenheiro de Telecomunicações dos laboratórios Bell, estes mapas permitem a simplificação e minimização de equações booleanas de uma forma mais simples do que a aplicação de teoremas ou axiomas (Cajueiro, 2009). Este método pode ser utilizado para simplificar funções com qualquer número de variáveis de entrada, mas de modo a que continue prático e simples de se utilizar, pode dizer-se que a sua utilidade prática está limitada a 6 variáveis de entrada (Tocci & Widmer, 2001). Para o âmbito deste trabalho abordam-se os mapas de Karnaugh de 3 a 6 variáveis.

Antes de proceder com a explicação do funcionamento dos mapas de Karnaugh, alguns conceitos deverão ser clarificados de modo a tornar a compreensão dos mapas de Karnaugh perceptível.

a) Funções Algébricas Booleanas

Também conhecidas por funções lógicas ou funções binárias, estas funções são resolvidas através da Álgebra de Boole Binária. As variáveis presentes nestas funções são variáveis que só podem assumir dois valores lógicos 1 (*True*) e 0 (*False*), daí a sua denominação de variáveis Binárias ou Lógicas (Sêrro, Carvalho, & Arroz, 2005). Veja-se então o exemplo de uma equação booleana:

$$\bar{A} + \bar{B} + B.C$$

A equação booleana apresentada tem duas variáveis negadas (A e B), uma **negação** significa que uma afirmação verdadeira passa a falsa e vice-versa, ou seja, se A tinha valor lógico 1 ao ser negada (\bar{A}) passou a ter valor lógico 0. Também é possível observar a presença de operadores lógicos, existe uma **conjunção** em $B.C$ e uma **disjunção** em $\bar{A} + \bar{B}$. A uma Conjunção pode-se também chamar **Produto Lógico** ou **AND** e a uma Disjunção pode-se chamar **Soma Lógica** ou **OR** (Sêrro, Carvalho, & Arroz, 2005).

Para o âmbito deste projeto não será necessário explorar em mais detalhe a Álgebra de Boole e as funções lógicas.

b) Tabelas de Verdade

Uma tabela de verdade consiste numa listagem de todas as possíveis entradas e correspondentes saídas de um circuito lógico, ou seja, é uma tabela da qual se pode extrair a informação de como as saídas de um circuito lógico dependem das entradas. Começando por analisar uma pequena tabela de Verdade (**Tabela 1**) para um circuito lógico de duas entradas. Esta tabela contém todas as combinações possíveis para os valores lógicos de entrada A e B , e os correspondentes valores de saída F . Os valores de saída dependerão dos valores de entrada e do tipo de circuito lógico. Por exemplo pode-se verificar na **Tabela 2** que quando ambas as entradas A e B têm valor lógico 0, a saída F será igualmente 0. A **Tabela 2** consiste numa tabela de verdade para um circuito lógico de 3 entradas, onde novamente pode-se verificar todas as possíveis combinações de entradas

e correspondentes saídas, mas neste caso existem mais quatro combinações possíveis do que na tabela de 2 entradas.

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Tabela 1 - Exemplo de tabela de Verdade com duas entradas.

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |

Tabela 2 - Exemplo de tabela de Verdade com três entradas

Para uma tabela de verdade de 2 entradas existem 4 combinações possíveis, para uma de 3 entradas existem 8 combinações possíveis e para uma de 4 entradas existiriam 16 combinações possíveis, ou seja o número de combinações possíveis das variáveis de entrada será igual 2^N , sendo N o número de entradas do circuito lógico que a tabela de verdade representa. Note-se também que as combinações possíveis aparecem seguindo a contagem binária o que torna a listagem de todas as combinações mais simples (Tocci & Widmer, 2001).

1.1.1. ORGANIZAÇÃO DO MAPA DE KARNAUGH

A aplicação do método dos mapas de Karnaugh consiste numa representação gráfica do espaço que representa a função lógica (Dias, 2010). A transferência dos valores de saída de uma tabela de verdade para um mapa de Karnaugh é feita seguindo os princípios do **Código de Gray**¹, onde apenas um bit é alterado de célula para célula do mapa. Deste modo para manter a adjacência entre células, as entradas são dispostas na seguinte sequência 00, 01, 11, 10, assim de célula para célula existe uma alteração de apenas uma variável (Dias, 2010). Sendo assim as disposições de mapas de Karnaugh para 3 e 4 variáveis de entrada podem ser observadas nas **Fig. 1** e **Fig. 2** respetivamente.

| C/AB | 00 | 01 | 11 | 10 |
|------|----|----|----|----|
| 0 | 0 | 1 | 3 | 2 |
| 1 | 4 | 5 | 7 | 6 |

| AB/C | 0 | 1 |
|------|---|---|
| 00 | 0 | 1 |
| 01 | 2 | 3 |
| 11 | 6 | 7 |
| 10 | 4 | 5 |

Fig. 1 - Disposições possíveis para um mapa de Karnaugh com 3 variáveis de entrada.

¹**Código de Gray** – Código binário inventado por Frank Gray, no qual de um número para o seguinte apenas um bit varia (Doran, 2007).

| AB/CD | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| 00 | 0 | 1 | 3 | 2 |
| 01 | 0 | 5 | 7 | 6 |
| 11 | 12 | 13 | 15 | 14 |
| 10 | 8 | 9 | 11 | 10 |

| CD/AB | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| 00 | 0 | 4 | 12 | 8 |
| 01 | 1 | 5 | 13 | 9 |
| 11 | 3 | 7 | 14 | 11 |
| 10 | 2 | 6 | 15 | 10 |

Fig. 2 - Disposições possíveis para um mapa de Karnaugh com 4 variáveis de entrada.

Uma vez analisada a Fig. 1 pode-se observar que um mapa com 3 variáveis de entrada pode ser desenhado na horizontal ou na vertical, sendo que quando desenhado na horizontal a ordem das variáveis de entrada é distinta de quando desenhado na vertical. O mesmo sucede num mapa com 4 variáveis de entrada, em que a ordem destas mesmas pode ser disposta de duas formas, como é possível verificar pela Fig. 2. De referir que a alteração da ordem das variáveis de entrada provoca igualmente uma alteração da ordem das posições das células do mapa.

Observe-se o exemplo da Fig. 3 que representa uma tabela de verdade para uma função lógica de 3 entradas (A,B e C).

| Valor Decimal | A | B | C | F |
|---------------|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 |

| C/AB | 00 | 01 | 11 | 10 |
|------|----|----|----|----|
| 0 | 0 | 1 | 3 | 2 |
| 1 | 4 | 5 | 7 | 6 |

| C/AB | 00 | 01 | 11 | 10 |
|------|----|----|----|----|
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

Fig. 3 - Tabela de Verdade e respetivo mapa de Karnaugh.

Neste exemplo pode-se analisar como é feito o preenchimento de um mapa de Karnaugh. Cada célula do mapa corresponde a um valor de saída de uma certa combinação de entradas. A azul na tabela de verdade está o número decimal correspondente à combinação de entradas em binário, por ex.: 0 → 000, 1 → 001; assim a partir destes números pode-se observar no primeiro mapa de Karnaugh da Fig. 3 que combinação corresponde a cada célula. Já no segundo mapa da Fig. 3 está representado em cada célula o valor da função para a combinação correspondente.

Na Fig. 4 é possível verificar que Minitermo corresponde a cada posição do mapa de Karnaugh, neste caso para 3 variáveis de entrada.

| C/AB | 00 | 01 | 11 | 10 |
|------|-------------------------|-------------------|-------------|-------------------|
| 0 | $\bar{A}\bar{B}\bar{C}$ | $\bar{A}B\bar{C}$ | $AB\bar{C}$ | $A\bar{B}\bar{C}$ |
| 1 | $\bar{A}\bar{B}C$ | $\bar{A}BC$ | ABC | $A\bar{B}C$ |

$\bar{A}B\bar{C} + AB\bar{C} = (\bar{A} + A)(B\bar{C}) = B\bar{C}$

Fig. 4 - Mapa de Karnaugh de 3 variáveis de entrada, com os Minitermos correspondentes a cada célula do mapa.

1.1.2. ESCOLHA DE IMPLICANTES

Um implicante de uma função lógica consiste num agrupamento de 1's (Minitermos) que satisfaça as regras de associação de termos, sendo estas a de adjacência e agrupar 2^n termos (Dias, 2010).

Posteriormente ao preenchimento do mapa de Karnaugh, o objetivo passa por conseguir extrair a fórmula simplificada. Para isso é necessário formar os implicantes, o que é realizado através de associações feitas com os termos que tomam valor 1. Nesta secção de modo a explicar a formulação de implicantes apenas serão consideradas as associações feitas de Minitermos, ou seja, os termos que têm valor lógico 1. Formar associações acaba por ser um processo simples onde apenas é necessário agrupar os 1's respeitando duas regras, os termos agrupados têm que ser adjacentes e os grupos devem conter 2^n termos.

Adjacência - Como já foi referido num mapa de Karnaugh as células são adjacentes uma a uma, sendo também adjacentes nas extremidades. Olhando para a Fig. 4 Pode-se verificar que o termo do canto superior esquerdo ($\bar{A}\bar{B}\bar{C}$) é adjacente ao termo do canto superior direito ($A\bar{B}\bar{C}$) visto que só variam na variável A. Sendo assim pode-se verificar a adjacência entre células como se a extremidade direita do mapa estivesse "colada" à extremidade esquerda e do mesmo modo a extremidade superior estivesse colada à extremidade inferior do mapa (Tocci & Widmer, 2001).

| C/AB | 00 | 01 | 11 | 10 |
|------|----|----|----|----|
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

Fig. 5 - Mapa de Karnaugh de 3 variáveis com dos implicantes.

Observe-se o mapa de Karnaugh da Fig. 5 onde estão assinalados dois Implicantes a verde e a vermelho. No Implicante vermelho pode-se verificar a adjacência entre os termos das extremidades direita e esquerda, sendo este constituído por $2^2 = 4$ termos. Já o implicante a verde é formado por $2^1 = 2$ termos que são adjacentes, pois estão lado a lado.

Analisando agora os mapas da Fig. 6 pode-se verificar que no mapa da esquerda existe um Implicante corretamente formado de $2^2 = 4$ termos sendo todos estes adjacentes, já no mapa da direita ainda que seja constituído por $2^1 = 2$ termos, ou seja é uma potência de 2, este não está corretamente formado pois os termos não são adjacentes.

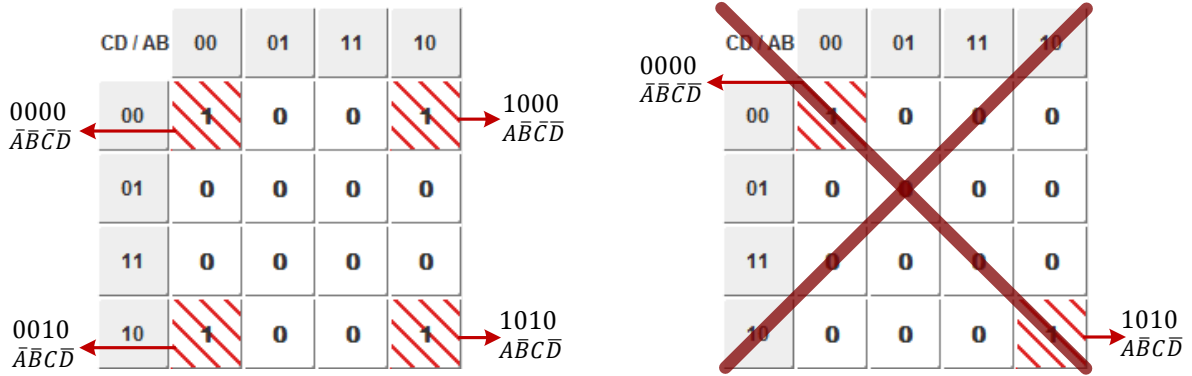


Fig. 6 - Mapas de Karnaugh de 4 variáveis.

Observe-se os termos do mapa da Esquerda:

- O termo $\bar{A}\bar{B}\bar{C}\bar{D}$ é adjacente com $\bar{A}\bar{B}C\bar{D}$ e com $A\bar{B}\bar{C}\bar{D}$, uma vez que apenas diferem numa variável. E por sua vez o termo $A\bar{B}C\bar{D}$ é também adjacente com $\bar{A}\bar{B}C\bar{D}$ e $A\bar{B}C\bar{D}$, porque também diferem numa única variável.

Agora analisando o mapa da esquerda:

- Os termos $\bar{A}\bar{B}\bar{C}\bar{D}$ e $A\bar{B}C\bar{D}$ não são adjacentes pois diferem em 2 variáveis A e C , logo o implicante deste mapa não está corretamente formado.

1.1.3. EXTRAINDO A FÓRMULA SIMPLIFICADA DOS IMPLICANTES

Observe-se a fórmula correspondente ao mapa da esquerda da Fig. 6:

$$F = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} = \bar{B}\bar{D}$$

O Implicante formado corresponde a todos os termos existentes neste mapa, e extraíndo a sua fórmula simplificada obtém-se $\bar{B}\bar{D}$. Para chegar a esta simplificação retiram-se as variáveis que mudam de valor dentro de todos os termos utilizados pelo Implicante, ou seja, nos quatro termos deste Implicante as variáveis que mudavam de valor eram A e C , logo estas foram retiradas ficando-se com $\bar{B}\bar{D}$.

1.1.4. SIMPLIFICAÇÃO DE FUNÇÕES COM MAPA DE KARNAUGH

A simplificação de funções lógicas através dos mapas de Karnaugh pode seguir duas estratégias, a Soma de Produtos ou o Produto de Somas. No Soma de Produtos pretende-se formar os maiores grupos possíveis de Minitermos, por outro lado no Produto de Somas pretende-se formar os maiores grupos possíveis de Maxitermos. Um exemplo de uma fórmula resultante da aplicação da Soma de Produtos seria:

$$F = \bar{A}B + \bar{A}\bar{C}$$

A mesma solução mas pelo Produto de Somas seria:

$$F = (B + \bar{C}).\bar{A}$$

A diferença é que na Soma de Produtos cada grupo formado corresponde a um produto na fórmula (por ex.: $\bar{A}B$), já no Produto de Somas cada grupo formado corresponde a uma Soma na fórmula (por ex.: $B + \bar{C}$).

1.1.5. RESOLVER UM MAPA DE KARNAUGH

A resolução de um mapa de Karnaugh de modo a se obter a *solução mínima* passa por um conjunto de etapas, etapas essas que serão descritas seguidamente.

Solução mínima – Consiste na obtenção de uma solução que seja formada pelo mínimo de implicantes possíveis e que estes contenham o mínimo de variáveis possíveis.

1ª Etapa – Encontrar todos os implicantes primos

Numa primeira fase o objetivo passa por identificar todos os Implicantes Primos presentes no mapa de Karnaugh. Este primeiro passo está ilustrado na **Fig. 7** onde todos os implicantes Primos estão identificados no mapa com a respetiva fórmula do lado direito.

Implicante Primo – Um implicante primo não é mais do que um implicante que agrupa o máximo número de termos possível, respeitando as regras de associação de termos.

| F AB / CD | 00 | 01 | 11 | 10 |
|--------------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 1 | 1 | 0 | 0 |

$$F = \bar{A}CD + \bar{A}BC + A\bar{C} + BCD + ABD$$

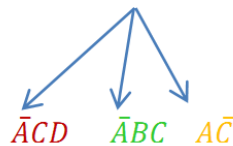
Fig. 7 - Mapa de Karnaugh de 4 variáveis com todos os implicantes Primos assinalados e respetiva fórmula.

2ª Etapa – Identificar quais dos implicantes Primos são Essenciais

Posteriormente à identificação de todos os implicantes Primos deve-se verificar quais destes são Essenciais. Na **Fig. 8** está representado o mesmo mapa de Karnaugh, mas desta vez com os implicantes Primos Essenciais identificados. Os termos que são utilizados por um único Implicante têm o fundo cinzento.

| AB / CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 1 | 1 | 0 | 0 |

Implicantes Primos Essenciais



Implicantes Primos



Fig. 8 – Mapa de Karnaugh de 4 variáveis com todos os implicantes Primos assinalados e identificação dos Primos Essenciais.

Implicante Primo Essencial – Consiste num implicante Primo que implica um termo de valor lógico 1, que não é coberto por nenhum outro Implicante.

3ª – Etapa – Escolher os Implicantes Primos Essenciais

Tendo identificado todos os Implicantes Primos Essenciais, verifica-se se estes são suficientes para se atingir a solução mínima. Se os Implicantes Primos Essenciais forem suficientes para se cobrir todos os termos, então a solução mínima será dada pelos Implicantes Primos Essenciais, caso contrário deve-se escolher os Implicantes Primos necessários para que todos os termos sejam utilizados. Esta seleção deve ser feita escolhendo o mínimo de Implicantes possível e com mínimo de variáveis possíveis. No caso ilustrado pela **Fig. 9** pode-se verificar pelo mapa da esquerda que os Implicantes Primos Essenciais não são suficientes para se atingir a solução do mapa, uma vez que resta um termo por utilizar. Sendo assim é necessário escolher um dos Implicantes Primos não Essenciais. Neste caso qualquer um dos implicantes Primos não Essenciais restantes pode ser utilizado para se atingir a solução mínima, visto que ambos têm a mesma quantidade de variáveis e ambos utilizam o termo que resta implicar.

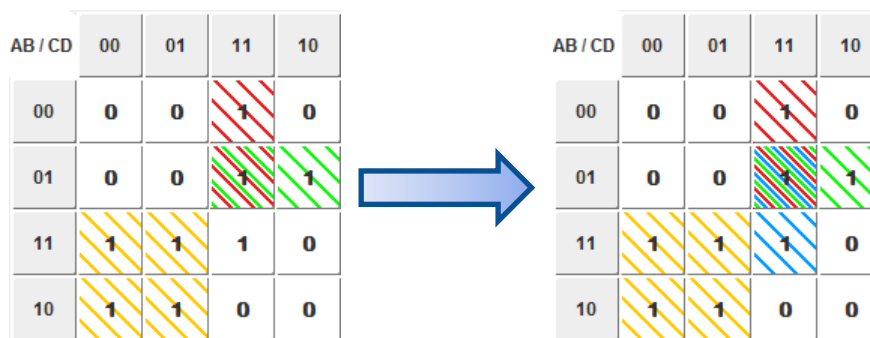


Fig. 9 - Mapas de Karnaugh de 4 variáveis com todos os implicantes Primos assinalados e identificação dos Primos Essenciais

Sendo assim a solução mínima deste mapa de Karnaugh é dada por:

$$F = \bar{A}CD + \bar{A}BC + A\bar{C} + BCD$$

Note-se que apesar do implicante Primo não Essencial escolhido ter sido BCD , a escolha também poderia ter passado pelo implicante ABD , visto que este também satisfaz os critérios referidos anteriormente.

De referir ainda que para atingir as soluções mínimas de Mapas de Karnaugh nem sempre é necessário passar por todas estas etapas. Como foi referido anteriormente o mapa pode ser solucionado só pelos Implicantes Primos Essenciais, não havendo necessidade de se escolher Primos não Essenciais e também podem existir Mapas de Karnaugh que não têm Implicantes Primos Essenciais, ou seja, não existem termos que sejam implicados por um único Implicante Primo.

1.1.5.1. MAPAS DE KARNAUGH SEM IMPLICANTES PRIMOS ESSENCIAIS

Alguns mapas de Karnaugh podem não conter Implicantes Primos Essenciais. Nestes casos para chegar à solução mínima deve-se identificar todos os Implicantes Primos e posteriormente escolher para a solução somente os necessários para que todos os termos sejam implicados, tendo como objetivo o mínimo de implicantes possíveis, com o mínimo de variáveis para se atingir a solução Mínima. A **Fig. 10** representa um

exemplo de um mapa de Karnaugh sem Implicantes Primos Essenciais. No mapa da esquerda estão todos os Implicantes Primos identificados enquanto os mapas da direita apresentam as duas possíveis soluções deste mapa, com os Implicantes Primos necessários.

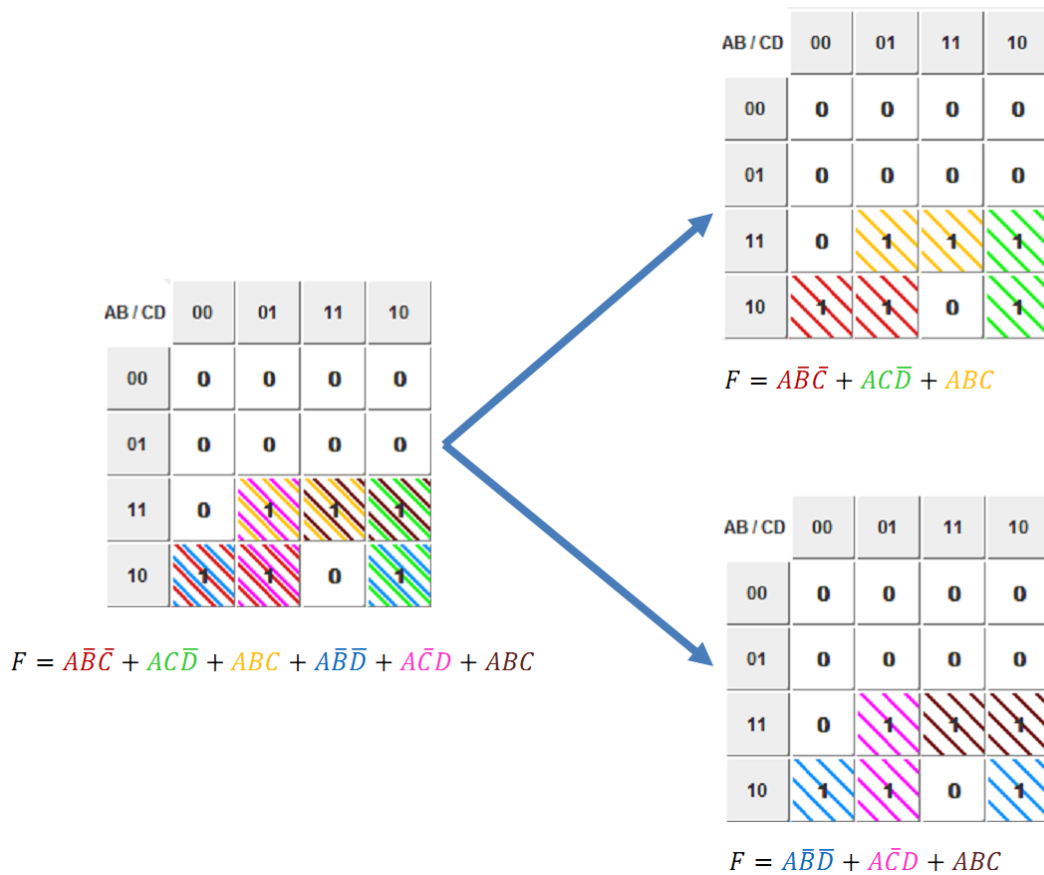


Fig. 10 - Mapa de Karnaugh de 4 variáveis cuja solução não tem implicantes Primos Essenciais.

1.1.5.2. RESOLUÇÃO DE MAPAS DE KARNAUGH PELA FORMA DO PRODUTO DE SOMAS

A resolução de Mapas de Karnaugh pela forma do Produto de Somas é em quase tudo semelhante à forma Soma de Produtos (forma que até agora tem sido utilizada nos exemplos). As diferenças são as seguintes:

- Em vez de Implicantes têm-se Implicados, sendo as definições de Implicado, implicado Primo e implicado Primo Essencial semelhantes às de implicante mas com a diferença que neste caso são formados agrupamentos de termos de valor Lógico 0 (Maxitermos) (Dias, 2010)
- A solução mínima será dada por um Produto de Somas, sendo que cada Implicado corresponderá a uma soma.

A Fig. 11 representa um exemplo de uma solução do mapa de Karnaugh pela forma do Produto de Somas, como pode-se constatar, cada Implicado é dado por uma soma sendo a solução dada pelo produto de todas as somas.

| AB / CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 1 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 1 | 0 | 1 | 1 |

$$F = (\bar{B} + D).(\bar{B} + \bar{C}).(\bar{A} + C + \bar{D}).(A + B + \bar{D})$$

Fig. 11 - Mapa de Karnaugh de 4 variáveis resolvido pela forma Produto de Somas.

1.1.6. MAPAS DE KARNAUGH DE CINCO E SEIS VARIÁVEIS

Os mapas de Karnaugh são normalmente utilizados para resolver funções lógicas de até quatro variáveis, contudo estes podem ainda ser utilizados para a simplificação de funções lógicas de cinco e seis variáveis.

Para simplificar funções de cinco variáveis utilizam-se dois mapas de Karnaugh, como pode-se observar pela Fig. 12, o preenchimento do mapa é feito de forma idêntica sendo que neste caso o mapa da esquerda corresponde aos casos em que a variável E toma valor 1 e o mapa da direita os casos em que a variável E toma valor 0. Posteriormente ao preenchimento do mapa, pretende-se formar as associações, uma vez mais funcionará de forma idêntica às situações em que era utilizado um único mapa com a adição de adjacência entre as células que ocupem a mesma posição em ambos os mapas.

| AB / CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 1 | 1 | 1 | 0 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 0 | 0 | 0 | 1 |

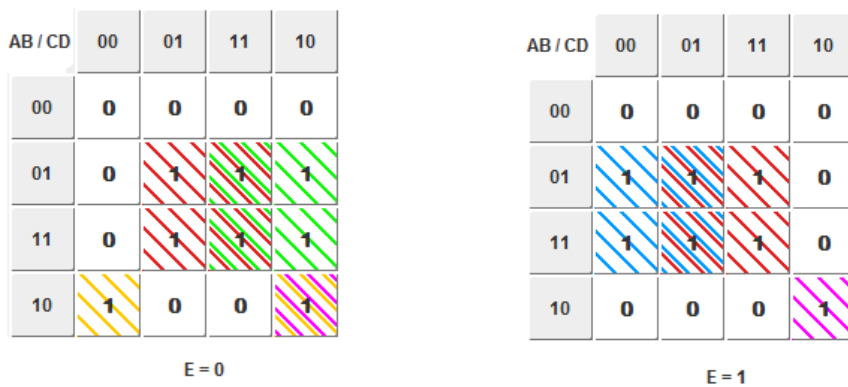
E = 1

| AB / CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | 1 | 1 |
| 10 | 1 | 0 | 0 | 1 |

E = 0

Fig. 12 - Mapas de Karnaugh para função lógica de 5 variáveis

Levando em consideração esta nova adjacência são obtidos os Implicantes assinalados na Fig. 13. Pode-se constatar que os implicantes BD e $\bar{A}\bar{B}\bar{C}\bar{D}$ beneficiam desta nova adjacência, visto que estão na mesma posição em ambos os mapas. Pode-se ainda verificar que os Implicantes formados num único mapa terão a variável E com o valor correspondente ao mapa em que se encontram. Se está no mapa em que $E = 0$, a expressão do implicante terá a variável E negada como é o caso dos Implicantes $BC\bar{E}$ e $AB\bar{D}\bar{E}$. No caso do implicante ser formado no mapa em que $E = 1$, o Implicante terá a variável E positiva como acontece com $B\bar{C}E$, por fim se o Implicante ocupar ambos os mapas a variável E é retirada uma vez que esta difere em ambos os mapas, como acontece com BD e $\bar{A}\bar{B}\bar{C}\bar{D}$.



$$F = BD + BC\bar{E} + A\bar{B}\bar{D}\bar{E} + \bar{B}\bar{C}E + A\bar{B}C\bar{D}$$

Fig. 13 - Mapas de Karnaugh para função lógica de 5 variáveis com os implicantes assinalados.

Para se simplificar uma função lógica de seis variáveis são utilizados quatro mapas de Karnaugh como pode-se observar na Fig. 14. Nestes mapas existe adjacência entre os termos das mesmas células como acontece nos mapas de cinco variáveis sendo esta adjacência presente entre os mapas posicionados esquerda/direita e acima/abaixo, ou seja, os mapas posicionados diagonalmente não são adjacentes. É ainda possível verificar que os mapas de cima terão $F = 0$, os de baixo $F = 1$, os da esquerda $E = 0$ e dos da direita $E = 1$.

A Fig. 15 representa os mapas de Karnaugh para uma função lógica de 6 variáveis com os implicantes devidamente assinalados e a respetiva solução.

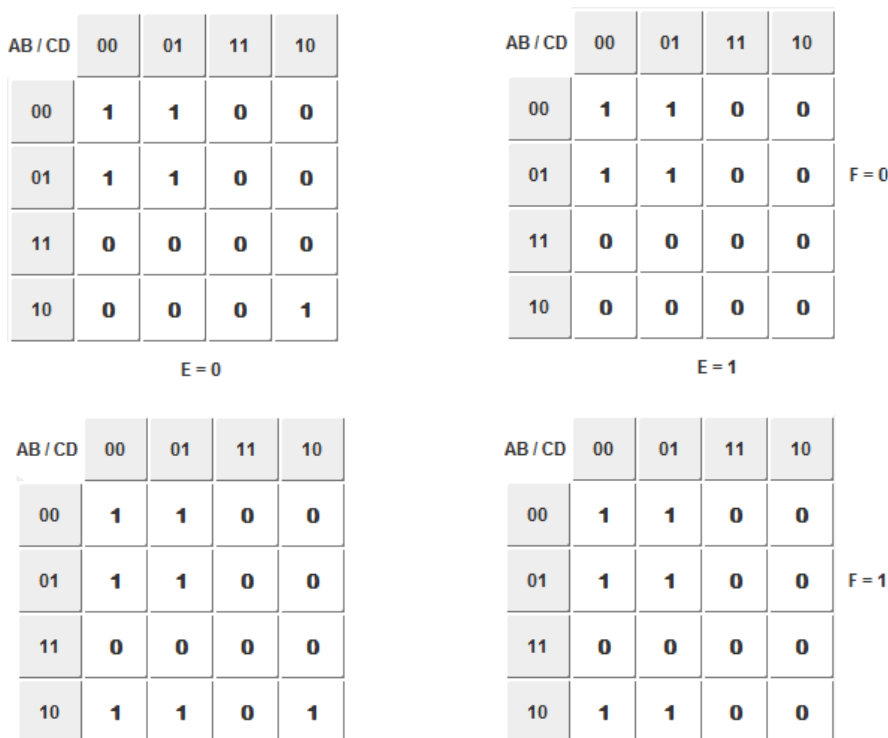


Fig. 14 - Mapas de Karnaugh para função lógica de 6 variáveis.



Fig. 15 - Mapas de Karnaugh para função lógica de 6 variáveis com os implicantes assinalados.

1.2. ALGORITMO DE QUINE-MCCLUSKEY

O Algoritmo de Quine-McCluskey, também denominado de Método dos Implicantes Primos, é um método utilizado para a simplificação de funções lógicas booleanas. Este método foi criado por W.V. Quine e Edward J. McCluskey (Tomaszewski, U.Celik, & Antoniou, 2003). Este método ainda que tenha a mesma funcionalidade que os mapas de Karnaugh, não é tão intuitivo como os mapas de Karnaugh para ser utilizado por humanos, mas por outro lado é mais adequado na vertente computacional sendo mais eficiente quando utilizado de forma automática por exemplo na área de programação (Crenshaw, 2004).

De modo a simplificar uma função lógica booleana fazendo uso do Método de Quine-McCluskey vários passos têm que ser seguidos, passos esses que serão ilustrados ao longo da sua especificação. De modo a facilitar a percepção, este algoritmo será aplicado à tabela seguinte (**Tabela 3**) fazendo uma descrição de cada passo.

1º Passo – Tabela Inicial

Numa primeira fase pretende-se a partir da **Tabela 3**, gerar uma outra tabela que disponha os termos de valor lógico 1 ou indiferente (X) consoante o número de 1's que o termo contenha. Por exemplo o termo 0101

ficaria na secção de Minitermos com dois 1's. Deste modo obtém-se a **Tabela 4** a qual contém todos os termos de valor lógico 1 ou indiferente (X), organizados pelo número de 1's (Dias, 2010).

| Minitermo | A | B | C | D | F |
|-----------|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | X |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 0 |

Tabela 3 - Tabela de verdade para uma função Lógica a simplificar pelo Algoritmo de Quine-McCluskey.

| Índice (Número de 1's) | Minitermo | Representação Binária |
|---------------------------|-----------|--------------------------|
| 1 | M1 | 0001 |
| | M2 | 0010 |
| | M4 | 0100 |
| 2 | M5 | 0101 |
| | M6 | 0110 |
| | M12 | 1100 |
| 3 | M7 | 0111 |
| | M13 | 1101 |
| | M14 | 1110 |

Tabela 4 - Tabela correspondente ao primeiro passo do Algoritmo de Quine-McCluskey.

2º Passo – Associação de Minitermos

Posteriormente à organização dos Minitermos consoante a quantidade de 1's (índice na tabela), obtém-se uma tabela a partir da qual será mais simples verificar que termos poderão ser associados, uma vez que para tal devem-se comparar os termos de índice k com os de $k+1$. Se destas comparações resultarem termos que variem somente num bit, ou seja termos adjacentes, é então possível simplifica-los, esta simplificação é feita colocando um traço (-) no bit que varia. Da execução deste passo resulta então a metade direita da **Tabela 5** que contém todos os pares de Maxitermos adjacentes e a simplificação resultante (Dias, 2010).

| 1ª Iteração | | | 2ª Iteração | |
|---------------------|------------|-----------|-------------------|-----------|
| Comparações Índices | Minitermos | Resultado | Minitermos | Resultado |
| 1 e 2 | M1 e M5 | 0 - 0 1 * | | |
| | M2 e M6 | 0 - 1 0 * | | |
| | M4 e M5 | 0 1 0 - | | |
| | M4 e M6 | 0 1 - 0 | | |
| | M4 e M12 | - 1 0 0 | | |
| 2 e 3 | M5 e M7 | 0 1 - 1 | M4, M5, M6 e M7 | 0 1 - - * |
| | M5 e M13 | - 1 0 1 | | |
| | M6 e M7 | 0 1 1 - | M4, M5, M12 e M13 | - 1 0 - * |
| | M6 e M14 | - 1 1 0 | | |
| | M12 e M13 | 1 1 0 - | M4, M6, M12 e M14 | - 1 - 0 * |
| | M12 e M14 | 1 1 - 0 | | |

Tabela 5 - Tabela correspondente ao segundo passo do Algoritmo de Quine-McCluskey.

Seguidamente deverá ser executado o mesmo processo, mas neste caso comparando as simplificações obtidas, as de índices 1 e 2 com as de 2 e 3 e assim sucessivamente caso houvessem índices maiores. Desta comparação devem novamente ser simplificados os termos que forem adjacentes, note-se que nesta iteração é mais simples verificar os termos adjacentes, visto que estes serão adjacentes se tiverem o traço (-) na mesma posição.

No caso deste exemplo apenas estas iterações são necessárias para completar o 2º passo, na eventualidade de existir um maior índice de 1's, mais iterações poderiam ser necessárias.

Quando se verificar que não é possível executar mais nenhuma iteração, devem-se assinalar com um * os termos que não foram utilizados em nenhuma associação, o que inclui todos termos resultantes da última iteração uma vez que não é feita mais nenhuma associação sobre estes.

3º Passo - Tabela dos Implicantes Primos

Com a simplificação dos Maxitermos no passo anterior alcançou-se a lista dos Implicantes Primos desta função. Neste passo final pretende-se identificar quais os Implicantes Primos necessários para se obter a fórmula simplificada da função lógica. Para tal é construída uma tabela que contém todos os implicantes Primos obtidos no passo anterior em cada linha e todos os Maxitermos em cada coluna. Para preencher a tabela assinalam-se quais os Maxitermos implicados por cada Implicante Primo. Um vez com a tabela preenchida torna-se fácil identificar quais os Implicantes Primos que são Essenciais, estes serão os que tiverem um Maxitermo que só é

utilizado pelo Implicante Primo em questão. A **Tabela 6** consiste na tabela de Implicantes Primos, estando os Implicantes Primos Essenciais assinalados com uma estrela (★). Note-se ainda que o Maxitermo M12 tem a sua coluna assinalada a cinzento, isto porque como este termo tem valor lógico indiferente, não deve ser contabilizado na procura por implicantes Primos Essenciais. Olhando para a tabela e posteriormente à identificação dos Implicantes Primos Essenciais, segue-se a escolha dos Implicantes Primos a utilizar de forma idêntica ao que acontecia nos Mapas de Karnaugh. Os Primos Essenciais terão que fazer parte da expressão lógica simplificada e posteriormente verifica-se se estes são suficientes para implicar todos os Maxitermos da função. Na eventualidade de serem suficientes, significa que atingiu-se a fórmula simplificada, caso contrário escolhem-se o ou os Implicantes Primos necessários a utilizar levando sempre em consideração os critérios referidos anteriormente para que se obtenha uma solução Mínima (Dias, 2010).

| Implicantes Primos | | M1 | M2 | M4 | M5 | M6 | M7 | M12 | M13 | M14 |
|--------------------|---|---------|----|----|----|----|----|-----|-----|-----|
| M1 e M5 | ★ | 0 - 0 1 | X | | X | | | | | |
| M2 e M6 | ★ | 0 - 1 0 | | X | | X | | | | |
| M4, M5, M6 e M7 | ★ | 0 1 - - | | X | X | X | X | | | |
| M4, M5, M12 e M13 | ★ | - 1 0 - | | X | X | | | X | X | |
| M4, M6, M12 e M14 | ★ | - 1 - 0 | | X | | X | | X | | X |

Tabela 6 - Tabela de Implicantes Primos.

Seguindo os critérios referidos e uma vez que todos os Implicantes Primos encontrados são Essenciais a solução é dada por:

$$F = \bar{A}B + B\bar{C} + B\bar{D} + \bar{A}\bar{C}D + \bar{A}\bar{C}\bar{D}$$

Ainda que no exemplo utilizado para a demonstração do Algoritmo de Quine-McCluskey, tenha sido aplicado de modo a se obter a solução na forma de Soma de Produtos, ou seja fazendo uso dos Minitermos, este algoritmo pode ser igualmente aplicado caso de pretenda obter a solução na forma de Produto de Somas. Para tal basta considerar os Maxitermos em vez de Minitermos, seguindo exatamente os mesmos passos.

1.3. TÉCNICA DAS PARTIÇÕES

A Técnica das Partições consiste numa ferramenta de Sistemas Digitais que tem por objetivo a eliminação de estados redundantes numa máquina de estados (Dias, 2010).

1.3.1. MÁQUINAS DE ESTADOS

Também denominadas por circuitos sequenciais, são circuitos onde as saídas não dependem somente das entradas do estado atual, dependem também de entradas anteriores. Como o próprio nome deste circuito indica, uma máquina de estados será então composta por vários estados, sendo que cada estado contém informações como estados anteriores, dados internos e possíveis ações seguintes (Dias, 2010).

Para o âmbito deste trabalho dois tipos de máquinas de estados foram considerados, as Máquinas de Moore e as Máquinas de Mealy, cujos criadores são E. F. Moore e G. H. Mealy respetivamente. A diferença entre estas máquinas reside no facto de que, na máquina de Moore, o valor de saída de cada estado depende somente do estado atual, enquanto na máquina Mealy a saída depende do estado atual, mas também da entrada (Dias, 2010). A **Tabela 7** corresponde a uma máquina de Moore com uma entrada e uma saída e a **Tabela 8** diz respeito a uma máquina de Mealy de uma entrada e uma saída.

| Estado Actual | Saída | Estados Seguintes | |
|---------------|-------|-------------------|-----|
| | | X=0 | X=1 |
| A | 0 | E | C |
| B | 0 | A | B |
| C | 1 | C | D |
| D | 0 | E | C |
| E | 1 | E | A |

Tabela 7 - Tabela de estados de uma máquina de Moore.

| Estado Actual | Estados Seguintes | |
|---------------|-------------------|-------|
| | X=0 | X=1 |
| A | B / 0 | E / 1 |
| B | B / 1 | C / 1 |
| C | E / 0 | A / 1 |
| D | A / 0 | A / 0 |
| E | B / 0 | A / 1 |

Tabela 8 - Tabela de estados de uma máquina de Mealy.

As máquinas de estados podem ser representadas na forma de tabelas, mas podem ainda ser representadas sob a forma de diagrama de estados. As **Fig. 16** e **Fig. 17** representam os diagrama de estados correspondentes às **Tabela 7** e **Tabela 8** respetivamente.

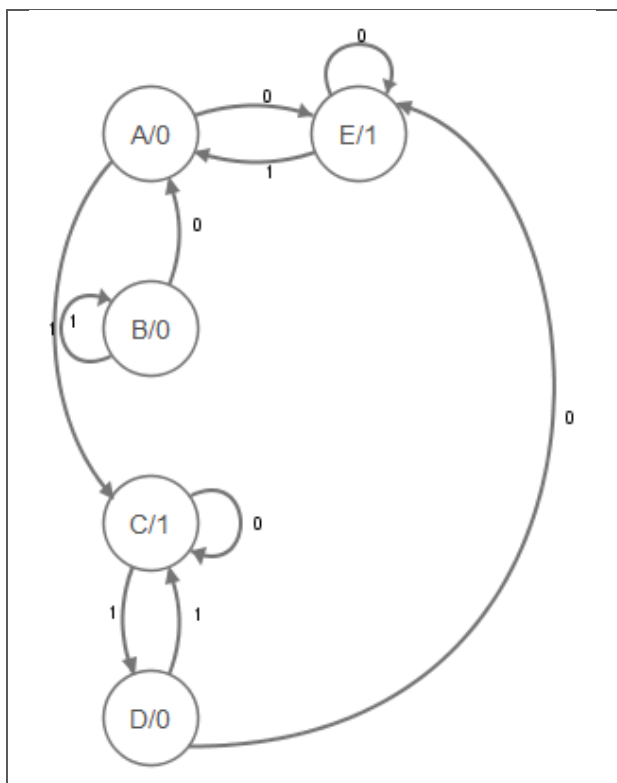


Fig. 16 - Diagrama de estados de uma máquina de Moore.

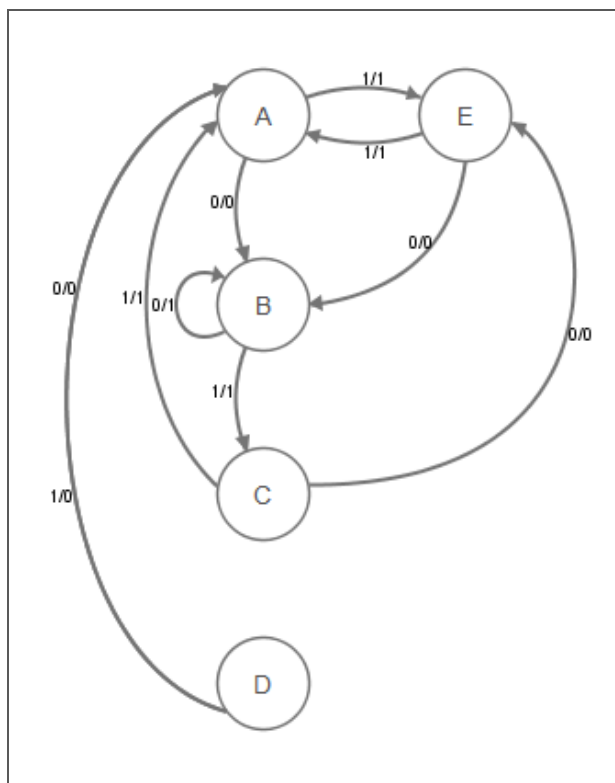
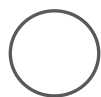


Fig. 17 - Diagrama de estados de uma máquina de Mealy.

Os diagramas de estados seguem a seguinte notação:



Estado – Caso o diagrama de estados represente uma máquina de Moore o círculo que representa um estado contém as seguintes informações: a designação do estado e o valor de saída. Caso se trate de uma máquina de Mealy contém somente a designação do estado.



Transição para estado Seguinte – Num diagrama de uma máquina de Moore conta com indicação do estado seguinte. Caso se trate de uma máquina de Mealy à indicação do estado seguinte junta-se o valor de saída para essa situação de estado seguinte.

As máquinas de estados podem variar no número de entradas e de saídas. O número de entradas tem uma relação diretamente proporcional com o número de estados seguintes, por exemplo: 1 entrada dá origem a 2 estados seguintes, 2 entradas dá origem a 4 estados seguintes, 3 entradas dá origem a 8 estados seguintes, ou seja, N entradas dá origem a 2^N estados seguintes, embora possam existir situações em que há combinações das entradas que não são possíveis, reduzindo assim o número de estados seguintes possíveis. A **Fig. 18** representa uma Tabela de estados com 2 entradas e conseqüentemente 4 estados seguintes.

| Estado Actual | Saída | Estados Seguintes | | | |
|---------------|-------|-------------------|-------|-------|-------|
| | | xy=00 | xy=01 | xy=10 | xy=11 |
| A | 0 | A | A | A | A |
| B | 0 | A | A | A | A |
| C | 0 | A | A | A | A |

Fig. 18 - Tabela de estados de uma máquina de Moore com 2 entradas.

O número de variáveis de saída, por outro lado, tem uma relação direta com a quantidade de valores disponíveis para as saídas visto que para N variáveis de saída estão disponíveis 2^N valores possíveis para a saída. Na Fig. 19 é possível verificar que para 2 variáveis de saída estão disponíveis 4 valores possíveis (00, 01, 10, 11) para a saída, embora também aqui possam existir combinações de saída que não são utilizadas.

| Estado Actual | Estados Seguintes | |
|---------------|-------------------|--------|
| | x=0 | x=1 |
| A | A / 00 | A / 00 |
| B | A / 00 | A / 00 |
| C | A / 00 | A / 00 |
| D | A / 10 | A / 00 |
| E | A / 11 | A / 00 |

Fig. 19 - Tabela de estados de uma máquina de Mealy com 2 variáveis de saída.

1.3.2. ELIMINAÇÃO DE ESTADOS REDUNDANTES PELA TÉCNICA DAS PARTIÇÕES

Ao desenvolver uma máquina de estados, é possível introduzir, inadvertidamente, estados desnecessários. Se existirem duas máquinas de estado que desempenham a mesma função e uma dispuser de K estados e a outra de J estados, com $J > K$, pode afirmar-se que a máquina com J estados tem estados redundantes.

Estes estados redundantes resultam frequentemente na utilização de mais *hardware*, pelo que é necessário eliminar os estados redundantes antes de se fazer a implementação da máquina de estados. Uma das técnicas que se pode utilizar para fazer a eliminação dos estados redundantes é a Técnica das Partições (Dias, 2010).

A Técnica das Partições parte do pressuposto que todos os estados são idênticos e que posteriormente se devem identificar as diferenças entre estados por forma a separá-los por grupos, denominados de Partições. Uma vez concluída esta separação em Partições, cada Partição deverá ser composta somente por estados equivalentes (Dias, 2010).

Esta técnica é composta essencialmente por quatro passos:

1. Separação inicial dos estados em Partições consoante os valores de saída;
2. Identificação das diferenças entre estados de uma mesma Partição, ou seja, identificar os estados que embora pertençam a uma mesma Partição transitam para estados de Partições diferentes;

3. Atualizar as Partições, ou seja, os estados identificados no passo anterior darão origem a uma nova Partição;
4. Repetição dos passos 2 e 3 até que não seja possível proceder a mais nenhuma alteração na tabela.

Aplicando a Técnica das Partições à **Tabela 8** obtêm-se os seguintes passos.

- A **Tabela 9** representa a aplicação do 1º passo. Como se pode verificar foram identificadas 3 Partições, a primeira Partição com saída a 0 e 1, composta pelos estados AB e E. A segunda partição com saída a 1 e 1, composta pelo estado B. Por fim a terceira Partição com saída a 0 e 0, composta pelo estado D.

| Estado Actual | Estados Seguintes | | |
|---------------|-------------------|-------|------------------------------|
| | X=0 | X=0 | |
| A | B / 0 | E / 1 | Partição 1 (P ₁) |
| C | E / 0 | A / 1 | |
| E | B / 0 | A / 1 | |
| B | B / 1 | C / 1 | Partição 2 (P ₂) |
| D | A / 0 | A / 0 | Partição 3 (P ₃) |

Tabela 9 - Aplicação do Primeiro passo da Técnica das Partições.

- Uma vez identificadas três Partições iniciais aplica-se o passo 2, do qual se obtém a **Tabela 10**. Como é possível verificar na Partição 1 foi identificado um estado que transita para Partições diferentes dos restantes estados. O Estado C transita em ambas as situações para a Partição 1, enquanto os estados A e E transitam para as Partições 2 e 1.
- Tendo identificado o estado C como um estado distinto dos restantes estados da mesma Partição, verifica-se que o estado C dá origem a uma nova Partição como se pode ver pela **Tabela 11**. É importante referir que todos estados que transitavam para o estado C devem ser atualizados igualmente com a designação da nova partição como se pode ver no estado B.

| Estado Actual | Estados Seguintes | | |
|----------------|-------------------|----------------|-----------------|
| | X=0 | X=0 | |
| A ₁ | B ₂ | E ₁ | ← Nova partição |
| C ₁ | E ₁ | A ₁ | |
| E ₁ | B ₂ | A ₁ | |
| B ₂ | B ₂ | C ₁ | |
| D ₃ | A ₁ | A ₁ | |

Tabela 10 - Aplicação do 2º passo da Técnica das Partições.

| Estado Actual | Estados Seguintes | |
|----------------|-------------------|----------------|
| | X=0 | X=0 |
| A ₁ | B ₂ | E ₁ |
| E ₁ | B ₂ | A ₁ |
| B ₂ | B ₂ | C ₄ |
| D ₃ | A ₁ | A ₁ |
| C ₄ | E ₁ | A ₁ |

← Nova partição

Tabela 11 - Aplicação do 3º passo da Técnica das Partições.

- Segue-se o passo 4, que consiste em repetir os passos 2 e 3 até que não se verifique mais nenhuma alteração na tabela.

Uma vez que, para esta tabela, não é possível proceder a mais nenhuma alteração resta criar a tabela resumida a partir da **Tabela 11**. A tabela resumida (**Tabela 12**) consiste numa tabela em que cada partição dá origem a um único estado equivalente, ou seja todos os estados de uma partição resumem-se a somente um estado saída (Dias, 2010). Os estados A e E darão origem ao estado α , o estado B dará origem a β , o D a γ e finalmente o C a δ .

| Estado Actual | Estados Seguintes | |
|---------------|-------------------|--------------|
| | X=0 | X=0 |
| α | $\beta / 0$ | $\alpha / 1$ |
| β | $\beta / 1$ | $\delta / 1$ |
| γ | $\alpha / 0$ | $\alpha / 0$ |
| δ | $\alpha / 0$ | $\alpha / 1$ |

Tabela 12 - Tabela resumida que obtida posteriormente aplicação da Técnica das Partições.

2. ESTADO DE ARTE

Esta dissertação tem como objetivo o desenvolvimento de duas ferramentas de *software*, para PC e para terminais móveis, com fins pedagógicos na área de Sistemas Digitais é relevante analisar em que área da Informática se insere este trabalho, assim como investigar o trabalho desenvolvido até ao momento dentro do mesmo âmbito.

As aplicações a desenvolver neste trabalho têm por objetivo oferecer ao seu utilizador uma forma de adquirir conhecimentos fazendo uso das tecnologias, neste caso particular dos computadores e dos terminais móveis. Levando em consideração os fins pedagógicos das aplicações a desenvolver e a forma como estas chegam ao utilizador final, torna-se possível inseri-las no âmbito do “*E-Learning*” (*Electronic Learning*).

2.1. E-LEARNING

E-Learning é um termo estrangeiro utilizado para se referir a “aprendizagem eletrónica” ou “formação à distância via Internet” (Cação & Dias, 2003). São várias as definições existentes para *E-Learning* sendo que estas mudam de autor para autor e algumas são até contraditórias, visto que alguns autores só consideram como fazendo parte deste modelo de ensino os métodos de aprendizagem que façam uso de redes que possibilitem comunicação de dados como a internet ou intranet (Bowles, 2004). Já outros autores consideram como *E-Learning* qualquer forma de aprendizagem feita por meio eletrónico, seja ele computador, *tablet* PC, telemóvel entre outros (Bowles, 2004).

Pode-se dizer que o *E-Learning* teve origem nos anos 60 na Universidade de Stanford onde os professores de Psicologia Patrick Suppes e Richard C. Atkinson testaram a utilização de computadores para ensinar matemática a crianças assim como ensiná-las a ler. Desta experiência surgiu mais tarde um programa educacional que viria a ser conhecido como *Stanford’s Educational Program for Gifted Youth*. Desde então, várias têm sido as definições aplicadas ao conceito de *E-Learning*. Nos anos 80 foi definido como “Aprendizagem utilizando qualquer meio eletrónico, especialmente computadores” (Bowles, 2004). Em 2003 no Reino Unido o Departamento para Educação e Capacidades afirmou no documento *Towards a Unified E-Learning Strategy* “Se alguém está a aprender de uma forma que utilize Tecnologias da Informação e Comunicação (ICTs) está a utilizar *E-Learning*” (Bowles, 2004). Contudo esta última afirmação não foi bem aceite, pois para alguns autores, como é o caso de Marc Rosenborg, para ser considerado *E-Learning* teria que ser um método de aprendizagem baseado numa rede, de modo a que fossem possíveis atualizações, armazenamento, distribuição e partilha de conteúdos de forma instantânea. Logo para Rosenberg teria que ser um método que chegasse ao utilizador final através da internet ou intranet (Bowles, 2004). Outra definição para *E-Learning* surge por parte de Ruth Clark no seu artigo “Six principles of Effective e-Learning: What works and Why”, para o autor *E-Learning* define-se como “conteúdo e métodos instrutivos fornecidos por um computador (seja por CD-ROM, pela internet ou intranet) com fim de oferecer conhecimento e capacidades relacionadas com objetivos individuais ou organizacionais” (Clark, 2002).

Todas as definições anteriormente referidas utilizam alguma tecnologia em particular como método a partir do qual é transmitido o conhecimento, o que de certa forma restringe tudo aquilo que realmente poderia ser visto como *E-Learning* e ainda exclui tecnologias emergentes que poderiam ser utilizadas para o mesmo fim (Bowles, 2004). Como tal para o âmbito deste projeto utilizar-se-á a definição dada por Marcus Bowles que define *E-Learning* como “experiência instrutiva que envolve a aquisição ou transferência de conhecimento

fornecida ou transmitida através de meios eletrônicos” (Bowles, 2004). Esta definição é mais ampla e faz questão de “abrir as portas” a todo o tipo de tecnologia que possa ser utilizado para o mesmo fim.

2.1.1. TIPOS DE E-LEARNING

O *E-Learning* é uma atividade educacional praticada por indivíduos ou grupos, trabalhando *online* ou *offline*, de forma síncrona ou assíncrona. Esta atividade pode ser realizada por via da internet ou isoladamente num computador ou qualquer outro dispositivo eletrônico (Naidu, 2006). Levando isto em consideração é possível diferenciar quatro formas distintas de *E-Learning*:

- **Online individual** – Neste caso os indivíduos aprendem fazendo uso de recursos disponibilizados via Internet ou Intranet.
- **Offline individual** – Para situações em que a aprendizagem é feita individualmente através de recursos educacionais não disponibilizados pela internet, como por exemplo CDs ou DVDs.
- **Em grupo de forma síncrona** – Neste caso um grupo de alunos trabalham conjuntamente em tempo real através da internet ou intranet, isto pode ser feito através de conferências (vídeo ou áudio) ou até mesmo *chat*.
- **Em grupo de forma assíncrona** – quando um grupo de alunos trabalho através de uma rede (internet ou intranet) mas em que as comunicações entre os participantes ocorre com atraso de tempo. Este tipo de *E-Learning* corresponde por exemplo a trocas de *E-mails*, fóruns, etc.

2.1.2. COMPOSIÇÃO DO E-LEARNING

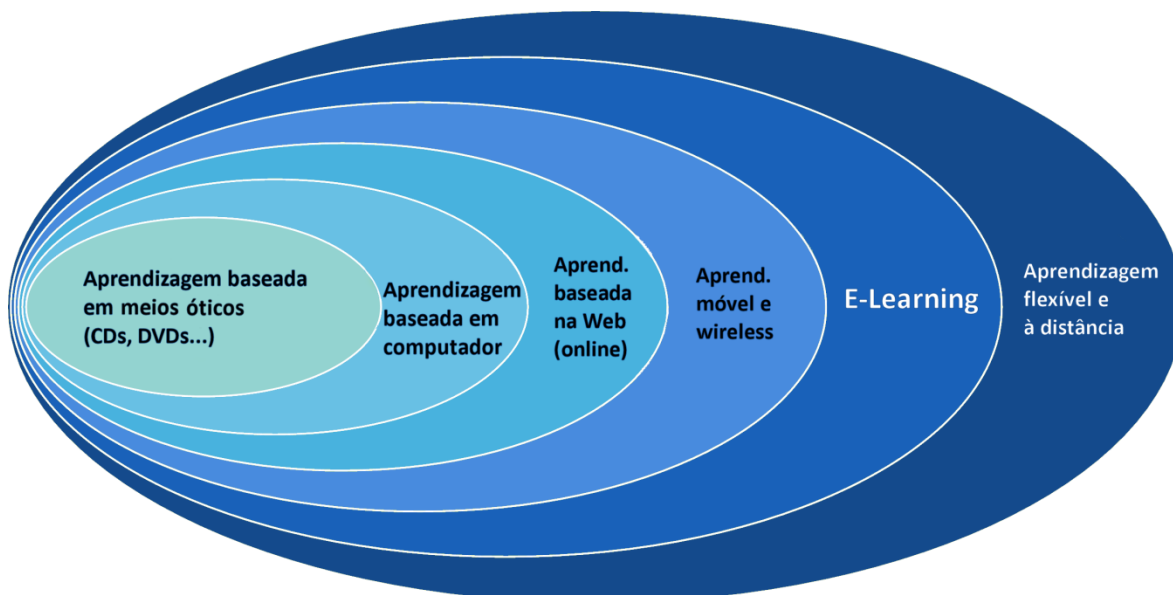


Fig. 20 - Tecnologias de Aprendizagem e suas relações (Bowles, 2004).

Através da Fig. 20 é possível verificar que conceitos engloba o *E-Learning* assim como onde este último se insere. Sendo assim é possível atestar que *E-Learning* não é mais de que um método de Aprendizagem à distância na qual são utilizadas as mais variadas tecnologias. O *E-Learning* inclui:

- **Aprendizagem baseada em meios óticos** – Os recursos educativos podem chegar ao utilizador final de várias formas, quando existe uma situação em que o *E-Learning* é feito de forma individual e *offline*, as médias óticas são uma das formas mais utilizadas de fazer chegar estes recursos educativos aos alunos;
- **Aprendizagem baseada no computador** – O computador permite a divulgação de conhecimentos das mais variadíssimas formas, desde a simples leitura de um documento educativo, troca de *E-mails* entre aluno-formador ou entre investigadores de uma mesma área, programas educativos, conferências, etc.;
- **Aprendizagem baseada na web** – Através da *World Wide Web (WWW)* torna-se possível a transferência de informação sem quaisquer restrições temporais ou locais. Deste modo as tecnologias *web* tornam-se numa ferramenta muito útil e potente para o *E-Learning*;
- **Aprendizagem móvel e wireless** – A tecnologia móvel tem evoluído de forma muito rápida nos últimos tempos como tal esta deve ser aproveitada para fins pedagógicos devido ao seu potencial e à forma como esta tecnologia atrai os mais jovens. Atualmente quase todos os equipamentos móveis estão equipadas com dispositivos *Wi-fi* ou rede móvel de terceira geração (3G) permitindo a transmissão de dados a alta velocidade o que oferece os vasto leque de opções para o *E-Learning*.

2.1.3. POSSÍVEIS FORMAS DE *E-LEARNING*

No *E-Learning* existem várias formas de fazer chegar o conhecimento aos formandos, todas elas com o intuito final de maximizar a capacidade de absorção de conhecimentos (Passos, 2010). Os recursos educativos podem então surgir das seguintes formas:

- **Tutoriais** – Estes apresentam o conhecimento de forma faseada e permitem que os alunos compreendam cada instrução antes de prosseguir para a próxima. Um bom tutorial de *E-Learning* deve permitir interação por parte do utilizador, assim como fornecer clarificação dos seus conteúdos.
- **Simulações** – Estas têm como fim a recriação de situações reais assim como situações cuja explicação seja mais complexa. Os simuladores devem conter exercícios práticos e fornecer *feedback* acerca do desempenho do utilizador.
- **Jogos Educativos** – Os jogos possibilitam uma forma mais divertida de adquirir conhecimento, como tal representam um dos modos mais atrativos para qualquer formando. Este método deve de certa forma recompensar os seus utilizadores pelos objetivos atingidos visto que cada objetivo corresponde a uma etapa do processo de aprendizagem.
- **Testes e Registos** – A realização de testes *online* pode ser utilizada tendo como finalidade a autoavaliação. Também é possível fornecer uma plataforma que avalie os testes enviando no fim o resultado ao tutor ou ao formando. Este tipo de funcionalidade estará associado normalmente a testes de múltipla escolha.

A combinação dos métodos anteriormente descritos para a elaboração de uma ferramenta que tenha como finalidade o *E-Learning* só beneficiará os formandos (Passos, 2010). Na ferramenta, a desenvolver neste projeto, relativamente aos Mapas de Karnaugh pretende-se fornecer um modo tutorial a partir do qual será

possível ao utilizador adquirir o conhecimento relativamente aos vários passos a executar na utilização dos mapas de Karnaugh para simplificar funções Algébricas Booleanas. A transação entre cada passo será interativa fornecendo uma descrição dos objetivos de cada etapa assim como esses objetivos são alcançados. A aplicação ainda contará com um modo que permitirá ao utilizador resolver os mapas passos a passo assim como resolução rápida fornecendo *feedback* acerca do seu desempenho combinando assim elementos de jogo, teste e simulação.

2.1.4. E-LEARNING 2.0

Com o evoluir da *Web*² e aparecimento da *Web 2.0*³ surgiu o *E-Learning 2.0*. Este último não é mais do que uma evolução do *E-Learning* usando como recursos um novo conjunto de ferramentas que surgiram com a *Web 2.0* (Bottentui & Coutinho, 2009). Com esta evolução tornou-se possível a comunicação entre formandos de partes opostas do globo, a consulta e partilha de conteúdos. A introdução destas novas ferramentas no *E-Learning* surge com o objetivo de eliminar qualquer barreira seja ela social, geográfica, cultural e fazer chegar o conhecimento a qualquer sítio, aumentando assim o prazer pela aprendizagem e aproximando os formando através da criação de comunidades virtuais (Bottentui & Coutinho, 2009).

A evolução para *E-Learning 2.0* surge com a introdução de novos instrumentos da *Web 2.0* como é o caso do *weblog*, com este instrumento tornou-se possível a atualização frequentes de conteúdos, estes permitem igualmente a troca de ideias, podem ser utilizados de forma individual e coletiva, e ainda têm a característica de serem muito simples de criar e atualizar (Bottentui & Coutinho, 2009). Um outro instrumento que passou a ser utilizado são os *wikis*, estes não são mais do que um sítio na *web* que tem como finalidade o trabalho coletivo, onde é permitido que qualquer pessoa altere ou acrescente conteúdos possibilitando assim a realização de trabalhos colaborativos e uma interação dinâmica entre colegas e professor (Bottentui & Coutinho, 2009). Muitos outros instrumentos foram introduzidos no *E-Learning* como é o caso das várias ferramentas disponibilizadas pelo *Google*, como o *Google Calender* que oferece a possibilidade de coordenar eventos entre várias pessoas, o *Google Docs* que permite a partilha de documentos entre comunidades os quais podem ser consultados a editados pelos membros consoante um conjunto de permissões definidos pelo tutor ou membro da comunidade com permissão para tal.

Com a *Web 2.0* surgiu uma vasta quantidade de instrumentos dos quais muitos foram aproveitados para potencializar o *E-Learning*.

2.1.5. VANTAGENS E DESVANTAGENS DO E-LEARNING

A utilização do *E-Learning* apresenta as seguintes vantagens:

- **Disponibilização online** – torna a aprendizagem flexível possibilitando ao formando aprender em qualquer sítio e a qualquer hora (Passos, 2010);
- **Interatividade** – Interação é sempre uma mais-valia para qualquer aplicação com fins pedagógicos pois aumenta a motivação de quem a utiliza (Passos, 2010);

² *Web* – Primeiro estado da *Word Wide Web*, onde as páginas *web* eram estáticas, não havia interatividade e não era aplicada a filosofia do *Open Source*, impedindo assim que os utilizadores pudessem criar as suas próprias modificações e expansões para as aplicações já existentes (Strickland, 2011).

³ *Web 2.0* – Evolução da *Web* com a introdução de aplicações que permitem a partilha de informação, colaboração, comunidades virtuais, etc.. Redes sociais, *blogs*, *wikis* entre outros são bons exemplos de *Web 2.0* (Musser & O'Reilly, 2006).

- **Disponibilidades da internet** – quando os conteúdos educativos são disponibilizados pela internet tornam-se acessíveis a todo mundo, uma vez que a internet já faz parte do quotidiano de grande parte da população mundial (Namahn, 2011);
- **Distribuição mundial sem custos** – Através das múltiplas tecnologias e ferramentas disponíveis de forma gratuita é possível partilhar conteúdos de sua autoria na Internet sem qualquer custo (Namahn, 2011);
- **Facilidades de atualização da informação** – quando existe necessidade de atualizar os conteúdos previamente fornecidos, estas alterações são feitas somente no servidor estando posteriormente disponíveis na internet em todo o mundo (Namahn, 2011);
- **Eficácia** - Uma formação personalizada tem normalmente uma maior probabilidade de retenção dos conteúdos, assim como a possibilidade de cada formando poder manter o seu próprio ritmo de aprendizagem o que contribui para uma melhor perceção (Cação & Dias, 2003).

Apesar da sua grande utilidade e dos benefícios já enumerados, algumas desvantagens também estão presentes no *E-Learning*, sendo estas:

- **Dependência da largura de banda** – Estando o *E-Learning* cada vez mais associado à internet, este pode depender muito dos limites da largura de banda principalmente quando os recursos estão na forma de vídeo ou áudio (Namahn, 2011).
- **Perca do contacto humano** – Começa a ser uma preocupação o facto de cada vez mais, existir uma grande focalização na utilização de tecnologias, chegando estas, por vezes, a substituir o contacto humano (Namahn, 2011).
- **Nem todas as matérias são facilmente apresentadas pelo *E-Learning*** – Algumas matérias requerem um toque humano para serem corretamente absorvidas, como por exemplo assuntos emocionais (Namahn, 2011).
- **Resistência à mudança** – Quando o assunto é a introdução de novas tecnologias, a resistência à mudança está sempre presente, pois qualquer mudança requer sempre um processo de adaptação (Namahn, 2011).
- **Sobrevalorização dos aspetos tecnológicos** – Por vezes os fatores técnicos podem-se sobrepor ao elemento pedagógico, ficando este último para 2º plano.

2.2. M-LEARNING

Os terminais móveis estão a demonstrar ser uma componente viável para fins Pedagógicos visto que oferecem a possibilidade de apoiar a aprendizagem convencional assim como potencializar o *E-Learning* (Traxler, 2005).

O *M-Learning* (*Mobile Learning*) pode ser definido como “ fonte educativa, onde a única tecnologia ou tecnologias dominantes são dispositivos portáteis” (Traxler, 2005). A partir desta definição conclui-se que no *M-Learning* incluem-se dispositivos como telemóveis, *smartphones*, *PDA*s, *tablets* PCs e até computadores portáteis (Traxler, 2005).

O *M-Learning* é igualmente considerada uma extensão do *E-Learning*, pois os seus objetivos são os mesmos havendo a especificação de serem utilizados os terminais móveis. Esta vertente do *E-Learning* teve origem nos avanços da computação móvel (telemóveis, *smartphones*, *PDA*s,...), das interfaces inteligentes e das comunicações sem fios (*Wi-Fi*, *Bluetooth*, *GPS*, *GSM*, *GPS*, *3G*) (Muyinda, 2007).

2.2.1. VANTAGENS E DESVANTAGENS DO *M-LEARNING*:

O *M-Learning* beneficia de alguns atributos próprios que proporcionam um novo conjunto de possibilidades (Park, 2011):

- **Portabilidade** – A utilização de terminais móveis permite a Aprendizagem em qualquer sítio, a qualquer horário, tornando assim a sua utilização muito flexível;
- **Pequena dimensão dos dispositivos** – Esta característica vem reforçar a portabilidade do *M-Learning*, pois a sua pequena dimensão faz com que sejam mais fáceis de transportar para qualquer lado;
- **Poder de computação** – A maioria dos terminais móveis recentemente lançados tem um grande poder de computação diminuindo assim as limitações do *M-Learning*;
- **Comunicações sem fios** – Atualmente estes dispositivos dispõem de uma variedade de ligações sem fios (*Wi-Fi*, *Bluetooth*, *GPS*, *GSM*, *GPS*, *3G*), fazendo com que seja possível estar *online* maioria do tempo;
- **Vasto leque de aplicações** – A variedade de aplicações atualmente disponíveis para estes dispositivos só vem aumentar o potencial do *M-Learning*.
- **Sincronização de dados com computador** – Torna a atualização de conteúdos muito mais acessível.
- **“Stylus pen” como forma de input** – Este método de input oferece a possibilidade de escrita à mão, desenho, tudo o que tradicionalmente era feito numa folha de papel.

Apesar do novo conjunto de possibilidades que o *M-Learning* proporcionou, existem também alguns problemas e desafios na sua utilização:

- **Dimensão do ecrã** – A maioria dos dispositivos móveis têm um ecrã de pequena dimensão o que obriga a um cuidado redobrado quando concebendo recursos a utilizar nesta modalidade de ensino, pois existe um limite de conteúdo a mostrar em cada ecrã e deve ser levado em consideração os mais diversos tamanhos e diferentes resoluções dos ecrãs dos dispositivos móveis.
- **Tempo de “vida” das baterias** – O facto de utilizar dispositivos móveis apresenta a vantagem de ser utilizado em qualquer sítio, mas contudo este facto vem associado ao facto de nem todos os sítios terem uma fonte de energia disponível, logo os dispositivos estão sujeitos à duração da bateria que em muitos casos se revela curta.
- **Resistência das organizações em implementarem esta modalidade de ensino** - A resistência à mudança é um fator que pesa sempre quando se consideram metodologias inovadoras.

2.3. TRABALHO RELACIONADO

O desenvolvimento de ferramentas informáticas com fins pedagógicos tem tido um crescimento considerável no decorrer das duas últimas décadas. Como tal já algum trabalho foi realizado relativamente às duas ferramentas base de Sistemas Digitais abordadas nesta Dissertação, os “Mapas de Karnaugh” e a “Técnica das Partições”. Ainda que algumas das aplicações que serão aqui apresentadas estejam mais focadas na apresentação de resultados e não na Pedagogia dos temas em questão, todas elas são importantes pois realçam diferentes componentes destas ferramentas de Sistemas Digitais.

2.3.1. APLICAÇÕES JÁ DESENVOLVIDAS NA ÁREA DE MAPAS DE KARNAUGH

Nesta secção serão enumeradas as aplicações dedicadas aos mapas de Karnaugh já existentes sendo feita uma referência aos aspetos nos quais estas se distinguem.

2.3.1.1. 4X4 KARNAUGH MAP MINIMIZER (4x4 KMM)

A abordagem aos Mapas de Karnaugh representada na Fig. 21 foi desenvolvida por Edans Sandes e permite obter a expressão booleana mínima através da introdução do valor de cada termo. A introdução por parte do utilizador do valor de cada termo pode ser feita no próprio mapa de Karnaugh ou através de duas colunas onde a coluna da esquerda corresponde a valor 1 para o termo e a coluna da direita corresponde a indiferente. Aos termos não selecionados, no mapa ou nas colunas, é-lhes atribuído o valor 0. Atualmente a ferramenta 4x4 Karnaugh Map Minimizer vai na versão 1.0 (Sandes, 2002).

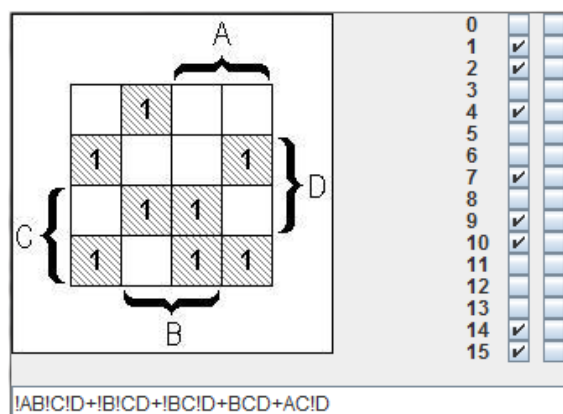


Fig. 21 - 4x4 Karnaugh Map Minimizer v1.0 (Sandes, 2002).

2.3.1.2. KARNAUGH APPLLET (KA)

A abordagem feita por Michael Keppler é um pouco mais completa que a anteriormente referido, pois esta conta com vários métodos de introduzir o valor dos termos dos mapas de Karnaugh. Esta ferramenta apresenta a representação gráfica do mapa assinalando de forma distinta cada implicante. Outra funcionalidade assinalável reside na possibilidade de obter mapas de 1 até 6 variáveis, ainda que para mapas de 6 variáveis com um elevado número de Implicantes a sua identificação torna-se complicado e a aplicação lenta. Como é possível verificar na Fig. 22, este software fornece ainda o esquema do circuito correspondente à solução. De referir que esta aplicação está disponível em forma de *applet*, fazendo com que esta só possa ser utilizada *online* (Keppler, 2010).

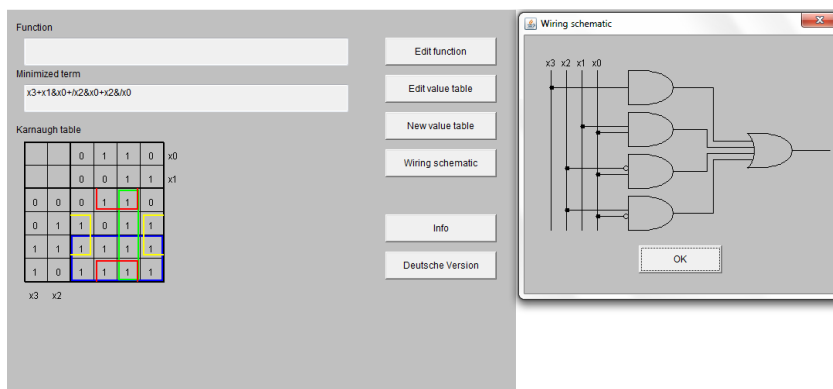


Fig. 22 - Karnaugh Applet (Keppler, 2010).

2.3.1.3. KARNAUGHMAP (KM)

A aplicação KarnaughMap corresponde à abordagem de Russel Sasamori de uma ferramenta que resolva mapas de Karnaugh. Tal como as restantes ferramentas analisadas até ao momento, a aplicação representada na **Fig. 23** limita-se a fornecer a solução mínima do Mapa Karnaugh. Este *software* encontra-se na versão 1.2 (Sasamori, 1999).

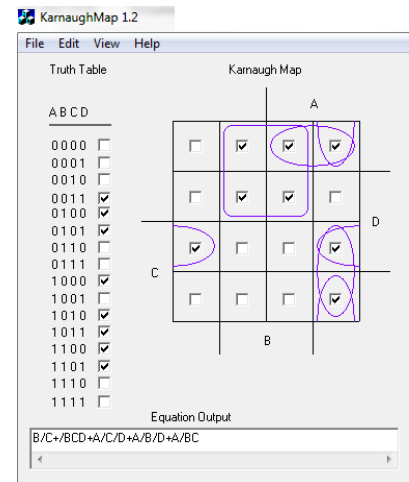


Fig. 23 - KarnaughMap (Sasamori, 1999).

2.3.1.4. KARNAUGH MAP MINIMIZER (KMM)

A abordagem de Robert Kovacevic tem como principais focos fornecer a solução do mapa de Karnaugh e a representação gráfica da solução, identificando distintamente cada Implicante/Implicado, ainda que esta identificação seja feita à vez, ou seja, apenas um Implicante/Implicado pode ser representado no mapa de cada vez, como se pode verificar na **Fig. 24**.

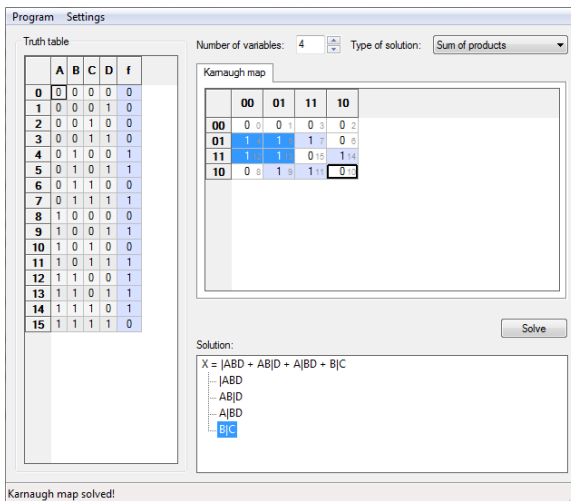


Fig. 24 - Karnaugh Map Minimizer (Kovacevic, 2005).

24. Relativamente às

abordagens anteriores introduz uma nova funcionalidade, a de permitir obter a solução do mapa através do método do Produto de Somas e também pelo método da Soma de Produtos. Esta aplicação encontra-se de momento na versão 0.4 (Kovacevic, 2005).

2.3.1.5. KARNAUGH MINIMIZER (KM2)

O Karnaugh Minimizer, representado na **Fig. 25**, é uma ferramenta desenvolvida pela Shuriksoft que oferece um vasto conjunto de funcionalidades. De referir que esta ferramenta está disponível de duas formas, uma versão *trial* gratuita e uma versão completa paga, tendo esta última um conjunto adicional de funcionalidades quando comparado com a primeira. Relativamente às ferramentas apresentadas anteriormente esta ferramenta introduz um novo conjunto de opções entre as quais a apresentação de um relatório com os dados do mapa e a hipótese de gravar e carregar mapas previamente guardados. Este *software* encontra-se de momento na versão 2.0 (Karnaugh Minimizer, 2008).

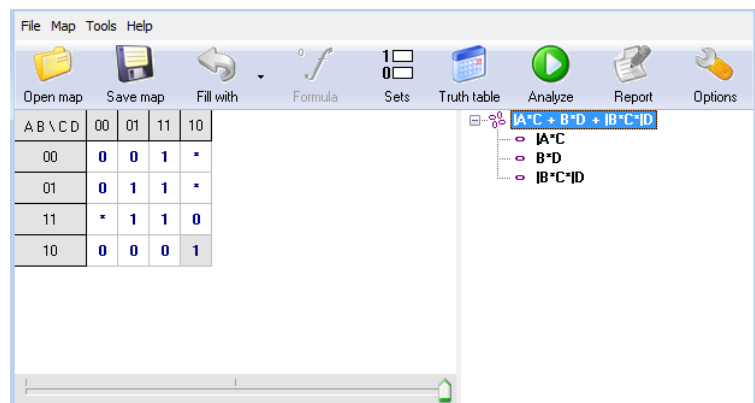


Fig. 25 - Karnaugh Minimizer (Karnaugh Minimizer, 2008).

2.3.1.6. EASYKARNAUGH (EK)

A abordagem feita aos Mapas de Karnaugh por Adriano Sena e Martha Torres representada na Fig. 26, destaca-se das restantes aplicações apresentadas até ao momento por estar direcionada para o ensino. Esta aplicação tem por objetivo ensinar o seu utilizador como resolver um mapa de Karnaugh. Levando em consideração o objetivo desta aplicação é possível afirmar que a sua conceção foi feita tendo como objetivo o *E-Learning* visto que a descrição feita pelos autores da aplicação é a seguinte "Uma ferramenta computacional para auxílio no ensino de Mapas de Karnaugh em Lógica Computacional". Esta aplicação tem quatro componentes principais, uma que apresenta a teoria dos Mapas de Karnaugh, outra que permite visualizar a solução apresentando um mapa com os Implicantes/Implicados devidamente assinalados, um terceiro componente no qual o utilizador pode tentar deduzir a expressão booleana correspondente à solução do mapa e o último componente no qual o utilizador tenta formar Implicantes/Implicados nos mapas. Atualmente esta aplicação está na versão 3.0 (Sena & Torres, 2009).

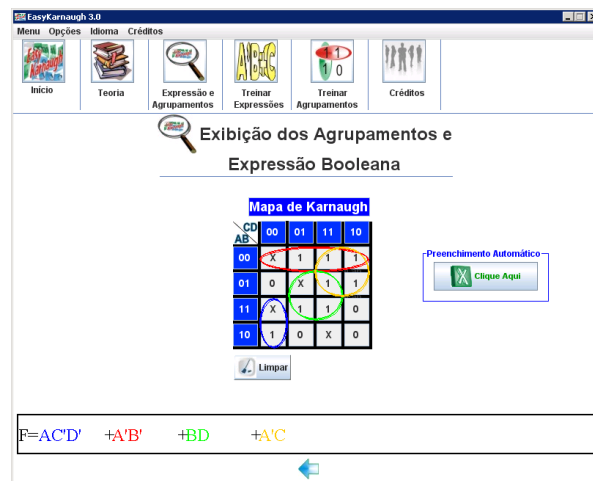


Fig. 26 - EasyKarnaugh (Sena & Torres, 2009).

2.3.1.7. KARMA

A ferramenta Karma desenvolvida por André Reis e Renato Ribas, é uma aplicação potente que de entre as suas funcionalidades dispõe de uma componente dedicada aos mapas de Karnaugh. Esta componente permite ao utilizador obter a solução de um mapa de Karnaugh incluindo dados básicos como a fórmula correspondente à solução mínima, o mapa devidamente preenchido, mas fornece também outros dados mais detalhados como os passos do método de Quine McCluskey e a respetiva tabela dos implicantes Primos. Ainda dentro da componente dedicada aos Mapas de Karnaugh a aplicação fornece ao utilizador um modo didático

onde é possível por exemplo verificar a adjacência em todo o mapa, formar Implicantes, dado um conjunto de mapas tentar identificar que Implicantes são primos, etc. (Reis & Ribas, 2011). Aquando da escrita desta dissertação a aplicação representada na Fig. 27 encontrava-se na versão 3.61.

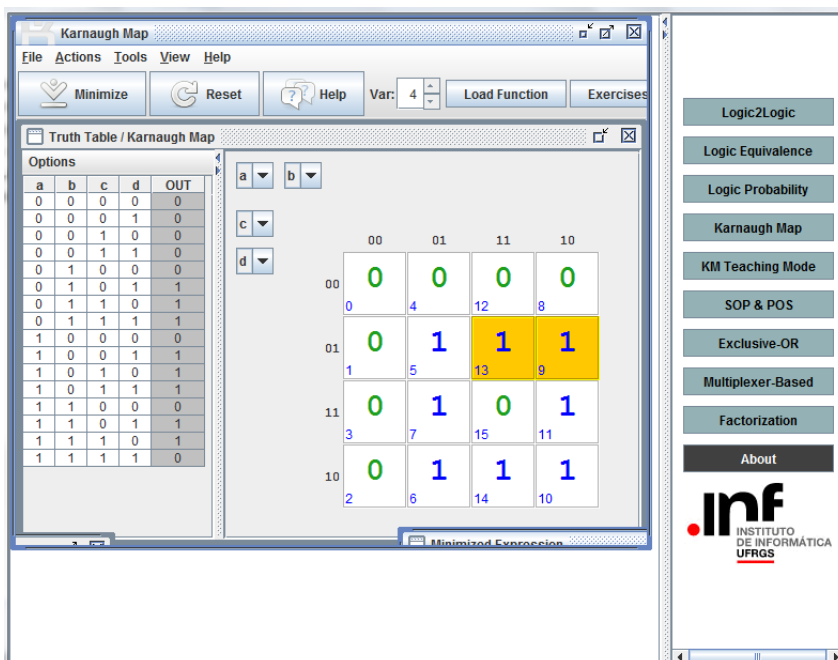


Fig. 27- Karma (Reis & Ribas, 2011).

2.3.1.8. BOOLE-DEUSTO

A aplicação Boole-Deusto foi desenvolvida na Universidade de Deusto em Espanha e tem como objetivo o desenho a análise de circuitos eletrónicos digitais simples. Esta aplicação inclui várias ferramentas de Sistemas Digitais entre as quais mapas de Karnaugh e Diagramas de Estados de máquinas de Moore e Mealy. Esta aplicação é ainda capaz de gerar código LDH (Linguagens de Descrição de Hardware) e criar ficheiros JEDEC (*Joint Electron Device Engineering Council*). É importante referir que esta aplicação se destina a servir de complemento no ensino de Sistemas Digitais tendo já sido galardoada como o prémio de melhor software no IV Congreso Iberoamericano de Tecnologías Aplicadas a la Enseñanza de la Electrónica (Zubía, 2003).

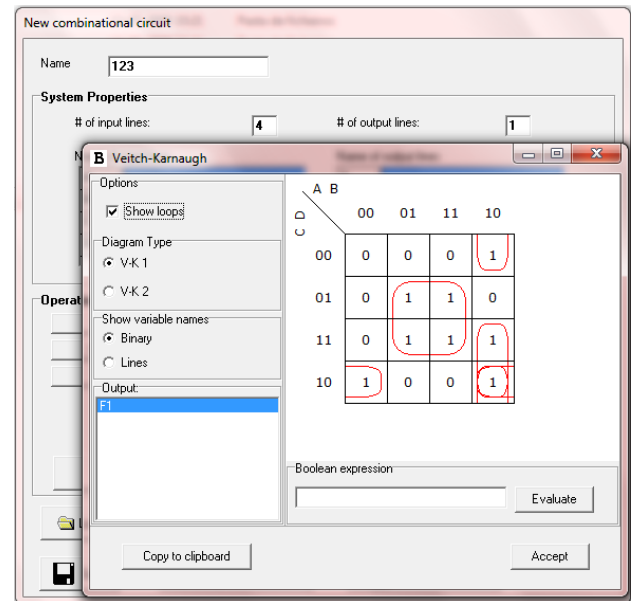


Fig. 28 - Boole-Deusto (Zubía, 2003).

2.3.1.9. KARNAUGH MAP OPTIMIZER (KMO)

Desenvolvida pela KoperSoft, o Karnaugh Map Optimizer é uma aplicação de Mapas de Karnaugh para *Iphone*. Esta aplicação exibida na Fig. 29, destaca-se das apresentadas até ao momento por ser destinada a um dispositivo móvel, o *Iphone*. Nesta ferramenta é possível obter a solução de uma Mapa de Karnaugh com a respetiva expressão e identificar os implicantes. De referir ainda que esta aplicação não é gratuita (Karnaugh Map Optimizer, 2011).

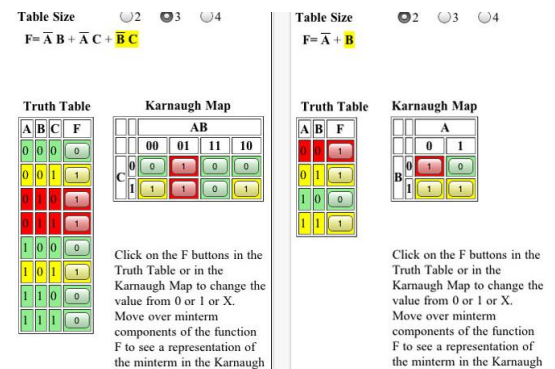


Fig. 29 - Karnaugh Map Optimizer (Karnaugh Map Optimizer, 2011).

2.3.1.10. POCKET KARNAUGH MAP (PKM)

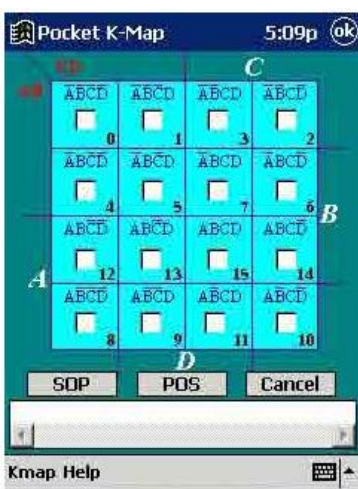


Fig. 30 - Pocket Karnaugh Map (Pocket Karnaugh Map, 2002).

Tal como a aplicação referida anteriormente, a aplicação retratada pela Fig. 30 também se destina a dispositivos móveis, nesta caso particular a dispositivos com o Sistema Operativo Windows Mobile. Esta ferramenta permite obter o resultado de um mapa de Karnaugh e a respetiva expressão. Atualmente esta aplicação encontra-se na versão 1.0 (Pocket Karnaugh Map, 2002).

2.3.1.11. KARNAUGH MAP ADVANCED (KMA)

Desenvolvida por Ivan Cigic esta aplicação para terminal móvel (*iPhone*) destaca-se das restantes, para o mesmo tipo de plataforma, por permitir resolver mapas até 8 variáveis. O *software* representado pela **Fig. 31** permite ainda a inserção da fórmula a simplificar por mapa de Karnaugh, tabela de verdade e expressão booleana. Atualmente esta aplicação encontra-se na versão 1.1 (Cigic, 2011).

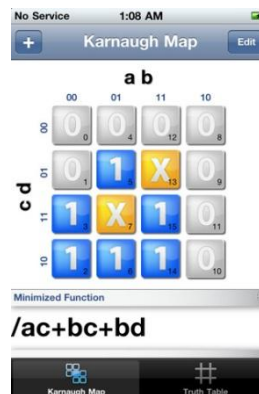


Fig. 31 - Karnaugh Map Advanced (Cigic, 2011).

2.3.1.12. VISÃO GERAL DAS FERRAMENTAS DE MAPAS DE KARNAUGH

A partir da **Tabela 13** é possível verificar que características e funcionalidades relativas aos mapas de Karnaugh estão presentes nas aplicações referidas anteriormente. É possível verificar que atualmente existem muitas mais aplicações disponíveis para PC do que para dispositivos móveis e que, das disponíveis para esta última plataforma nenhuma fornece suporte didático ou dispõe de exercícios para o utilizador desenvolver os seus conhecimentos nesta matéria. No que toca às aplicações disponíveis para PC apenas três contêm suporte didático, as aplicações Karma, EasyKarnaugh (EK) e Boole-Deusto, sendo a primeira uma ferramenta mais completa pois oferece uma maior quantidade de opções ao seu utilizador.

Num âmbito geral pode-se também afirmar que a aplicação mais completa é o Karnaugh Minimizer (KM2) principalmente na sua versão paga (KM2*), ainda que a versão gratuita também ofereça um vasto leque de funcionalidades.

A partir da análise à **Tabela 13**, verificou-se que as ferramentas mais relevantes no âmbito de Mapas de Karnaugh são o Karnaugh Minimizer (KM2), pelo seu vasto leque de funcionalidades, o Karma e o EasyKarnaugh pela oferta de componentes didáticas e exercícios, como tal estas aplicações servirão de referência para o trabalho a desenvolver nesta dissertação. Levando em consideração as aplicações de referências pretende-se que a aplicação a desenvolver disponibilize as funcionalidades mais relevantes das ferramentas analisadas, sendo estas:

- Identificação dos Implicantes/Implicados da solução (solução completa e um elemento de cada vez);
- Resolução mapas de Karnaugh com um máximo de 6 variáveis;
- Resolução de mapas de Karnaugh pelos métodos da Soma de Produtos e Produto de Somas;
- Mais do que um método de introdução da expressão a minimizar;
- Possibilidade de alterar a ordem das variáveis de entrada;
- Presença de uma componente didática;
- Geração automática de um relatório com os dados do Mapa de Karnaugh;
- Geração de Códigos *Verilog* e *VHDL* relativamente à solução encontrada;

| Características | PC | | | | | | | | | Telemóvel | | |
|---|---------|----|----|-----|-----|-----------------------------|----|-------|------------------|-----------|-----|-----|
| | 4x4 KMM | KA | KM | KMM | KM2 | KM2 (versão Completa) | EK | Karma | Boole- Deusto | KMO | PKM | KMA |
| Assinala os Implicantes/Implicados | - | X | X | X | X | X | X | X | X | X | X | X |
| Introdução pelo Mapa | X | X | X | X | X | X | X | X | X | X | X | X |
| Introdução por Tabela de Verdade | X | X | X | X | X | X | - | X | X | X | - | X |
| Introdução por expressão algébrica | - | X | - | - | - | X | - | X | X | - | - | - |
| Suporta termos de valor indiferente | X | - | X | - | X | X | X | X | X | X | X | X |
| Expressão simplificada | X | X | X | X | X | X | X | X | X | X | X | X |
| Número mínimo de variáveis | 4 | 1 | 4 | 1 | 4 | 1 | 3 | 2 | 1 | 1 | 4 | 1 |
| Número máximo de variáveis | 4 | 6 | 4 | 8 | 4 | 8 | 4 | 8 | - | 4 | 4 | 8 |
| Resolução SOP | X | X | X | - | X | X | X | X | X | X | X | X |
| Resolução POS | - | X | - | - | X | X | X | - | X | - | X | X |
| Opções de preenchimento rápido do mapa | - | - | - | - | X | X | - | - | X | - | - | - |
| Apagar todo o mapa | - | - | - | - | X | X | X | X | X | - | - | - |
| Alterar ordem das variáveis de entrada | - | X | - | - | - | - | - | X | X | - | - | - |
| Personalizar nomes das variáveis | - | - | - | - | X | X | - | X | - | - | - | - |
| Número de ordem dos Minitermos | - | - | - | X | X | X | - | X | - | - | - | - |
| Tabela dos implicantes (Quine McCluskey) | - | - | - | - | - | - | - | X | - | - | - | - |
| Guardar e abrir Mapas | - | - | - | - | X | X | - | X | X | - | - | - |
| Gerar relatório do Mapa | - | - | - | - | X | X | - | - | X | - | - | - |
| Gerar código VHDL | - | - | - | - | - | X | - | - | X | - | - | - |
| Gerar código Verilog | - | - | - | - | - | X | - | - | X | - | - | - |
| Exportar mapa em HTML | - | - | - | - | - | X | - | - | - | - | - | - |
| Gerar esquema elétrico | - | X | - | - | X | X | - | - | X | - | - | - |
| Utilizador pode resolver mapas de Karnaugh | - | - | - | - | - | - | X | - | X | - | - | - |
| Suporte didático | - | - | - | - | - | - | X | X | X | - | - | - |
| Contém Exercícios | - | - | - | - | - | - | X | X | X | - | - | - |
| Suporta mais que um idioma | - | X | - | X | X | X | X | X | X | - | - | - |

Tabela 13 - Características das Aplicações de Mapas de Karnaugh.

De modo tornar a ferramenta mais completa pretende-se ainda disponibilizar uma componente didática a partir da qual um utilizador possa adquirir conhecimentos acerca dos passos a executar na resolução dum mapa de Karnaugh, assim como a possibilidade executar esses mesmos passos. Pretende-se ainda que a aplicação seja capaz de disponibilizar mais do que uma solução para um mapa de Karnaugh e verificar todas as possíveis, visto que muitas vezes um mapa de Karnaugh pode ter mais do que uma solução mínima.

2.3.2. DESENVOLVIMENTO DE SOFTWARE NA ÁREA DA TÉCNICA DAS PARTIÇÕES

Até ao momento da escrita desta dissertação, não foi identificado qualquer *software* que abordasse esta ferramenta de Sistemas Digitais.

Dada a inexistência de aplicações nesta área, fica igualmente a faltar uma referência a seguir aquando do desenvolvimento de uma nova aplicação. Ainda assim o objetivo passa por criar uma aplicação Informática interessante que possa ser utilizada como ferramenta complementar na tarefa de lecionar funcionamento da Técnica das Partições.

2.4. CONCLUSÃO

As metodologias de ensino apresentadas neste capítulo aparecem num contexto de tornar a aprendizagem mais acessível, motivante e divertida apresentando um conjunto de benefícios que sobrepõem-se às suas limitações. Com a evolução e desenvolvimento destas áreas as suas limitações acabarão por ser contornadas, contribuindo para que o *E-Learning* e *M-Learning* sejam cada vez mais um complemento benéfico a conjugar com os métodos de ensino tradicionais.

No que toca ao trabalho já desenvolvido na área abordada nesta dissertação é possível verificar que, relativamente à ferramenta Mapas de Karnaugh, já existem diversas aplicações, mas que poucas têm como objetivo ensinar ao utilizador o seu funcionamento. Esta situação é igualmente verificada quando analisadas as aplicações disponíveis para dispositivos móveis, visto que estas têm por objetivo somente apresentar a solução de um mapa de Karnaugh sem indicar como se chega a tal solução.

Finalmente na área da Técnica das Partições não foi identificado qualquer *software* desenvolvido até ao momento, não existindo assim qualquer ponto de referência. Como tal o objetivo passa por criar uma aplicação capaz de satisfazer as necessidades existentes nesta área.

3. DESENVOLVIMENTO DO SOFTWARE

3.1. TECNOLOGIAS UTILIZADAS

Tendo em consideração que o objetivo desta tese consiste no desenvolvimento de duas aplicações capazes de auxiliar o ensino de ferramentas de Sistema Digitais e posteriormente a sua publicação na *internet*, os critérios para a escolha das tecnologias a utilizar incidiram nos seguintes pontos:

- As tecnologias utilizadas devem fazer parte do quotidiano da comunidade à qual se destinam, sendo que a escolha deve incidir sobre uma tecnologia que esteja disponível à maior percentagem possível de indivíduos desta comunidade;
- Constatando que trata de uma comunidade académica, as tecnologias utilizadas devem assentar no princípio do *software* livre. Desta forma as aplicações desenvolvidas não devem exigir a pré-instalação de qualquer *software* comercial.

Por forma a respeitar os critérios acima referidos as escolhas das tecnologias a utilizar foram as seguintes:

Aplicação PC – Para este formato a aplicação foi desenvolvida em *Java*. Atualmente esta tecnologia encontra-se na versão 6. A escolha recaiu sobre o *Java* pois oferece versatilidade, eficiência e portabilidade. Para que a aplicação desenvolvida possa funcionar num computador basta que este tenha instalado uma máquina virtual *Java*, mais conhecida como *Java Runtime Environment (JRE)*. De referir ainda que a utilização desta tecnologia é gratuita (ORACLE, 2012).

Aplicação Telemóvel – Para este formato a escolha da tecnologia a utilizar recaiu sobre o *Java Micro Edition (Java ME)*. Atualmente esta tecnologia encontra-se na versão 3.0.5 (ORACLE, 2012). O principal fator para a escolha desta tecnologia reside no facto de que atualmente, a nível mundial, 80% dos telemóveis suportam *Java ME* (Enough Software, 2011). Ainda assim, por forma a garantir uma maior disponibilidade da aplicação, esta foi adaptada a *Android*, devido ao elevado crescimento que esta tecnologia tem vindo a demonstrar nos últimos tempos. Atualmente o *Android* encontra-se na versão 4.0 (Google, 2012).

Página Web – Para a criação de uma página *web* foi escolhida a tecnologia *Joomla*. O *Joomla* é um Sistema de Gestão de Conteúdos (*CMS – Content Management System*) que permite a criação e gestão de sítios *web* dinâmicos. Foi escolhido um *CMS* para esta tarefa, visto que este permite a criação de uma página *web* de forma rápida e eficiente, com o extra de permitir criar um *web site* apelativo sem grandes esforços dado o vasto leque de extensões, distribuídas de forma gratuita, que o *Joomla* disponibiliza para criar funcionalidades específicas e personalizar os conteúdos (JoomlaPT, 2012).

3.1.1. TECNOLOGIAS UTILIZADAS NA IMPLEMENTAÇÃO

Para o desenvolvimento das aplicações o *IDE (Integrated Development Environment)* escolhido foi o Eclipse. Este *IDE* encontra-se atualmente na versão *Indigo (3.7.1)* (Eclipse Foundation, 2012). Para esta escolha contribuíram os factos de este ser altamente eficiente, ser compatível com todas as tecnologias referidas anteriormente e ainda ser de utilização gratuita, uma vez que também obedece ao princípio de *software* livre.

Desenvolvimento para PC – No desenvolvimento com destino ao PC, para além das tecnologias previamente descritas, foram ainda utilizadas as seguintes *APIs (Application Programming Interface)*:

- *iText* – reúne um conjunto de métodos que possibilitam a criação de ficheiros *PDF* (iText Software Corp, 2011);
- *JDOM* – reúne um conjunto de métodos para a escrita e leitura de ficheiros *XML* (Hunter & McLaughlin, 2012).

Desenvolvimento para telemóvel – No desenvolvimento para esta plataforma também foram utilizadas mais algumas *APIs*:

- *LWUIT – Lightweight User Interface Toolkit*. Esta *API* não é mais do que uma *framework* que tem como objetivo facilitar a criação de uma *Interface*, garantindo ainda que o aspeto da aplicação será independente do telemóvel onde esta seja executada, ou seja, garante um aspeto idêntico em qualquer telemóvel (Knudsen, 2008).
- *LWUITPortAndroid* – Esta *framework* permite a adaptação para *Android* de uma aplicação desenvolvida em *Java ME* que utilize *framework LWUIT* (Chirico, 2011).

Nota: Os detalhes de utilização das tecnologias referidas serão descritos nas secções seguintes.

3.2. REQUISITOS DAS APLICAÇÕES

Nesta secção serão apresentados os requisitos aos quais as aplicações devem obedecer por forma a cumprir com os objetivos pretendidos e de modo a que estas sejam intuitivas proporcionando uma fácil utilização.

Primeiramente serão apresentados os requisitos Funcionais, estes resumem-se a uma descrição daquilo que as aplicações devem ser capazes de fazer, seguidamente serão apresentados os requisitos não Funcionais, os quais se referem à eficácia e eficiência com que as aplicações devem cumprir os requisitos Funcionais (Kotonya & Sommerville, 1996). No que toca à ferramenta dedicada aos Mapas de Karnaugh, considerando que existem duas versões da ferramenta a desenvolver, ou seja, uma para PC e uma outra para terminal móvel, ambas as versões partilharão parte dos requisitos, mas por outro lado alguns requisitos serão específicos de cada versão, como tal a apresentação dos requisitos será feita da seguinte forma: em primeiro lugar os requisitos partilhados por ambas as versões e seguidamente os requisitos específicos a cada versão, isto para os requisitos Funcionais e Não Funcionais. Para a aplicação da Técnica das Partições os requisitos serão apresentados de forma normal, ou seja primeiro os Requisitos Funcionais seguidos pelos Requisitos Não Funcionais

3.2.1. REQUISITOS - KARNUMA

Seguem-se os requisitos da aplicação KarnUMa.

3.2.1.1. REQUISITOS FUNCIONAIS GERAIS (RFG):

RFG 1. O sistema deve permitir a escolha de entre dois idiomas, sendo estes o Português e o Inglês.

RFG 2. O sistema deve permitir a escolha do número de variáveis do mapa de Karnaugh.

RFG 3. O sistema deve permitir a escolha do método de resolução do mapa de Karnaugh, podendo ser Soma de Produtos ou Produto de Somas.

RFG 4. O sistema deve fornecer dois métodos de preenchimento do mapa de Karnaugh, sendo estes pelo próprio mapa ou alternativamente por uma tabela de Verdade.

RFG 5. O sistema deve permitir a escolha da ordem das variáveis de entrada para o mapa de Karnaugh;

- RFG 6.** O sistema deve permitir o preenchimento do mapa de Karnaugh por parte do utilizador ou automaticamente por parte do sistema de forma aleatória.
- RFG 7.** O sistema deve fornecer o mapa de Karnaugh e a tabela de verdade devidamente preenchidas por forma a verificar a sua correspondência, isto após o preenchimento do mapa de Karnaugh que deverá ser feito de acordo com RFG 4 e RFG 6.
- RFG 8.** O sistema deve permitir ao utilizador resolver o mapa de Karnaugh.
- RFG 9.** O sistema deve permitir obter a solução do mapa de Karnaugh.
- RFG 10.** O sistema deve, aquando da resolução do mapa de Karnaugh por parte do utilizador (RFG 8), fornecer *feedback* do estado da resolução.
- RFG 11.** O sistema deve, aquando da resolução do mapa de Karnaugh por parte do utilizador (RFG 8), validar cada passo da resolução.
- RFG 12.** O sistema deve, aquando da resolução do mapa de Karnaugh por parte do utilizador (RFG 8), identificar as possíveis falhas do utilizador neste processo, indicando as possíveis causas desta falha.
- RFG 13.** O sistema deve, aquando da apresentação da solução do mapa de Karnaugh (RFG 9), disponibilizar a fórmula correspondente à solução apresentada.
- RFG 14.** O sistema deve, aquando da apresentação da solução do mapa de Karnaugh (RFG 9), permitir a identificação de cada Implicante/Implicado no mapa de Karnaugh simultaneamente com os restantes Implicantes/Implicados ou exclusivamente, sendo que neste caso apenas o grupo pretendido será apresentado no mapa.
- RFG 15.** O sistema deve, aquando da apresentação da solução do mapa de Karnaugh (RFG9), fornecer sempre que se verifique, até 2 soluções para o mapa.
- RFG 16.** O sistema deve permitir iniciar um novo mapa de Karnaugh sempre que o utilizador assim o entenda.
- RFG 17.** O sistema deve permitir alterar a orientação das variáveis de entrada do Mapa de Karnaugh sempre que o utilizador assim o entenda.
- RFG 18.** O sistema deve permitir a alteração do método de resolução aquando da execução do RFG8.
- RFG 19.** O sistema deve permitir a alteração do idioma da aplicação a qualquer momento da sua execução.
- RFG 20.** O sistema deve permitir, a qualquer momento da sua execução, voltar ao estado anterior ou ao estado seguinte.
- RFG 21.** O sistema deve permitir o término da aplicação a qualquer momento, desde que o utilizador assim o entenda.

3.2.1.2. REQUISITOS FUNCIONAIS DA VERSÃO PC (RFPC)

- RFPC 1.** O sistema deve permitir escolher de entre 3 até 6 variáveis para os mapas de Karnaugh.
- RFPC 2.** O sistema deve permitir a observação da resolução do mapa de Karnaugh ao detalhe, indicando os vários passos necessários para se obter tal solução.
- RFPC 3.** O sistema deve fornecer a solução do mapa de Karnaugh expressa, ou seja, sem passos intermédios.
- RFPC 4.** O sistema deve permitir ao utilizador resolver o mapa de Karnaugh detalhadamente, ou seja num processo que engloba vários passos.
- RFPC 5.** O sistema deve permitir ao utilizador resolver o mapa de Karnaugh diretamente, ou seja numa única etapa na qual deverá atingir a solução final.
- RFPC 6.** O sistema deve permitir, aquando apresentação da solução do mapa de Karnaugh (RFG 8), a exportação de código fonte nas Linguagens de Descrição de Hardware *VHDL* e *Verilog*;
- RFPC 7.** O sistema deve permitir, aquando apresentação da solução do mapa de Karnaugh (RFG) a exportação de um relatório correspondente ao mapa de Karnaugh com todos os seus dados relevantes.

3.2.1.3. REQUISITOS FUNCIONAIS DA VERSÃO TELEMÓVEL (RFT)

RFT 1. O sistema deve permitir escolher de entre 3 e 4 variáveis para os mapas de Karnaugh.

3.2.1.4. REQUISITOS NÃO FUNCIONAIS GERAIS (RNFG):

RNFG 1. O sistema deve recordar os dados de utilizador.

RNFG 2. O sistema deve fornecer soluções corretas para os mapas de Karnaugh.

RNFG 3. O sistema deve estabelecer uma correta correspondência entre a tabela de verdade e o mapa de Karnaugh.

RNFG 4. O sistema deve ser intuitivo fornecendo uma *interface* de fácil utilização.

RNFG 5. O sistema deve ter um desempenho razoável não levando mais do que 5 segundos a efetuar cada operação.

3.2.1.5. REQUISITOS NÃO FUNCIONAIS DA VERSÃO PC (RNFGPC)

RNFGPC 1. O sistema deve correr em computadores de baixa performance.

RNFGPC 2. O sistema deve fornecer o código fonte nas linguagens *VHDL* e *Verilog* com correta sintaxe e semântica.

3.2.1.6. REQUISITOS NÃO FUNCIONAIS DA VERSÃO TELEMÓVEL (RNFT)

RNFT 1. O sistema deve garantir uma correta apresentação independente das dimensões do ecrã, desde que estas satisfaçam as dimensões mínimas exigidas.

RNFT 2. O sistema deve garantir portabilidade entre os dispositivos, desde que estes contenham os requisitos exigidos.

Nota: Garantir a portabilidade entre diferentes Telemóveis como referido nos requisitos Não Funcionais RNFT 1 e RNFT 2 implica que sejam feitos alguns “sacrifícios” no que toca ao desempenho (RNFG 5), visto que a portabilidade e o desempenho estão sempre relacionados, sendo que elevado desempenho normalmente implica uma baixa portabilidade e vice-versa.

3.2.2. REQUISITOS – PARTEK

Seguem-se os requisitos da aplicação Partek

3.2.2.1. REQUISITOS FUNCIONAIS

RF 1. O sistema deve permitir a escolha de entre dois idiomas, sendo estes o Português e o Inglês.

RF 2. O sistema deve permitir a escolha do número de estados, entradas e saídas de cada Tabela de Estados;

RF 3. O sistema deve permitir trabalhar com dois tipos de máquinas de estados, sendo estas as máquinas de Moore e máquinas de Mealy;

RF 4. O sistema deve permitir o preenchimento das Tabelas de Estados por parte do utilizador ou automaticamente por parte do sistema de forma aleatória.

RF 5. O sistema deve permitir obter a solução da aplicação da Técnica das Partições.

RF 6. O sistema deve, aquando da apresentação da solução (RF 5), identificar corretamente cada componente da solução apresentada.

- RF 7.** O sistema deve, aquando da apresentação da solução (RF 5), permitir a visualização detalhada dessa mesma solução.
- RF 8.** O sistema deve, aquando da apresentação da solução (RF 5), permitir a visualização dos diagramas de estados correspondentes às Tabelas de Estados apresentadas.
- RF 9.** O sistema deve permitir iniciar uma nova Tabela de Estados sempre que o utilizador assim o entenda.
- RF 10.** O sistema deve permitir a alteração do idioma da aplicação a qualquer momento da sua execução.
- RF 11.** O sistema deve permitir, a qualquer momento da sua execução, voltar ao estado anterior ou ao estado seguinte.
- RF 12.** O sistema deve permitir o término da aplicação a qualquer momento, desde que o utilizador assim o entenda.
- RF 13.** O sistema deve permitir a memorização de Tabelas de Estados.

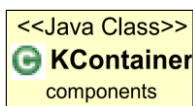
3.2.2.2. REQUISITOS NÃO FUNCIONAIS

- RNF 1.** O sistema deve recordar os dados de utilizador.
- RNF 2.** O sistema deve fornecer soluções corretas para a Técnica das Partições.
- RNF 3.** O sistema deve estabelecer uma correta correspondência entre as Tabelas de Estados e os Diagrama de Estados.
- RNF 4.** O sistema deve ser intuitivo fornecendo uma *interface* de fácil utilização.
- RNF 5.** O sistema deve ter um desempenho razoável não levando mais do que 5 segundos a efetuar cada operação.
- RNF 6.** O sistema deve correr em computadores de baixa performance.

3.3. MODELAÇÃO E IMPLEMENTAÇÃO

Para a apresentação da estrutura e características do *software* desenvolvido será utilizada a linguagem *UML (Unified Modeling Language)*, esta dispõe de diagramas padronizados os quais facilitarão a compreensão da estrutura por detrás das aplicações finais.

Um dos elementos da *UML* a utilizar será o Diagrama de Classes. Este Diagrama é capaz de representar a estrutura e relações existentes entre as várias classes criadas no desenvolvimento de uma aplicação. De referir que os Diagramas de classes que serão apresentados serão diagramas simplificados e seguem a seguinte notação:



Representa uma Classe em que *KContainer* é o nome da classe e *components* o pacote ao qual a classe pertence. Uma vez que se trata de um diagrama simplificado não serão apresentados nem os métodos nem os atributos de cada classe.



Representa uma Associação em que *0..1* indica a multiplicidade da associação e *Instance* o seu nome.



Representa uma Generalização, ou seja quando existem classes que derivam de uma outra classe, seguindo da classe derivada (subclasse) para a classe “mãe” (superclasse) existe uma Generalização.

A decisão de apresentar Diagramas de Classes simplificados deve-se ao facto de facultar um diagrama legível e perceptível, uma vez que facultando mais dados como os métodos e atributos de cada classe os diagramas tornar-se-iam demasiado complexos e confusos.

3.3.1. KARNUMA VERSÃO PC

Como principal característica do desenvolvimento desta versão do *software* KarnUMa destaca-se a implementação utilizando o paradigma da programação Orientada a Objetos. Este paradigma facilita a interpretação do “mundo real” do *software*, uma vez que o seu princípio consiste na composição e interação entre diversos elementos, elementos esses que são denominados de objetos.

A **Fig. 32** apresenta o Diagrama de Classes para a versão PC da aplicação KarnUMa. Neste diagrama pode-se observar que classes compõem o programa, e ainda as relações existentes entre as diversas classes. Como exemplo de uma relação refira-se a Associação denomina “*truthTable*” existente entre a classe *Formula* e a classe *Term*, esta Associação indica que um objeto do tipo *Formula* terá na sua composição 0 ou mais instâncias do tipo de objeto *Term*, esta multiplicidade é verificada pela indicação 0...*, ou seja o atributo *truthTable* (correspondente a uma tabela de verdade) da classe *Formula* será composto por 0 ou mais instâncias do tipo *Term*. Uma outra relação que pode ser observada no Diagrama de Classes da **Fig. 32**, é a de Generalização de *MapContainer* e *TableContainer* para *KContainer*, isto significa que ambas as classes *mapContainer* e *TableContainer* são Especializações da classe *KContainer* e que esta mesma classe é uma Generalização das classes mencionadas. Chama-se superclasse à classe da qual são feitas as Especializações, por sua vez estas últimas são denominadas subclasse. Esta relação indica que uma instância da subclasse é também uma instância da superclasse. Ainda sobre diagrama de Classes da **Fig. 32**, resta dizer que todas as classes que não têm qualquer tipo de relação representada não são objetos. Estas classes contêm um conjunto de funções estáticas que estão inseridas em cada uma das classes unicamente por uma questão de organização.

A descrição da estrutura do código fonte da aplicação KarnUMa na versão PC será feita apresentado cada pacote e respetivas classes fazendo uma alusão aos seus conteúdos:

QuineMcCluskey – Ambas as classes contidas neste pacote têm por base código fonte obtido em *Literate Programs* (LiteratePrograms, 2010). Ambas as classes foram alteradas de forma a cumprirem com os objetivos pretendidos para aplicação. Este pacote é composto pelas seguintes classes:

- **Formula** – Responsável por obter a fórmula booleana e simplificá-la através do método de Quine McCluskey. Para além do conteúdo inicial que permitia aplicação do método de Quine McCluskey foram acrescentados métodos por forma a colmatar mais algumas necessidades como a interpretação do conteúdo da fórmula inicial de forma a preencher o mapa de Karnaugh, fornecimento de até duas soluções quando possível, obtenção os grupos de termos correspondentes aos implicados ou implicantes, a verificação da solução obtida pelo utilizador;
- **Term** – Esta classe corresponde a cada termo de uma fórmula booleana. Inicialmente era simplesmente uma classe auxiliar, visto que cada fórmula, seja esta inicial ou mesmo a solução é composta por uma lista de termos. Posteriormente foram acrescentados métodos que permitem verificar se um termo em particular está presente num determinada fórmula, verificar que outros termos estariam contidos num termo simplificado, entre outros sempre com a finalidade de cumprir com os objetivos finais da aplicação.

components – Este pacote é composto por classes que correspondem a componentes gráficas da aplicação como por exemplo o desenho gráfico do mapa de Karnaugh e da tabela de verdade. Fazem parte deste pacote as seguintes classes:

- **KContainer** – Esta classe está na base do mapa de Karnaugh e da tabela de verdade visto que é superclasse de ambas, contendo vários atributos e métodos partilhados pelas classes dos elementos referidos, como por exemplo a leitura dos seus conteúdo para se obter a expressão inicial a simplificar;
- **MapContainer** – Corresponde à representação gráfica do mapa de Karnaugh, e é uma especialização de *KContainer*, sendo responsável por todas as representações gráficas dos mapas, como o seu preenchimento, assinalar implicantes/implicados, etc.;
- **TableContainer** – Representação gráfica da tabela de verdade, como acontece com *MapPanel*, esta classe é também uma especialização de *KContainer*;
- **TermLabel** – Corresponde a cada célula do mapa de Karnaugh, quando este não é interativo;
- **TermToggleButton** – Corresponde também a cada célula do mapa de Karnaugh, mas neste caso para as situações em que o mapa é interativo, como o preenchimento ou resolução, ambos por parte do utilizador.

userInterface – Este é o pacote responsável pelo desenho da *interface* gráfica, e por lidar com a interação com o utilizador. Este pacote é composto pelas seguintes classes:

- **UI** – Classe principal do programa, esta é a classe que determina o fluxo do programa consoante as escolhas do utilizador e apresenta os conteúdos respetivos de cada elemento da aplicação. É esta classe também que dá resposta a todas as interações do utilizador, com as exceções daquelas que são específicas do mapa de Karnaugh e da tabela de verdade como o seu preenchimento ou seleção de células.
- **UndoRedo** – Tem por objetivo o armazenamento das interações do utilizador de forma a permitir a execução das funcionalidades de “voltar atrás” e “repetir”;
- **UserDataXML** – Classe responsável pela escrita e leitura de um ficheiro *XML* com informações do utilizador que devem ser “memorizadas” para cada execução da aplicação. Esta classe faz uso da *API JDOM* para escrita e interpretação do ficheiro *XML*;
- **WindowsClassicThemeDetector** – Classe responsável por detetar, quando executado em ambiente Windows, se este se encontra a utilizar o tema Clássico por forma a lidar com algumas falhas gráficas geradas por este tema na aplicação.

languagePack – Pacote onde se encontram os idiomas disponíveis para a aplicação. Segue-se o seu conteúdo:

- **Languages** – Superclasse para qualquer idioma que possa ser introduzido na aplicação. Esta contém todas declarações dos métodos que contêm o texto, estando a cargo de cada classe de idioma implementá-los com o conteúdo no respetivo idioma;
- **English** – Conteúdo do idioma Inglês;
- **Portuguese** – Conteúdo do idioma Português.

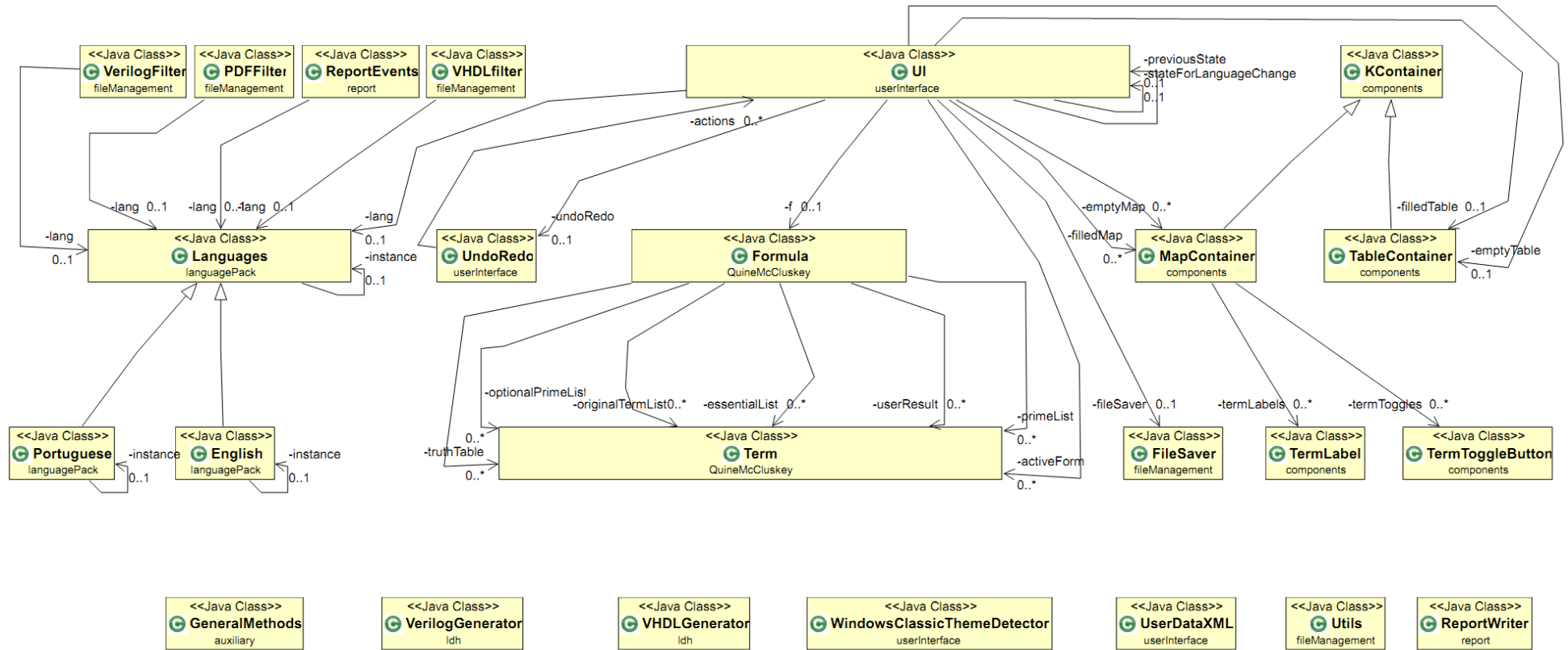


Fig. 32 - Diagrama de Classes da aplicação KarnUMA na versão PC.

Report – Este pacote contém um par de classes que têm como finalidade a escrita de um relatório em formato *PDF* com todo o conteúdo relevante de cada mapa de Karnaugh. Ambas as classes deste pacote fazem uso da *API iText*. Este pacote é composto por:

- **ReportWriter** – Obtém o conteúdo pretendido e cria o relatório em *PDF*;
- **ReportEvents** – Responsável por lidar com eventos repetidos ao longo da escrita do relatório em *PDF*, como a numeração das páginas e a escrita dos cabeçalhos.

ldh – Neste pacote encontram-se as classes responsáveis pela escrita de código fonte em *VHDL* e *Verilog*.

- **VerilogGenerator** – Constrói um ficheiro com código fonte *Verilog* da solução do mapa de Karnaugh que está a ser apresentada na aplicação;
- **VHDLGenerator** – Constrói um ficheiro com código fonte *VHDL* da solução do mapa de Karnaugh que está a ser apresentada na aplicação.

fileManagement – Este é o pacote responsável por exportar os ficheiros *PDF* do relatório e os ficheiros de código fonte *VHDL* e *Verilog*. Parte do conteúdo das classes neste pacote foi gerada tendo por base código fonte obtido nos tutoriais da Oracle (Oracle, 2012). A constituição deste pacote é formado por:

- **FileSaver** – Classe que tem como o objetivo a aparecimento da janela que permite gravar ficheiros;
- **PDFFilter** – Cria um filtro para lidar com os ficheiros do tipo *PDF* cuja extensão é *.pdf*;
- **VerilogFilter** – Cria um filtro para lidar com os ficheiros *Verilog* cuja extensão é *.v*;
- **VHDLFilter** – Cria um filtro para lidar com ficheiros *VHDL* cuja extensão é *.vhdl*;
- **Utils** – Classe auxiliar à exportação de ficheiros, tem por objetivo obter a extensão dos ficheiros.

auxiliary – Pacote auxiliar que contém um conjunto de métodos gerais a serem utilizados ao longo de todo o programa pelas vários pacotes.

- **GeneralMethos** – Classe onde estão implementados os vários métodos gerais.

3.3.2. POCKET KARNUMA VERSÃO MIDLET (JAVA ME)

Tal como no desenvolvimento da versão PC, o desenvolvimento do *software* na versão *Midlet* obedeceu ao paradigma da programação Orientada a Objetos.

Ainda que o desenvolvimento da *Midlet* tenha sido realizado com as *APIs* do *Java ME*, este mesmo desenvolvimento recorreu à utilização da *framework Lwuit*. Esta *framework* disponível para auxiliar a criação de aplicações *Java ME* permite a utilização dos mais variados componentes gráficos que dela fazem parte, muitos dos quais, não estão presentes por defeito nas *APIs* do *Java ME*. Todos os componentes desta *framework* podem ainda ser personalizados através de uma ferramenta disponível para tal efeito, denominada “*Resource Editor*” (Knudsen, 2008).

Como característica específica desta versão destaca-se a implementação de várias classes recorrendo ao padrão de Desenho de *software Singleton*, com o objetivo de diminuir a quantidade de memória utilizada. Esta característica vem assegurar um correto funcionamento mesmo em telemóveis que dedicam pouca memória às aplicações do tipo *Midlet*. A utilização do padrão de desenho *Singleton*, ao conceber uma classe, estabelece que apenas uma instância dessa classe será criada ao longo da execução do programa, garantindo assim uma gestão mais eficiente da memória.

Outro cuidado específico da implementação desta versão residiu em utilizar dimensões relativas para todos os componentes gráficos criados. Esta medida assegura que a aplicação terá um aspeto semelhante independente das dimensões dos ecrãs dos dispositivos móveis. Ainda relacionado com as dimensões do ecrã, é importante referir que a aplicação é capaz de lidar com ecrãs de configuração vertical e horizontal, visto que a disposição dos elementos gráficos varia consoante a configuração é detetada.

Na **Fig. 33** está representado o Diagrama de Classes para a aplicação KarnUMa na versão *Midlet*. Olhando para o diagrama é possível identificar um conjunto de classes idênticas ao diagrama da versão PC, isto deve-se ao aproveitamento que foi feito ao nível de código fonte, com a exceção da classe principal *UI* que apesar de ter o mesmo objetivo, a sua implementação foi realizada de raiz visto se tratarem de plataformas diferentes. Olhando para o diagrama da **Fig. 33** podem-se identificar todas as relações de Associação e Generalização existentes nesta versão da implementação como por exemplo a Associação “*UndoRedo*”, na qual é possível verificar que a classe *UI* pode conter até uma instância do tipo de objeto *UndoRedo*. Uma outra relação a destacar é a de Generalização das classes *Portuguese* e *English* para *Languages*, que indica que *Languages* será superclasse de *English* e *Portuguese*, e estas suas subclasses. Pode-se então dizer que *English* e *Portuguese* derivam de *Languages*, sendo que uma instância de *English* ou *Portuguese* será também instância de *Languages*. Ainda ao analisar o diagrama da **Fig. 33** podem-se identificar as várias classes em que foi seguido o padrão de Desenho *Singleton*. Olhando para as classes *Languages*, *Portuguese* e *English* pode-se identificar a presença deste padrão ao verificar que cada uma destas classes contém uma associação com a própria classe de multiplicidade 0...1 denominada *Instance*, o que indica que apenas uma instância destes objetos será criada durante a execução da aplicação. Ainda relativamente ao padrão de desenho *Singleton* é possível identificar no diagrama da **Fig. 33** uma derivação deste padrão nas classes *TableContainer* e *MapContainer*, visto que estas também limitam a quantidade de instâncias de cada uma. Isto pode ser observado por exemplo na classe *TableContainer* através das associações *emptyTableInstance* e *filledTableInstance*, ambas de multiplicidade 0...1.

O código fonte da aplicação Pocket KarnUMa (versão *Midlet*) encontra-se estruturado de forma semelhante ao da aplicação KarnUMa na versão PC. Contudo existem algumas diferenças que serão seguidamente apresentadas. A descrição da estrutura do código fonte desta versão fará referência somente às situações em a estrutura é diferente à versão PC. Como tal deve-se assumir que os pacotes não referidos têm o mesmo conteúdo da versão descrita anteriormente.

componentes – Pacote de conteúdo semelhante ao da versão anteriormente referida. Contém as mesmas classes da versão congénere, tendo sido adicionada a classe:

- **GroupPainter** – Classe destinada definir a “pintura” das células dos mapas de Karnaugh para assinalar os Implicantes/Implicados. A criação desta classe deve-se às diferenças de implementação entre ambas as versões até agora mencionadas, enquanto na versão PC a “pintura” era definida nos componentes, na versão *Midlet* foi criada uma classe para este fim, sendo os seus métodos posteriormente utilizados nos componentes onde é suposto efetuar esta “pintura”.

userInterface - o conteúdo deste pacote é semelhante ao que acontece na versão PC, contudo a sua implementação é distinta visto que tratam-se de duas plataformas diferentes. Relativamente ao pacote da versão PC apenas partilha as classes *UI* e *UndoRedo*, sendo que ambas partilham as mesmas funcionalidades da versão PC. Seguem-se as classes específicas desta versão:

- **Uclone** – Classe que contém uma cópia dos objetos relevantes para as funcionalidade de “Voltar atrás” e “Repetir”. Tornou-se necessária criação desta classe uma vez que nesta versão do *Java* não

está disponível o método de colonizar objetos, sendo então necessário criar uma classe que guarda os objetos necessários às funcionalidades já referidas;

- **UserData** – Iguamente ao que acontecia com a classe *UserDataXML* da versão PC, esta classe tem por objetivo guardar os dados de utilizador relevantes. Mas neste caso não é utilizado um ficheiro XML para este fim, mas sim, a memória do telemóvel destinada às aplicações. Recorrendo a métodos específicos do *Java ME* são guardados e carregados os dados de utilizador na memória do telemóvel.

com.sun.lwuit – Este é um pacote específico da *framework Lwuit* utilizada nesta versão da aplicação. Foram introduzidas novas classes neste pacote com o objetivo de criar especializações de classes já nelas definidas. Estas especializações consistem em objetos da *framework* utilizada, aos quais pretendia-se adicionar novas funcionalidades. As classes implementadas neste pacote foram as seguintes:

- **MyDialog e MyForm** – Especialização dos objetos *Dialog* e *Form* da *framework Lwuit*. Estas classes consistem numa implementação com características adicionais às classes originais. A característica mais relevante está na implementação destas classes com o padrão de desenho *Singleton*.
- **ProgressBar** – Esta classe consiste na implementação gráfica de uma barra de progresso.

Auxiliary – Como acontece na versão congénere, este pacote está munido de métodos auxiliares que são utilizados pelos vários pacotes ao longo da execução do programa. Foram ainda adicionados alguns métodos que fazem parte das *APIs* do *Java*, mas que não faziam parte das *APIs* do *Java ME*. Este pacote é constituído pelas classes:

- **GeneralMethods** – Contém métodos auxiliares e ainda alguns métodos copiados do *Java*;
- **Arrays** – Contém a implementação do método “*equals*” para *Arrays* do *Java*;
- **InputCheck** – Esta classe tem por objetivo identificar a origem da interação por parte do utilizador. Basicamente faz a distinção entre dois tipos de objetos responsáveis por interagir com o utilizador e fornece esta informação à classe responsável por lidar com a interação.

De referir ainda que esta versão partilha dos pacotes **QuineMcCluskey** e **langaugePack** com a versão PC. Estes pacotes são semelhantes em parte do conteúdo, mas contêm algumas diferenças relacionadas com o facto de serem implementações de diferentes plataformas e também relacionadas com algumas funcionalidades exclusivas da versão PC. Os restantes pacotes da versão PC não mencionados são exclusivos dessa versão visto que estão relacionados com funcionalidades exclusivas.

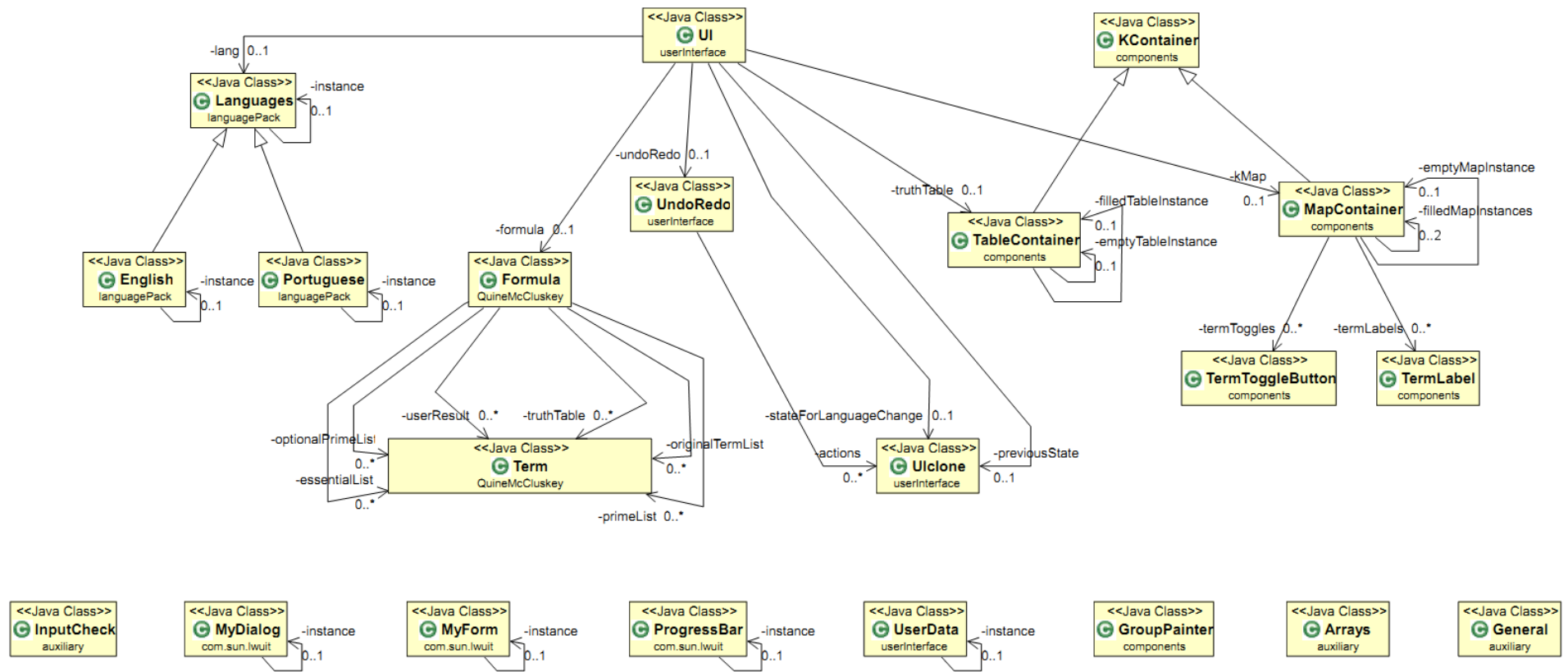


Fig. 33 - Diagrama de Classes da aplicação Pocket KarnUMA (versão Midlet).

3.3.3. POCKET KARNUMA VERSÃO ANDROID (ANDROID PACKAGE)

Esta versão do *software* é em tudo semelhante à versão *Midlet*, apenas foi necessária a utilização da *framework LWUITPortAndroid*, que tornou possível o aproveitamento do código fonte desenvolvido para a *Midlet*, para a criação de um *Android Package*. Sendo assim a *framework LWUITPortAndroid* tem como tarefa a interpretação das bibliotecas provenientes da *framework Lwuit*, de modo a que estas sejam corretamente executadas na Aplicação *Android* (Chirico, 2011). O restante código fonte não requer qualquer tipo de interpretação, uma vez que o código fonte para a criação de um *Android Package* é também baseado em *Java*.

Como foi referido o código fonte desenvolvido para esta versão da aplicação KarnUMa é semelhante à versão *Midlet*, contendo somente pequenas diferenças relacionados com a apresentação, como tamanhos de tipo de letra. Comparando à aplicação na versão *Midlet*, a estrutura desta versão apenas se distingue da estrutura da *Midlet* pela introdução de uma novo pacote com a classe responsável por utilizar *framework LWUITPortAndroid*. O novo pacote introduzido nesta versão foi:

com.activity – Responsável por utilizar a *framework LWUITPortAndroid*. Este pacote é constituído pela classe:

- **KarnUMaActivity** – Inicia a execução do código fonte da *Midlet* interpretado pela *framework* já mencionada.

3.3.4. PARTEC

Esta aplicação foi desenvolvida exclusivamente para PC. Tal como aconteceu com todas as aplicações referidas até ao momento, o desenvolvimento deste *software* também obedeceu ao paradigma da programação Orientada aos Objetos.

A **Fig. 34** representa o Diagrama de Classes desta aplicação. Pode-se verificar pelo diagrama que o desenvolvimento desta aplicação seguiu a mesma estrutura das aplicações até aqui apresentadas, o que é visível pela presença de classes como *UI*, *UIClone*, *UndoRedo*, *Languages* e suas subclasses.

Analisando o Diagrama pode-se observar a relação de Generalização existente das classes *TableToFill*, *InteractionTable* e *FinalTable* para *TableContainer*. Isto acontece porque ainda que cada uma das subclasses represente um tipo de tabela distinto, todas elas obedecem ao mesmo padrão para o desenhar de uma Tabela.

Segue-se a descrição da estrutura do código fonte da aplicação ParTec:

partitionsTechniqueAlgorithm – Este pacote contém um total de 3 classes que têm por objetivo a execução de um algoritmo que aplica a Técnica das Partições.

- **Iterator** – Considerando os vários passos da Técnica das Partições como Iterações, esta classe executa e guarda os dados das iterações necessárias para completar o processo de eliminação de estados redundantes pela Técnica das partições.
- **Partition** – Corresponde a cada partição criada com a execução da Técnica das Partições. Faz parte deste objeto uma lista dos estados pertencentes a esta partição assim como um conjunto de métodos para lidar com os mesmos, como por exemplo um método para identificar qual o estado ou estados redundantes dentro desta partição.

- **State** – Este objeto corresponde a um estado da Técnica das Partições, contém elementos como os estados seguintes, valores de saída, partição a qual o estado pertence entre outros.

components – Este pacote é composto por classes que correspondem a componentes gráficas da aplicação como por exemplo o desenho gráfico das tabelas de estados. Fazem parte deste pacote as seguintes classes:

- **Arrow** – Desenha uma seta que é utilizada para assinalar linhas nas tabelas.
- **TableContainer** – Corresponde representação gráfica das tabelas de estados. Contém os métodos que permitem desenhar as tabelas de estados. Todos os tipos de tabelas de estados recorrem a este objecto.
- **TableToFill** – É uma especialização de *TableContainer* e corresponde à tabela de estados que pode ser preenchida pelo utilizador. Fazem parte dos seus métodos por ex. métodos para extrair os conteúdos preenchidos na tabela.
- **MealyBeginTable** – É uma especialização de *TableToFill*, e corresponde à especificação tabela de estados de uma máquina de *Mealy*. Conta com os mesmos métodos de *TableToFill*, mas desta feita especificados para este tipo de tabela.
- **MooreBeginTable** – Corresponde a uma situação idêntica a *MealyBeginTable*, mas neste caso para uma tabela de estados de uma máquina de Moore.
- **IterationTable** – Consiste numa especialização de *TableContainer* para apresentar cada iteração da Técnica das Partições.
- **FinalTable** – É um especialização de *TableContainer* para representar tabelas iniciais ou tabelas simplificadas (tabelas já preenchidas).
- **StateCell** – Corresponde a uma célula de estado seguinte nas tabelas de estados. Nestas células o utilizador pode escolher o estado bem como o valor de saída.
- **StatePartitionCell** – Corresponde à representação de uma célula na qual é indicada a partição a que um estado pertence. A característica principal desta classe consiste em representar o índice da partição numa posição “inferior à linha”.
- **PartitionsTable** – Consiste na representação de uma tabela que indica a correspondência entre a tabela inicial e a tabela simplificado.

stateDiagram – Este é o pacote responsável pelo desenho dos diagramas de Estados. Fazem parte deste pacote as seguintes classes:

- **Arrows** – Desenha as setas que representam as transições de estado.
- **StateCircle** – Desenha os círculos correspondentes a cada estado.
- **Diagram** – Corresponde ao diagrama no seu todo, fazendo uso das classes *Arrows* e *StateCircle* desenha todo o diagrama de estados.

Resta referir que os pacotes **auxiliar**, **languagePack** e **userInterface** são semelhantes aos existentes nas aplicações referidas anteriormente, de modo que desempenham os mesmos papéis no desenvolvimento da aplicação.

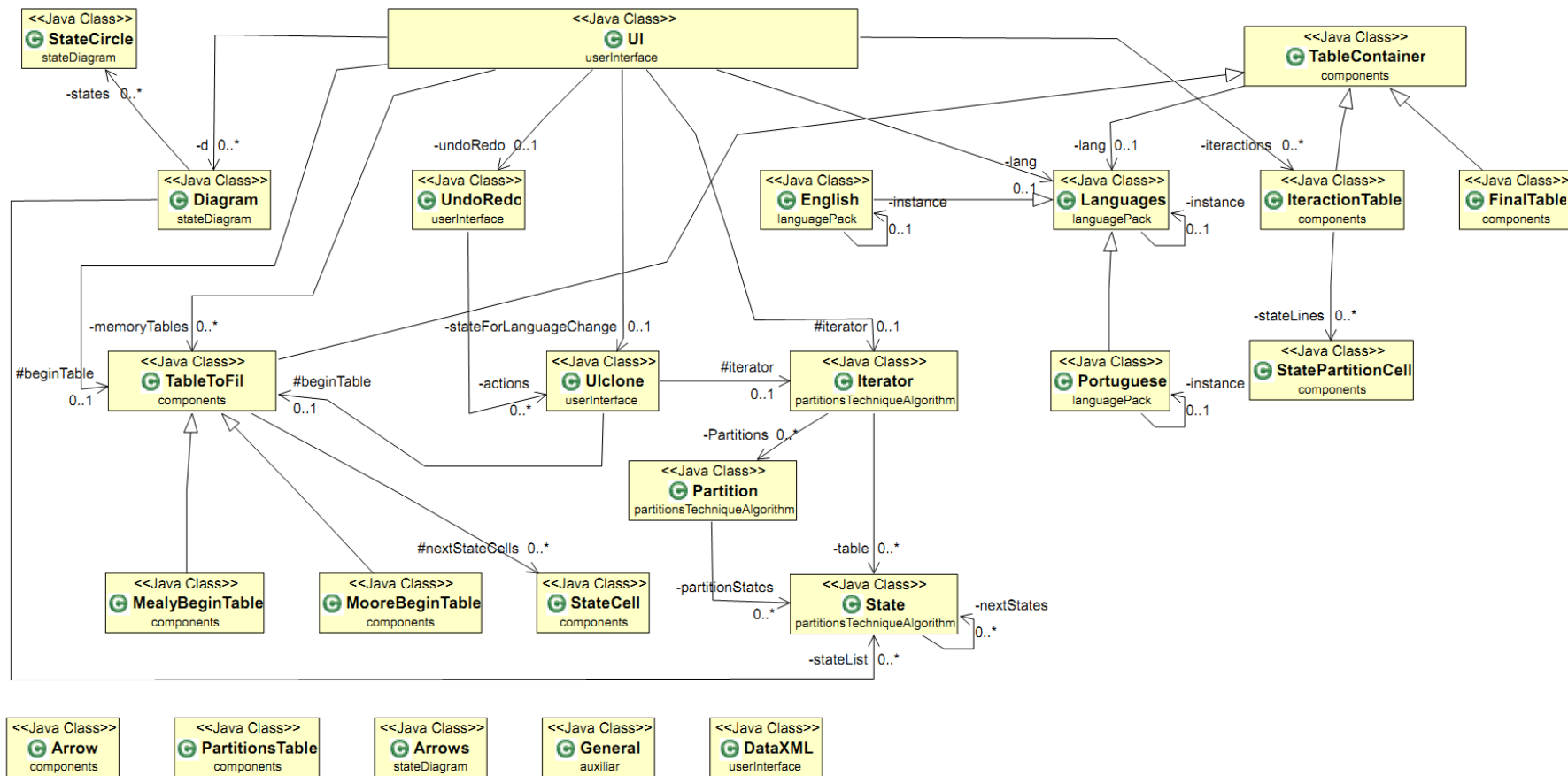


Fig. 34 - Diagrama de Classes da aplicação ParTec.

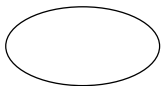
3.4. ANÁLISE DE TAREFAS

A análise de tarefas será feita através de diagramas de Casos de Utilização, estes diagramas são um elemento do *UML* que representa as diferentes tarefas que os utilizadores do sistema podem efetuar. Com os diagramas de Casos de Utilização constatar-se-á que tarefas podem ser realizadas por cada utilizador de sistema.

A notação utilizada nestes diagramas é a seguinte:



Ator – Representa os atores do sistema;



Tarefa – Representa as ações de podem ser executadas sobre o sistema;



Associação – Reproduz uma associação entre o utilizador e o sistema;



Extends – Significa uma tarefa que está implícita através de outra.

Seguidamente nas **Fig. 35**, **Fig. 36** e **Fig. 37** estão representados os diagramas de Casos de Utilização para as aplicações desenvolvidas.

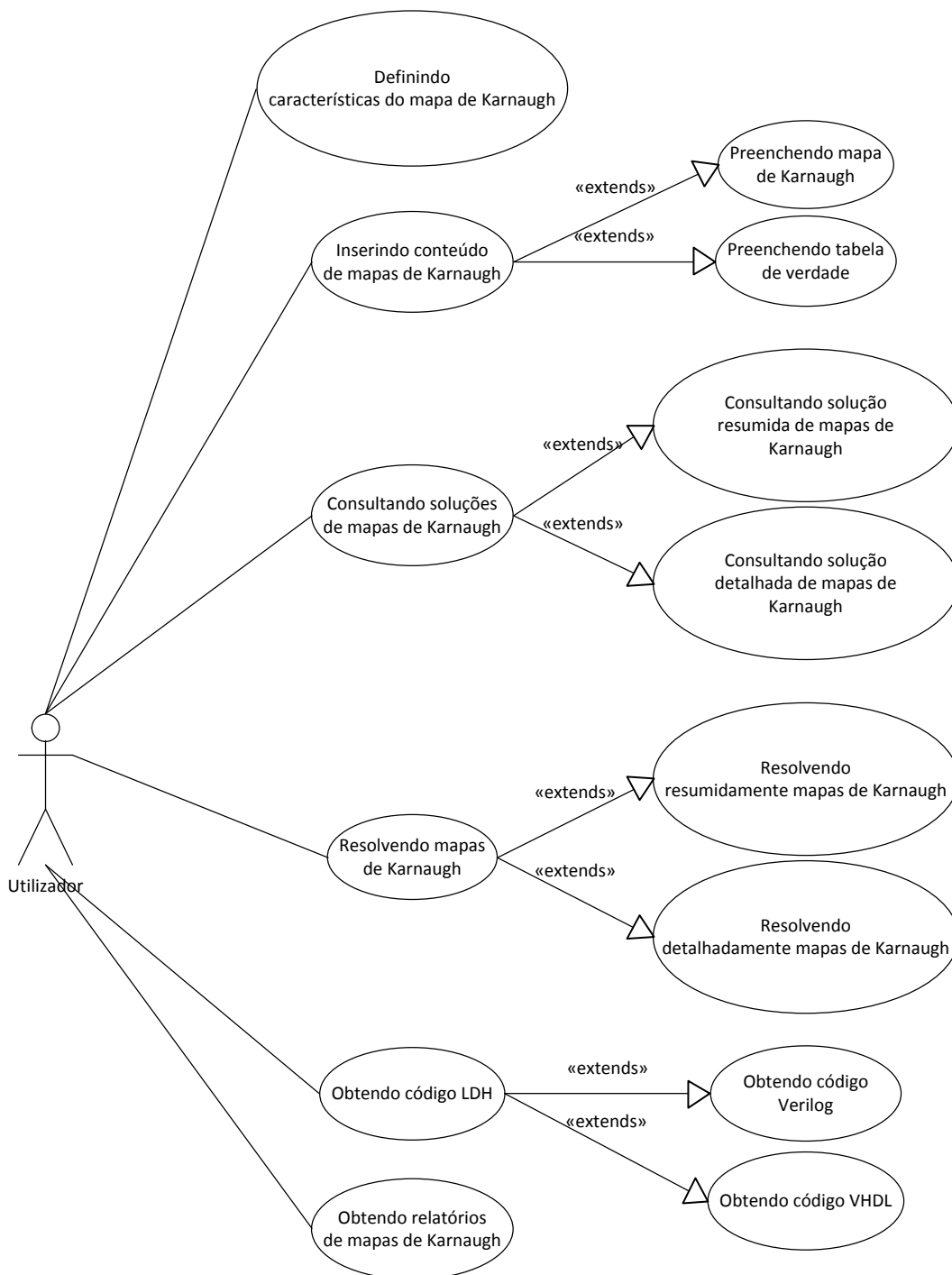


Fig. 35 - Diagrama de Casos de Utilização para KarnUMA na versão PC.

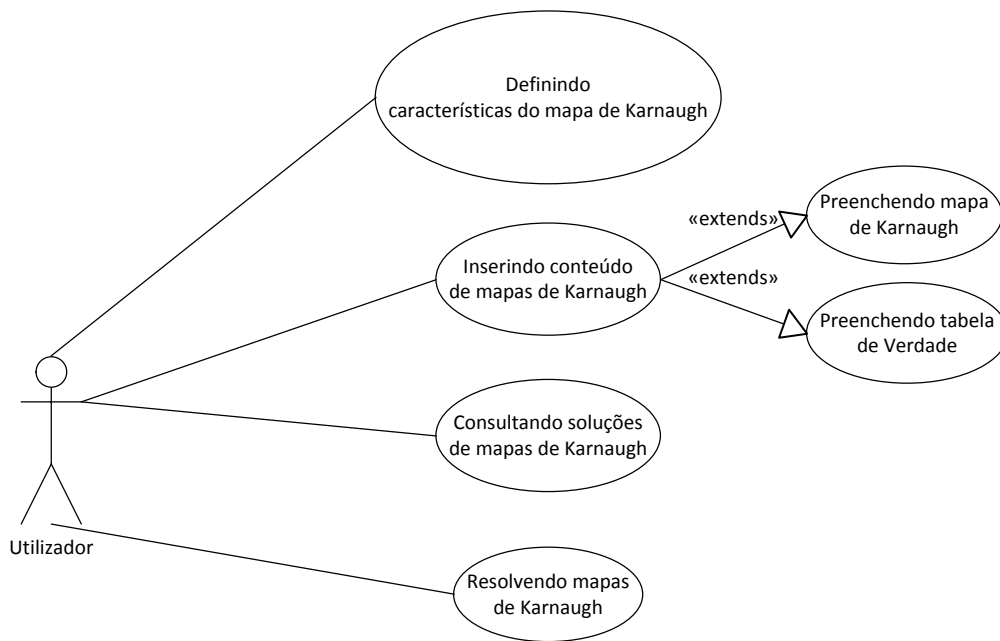


Fig. 36 - Diagrama de Casos de Utilização para KarnUMa na versão telemóvel.

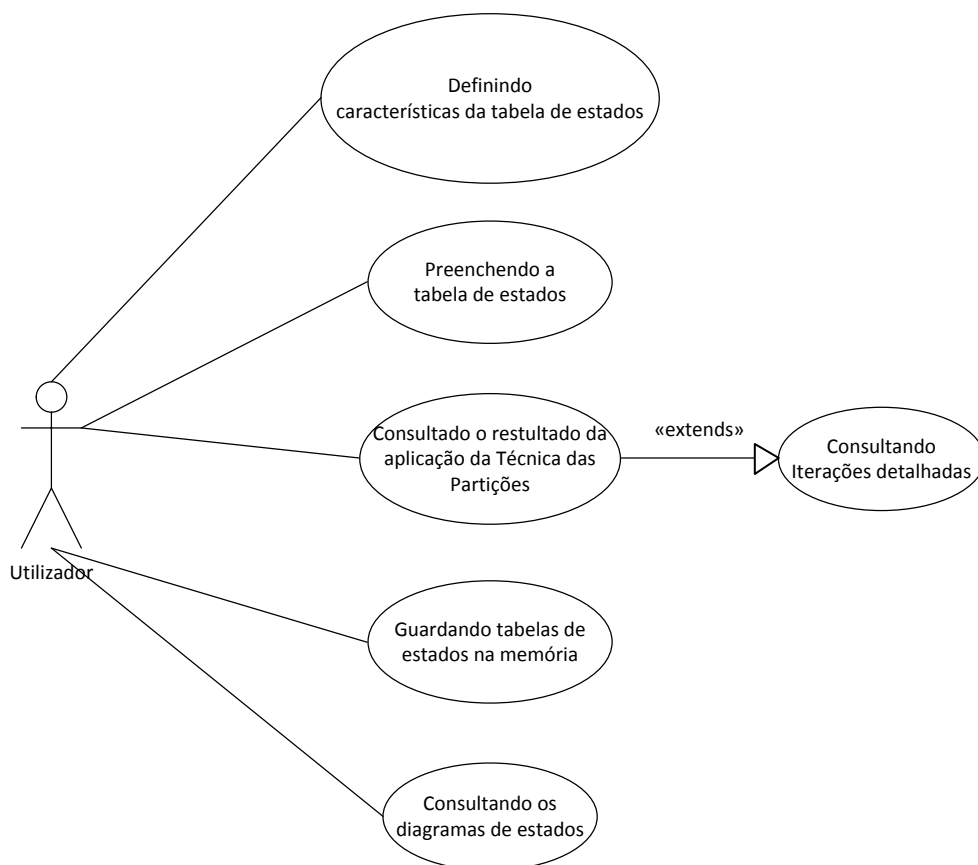


Fig. 37 - Diagrama de Casos de Utilização para a aplicação ParTec.

3.5. APRESENTAÇÃO DAS APLICAÇÕES

Por forma a demonstrar as aplicações, serão apresentados alguns *Storyboards* seguidamente, onde para cada um destes será feita a correspondências com as Tarefas dos Casos de Utilização apresentados anteriormente.

3.5.1. KARNUMA (PC)

Storyboard 1 (Fig. 38 e Fig. 39) – Pretende-se resolver resumidamente um mapa de Karnaugh de 4 variáveis pelo método da Produto de Somas. Nas ilustrações deste *Storyboard* encontram-se as seguintes tarefas realizadas:

- **Ilustração 1:**
 - Definindo Características do mapa de Karnaugh;
- **Ilustração 2:**
 - Inserindo Conteúdo de mapas de Karnaugh;
 - Preenchendo mapa de Karnaugh.
- **Ilustrações 3, 4 e 5:**
 - Resolvendo mapas de Karnaugh;
 - Resolvendo resumidamente mapas de Karnaugh.

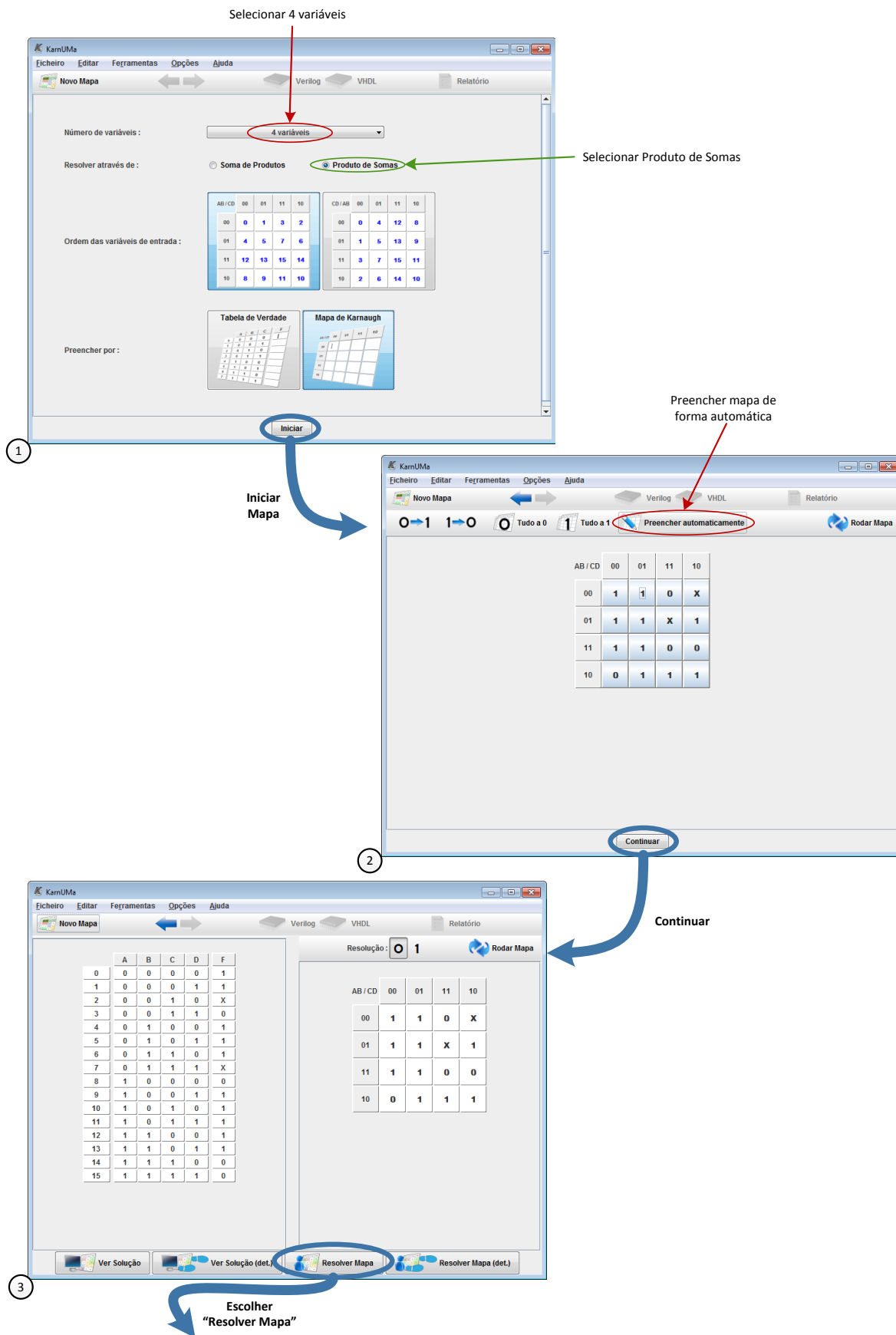


Fig. 38 –Storyboard 1 – KarnUMa.

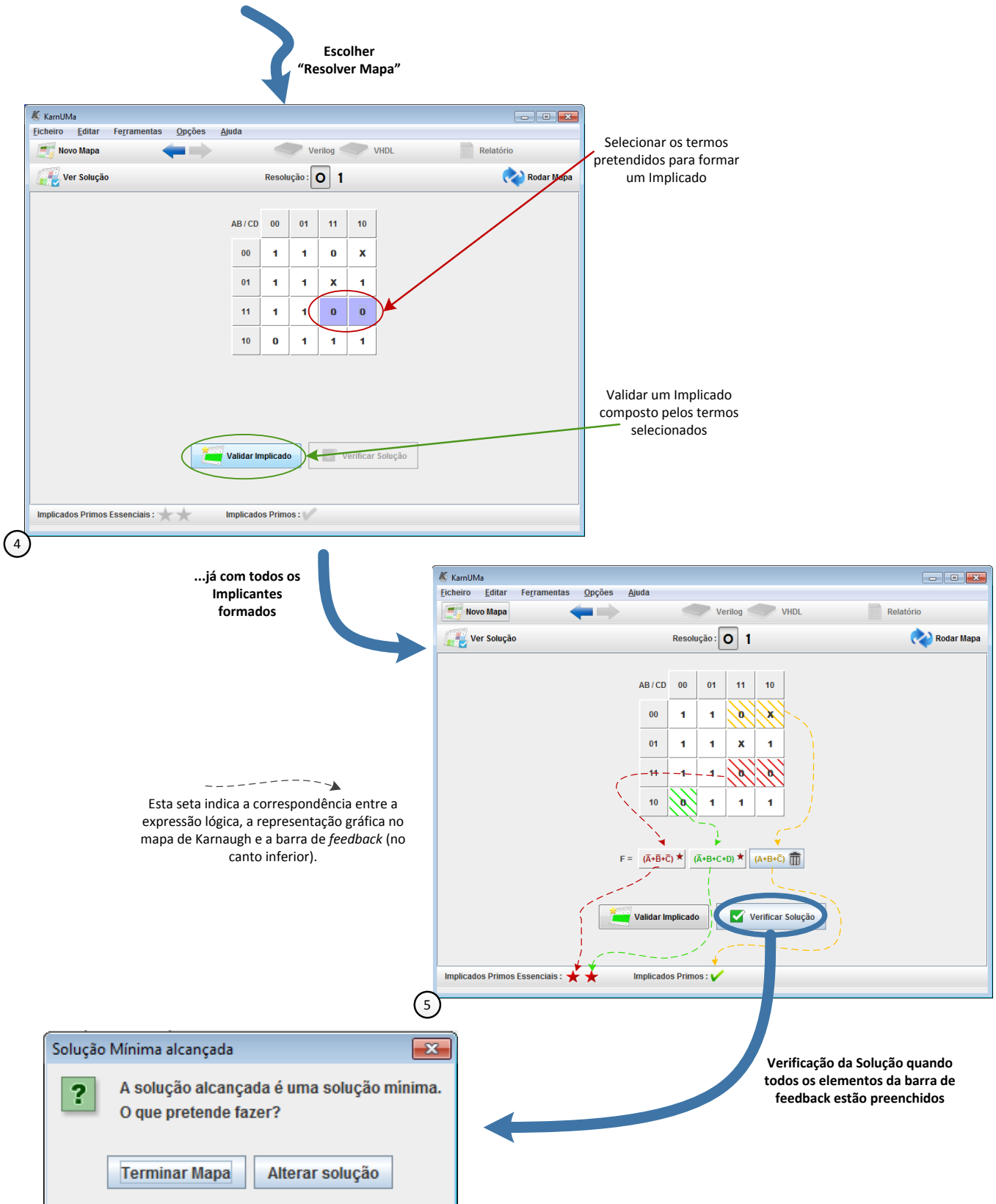


Fig. 39 - Storyboard 1 - KarnUMA (continuação).

Storyboard 2 (Fig. 40 e Fig. 41) – Pretende-se obter a solução detalhada de um mapa de Karnaugh de 5 variáveis, em que a ordem das variáveis de entrada seja “CD/AB” e a inserção do conteúdo do mapa seja feita por uma tabela de verdade. Nas ilustrações do *Storyboard 2*, representado nas **Fig. 40 e Fig. 41**, é possível encontrar as seguintes tarefas:

- **Ilustração 1:**
 - Definindo Características do mapa de Karnaugh.

- **Ilustração 2:**
 - Inserindo Conteúdo de mapas de Karnaugh;
 - Preenchendo tabela de Verdade.

- **Ilustrações 3, 4 e 5:**
 - Consultando soluções de mapas de Karnaugh;
 - Consultando solução detalhada de mapas de Karnaugh.

Storyboard 2.1 (Fig. 42) – A partir da ilustração onde terminou o *Storyboard 2*, pretende-se demonstrar em primeiro lugar como alterar a solução apresentada para o método da Soma de Produtos e seguidamente mostrar como obter o relatório do mapa de Karnaugh e os códigos nas Linguagens de Descrição de Hardware (LDH) *Verilog* e *VHDL*. Neste diagrama podem-se encontrar as seguintes tarefas:

- **Ilustração 1:**
 - Consultando soluções de mapas de Karnaugh;
 - Consultando solução detalhada de mapas de Karnaugh.

- **Ilustração 2:**
 - Obtendo código LDH;
 - Obtendo código *Verilog*;
 - Obtendo código *VHDL*;
 - Obtendo relatórios de mapas de Karnaugh.

Nota: Os anexos A e B contêm exemplos de relatórios gerados pela aplicação KarnUMa nos idiomas português e inglês. Nos mesmos anexos podem ainda ser consultados os exemplos de código *VHDL* e código *Verilog* nos respetivos pontos dos relatórios gerados.

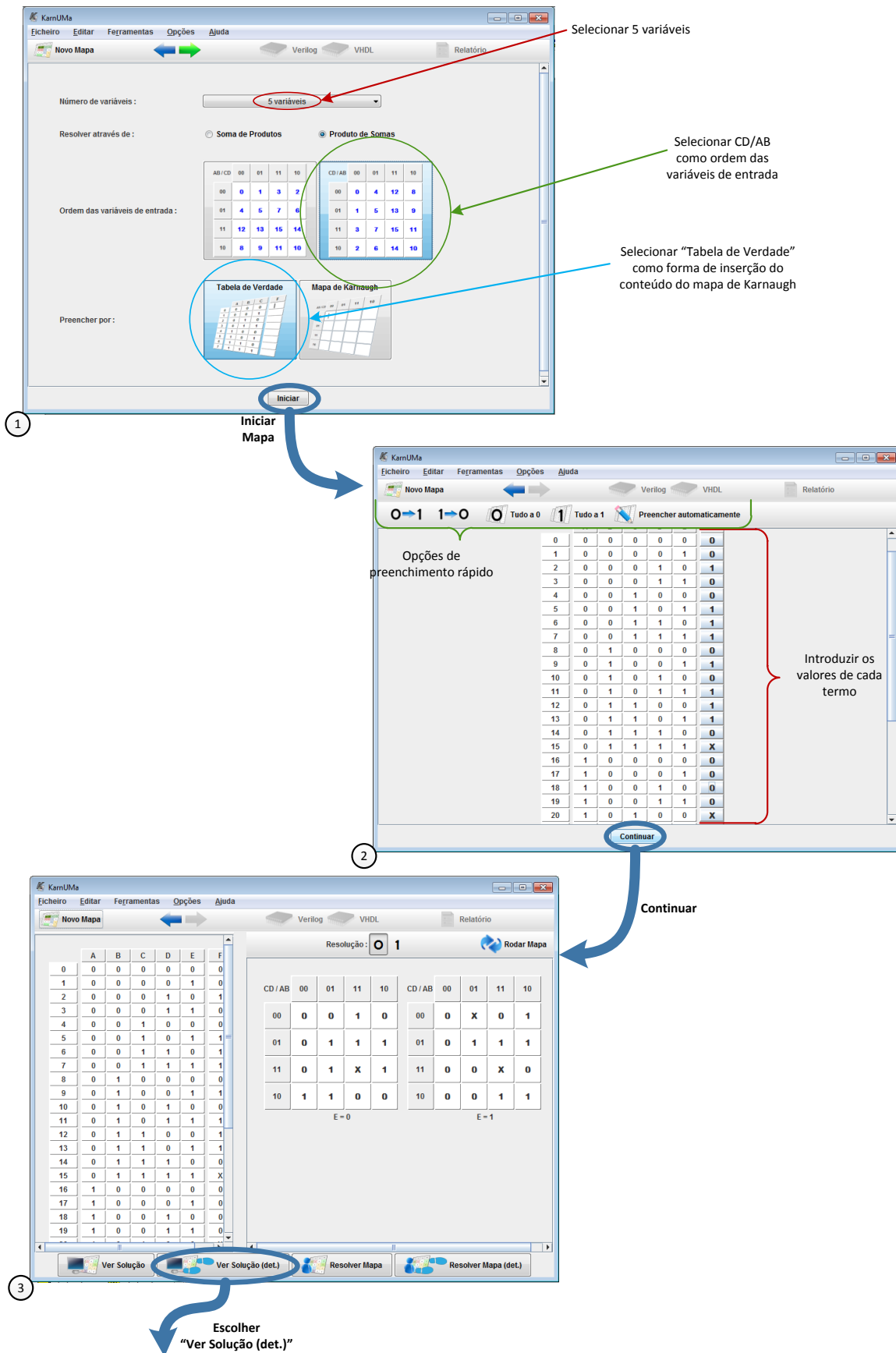
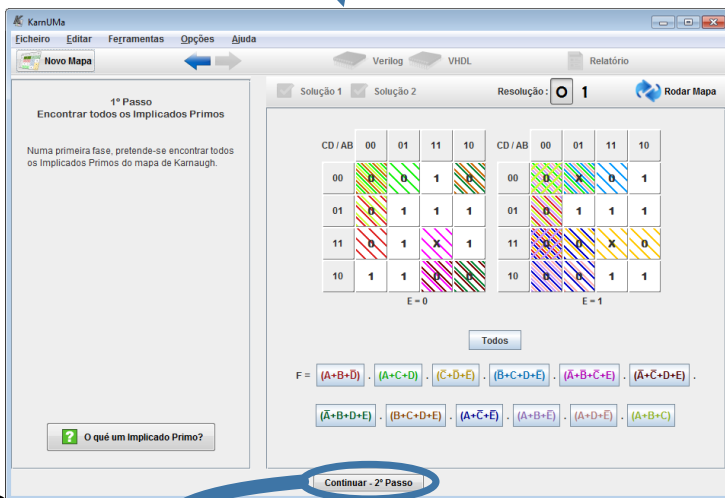


Fig. 40 - Storyboard 2 – KarnUMA.

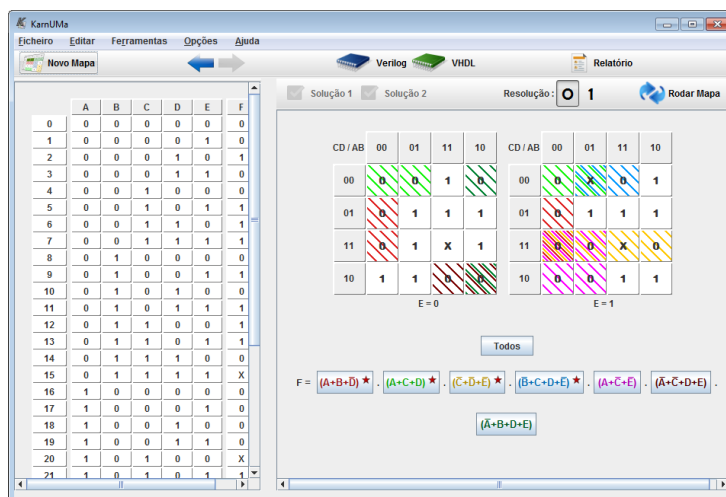
Escolher
"Ver Solução (det.)"



4

Continuar, passando pelos
vários passos da solução
detalhada

...



5

Solução final do mapa

Fig. 41 - Storyboard 2 - KarnUMa (continuação).

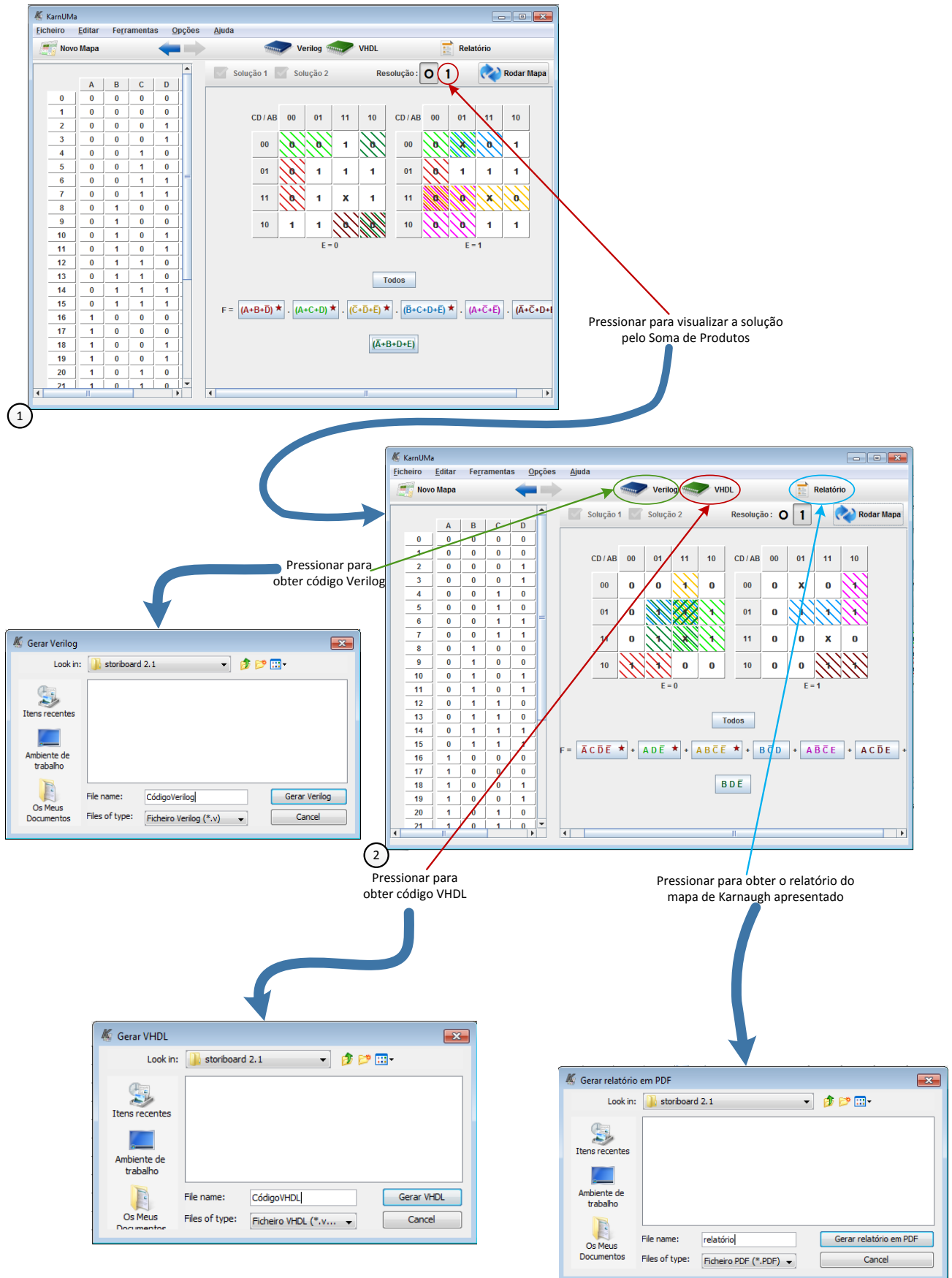


Fig. 42 - Storyboard 2.1 – KarnUMA.

3.5.2. POCKET KARNUMA (MIDLET E ANDROID PACKAGE)

Storyboard 1 (Fig. 43) – Pretende-se resolver um mapa de Karnaugh de 3 variáveis através do método da Soma de Produtos. Neste *Storyboard* podem-se identificar as seguintes tarefas:

- **Ilustração 1:**
 - Definindo características do mapa de Karnaugh.
- **Ilustração 2:**
 - Inserindo conteúdo de mapas de Karnaugh;
 - Preenchendo mapa de Karnaugh.
- **Ilustrações 3, 4, 5 e 6:**
 - Resolvendo mapas de Karnaugh.

Storyboard 2 (Fig. 44) – Pretende-se obter a solução pela Produto de Somas de um mapa de Karnaugh de 4 variáveis em que o conteúdo deste será inserido por uma tabela de verdade. Uma vez visualizada a solução pretende-se alterar a ordem das variáveis de entrada e posteriormente observar a solução do mesmo mapa pela Soma de Produtos. Neste *Storyboard* encontram-se as seguintes tarefas:

- **Ilustração 1:**
 - Definindo características do mapa de Karnaugh.
- **Ilustração 2:**
 - Inserindo conteúdo de mapas de Karnaugh;
 - Preenchendo tabela de Verdade.
- **Ilustrações 3, 4, 5 e 6:**
 - Consultando soluções de mapas de Karnaugh.

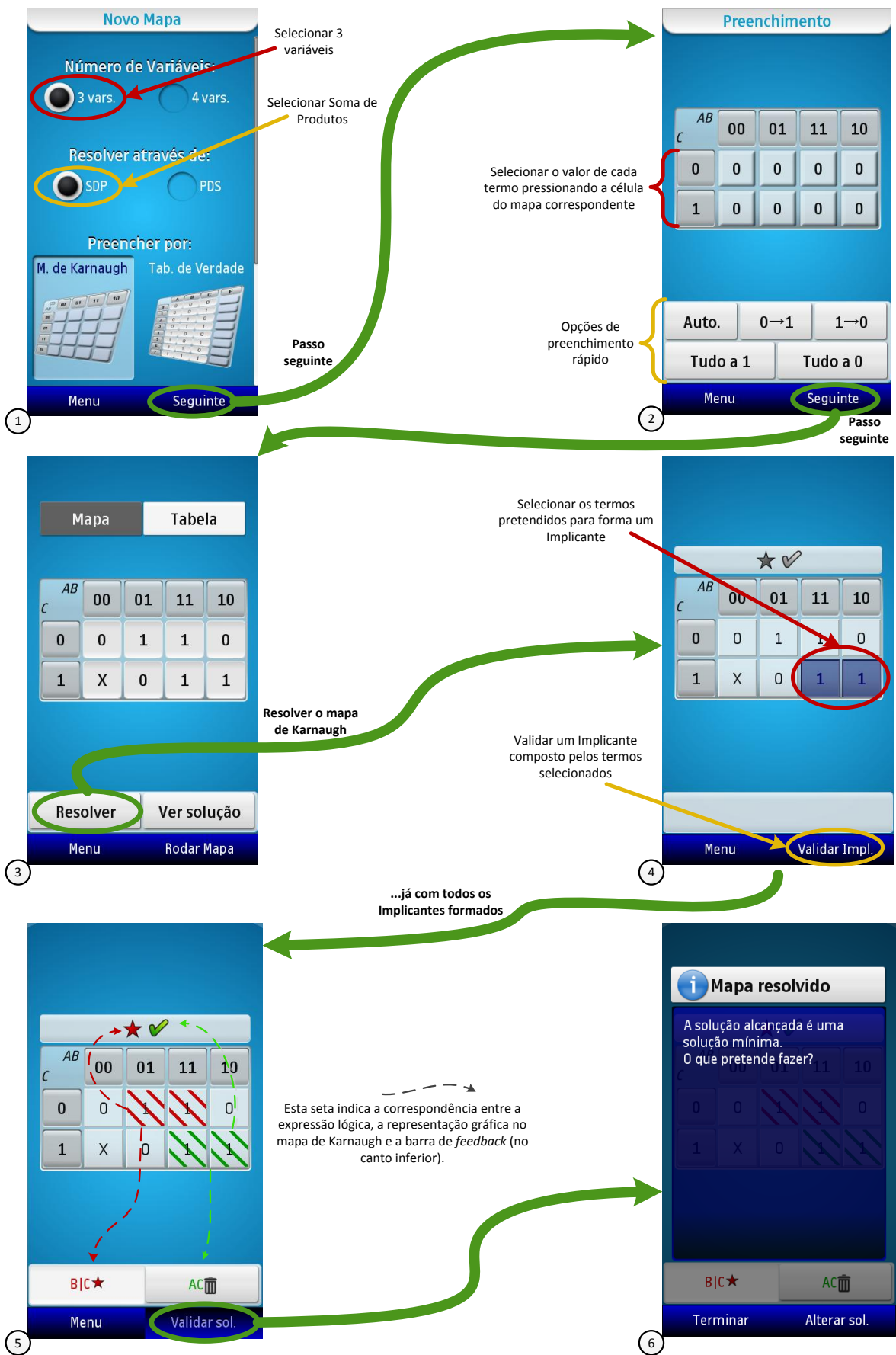


Fig. 43 - Storyboard 1 - Pocket KarnUMa num telemóvel Touch Screen.



Fig. 44 - Storyboard 2 - Pocket KarnUMA num telemóvel convencional.

3.5.3. PARTEC

Storyboard 1 (Fig. 45) – Pretende-se preencher uma tabela estados de uma máquina de Moore, com 7 estados, 1 entrada e 1 saída. Uma vez preenchida a tabela deve-se guarda-la como tabela 3. Neste Storyboard identificam-se seguintes tarefas:

- **Ilustração 1:**
 - Definindo características da tabela de estados;
 - Preenchendo a tabela de estados;
 - Guardando tabelas de estados na memória.

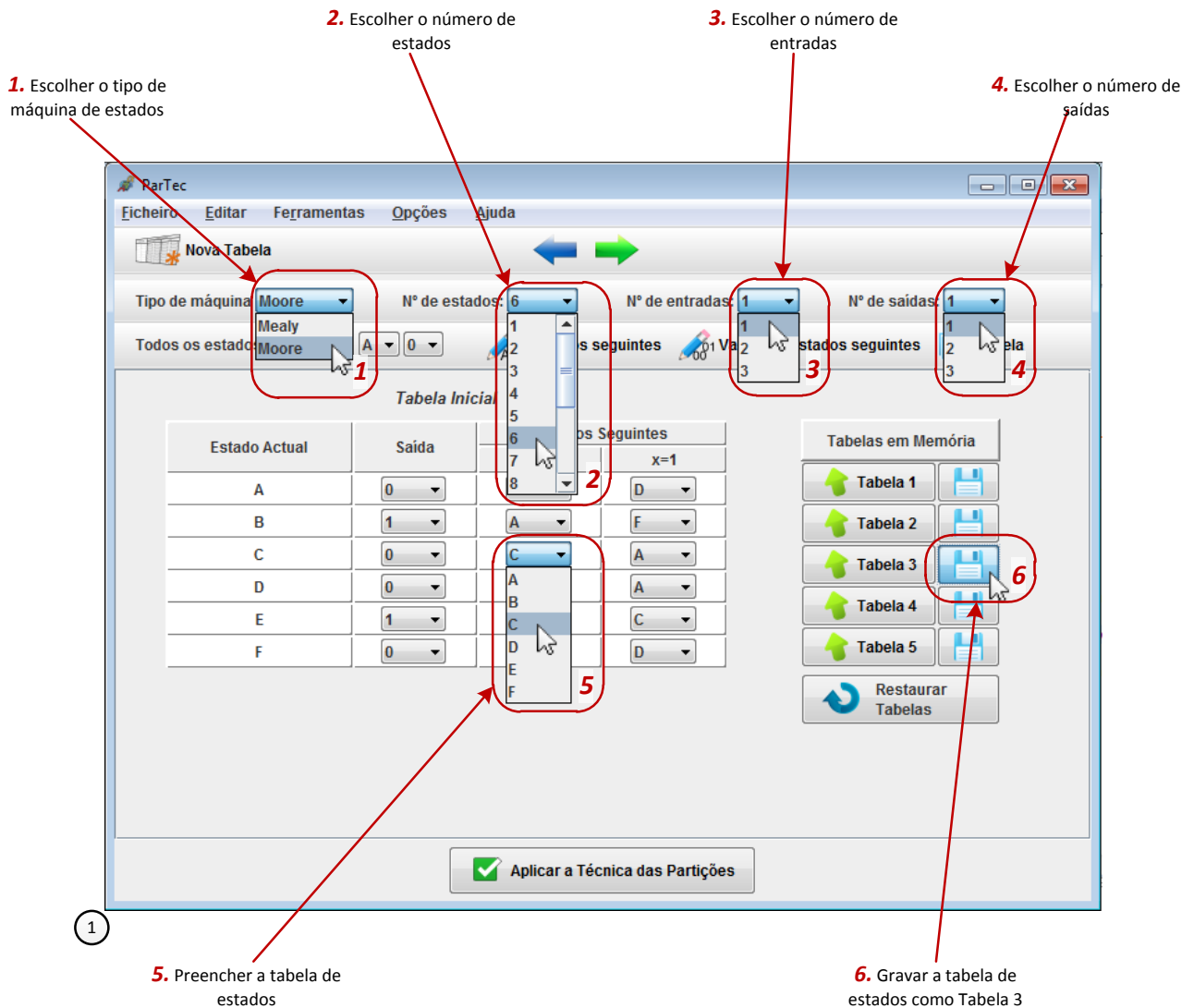


Fig. 45 - Storyboard 1 - Aplicação ParTec.

Storyboard 2 (Fig. 45) – Pretende-se visualizar o resultado da execução da Técnica das Partições à tabela preenchida no Storyboard 1 (memorizada como Tabela 3). Posteriormente pretende-se observar as iterações detalhadas da Técnica das Partições e por fim obter os diagramas de estados correspondentes à solução apresentada.

- **Ilustrações 1 e 2:**
 - Consultando o resultado da aplicação da Técnica as Partições.
- **Ilustração 3:**
 - Consultando Iterações detalhadas.
- **Ilustração 4:**
 - Consultando os diagramas de estados.



Fig. 46 - Storyboard 2 - Aplicação Partec.

3.6. ANÁLISE ÀS APLICAÇÕES DESENVOLVIDAS

Uma vez analisadas as várias etapas do desenvolvimento das aplicações produzidas nesta dissertação, resta, no caso das aplicações KarnUMa, estabelecer uma comparação com aquelas que foram consideradas como aplicações de referência. Já no que à aplicação ParTec diz respeito, identificar-se-ão as funcionalidades disponíveis, visto que não foi encontrada nenhuma aplicação com a qual se pudesse estabelecer uma comparação.

3.6.1. APLICAÇÕES KARNUMA

A **Tabela 14** estabelece uma comparação entre as aplicações de referência e as aplicações KarnUMa. Pode-se verificar que a aplicação KarnUMa conjuga as principais características das aplicações de referências da respetiva plataforma e ainda acrescenta novas funcionalidades relevantes nesta área. No fim da **Tabela 14** estão listadas as características que permitem às aplicações desenvolvidas se destacarem das demais.

A aplicação KarnUMa destaca-se das demais existentes pelo ênfase colocado na identificação dos Implicantes e Implicados Primos, identificando-os de forma inequívoca e destacando os Essenciais. A utilização dos Implicantes e Implicados permite uma abordagem determinística à minimização de funções, garantindo a obtenção de uma solução otimizada do mapa de Karnaugh. Uma outra característica relevante desta aplicação reside no facto de apresentar todos os passos que constituem o processo de resolver um mapa de Karnaugh, dando ainda a oportunidade ao utilizador de resolver o mapa deste mesmo modo, algo que vem reforçar a componente didática desta ferramenta. A aplicação KarnUMa é ainda capaz de fornecer até duas soluções por mapa e verificar qualquer solução do utilizador validando-a caso seja uma solução correta, independentemente de ser idêntica às duas soluções encontradas pela aplicação (basta que seja equivalente). Relativamente às restantes funcionalidades que já estavam presentes nas aplicações existentes, tentou-se tornar estas funcionalidades mais completas e intuitivas, como por exemplo na resolução de um mapa por parte do utilizador foi introduzida uma barra de *feedback*, que permite ao utilizador tomar consciência do seu desempenho neste processo. Ainda na resolução por parte do utilizador, a ferramenta é capaz de identificar os erros cometidos pelo utilizador fornecendo dicas para que esses mesmos erros não voltem a ser cometidos.

No que diz respeito à aplicação Pocket KarnUMa para telemóvel, sucede o mesmo que referido anteriormente, visto que esta aplicação também conjuga as principais características daquela que foi considerada a mais completa para esta plataforma. Relativamente à aplicação de referência o Pocket KarnUMa apresenta as vantagens de estar disponível para mais do que uma plataforma, uma vez está disponível como *Midlet* e como *Android Package*. Esta aplicação fornece ao utilizador a oportunidade de resolver mapas de Karnaugh e ainda partilha das características da versão PC que consistem em dar especial atenção aos Implicantes Primos Essenciais e fornecer até duas soluções por mapa, sendo capaz de validar qualquer solução correta alcançada pelo utilizador.

É importante realçar ainda que, apesar de tudo, em alguns aspetos, as aplicações desenvolvidas nesta tese não conseguem ser tão completas como as de referência. A ferramenta KarnUMa aceita como número máximo 6 variáveis enquanto ambos *Karnaugh Minimizer* e *Karma* conseguem ir até 8 variáveis. O mesmo sucede com a aplicação Pocket KarnUMa que aceita um máximo de 4 variáveis enquanto a aplicação *Karnaugh Map Advanced* aceita até 8 variáveis. Nenhuma das aplicações desenvolvidas disponibiliza a introdução do conteúdo dos mapas de Karnaugh por expressão booleana, algo que ambas as aplicações de referência na plataforma PC disponibilizam. Ainda é importante referir que também não foi implementada a funcionalidade de

gerar o esquema do circuito elétrico, mas esta falha pode ser colmatada pelo facto de fornecer código fonte *Verilog* e *VHDL*, a partir do qual diversas aplicações conseguem gerar esse mesmo esquema.

3.6.2. APLICAÇÃO PARTEK

A aplicação desenvolvida sobre a ferramenta de Sistemas Digitais, Técnica das Partições disponibiliza as seguintes funcionalidades:

- Trabalha com máquinas de Moore e Mealy;
- As tabelas de estados podem contar com um máximo de 15 estados;
- Pode ter até 3 entradas o que se traduz num máximo 8 estados seguintes;
- Pode ter até 3 variáveis de saída;
- Opções de preenchimento rápido da tabela de estados;
- Apresentação das iterações de forma simples;
- Apresentação das iterações de forma detalhada:
 - Identificação das Partições por cores;
 - Identificação dos estados redundantes e novas Partições a cada iteração.
- Apresentação da tabela resumida;
- Apresentação dos diagramas de estados correspondentes à tabela inicial e tabela resumida (somente disponível para máquinas de estados com um máximo 10 estados e uma única entrada);
- Permite guardar e carregar tabelas de estados memorizadas;
- Disponível em Português e Inglês.

| Características | PC | | | | Telemóvel | |
|--|--------------------|-------|--------------|---------|-----------------------|----------------|
| | Karnaugh Minimizer | Karma | EasyKarnaugh | KarnUMa | Karnaugh Map Advanced | Pocket KarnUMa |
| Assinala os Implicantes/Implicados | X | X | X | X | X | X |
| Introdução pelo Mapa | X | X | X | X | X | X |
| Introdução por Tabela de Verdade | X | X | - | X | X | X |
| Introdução por expressão algébrica | X | X | - | - | - | - |
| Suporta termos de valor indiferente | X | X | X | X | X | X |
| Expressão simplificada | X | X | X | X | X | X |
| Número mínimo de variáveis | 1 | 2 | 3 | 3 | 1 | 3 |
| Número máximo de variáveis | 8 | 8 | 4 | 6 | 8 | 4 |
| Resolução SOP | X | X | X | X | X | X |
| Resolução POS | X | - | X | X | X | X |
| Opções de preenchimento rápido do mapa | X | - | - | X | - | X |
| Apagar todo o mapa | X | X | X | - | - | - |
| Alterar ordem das variáveis de entrada | - | X | - | X | - | X |
| Personalizar nomes das variáveis | X | X | - | - | - | - |
| Número de ordem do minitermo | X | X | - | - | - | - |
| Tabela dos implicantes (Quine McCluskey) | - | X | - | - | - | - |
| Guardar e abrir Mapas | X | X | - | - | - | - |
| Gerar relatório do Mapa | X | - | - | X | - | - |
| Gerar código VHDL | X | - | - | X | - | - |
| Gerar código Verilog | X | - | - | X | - | - |
| Exportar mapa em HTML | X | - | - | - | - | - |
| Gerar esquema eléctrico | X | - | - | - | - | - |
| Utilizador pode resolver mapas de Karnaugh | - | - | X | X | - | X |
| Suporte didáctico | - | X | X | X | - | - |
| Contém Exercícios | - | X | X | X | - | - |
| Suporta mais que um idioma | X | X | X | X | - | X |
| Ênfase aos Implicantes Primos Essenciais | - | - | - | X | - | X |
| Apresentação das diversas etapas que compõem a resolução de um Mapa de Karnaugh | - | - | - | X | - | - |
| Utilizador pode resolver passando por todas as etapas da resolução de um mapa de Karnaugh | - | - | - | X | - | - |
| Fornecer até 2 soluções por mapa e é capaz de verificar todas as soluções alternativas | - | - | - | X | - | X |

Tabela 14 – Comparação das aplicações de referência com as aplicações desenvolvidas nesta tese sobre os mapas de Karnaugh.

4. AVALIAÇÃO DE USABILIDADE

Para realizar uma avaliação de usabilidade às aplicações KarnUMa recorreu-se a um conjunto de utilizadores que deveriam executar as tarefas apresentadas no ponto 3.5 *Apresentação das aplicações*. O desempenho dos utilizadores foi observado e com base nas Heurísticas de Nielsen (Nielsen, 1994) tentaram-se identificar as falhas de usabilidade existentes.

As 10 Heurísticas de Nielsen utilizadas para identificar as possíveis falhas de usabilidade são as seguintes:

1. **Tornar o estado do sistema visível** – O sistema deve manter sempre os utilizadores informados sobre o que está acontecer, fornecendo *feedback* apropriado dentro de um intervalo de tempo aceitável (Nielsen, 1994).
2. **Correspondência entre o sistema e o mundo real** – O sistema deve “falar” a linguagem dos utilizadores, com palavras, frases e conceitos familiares ao utilizador, ao invés de termos orientados ao sistema. Devem ser seguidas as convenções do mundo real para que as informações apareçam seguindo um ordem natural e lógica (Nielsen, 1994).
3. **Utilizador controla e livre-arbítrio** – Os utilizadores enganam-se frequentemente, logo é necessário fornecer um "saída de emergência" óbvia para minimizar este engano. Fornecer opções de retroceder e repetir (Nielsen, 1994).
4. **Consistência e adesão às normas** – Os utilizadores não devem ter a necessidade de adivinhar que diferentes palavras, situações ou ações têm o mesmo significado. Devem-se seguir as convenções de cada plataforma (Nielsen, 1994).
5. **Evitar erros** – Melhor do que boas mensagens de erro é evitar que estes sucedam. As condições passíveis de erros devem ser eliminadas ou corretamente identificadas apresentando uma mensagem de confirmação antes do utilizador executar uma ação comprometedora (Nielsen, 1994).
6. **Reconhecer, em vez de lembrar** – Minimizar a necessidade do utilizador recorrer à sua memória. O utilizador não deve precisar de se lembrar de informação de um estado para o outro da aplicação. Instruções de utilização do sistema devem estar visíveis e acessíveis quando necessárias (Nielsen, 1994).
7. **Flexibilidade e eficiência** – Fornecer atalhos, ainda que pouco ou nunca utilizados por um utilizador principiante, podem frequentemente acelerar a interação no caso de utilizadores mais experientes. Permitir aos utilizadores personalizar ações frequentes (Nielsen, 1994).
8. **Desenho de ecrã estético e minimalista** – Os ecrãs não devem conter informação irrelevante ou raramente necessária. Cada unidade extra de informação num ecrã está a competir com a informação relevante pela atenção do utilizador, tornando esta última menos visível (Nielsen, 1994).
9. **Ajudar o utilizador a reconhecer, diagnosticar e recuperar dos erros** – As mensagens de erro devem ser expressas em linguagem clara (sem códigos), indicando com precisão o problema e construtivamente sugerir uma solução (Nielsen, 1994).
10. **Dar ajuda e documentação** – Ainda que o ideal seja que o utilizador use o sistema sem recorrer a ajudas, esta deve estar sempre disponível. Tais informações devem ser facilmente acedidas, focando-se na tarefas do utilizador. Uma boa prática consiste em listar os passos a serem realizados não se alongando em demasia (Nielsen, 1994).

4.1. FALHAS DE USABILIDADE

Nesta secção serão apresentadas as falhas de usabilidade detetadas nos testes realizados, bem como as soluções adotadas para colmatá-las.

4.1.1. FALHAS DE USABILIDADE DA APLICAÇÃO KARNUMA (AMBAS AS VERSÕES)

As seguintes falhas de usabilidade, comuns a ambas as versões da aplicação KarnUMa, foram identificadas:

a) Na resolução por parte do utilizador (tanto resolução detalhada, como resolução rápida) quando um agrupamento não era considerado válido, após a mensagem de erro a aplicação mantinha toda a seleção de termos do mapa realizada por parte do utilizador. Esta situação foi considerada uma falha de usabilidade, uma vez que constantemente se verificou que o utilizador teria que desseleccionar os termos para voltar a realizar uma nova seleção para formar um agrupamento. Esta situação não respeita a **heurística 7** uma vez que diminui a eficiência e o desempenho do utilizador. Por forma a colmatar esta falha de usabilidade foram implementadas as seguintes modificações:

- Quando se verifica uma seleção de termos completamente incorreta, após a mensagem de erro todos os termos são desseleccionados;
- Quando se verifica uma seleção parcialmente incorreta, após a mensagem de erro são desseleccionados aqueles termos que nunca poderiam ser utilizado para a correta formação de um agrupamento.

A **Fig. 47** ilustra uma situação em que o método de resolução definido consiste na Soma de Produtos, e que por engano o utilizador fez uma seleção que seria só seria correta pelo Produto de Somas, como se pode ver na ilustração “Antes”, a aplicação mantém toda a seleção. Implementada a correção a aplicação apenas mantém o termo que ainda pode ser utilizado neste método de resolução como se pode ver na ilustração “Depois”.

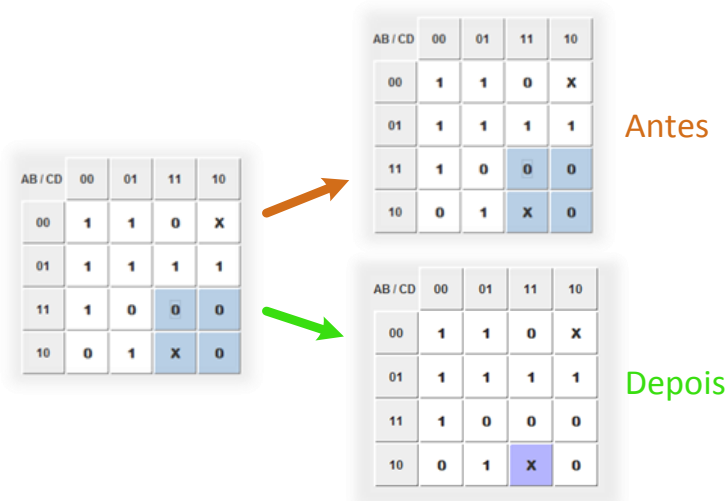


Fig. 47 – Falha de usabilidade da aplicação KarnUMa (ambas as versões) descrita em a).

b) A segunda falha de usabilidade, comum a ambas as versões, que foi identificada ocorria também aquando da resolução por parte do utilizador. Numa situação em que o utilizador já tivesse formado corretamente alguns agrupamentos e por engano alterasse o método de resolução, ou seja estando a resolver pela Soma de Produtos alterasse para Produtos de Somas, mesmo recorrendo à funcionalidade de “voltar atrás” os agrupamentos já feitos perdiam-se. Isto entra em desacordo com a **heurística 5** visto que a aplicação não apresentava forma de minimizar este erro ou mesmo até mesmo evitá-lo. Uma situação semelhante verificou-se quando o utilizador escolhia por engano a opção “Novo Mapa” ou ainda se escolhesse sair do programa em qualquer altura da sua execução. Para colmatar estas falhas foram introduzidas as mensagens de confirmação que podem ser visualizadas na **Fig. 48**.

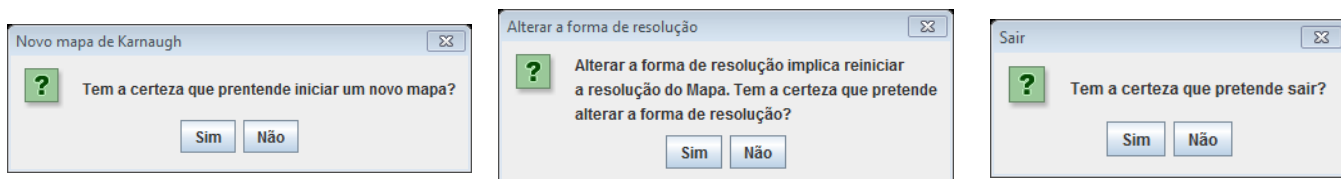


Fig. 48 - Mensagens de Confirmação introduzidas para colmatar a falha de usabilidade da aplicação KarnUMA (ambas as versões) descrita em b).

4.1.2. FALHAS ESPECÍFICAS DA VERSÃO PC DO KARNUMA

As seguintes falhas foram identificadas especificamente na versão PC do KarnUMA.

a) Na resolução por parte do utilizador constatou-se a situação ilustrada pela **Fig. 49**. Quando o utilizador selecionava um termo que fizesse parte da composição de um agrupamento já identificado, este perdia essa sinalização (as linhas diagonais que identificam o agrupamento desapareciam) como se pode verificar na ilustração “Antes” da **Fig. 49**. Esta falha entra em conflito com a **heurística 1** uma vez que omite a já utilização daquele termo. Esta falha tornou-se ainda mais evidente porque o mesmo não se verificava na versão para terminal móvel. Por forma a corrigir esta falha os termos selecionados passaram a manter as linhas diagonais que os identificam como se pode verificar pela ilustração “Depois”;

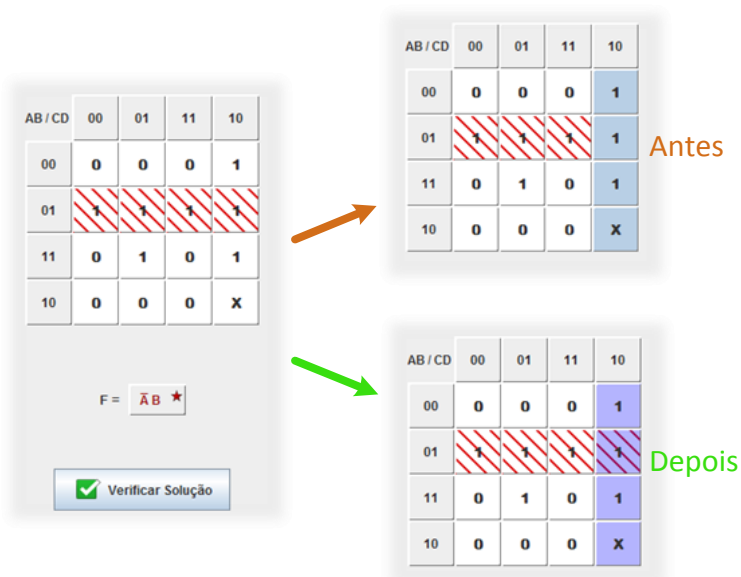


Fig. 49 - Falha de usabilidade específica da versão PC da aplicação KarnUMA descrita em a).

b) A Fig. 50 ilustra outra falha de usabilidade identificada desta feita, aquando da apresentação da solução de um mapa de Karnaugh. Numa situação em que utilizador seleccionava um agrupamento em particular de forma a visioná-lo em exclusivo no mapa e posteriormente pretendia voltar a ver todos os agrupamentos assinalados. O problema averiguado residia no facto de que para voltar a visionar o mapa com todos os grupos assinalados o utilizador voltava a carregar no botão do tipo *toggle* que se encontrava pressionado, e nada acontecia, visto que como se pode verificar pela ilustração “Antes” da Fig. 50, para a ação pretendida deveria pressionar no botão “todos”. Esta situação entra em conflito com a *heurística 4*, visto que no comportamento comum de um botão do tipo *toggle*, quando este deixa de estar pressionado a ação que este executa é igualmente cancelada. Para solucionar este problema, implementou-se o que está representado na ilustração “Depois”, ou seja para voltar a ver todos os agrupamentos ou “solta-se” o botão da fórmula que está pressionado ou pressiona-se “Todos”.

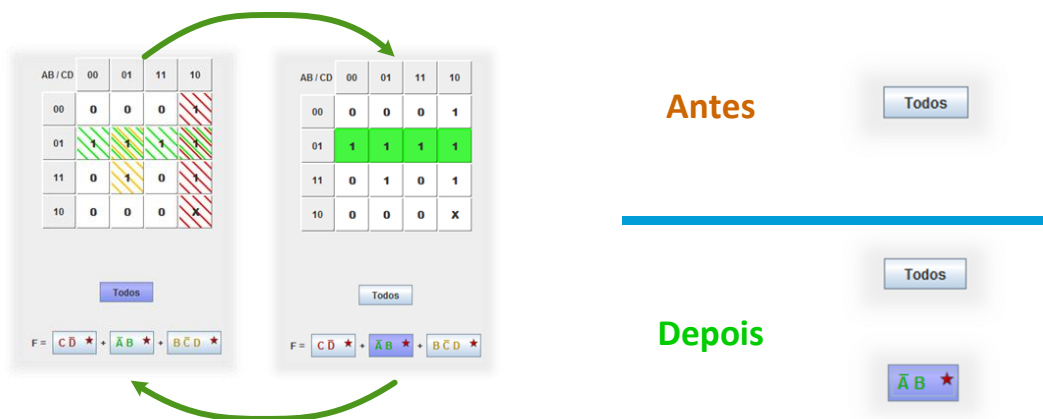


Fig. 50 - Falha de usabilidade específica da versão PC da aplicação KarnUMa descrita em b).

c) A Fig. 51 ilustra mais uma falha de usabilidade aquando da resolução por parte do utilizador. Desta feita consistia na posição do botão que permite a validação dos agrupamentos. Quando se tratava de uma primeira experiência com a aplicação verificou-se, muito frequentemente, que após a seleção dos termos pretendidos para formar um agrupamento os utilizadores ficavam “perdidos” sem saber o que fazer. Este problema residia principalmente pela posição que o botão “Validar Implicante” assumia no ecrã. Esta posição não respeitava a *heurística 8*, visto não era a mais visível. Para colmatar esta falha o botão “Validar Implicante” foi colocado numa posição onde passou a ser muito mais facilmente identificado como se pode verificar pela ilustração “Depois” da Fig. 51.

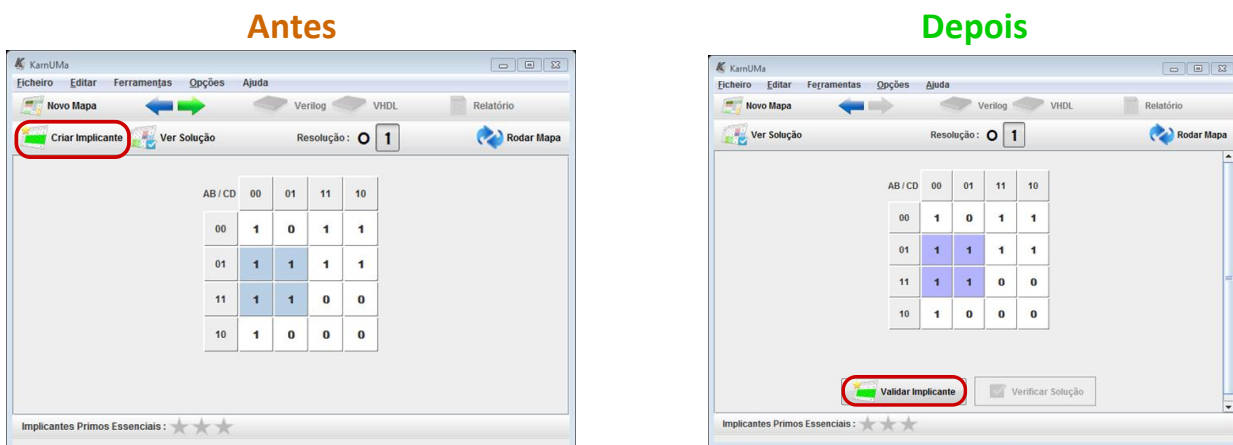


Fig. 51 - Falha de usabilidade específica da versão PC da aplicação KarnUMa descrita em c).

d) Finalmente a última falha de usabilidade encontrada, ilustrada na **Fig. 52**, foi mais facilmente identificada ao observar utilizadores que já tinham utilizado aplicação KarnUMa para terminal móvel. Isto porque o método de preenchimento de um mapa de Karnaugh ou tabela de verdade adotado por esta última versão provou ser mais eficiente e intuitivo. Esta falha foi detectada essencialmente porque após a utilização da versão terminal móvel os utilizadores tentam preenchê-lo da mesma forma, e deparavam-se que tal situação não era possível. O modo inicial de preenchimento exigia que o utilizador digitasse o valor pretendido para cada célula do mapa ou tabela, já na versão terminal móvel o valor de cada célula alterava-se a cada clique e ainda continha um conjunto de opções de preenchimento rápido. O modo de preenchimento inicial entrava em conflito com as **heurísticas 5 e 7**. Heurística 5 porque uma vez que o valor tinha que ser digitado, caso isto não acontecesse o mapa poderia ficar com célula sem qualquer valor, situação que originava um erro. Por fim a Heurística 6 porque o método de preenchimento que foi adotado na versão final comprovou ser mais flexível e eficiente. Esta falha de usabilidade foi ultrapassada com a adoção do mesmo método de preenchimento presente na aplicação para terminal móvel como se pode verificar pela ilustração “Depois” da **Fig. 52**.

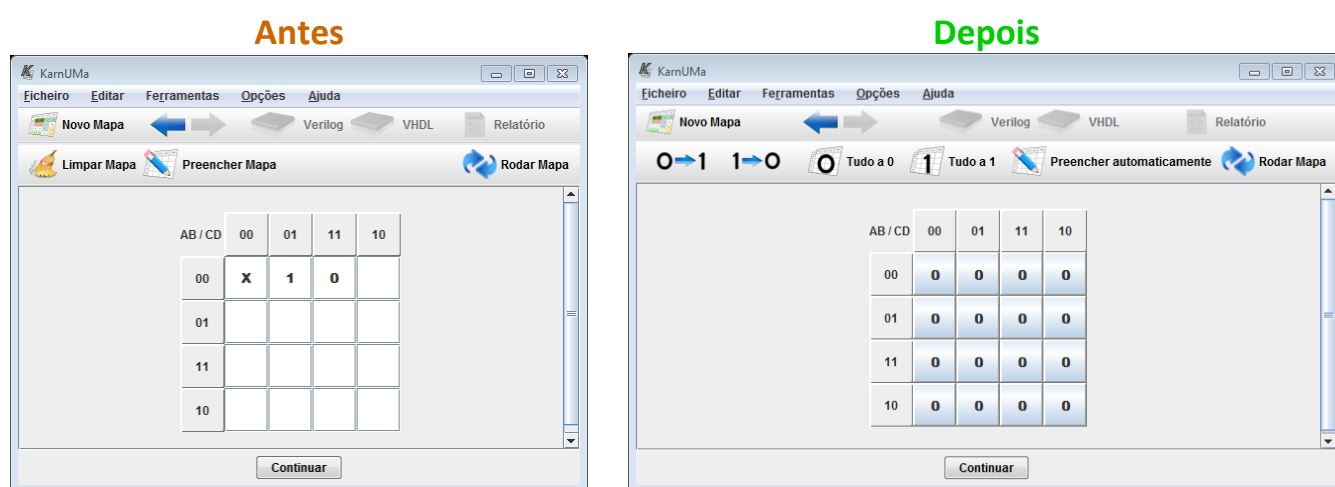


Fig. 52 - Falha de usabilidade específica da versão PC da aplicação KarnUMa descrita em d).

4.1.3. FALHAS ESPECÍFICAS DA VERSÃO POCKET KARNUMA (MIDLET E ANDROID PACKAGE)

Seguem-se as falhas de usabilidade identificadas nesta versão da aplicação Pocket KarnUMa:

a) Posteriormente à escolha das especificações do mapa de Karnaugh, a aplicação segue para o preenchimento do mapa de Karnaugh. Contudo esta última situação não era clara, uma vez que numa primeira utilização da aplicação quando se chegava a este ecrã onde deve ser feito o preenchimento do mapa, alguns utilizadores não sabiam o que fazer. Esta situação entra em conflito com as **heurísticas 1 e 6**, uma vez que os utilizadores não conseguiam identificar o estado da aplicação ou reconhecer o que deveriam fazer naquela situação. Foi considerado como origem deste problema o título presente neste ecrã, que caso o preenchimento fosse por mapa de Karnaugh o título seria “Mapa de Karnaugh”, como se pode verificar pela ilustração “Antes” da **Fig. 53**, e caso se tratasse de um preenchimento por tabela de Verdade o título seria “Tabela de Verdade”. Para ultrapassar esta falha decidiu-se colocar um título mais intuitivo neste ecrã para ajudar na tarefa de reconhecer o que deveria ser feito. O título adotado, como se pode ver pela ilustração “Depois” da **Fig. 53**, foi “Preenchimento”.

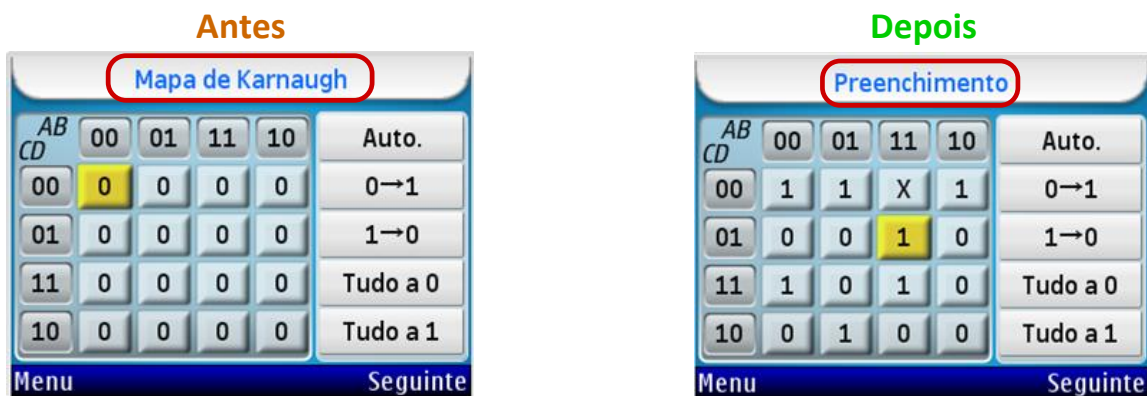


Fig. 53 - Falha de usabilidade específica da versão Pocket KarnUMA descrita em a).

b) Outro problema verificado consistia numa falha de usabilidade provocada pela utilização da *framework Lwuit*, isto porque por defeito esta *framework* coloca o menu e o seu botão de acesso no lado direito, como se pode constatar pela ilustração “Antes” da Fig. 54. Esta situação começou por provocar alguma confusão aos utilizadores e posteriormente verificou-se que a norma neste formato consiste em colocar o menu no lado esquerdo. Esta situação entra em conflito com a *heurística 4*, visto que a aplicação não estava a seguir um aspeto que é norma neste formato. Como se pode verificar pela ilustração “Depois” da Fig. 54, o menu foi colado no lado direito obedecendo à norma deste formato.

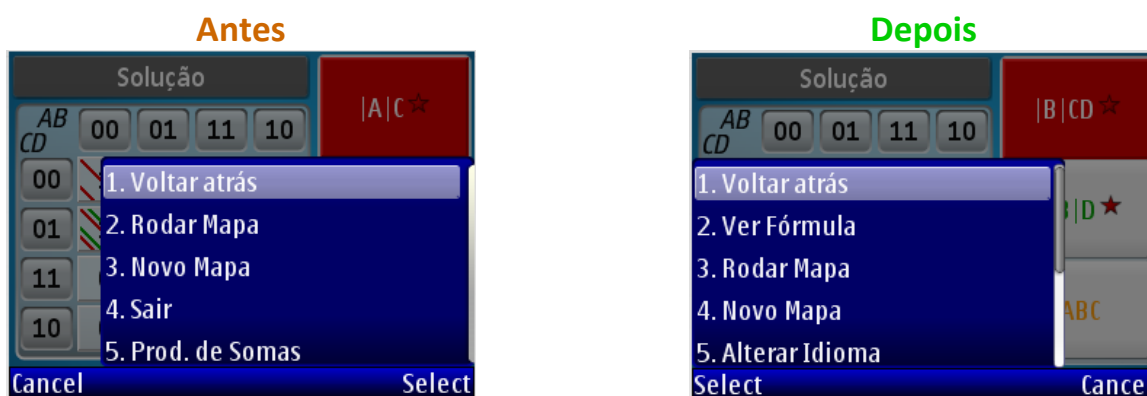


Fig. 54 - Falha de usabilidade específica da versão Pocket KarnUMA descrita em b).

4.1.4. FALHAS DE USABILIDADE DA APLICAÇÃO PARTEC

Nesta aplicação foi encontrado a seguinte falha de usabilidade:

a) A aplicação ParTec fornece a capacidade de memorizar e carregar da memória tabelas de estados através do menu representado na Fig. 55. Verificou-se que quando os utilizadores pretendiam memorizar uma tabela pressionavam mais do que uma vez o botão existente para essa tarefa. Verificou-se ainda que alguns utilizadores posteriormente a memorizar uma tabela carregavam essa mesma tabela da memória por forma a verificar se esta tinha realmente sido memorizada. As situações aqui descritas podem ser interpretadas como falhas de usabilidade de acordo com a *heurística 1*, visto que o sistema não permitia verificar que esta tarefa foi corretamente executada. Pode-se ainda considerar que a *heurística 5* foi igualmente desrespeitada visto que um utilizador podia guardar uma tabela de estados em cima de uma tabela já existente, sem pedir qualquer tipo de confirmação. A solução encontrada para estas falhas encontra-se na Fig. 56. Com esta mensagem de

confirmação o utilizador passa a receber informação de que posição de memória está a utilizar, bem como evita-se o erro de substituir involuntariamente uma tabela já existente.



Fig. 55 - Menu da aplicação ParTec que permite memorizar e carregar tabelas de estados.

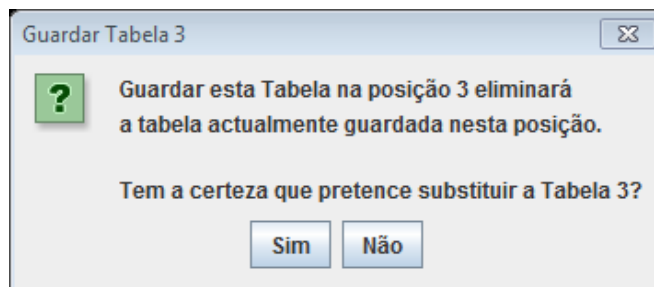


Fig. 56 - Mensagem de confirmação inserida para colmatar as falhas de usabilidade identificadas em a), da aplicação ParTec.

4.2. CONCLUSÃO

A avaliação de usabilidade consiste numa ferramenta essencial na conceção de *software*, visto que é a forma mais eficiente de garantir que uma aplicação será minimamente funcional. Realizando uma boa avaliação de usabilidade existe uma maior probabilidade de que o *software* disponibilizará uma boa experiência ao utilizador final, o que deve ser sempre o objetivo principal no processo de desenvolvimento de *software*.

5. PUBLICAÇÃO DAS APLICAÇÕES NUMA PÁGINA WEB

Com o objetivo de facilitar o acesso às aplicações criadas foi criada uma página *web*, onde é possível descarregar as aplicações bem como obter informações acerca das mesmas. A estrutura utilizada para conceção da página *web* está representada no diagrama da **Fig. 57** e conta com os seguintes conteúdos:

- **Home** – Esta é a página principal do *site* (*homepage*) onde encontra-se disponível uma breve introdução ao trabalho bem como uma pequena descrição de cada uma das aplicações desenvolvidas.
 - **KarnUMa** – Este é o menu no qual encontram-se todas as ligações referentes à aplicação KarnUma.
 - *Apresentação* – Nesta página é feita a apresentação à aplicação KarnUMa.
 - *KarnUma PC* – Nesta secção da página *web* encontram-se descritas as funcionalidades de que a aplicação KarnUMa na versão PC dispõe bem como uma galeria de imagens desta mesma versão da aplicação.
 - *Pocket KarnUMa - Telemóvel* – Conta com conteúdos idênticos aos referidos em KarnUMa PC, mas desta feita para a versão Pocket KarnUma (telemóvel).
 - **ParTec** – Menu de acesso às páginas referentes à aplicação ParTec.
 - *Apresentação* – Página onde se encontra a apresentação da aplicação ParTec.
 - *Características* – Secção onde se encontram as funcionalidades da aplicação ParTec e uma galeria de imagens deste *software*.
 - **Downloads** – Menu de acesso aos *downloads* das aplicações
 - *KarnUMa* – Página onde é possível descarregar todas versões aplicação KarnUma.
 - *ParTec* – Página onde é possível descarregar todas versões da aplicação ParTec.
 - **Autores** – Identificação e apresentação dos autores.

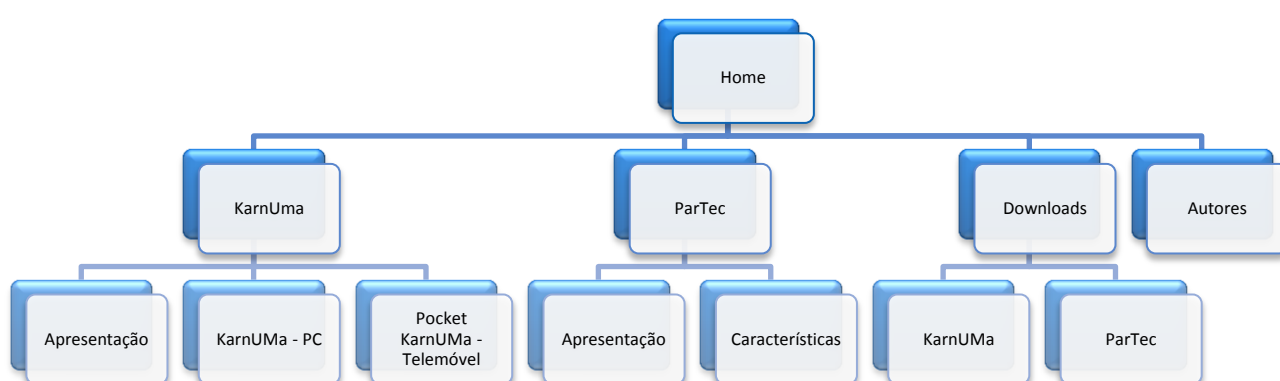


Fig. 57 - Diagrama Conceptual da página web.

5.1. CONCEÇÃO DA PÁGINA WEB

Com o objetivo de criar um *website* capaz de oferecer uma boa experiência de utilização tentou-se seguir os “10 princípios para um *web design* eficaz” (Friedman, 2008). Segundo Friedman a usabilidade e utilidade de uma página *web* é que são responsáveis por delinear o seu sucesso ou fracasso. Levando isto em consideração a conceção de um *site* deve ser centrada no utilizador.

Seguem os 10 princípios para um *web design* eficaz:

1. **Não obrigar os utilizadores a pensar** – O *design* de uma página *web* não deve levantar questões, ou seja este deve ser óbvio e autoexplicativo. Para seguir este princípio optou-se por disponibilizar na *homepage* (Fig. 58) o acesso a todos os conteúdos que o utilizador poderia procurar na página criada evitando assim que este tenha que navegar pelo *site* à procura do conteúdo X ou Y. Este acesso é feito através do menu superior que disponibiliza todos principais temas da página *web*.

2. **Não desperdice a paciência dos utilizadores** – Este princípio diz que é necessário aproveitar ao máximo cada segundo da atenção do utilizador, como tal na página *web* criada optou-se por fornecer um acesso rápido aos conteúdos pretendidos. Por exemplo se o utilizar pretender simplesmente descarregar uma das aplicações, este pode aceder à página de *downloads* executando apenas um passo visto que o seu acesso é possível através do menu principal.

3. **Gerir a concentração a atenção dos utilizadores** – Numa página *web* existem conteúdos dinâmicos bem como conteúdos estáticos, como é óbvio os primeiros chamam mais a atenção que os segundos bem como imagens recebem mais atenção do que texto. Na conceção da página devem-se apresentar os elementos mais importantes de forma a captar a maior atenção por parte do utilizador. Pode-se verificar a aplicação deste princípio na *homepage* que conta com uma animação em grande plano onde são apresentadas as aplicações KarnUMA e ParTec, os principais elementos deste *website*.

4. **Esforce-se por expor as funcionalidades** – Este princípio indica que é fundamental, para um *design* de sucesso, permitir que o utilizador perceba claramente que funcionalidades estão disponíveis. Uma vez mais tentou-se alcançar este objetivo através do menu principal que disponibiliza uma ligação a todos os conteúdos da página *web*.



Fig. 58 - Homepage da página web.

5. **Utilize uma escrita eficaz** – A *web* desempenha uma papel distinto dos documentos impressos como tal a escrita numa página *web* deve ser curta e concisa utilizando uma linguagem simples e objetiva, deve-se ainda separar os conteúdos por categorias. Na página *web* criada optou-se por apresentar pequenas porções de texto por cada página e por exemplo, ao apresentar uma aplicação optou-se por colocar as suas funcionalidades numa outra secção acompanhada por imagens ilustrativas.

6. **Mantenha a simplicidade** – Regra geral os utilizadores acedem a um *site* pelo seu conteúdo e não pelo seu especto, portanto deve-se procurar a simplicidade ao invés da complexidade. Com um visual carregado existe uma maior probabilidade que a informação essencial passe despercebida. Olhando para a **Fig. 58** pode-se verificar que a página tem um visual simples e não sobrecarregado.

7. **Não tenha medo dos espaços brancos** – Os espaços brancos são muito importantes, pois não só reduzem a carga cognitiva como facilitam a perceção da informação apresentada da no ecrã. A página criada conta com espaços brancos suficientes para facilitar a sua perceção o que acaba também por torna-la mais apelativa.

8. **Comunicar eficazmente com uma “linguagem visível”** – Por linguagem visível entende-se o conteúdo apresentado no ecrã. Este deve aparecer de forma organizada, deve-se fazer os possíveis por fornecer o máximo de informação com mínimo de elementos visíveis mantendo a simplicidade. A escolha dos tipos de letra bem como cores a utilizar na página devem garantir que esta será legível assegurando assim que a página consiga comunicar a informação pretendida. Na página criada optou-se por, à exceção dos logotipos, utilizar apenas dois tipos de letra e manter uma sintonia ao nível das cores utilizadas. Resta dizer ainda que todas as páginas do *site* obedecem à mesma estrutura.

As convenções são nossas amigas – A utilização de convenções diminui a “curva de aprendizagem”, pois se um utilizador encontrar uma *interface* com uma estrutura que lhe é familiar, ser-lhe-á mais fácil utilizar essa mesma *interface*. A partir do momento que a página *web* foi criada recorrendo a um gestor de conteúdos e como tal obedece a um *template* a sua estrutura segue um conjunto de convenções.

9. **Teste cedo, teste geralmente** – A realização de testes ajuda a deteção de falhas na conceção de uma *interface*. Assim como as aplicações desenvolvidas, a página *web* foi objeto de testes.

5.2. ALOJAMENTO DA PÁGINA WEB

O alojamento dá página *web* criada requer um servidor que disponibilize os seguintes serviços (JoomlaPT, 2012):

- Linguagem *PHP* numa versão mínima 5.2.4;
- Bases de dados *MySQL* numa versão mínima 5.0.4;
- Apache numa versão mínima 2.0;
- *ISS (Internet Information Services)* da *Microsoft* numa versão mínima 7.

Resta referir que esta página *web* pode ser acedida em: http://cee.uma.pt/morgado/aplicacoes_sd/.

CONCLUSÃO

Este projeto vem demonstrar as grandes potencialidades do recurso ao *E-Learning* no ensino. Recorrer ao *E-Learning* não é mais do que aproveitar uma “infraestrutura” já implementada na nossa sociedade. As tecnologias, cada vez mais, representam uma componente indispensável à nossa sociedade, desempenhando já inúmeras funcionalidades como instrumento de trabalho, meio de comunicação, forma de entretenimento, etc., porque não desempenhar mais um papel fundamental como é o caso da educação. O *M-Learning* assenta exatamente no mesmo princípio, não fosse este uma derivação do *E-Learning* que recorre às tecnologias móveis. O recurso a estas novas plataformas de ensino só vem reforçar e evoluir o ensino para um novo patamar.

Ao criar uma aplicação sobre a ferramenta de Sistemas Digitais, mapas de Karnaugh, pretendia-se algo mais do que um *software* que, dado um determinado problema, se limitasse a devolver a solução. O objetivo passava por uma aplicação que fosse capaz demonstrar ao seu utilizador como utilizar corretamente esta ferramenta de Sistemas Digitais, permitindo também que o próprio utilizador resolvesse mapas de Karnaugh, com a aplicação a validar os resultados. Analisando os objetivos pode-se afirmar que aplicação criada cumpre as expectativas visto que conta com um modo denominado “Ver solução (det.)” que apresenta a solução de um mapa de Karnaugh passo a passo, informando o utilizador do que se sucede em cada passo. Um outro aspeto importante a evidenciar é o facto de a aplicação contar com dois modos que permitem ao utilizador resolver um mapa. Em que no primeiro o utilizador resolve o mapa executando todos os passos da mesma forma que sucede quando é demonstrada a solução passo a passo, este modo é mais indicado para uma fase inicial da aprendizagem desta matéria. Já o segundo modo de resolução permite resolver num único passo, neste caso todos os passos serão executados mentalmente, tendo o utilizador que indicar somente a solução, este modo é mais indicado para verificar os conhecimentos adquiridos. Ainda relativamente a esta aplicação é importante referir que houve uma tentativa de implementar todas aquelas funcionalidades que foram consideradas mais relevantes nas aplicações referidas no estado de arte.

Terminada a aplicação sobre os mapas de Karnaugh pretendia-se transferi-la para um novo formato, o dos terminais móveis. Converter esta aplicação para este formato apresentava como principais desafios as próprias limitações dos equipamentos quando comparados com um computador, como é o caso das dimensões, resoluções e formatos do ecrã, a menor memória e principalmente conseguir desenvolver a aplicação numa plataforma que a tornasse disponível a uma maior percentagem da comunidade de destino, ou seja, a comunidade académica. Inicialmente a decisão recaiu sobre o *Java* para terminais móveis (*J2ME*), mas posteriormente verificando-se o elevado crescimento da plataforma *Android* nestes mesmos terminais e a facilidade de conversão, decidiu-se disponibilizar esta aplicação igualmente nesta plataforma. Dados os restantes desafios que a conversão implica, não foi possível disponibilizar todas as funcionalidades que existiam na versão PC, sendo as principais ausências os modos que permitem obter a resolução assim como o de resolver passo a passo. Ainda assim considera-se que o resultado foi uma aplicação interessante e completa dado o formato a que se destina, o dos terminais móveis.

Ao desenvolver uma aplicação sobre a Técnica das Partições os objetivos iniciais eram semelhantes aos da aplicação sobre os Mapas de Karnaugh, contudo por questões de calendário não foi possível desenvolver uma aplicação tão completa. Assim sendo esta aplicação permite ao seu utilizador obter a solução da correta utilização desta técnica, oferecendo ainda a possibilidade de verificar os detalhes da sua execução. Esta

aplicação conta ainda com outras funcionalidades interessantes, ficando a faltar um modo que permitisse ao próprio utilizador executar esta técnica.

Terminado o desenvolvimento das aplicações, pretendia-se disponibilizá-las à comunidade de destino, para tal foi criada uma página *web* a partir da qual é possível obter informações das mesmas bem como descarregá-las. Dado o elevado crescimento que os Sistemas de Gestão de Conteúdos têm demonstrado nos últimos tempos e dada a sua facilidade de utilização, construiu-se a página *web* recorrendo a esta tecnologia.

Resta referir que foi escrito um artigo referente a uma parte do trabalho aqui desenvolvido, nomeadamente as ferramentas sobre os mapas de Karnaugh. Este artigo concorreu à conferência Controló'12 tendo sido aceite para apresentação e encontra-se no anexo C. Espera-se que contribua para uma maior visibilidade deste trabalho.

REFERÊNCIAS

- Pocket Karnaugh Map*, Versão 1.0. (2002). (Emcon Emsys Technologies) Obtido em Novembro de 2010, de http://download.cnet.com/Pocket-Karnaugh-Map-for-Pocket-PC-2002-2003/3000-2251_4-10957176.html
- Karnaugh Minimizer*, Versão 2.0. (2008). (Shuriksoft) Obtido em Novembro de 2010, de <http://karnaugh.shuriksoft.com/>
- Karnaugh Map Optimizer*. (2011). (KoperSoft) Obtido em Outubro de 2011, de <http://www.kopersoft.com/Karnaugh.html>
- Ally, M. (2009). *Mobile Learning, Transforming the Delivery of Education and Training*. (M. Ally, Ed.) Athabasca University, Canada: Issues in Distance Education, AU PRESS.
- Bottentui, J. B., & Coutinho, C. P. (2009). *Do e-Learning tradicional ao e-Learning 2.0*. Obtido de http://prisma.cetac.up.pt/63_eLearning_Tradicional_ao_e_Learning_2.0_Joao_Junior_e_Clara_Coutinho.pdf
- Bowles, M. (2004). *Relearning to E-Learn - Strategies for electronic learning and knowledge* (Vol. Cap. 1). Melbourne University Publishing Ltd, Polar Design Pty Ltd.
- Cação, R., & Dias, P. J. (2003). *Introdução ao E-Learning* (1ª Edição ed.). (S. Sociedade Portuguesa de Inovação Consultadoria Empresarial e Fomento da Inovação, Ed.) Porto, Portugal: Principia, Publicações Universitárias e Científicas, Lda.
- Cajueiro, J. (24 de Agosto de 2009). Mapa de Karnaugh.
- Chirico, U. (Maio de 2011). *Ugo Chirico's Blog*. Obtido em 16 de Fevereiro de 2012, de <http://www.ugochirico.com/post/2011/05/31/LWUIT-on-Android.aspx>
- Cigic, I. (2011). *Karnaugh Map Advanced*, Versão 1.1. Obtido de <http://www.kmapadvanced.com/>
- Clark, R. (10 de Setembro de 2002). Six Principles of Effective e-Learning: What Works and Wh. *Learning Solutions, Practical Applications of Technology for Learning, Design Techniques*.
- Cochrane, T., & Bateman, R. (2010). Smartphones give you wings: Pedagogical affordances of mobile Web 2.0. *Australian Journal of Educational Technology*.
- Crenshaw, J. (19 de Agosto de 2004). *All about Quine-McCluskey*. Obtido em 14 de Março de 2011, de <http://www.eetimes.com/discussion/programmer-s-toolbox/4025004/All-about-Quine-McCluskey>
- Dias, M. (2010). *Sistemas Digitais Princípios e Prática*. FCA - Editora de Informática.
- Doran, R. W. (28 de Novembro de 2007). The Gray Code. *Journal of Universal Computer Science*, 13, 1573-1597.

- Eclipse Foundation. (2012). *About the Eclipse Foundation*. Obtido em 16 de Fevereiro de 2012, de <http://www.eclipse.org/org/>
- Enough Software. (2011). *Don't Panic Mobile Developer's Guide to the Galaxy* (9 ed.). Sogestrass, Breman, Alemanha: Enough Software.
- Friedman, V. (Janeiro de 2008). *10 Principles Of Effective Web Design*. Obtido em 25 de Abril de 2012, de <http://uxdesign.smashingmagazine.com/2008/01/31/10-principles-of-effective-web-design/>
- Google. (2012). *Android 4.0 Platform Highlights*. Obtido em 16 de Fevereiro de 2012, de <http://developer.android.com/sdk/android-4.0-highlights.html>
- Hunter, J., & McLaughlin, B. (2012). *JDOM*. Obtido em 16 de Fevereiro de 2012, de <http://www.jdom.org/index.html>
- iText Software Corp. (2011). *About iText: mission and Vision*. Obtido em 2012 de Fevereiro de 2012, de <http://itextpdf.com/about.php>
- JoomlaPT. (2012). *JoomlaPT!com Comunidade Joomla Portuguesa*. Obtido em 15 de Abril de 2012, de <http://www.joomlapt.com/sobre-o-joomla/o-que-e-joomla.html>
- Kepler, M. (2010). *Karnaugh-Minimierung*. (Technical University Ilmenau, Alemanha) Obtido em Novembro de 2010, de <http://www-ihs.theoinf.tu-ilmenau.de/~sane/projekte/karnaugh/>
- Knudsen, J. (Setembro de 2008). *The Lightweight User Interface Toolkit (LWUIT): An Introduction*. (Oracle, Editor) Obtido em 16 de Fevereiro de 2012, de http://java.sun.com/developer/technicalArticles/javame/lwuit_intro/
- Kotonya, G., & Sommerville, I. (Janeiro de 1996). Requirements Engineering Processes. *Software Engineering Journal*, XI(1), 5-18.
- Kovacevic, R. (2005). *Karnaugh Map Minimizer*, Versão 0.4. Obtido em Novembro de 2010, de <http://k-map.sourceforge.net/>
- LiteratePrograms. (2010). *Quine-McCluskey algorithm (Java)*. Obtido em 22 de Fevereiro de 2012, de [http://en.literateprograms.org/Quine-McCluskey_algorithm_\(Java\)](http://en.literateprograms.org/Quine-McCluskey_algorithm_(Java))
- Musser, J., & O'Reilly, T. (2006). *Web 2.0 Principles and Best Practices*.
- Muyinda, P. B. (2007). MLearning: pedagogical, technical and organisational hypes and realities. 24, 97-104.
- Naidu, S. (2006). *E-Learning - A Guidebook of Principles, Procedures and Practices* (2ª ed.). University of Melbourne, Australia: Commonwealth Educational Media Center for Asia (CEMCA).
- Namahn. (2011). *E-Learning - A research note*. Obtido de <http://www.namahn.com/resources/documents/note-E-Learning.pdf>
- Nielsen, J. (24 de Abril de 1994). Enhancing the Explanatory Power of Usability Heuristics. *Human Factors in Computing System*, pp. 152-158.

- Oracle. (2012). *How to Use File Choosers*. Obtido em 23 de Fevereiro de 2012, de <http://docs.oracle.com/javase/tutorial/uiswing/components/filechooser.html>
- ORACLE. (2012). *Java*. Obtido em 16 de Fevereiro de 2012, de http://www.java.com/pt_BR/about/
- ORACLE. (2012). *JAVA FOR MOBILE DEVICES*. Obtido em 16 de Fevereiro de 2012, de <http://www.oracle.com/technetwork/java/javame/javamobile/overview/getstarted/index.html>
- Park, Y. (Fevereiro de 2011). A Pedagogical Framework for Mobile Learning: Categorizing Educational Applications of Mobile Technologies into Four Types. *International Review of Research in Open and Distance Learning*.
- Passos, H. (2010). *Pequeno guia do E-Learning*. Rua das Indústrias 27 - A Venda Nova, Portugal: Cinel.
- Reis, A. I., & Ribas, R. P. (2011). *Karma*, Versão 3.61. (Logic Circuit Synthesis Labs, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Brasil) Obtido em Outubro de 2011, de http://www.inf.ufrgs.br/logics/index.php?option=com_docman&task=doc_download&gid=202&Itemid=58
- Sandes, E. (2002). *4x4 Karnaugh Minimizer*, Versão 1.0. Obtido em Outubro de 2011, de <http://www.programmersheaven.com/download/24979/1/ZipView.aspx>
- Sasamori, R. (1999). *KarnaughMap*, Versão 1.2. Obtido em Novembro de 2010, de http://www.puz.com/sw/karnaugh/karnaugh_12.htm
- Sena, A., & Torres, M. (2009). *EasyKarnaugh 3.0 - Uma ferramenta computacional para o auxílio no ensino de Mapas de Karnaugh em Lógica Digital*. Universidade Estadual de Santa Cruz, Departamento de Ciências Exatas e Tecnologias, Brasil.
- Sêrro, C., Carvalho, J., & Arroz, G. (Outubro de 2005). *Álgebra de Boole Binária, Funções booleanas*. Instituto Superior Técnico.
- Strickland, J. (2011). *Is there a Web 1.0?* Obtido em Novembro de 2011, de <http://computer.howstuffworks.com/web-101.htm>
- Tocci, R. J., & Widmer, N. S. (2001). *Digital Systems, Principles and Applications* (8º ed.). Prentice Hall.
- Tomaszewski, S. P., U.Celik, I., & Antoniou, G. E. (2003). WWW-BASED BOOLEAN FUNCTION MINIMIZATION. *BAE Systems Controls*, 13, pp. 577-583.
- Traxler, J. (2005). Defining Mobile Learning. *International Conference Mobile Learning*. Wolverhampton, Reino Unido: IADIS .
- Zubía, J. G. (2003). Educational Software for Digital Electronics: BOOLE-DEUSTO. *IEEE International Conference on Microelectronic Systems Education*.

ANEXOS

ANEXO A

Este anexo consiste num relatório gerado automaticamente pela aplicação KarnUMa.

Relatório

1. Função inicial

1.1. Tabela de Verdade:

| | A | B | C | D | E | F |
|----|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 |
| 9 | 0 | 1 | 0 | 0 | 1 | 1 |
| 10 | 0 | 1 | 0 | 1 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 | 1 | 1 |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 |
| 13 | 0 | 1 | 1 | 0 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 | 0 | 0 |
| 15 | 0 | 1 | 1 | 1 | 1 | X |
| 16 | 1 | 0 | 0 | 0 | 0 | 0 |
| 17 | 1 | 0 | 0 | 0 | 1 | 0 |
| 18 | 1 | 0 | 0 | 1 | 0 | 0 |
| 19 | 1 | 0 | 0 | 1 | 1 | 0 |
| 20 | 1 | 0 | 1 | 0 | 0 | X |
| 21 | 1 | 0 | 1 | 0 | 1 | 1 |
| 22 | 1 | 0 | 1 | 1 | 0 | 0 |
| 23 | 1 | 0 | 1 | 1 | 1 | 0 |
| 24 | 1 | 1 | 0 | 0 | 0 | 1 |
| 25 | 1 | 1 | 0 | 0 | 1 | 1 |
| 26 | 1 | 1 | 0 | 1 | 0 | 1 |
| 27 | 1 | 1 | 0 | 1 | 1 | 0 |
| 28 | 1 | 1 | 1 | 0 | 0 | 0 |
| 29 | 1 | 1 | 1 | 0 | 1 | 1 |
| 30 | 1 | 1 | 1 | 1 | 0 | 1 |
| 31 | 1 | 1 | 1 | 1 | 1 | X |

1.2. Mapa de Karnaugh:

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | X | 1 |
| 10 | 1 | 1 | 0 | 0 |

E = 0

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | X | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | X | 0 |
| 10 | 0 | 0 | 1 | 1 |

E = 1

2. Soluções

2.1. Soma de Produtos

2.1.1. Solução

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | X | 1 |
| 10 | 1 | 1 | 0 | 0 |

E = 0

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | X | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | X | 0 |
| 10 | 0 | 0 | 1 | 1 |

E = 1

$$F = \bar{A}C\bar{D}\bar{E} + A\bar{D}\bar{E} + AB\bar{C}\bar{E} + B\bar{C}D + A\bar{B}\bar{C}E + AC\bar{D}E + BD\bar{E}$$

Implicantes Primos Essenciais: $\bar{A}C\bar{D}\bar{E}$, $A\bar{D}\bar{E}$, $AB\bar{C}\bar{E}$

Implicantes Primos: $B\bar{C}D$, $A\bar{B}\bar{C}E$, $AC\bar{D}E$, $BD\bar{E}$

2.2. Produto de Somas

2.2.1. Solução

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | X | 1 |
| 10 | 1 | 1 | 0 | 0 |

E = 0

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | X | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | X | 0 |
| 10 | 0 | 0 | 1 | 1 |

E = 1

$$F = (A+B+\bar{D}) \cdot (A+C+D) \cdot (\bar{C}+\bar{D}+\bar{E}) \cdot (\bar{B}+C+D+\bar{E}) \cdot (A+\bar{C}+\bar{E}) \cdot (\bar{A}+\bar{C}+D+E) \cdot (\bar{A}+B+D+E)$$

Implicados Primos Essenciais: $(A+B+\bar{D})$, $(A+C+D)$, $(\bar{C}+\bar{D}+\bar{E})$, $(\bar{B}+C+D+\bar{E})$

Implicados Primos: $(A+\bar{C}+\bar{E})$, $(\bar{A}+\bar{C}+D+E)$, $(\bar{A}+B+D+E)$

3. Código LDH (Linguagem de Descrição de Hardware)

3.1. Código Verilog

3.1.1. Código Verilog - Soma de Produtos

```
/*-----
----- Código gerado por Karnuma V 0.9 -----
-----*/
```

```
module Karnough(A, B, C, D, E, F);
    input A;
    input B;
    input C;
    input D;
    input E;
    output F;

    and w1, !A, C, !D, !E);
    and w2, A, D, !E);
    and w3, A, B, !C, !E);
    and w4, B, !C, D);
    and w5, A, !B, !C, E);
    and w6, A, C, !D, E);
    and w7, B, D, !E);
    or (F, w1, w2, w3, w4, w5, w6, w7);

endmodule
```



3.1.2. Código Verilog - Produto de Somas

```
/*-----  
----- Código gerado por Karnuma V 0.9 -----  
-----*/  
  
module Karnaugh(A, B, C, D, E, F);  
  
    input A;  
    input B;  
    input C;  
    input D;  
    input E;  
    output F;  
  
    or (w1, A, B, !D);  
    or (w2, A, C, D);  
    or (w3, !C, !D, !E);  
    or (w4, !B, C, D, !E);  
    or (w5, A, !C, !E);  
    or (w6, !A, !C, D, E);  
    or (w7, !A, B, D, E);  
    and (F, w1, w2, w3, w4, w5, w6, w7);  
  
endmodule
```

3.2. Código VHDL

3.2.1. Código VHDL - Soma de Produtos

```
/*-----  
----- Código gerado por Karnuma V 0.9 -----  
-----*/  
  
library IEEE;  
use IEEE.SDT_LOGIC_1164.all;  
  
entity KarnaughFunction is  
    port (A, B, C, D, E : in STD_LOGIC;  
          F: out STD_LOGIC);  
end KarnaughFunction;  
  
architecture synth of KarnaughFunction is  
begin  
    F <=((not A) and C and (not D) and (not E)) or  
        (A and D and (not E)) or  
        (A and B and (not C) and (not E)) or  
        (B and (not C) and D) or  
        (A and (not B) and (not C) and E) or  
        (A and C and (not D) and E) or  
        (B and D and (not E));  
end;
```

3.2.2. Código VHDL - Produto de Somas

```
/*-----  
----- Código gerado por Karnuma V 0.9 -----  
-----*/  
  
library IEEE;  
use IEEE.SDT_LOGIC_1164.all;  
  
entity KarnaughFunction is  
    port (A, B, C, D, E : in STD_LOGIC;
```



```
F: out STD_LOGIC);  
end KarnaughFunction;  
  
architecture synth of KarnaughFunction is  
begin  
    F <=(A or B or (not D)) and  
        (A or C or D) and  
        ((not C) or (not D) or (not E)) and  
        ((not B) or C or D or (not E)) and  
        (A or (not C) or (not E)) and  
        ((not A) or (not C) or D or E) and  
        ((not A) or B or D or E);  
end;
```

ANEXO B

Este anexo consiste no mesmo relatório gerado pela aplicação que consta no Anexo A, mas desta feita utilizando o idioma Inglês.

Report

1. Initial function

1.1. Truth Table:

| | A | B | C | D | E | F |
|----|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 |
| 9 | 0 | 1 | 0 | 0 | 1 | 1 |
| 10 | 0 | 1 | 0 | 1 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 | 1 | 1 |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 |
| 13 | 0 | 1 | 1 | 0 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 | 0 | 0 |
| 15 | 0 | 1 | 1 | 1 | 1 | X |
| 16 | 1 | 0 | 0 | 0 | 0 | 0 |
| 17 | 1 | 0 | 0 | 0 | 1 | 0 |
| 18 | 1 | 0 | 0 | 1 | 0 | 0 |
| 19 | 1 | 0 | 0 | 1 | 1 | 0 |
| 20 | 1 | 0 | 1 | 0 | 0 | X |
| 21 | 1 | 0 | 1 | 0 | 1 | 1 |
| 22 | 1 | 0 | 1 | 1 | 0 | 0 |
| 23 | 1 | 0 | 1 | 1 | 1 | 0 |
| 24 | 1 | 1 | 0 | 0 | 0 | 1 |
| 25 | 1 | 1 | 0 | 0 | 1 | 1 |
| 26 | 1 | 1 | 0 | 1 | 0 | 1 |
| 27 | 1 | 1 | 0 | 1 | 1 | 0 |
| 28 | 1 | 1 | 1 | 0 | 0 | 0 |
| 29 | 1 | 1 | 1 | 0 | 1 | 1 |
| 30 | 1 | 1 | 1 | 1 | 0 | 1 |
| 31 | 1 | 1 | 1 | 1 | 1 | X |

1.2. Karnaugh Map:

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | X | 1 |
| 10 | 1 | 1 | 0 | 0 |

$E = 0$

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | X | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | X | 0 |
| 10 | 0 | 0 | 1 | 1 |

$E = 1$

2. Solutions

2.1. Sum-of-Products

2.1.1. Solution

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | X | 1 |
| 10 | 1 | 1 | 0 | 0 |

$E = 0$

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | X | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | X | 0 |
| 10 | 0 | 0 | 1 | 1 |

$E = 1$

$$F = \bar{A}C\bar{D}\bar{E} + A\bar{D}\bar{E} + AB\bar{C}\bar{E} + B\bar{C}D + A\bar{B}\bar{C}E + AC\bar{D}E + BD\bar{E}$$

Essential Prime null: $\bar{A}C\bar{D}\bar{E}$, $A\bar{D}\bar{E}$, $AB\bar{C}\bar{E}$

Prime null: $B\bar{C}D$, $A\bar{B}\bar{C}E$, $AC\bar{D}E$, $BD\bar{E}$

2.2. Product-of-Sums

2.2.1. Solution

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | X | 1 |
| 10 | 1 | 1 | 0 | 0 |

E = 0

| CD / AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | X | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | X | 0 |
| 10 | 0 | 0 | 1 | 1 |

E = 1

$$F = (A+B+\bar{D}) \cdot (A+C+D) \cdot (\bar{C}+\bar{D}+\bar{E}) \cdot (\bar{B}+C+D+\bar{E}) \cdot (A+\bar{C}+\bar{E}) \cdot (\bar{A}+\bar{C}+D+E) \cdot (\bar{A}+B+D+E)$$

Essential Prime null: $(A+B+\bar{D})$, $(A+C+D)$, $(\bar{C}+\bar{D}+\bar{E})$, $(\bar{B}+C+D+\bar{E})$

Prime null: $(A+\bar{C}+\bar{E})$, $(\bar{A}+\bar{C}+D+E)$, $(\bar{A}+B+D+E)$

3. HDL (Hardware Description Language) code

3.1. Verilog Code

3.1.1. Verilog Code - Sum-of-Products

```

/*-----
----- Code generated by Karnuma V 0.9 -----
-----*/
    
```

```

module Karnough(A, B, C, D, E, F);

    input A;
    input B;
    input C;
    input D;
    input E;
    output F;

    and w1, !A, C, !D, !E);
    and w2, A, D, !E);
    and w3, A, B, !C, !E);
    and w4, B, !C, D);
    and w5, A, !B, !C, E);
    and w6, A, C, !D, E);
    and w7, B, D, !E);
    or (F, w1, w2, w3, w4, w5, w6, w7);

endmodule
    
```

3.1.2. Verilog Code - Product-of-Sums



```
/*-----  
----- Code generated by Karnuma V 0.9 -----  
-----*/  
  
module Karnaugh(A, B, C, D, E, F);  
  
    input A;  
    input B;  
    input C;  
    input D;  
    input E;  
    output F;  
  
    or (w1, A, B, !D);  
    or (w2, A, C, D);  
    or (w3, !C, !D, !E);  
    or (w4, !B, C, D, !E);  
    or (w5, A, !C, !E);  
    or (w6, !A, !C, D, E);  
    or (w7, !A, B, D, E);  
    and (F, w1, w2, w3, w4, w5, w6, w7);  
  
endmodule
```

3.2. VHDL Code

3.2.1. VHDL Code - Sum-of-Products

```
/*-----  
----- Code generated by Karnuma V 0.9 -----  
-----*/  
  
library IEEE;  
use IEEE.SDT_LOGIC_1164.all;  
  
entity KarnaughFunction is  
    port (A, B, C, D, E : in STD_LOGIC;  
          F: out STD_LOGIC);  
end KarnaughFunction;  
  
architecture synth of KarnaughFunction is  
begin  
    F <=((not A) and C and (not D) and (not E)) or  
        (A and D and (not E)) or  
        (A and B and (not C) and (not E)) or  
        (B and (not C) and D) or  
        (A and (not B) and (not C) and E) or  
        (A and C and (not D) and E) or  
        (B and D and (not E));  
end;
```

3.2.2. VHDL Code - Product-of-Sums

```
/*-----  
----- Code generated by Karnuma V 0.9 -----  
-----*/  
  
library IEEE;  
use IEEE.SDT_LOGIC_1164.all;  
  
entity KarnaughFunction is  
    port (A, B, C, D, E : in STD_LOGIC;  
          F: out STD_LOGIC);  
end KarnaughFunction;  
  
architecture synth of KarnaughFunction is
```

*begin*

```
F <=(A or B or (not D)) and
(A or C or D) and
((not C) or (not D) or (not E)) and
((not B) or C or D or (not E)) and
(A or (not C) or (not E)) and
((not A) or (not C) or D or E) and
((not A) or B or D or E);
```

end;

ANEXO C

Artigo “*KARNUMA AN EDUCATIONAL TOOL FOR DIGITAL SYSTEMS*”, escrito para a 10ª conferência Portuguesa de Controlo Automático – Controlo’12.

KARNUMA AN EDUCATIONAL TOOL FOR DIGITAL SYSTEMS

Valentim Freitas and Fernando Morgado-Dias

*Madeira Interactive Technologies Institute and Centro de Competências de Ciências Exactas e da Engenharia,
Universidade da Madeira
Campus da Penteada, 9000-039 Funchal, Madeira, Portugal.
Tel: +351 291-705150/1, Fax: +351 291-705199
morgado@uma.pt*

Abstract: This paper presents an application called *KarnUMa* which aims to demonstrate the operation of the Karnaugh Maps which are used to simplify Boolean algebra expressions. This application is available in two variants with different target platforms, the first, *KarnUMa* available for computer and the second *Pocket KarnUMa* available for mobile terminals in either Midlet or Android Package. This paper addresses the existing applications in this area that served as inspiration, the technologies chosen for the development of this software, and finally it presents the *KarnUMa* applications identifying the new features introduced by them as well as new approaches that were made to already existing features in the reference applications. . Copyright CONTROLLO 2012.

Keywords: digital systems, computer software, educational aids, e-learning.

1. INTRODUCTION

In the last decade there has been a constant technological evolution in what regards computers and mobile terminals. As such it became advantageous to use these new technologies, which have become part of our everyday life, as sources of information for education. These technologies have shown great potential when harnessed for pedagogical purposes hence the emergence of concepts such as E-Learning and M-Learning. E-Learning is a term used to refer to "electronic learning" (Bowles, 2004), i.e. when there is the use of electronic technologies for educational purposes, on the other hand when talking about M-Learning (Mobile -Learning) we refer to the use of mobile terminals to this same end, therefore M-Learning can be defined as "educational source, where a single technology or dominant technologies are handheld devices" (Traxler, Defining Mobile Learning, 2005).

The work presented in this paper is a part of a master's thesis in Computer Science in which the purpose was to create an application to use as a component of E-Learning, since this application would aim to aid learning the Karnaugh Maps tool used in the subject of Digital Systems.

Aiming to help teach the use of Karnaugh maps tool, the application should not limit itself to presenting results, but be able to demonstrate how to achieve

these results. With this in mind the application should be able to display the whole process of solving a Karnaugh Map as well as providing detailed information for each step of this process. Another important aspect for teaching is the practice, as such the application should enable the user himself to perform the Karnaugh Map resolution process. Hence the motivation to develop a new application of Karnaugh maps lays in providing software from which the user could learn to use this tool and also put his knowledge to the test.

2. STATE OF THE ART

In a first stage of this work the aim was to check all the existing applications within the same area in order to determine what has already been achieved. This process consisted in testing the multiple applications found to identify features that made them stand out from each other giving primary importance to the ones that had some educational component. After this stage the following applications have been identified as reference cases to the development of the *KarnUMa* applications.

2.1. Karnaugh Minimizer

The application shown in Fig. 1 is available in both free and commercial releases whereas the last one offers more features. Since the applications presented in this article are intended to the academic community, only the free version of the application developed by (ShurikSoft, 2008) was taken into consideration. The Karnaugh Minimizer was regarded as a reference application for its wide range of features among which the following ones should be highlighted:

- Allows the introduction of the Karnaugh Map's content by the map itself and also by a truth table;
- Indifferent terms can be used;
- Solves maps up to four variables;
- Provides solutions using the Sum of Products method and the Product of Sums method;
- It has options to assist the map's filling such as inverting its content or deleting more than one cell at a time;
- It generates a report with the relevant information from the Karnaugh map such as the initial function, the Karnaugh Map itself, the Truth table and the map's Solution with the representation of each minimized function's term.
- It generates the circuit schematics corresponding to the solution of the map.

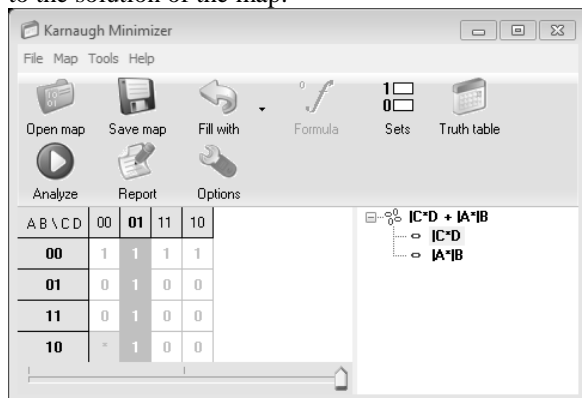


Fig. 1. Karnaugh Minimizer (ShurikSoft, 2008).

At the time of this paper's writing this software was in the 2.0 release.

2.2. EasyKarnaugh

The approach taken to Karnaugh maps by (Sena *et* Torres, 2009) is represented in Fig. 2. This approach stands out from the previous application for being targeted for education. This application aims to teach its users how to solve a Karnaugh map. Taking into consideration the purpose of this application it can be said that its design was made with the objective of E-Learning given that the description made by the authors of the application is as follows "A computational tool to aid in the teaching of Karnaugh maps in Computational Logic" (Sena *et* Torres, 2009). This application has four main components:

- Presentation of Karnaugh Map's Theory;
- Solution's visualization in which the solution's

expression is shown as well as each Implicant is properly marked in the Karnaugh map;

- A component that allows the user to try to deduce the Boolean expression that corresponds to the Karnaugh map's solution;
- A component in which the user tries to form the solution's Implicants of the Karnaugh map. A fault was detected in this section since it only accepts Implicants that are part of the solution set by the computer, so when there's more than one solution to a Karnaugh Map the program doesn't accept Implicants that only belong to an alternative solution.

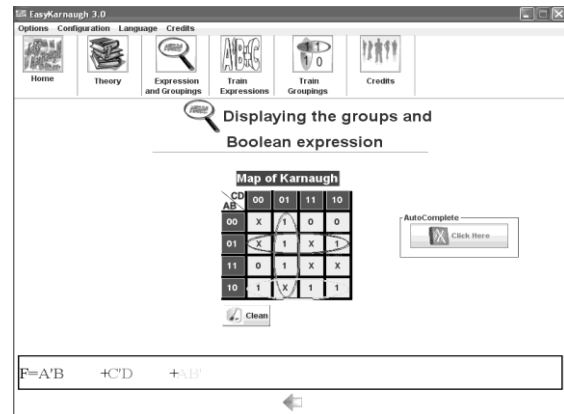


Fig. 2. EasyKarnaugh (Sena *et* Torres, 2009).

This application solves maps from 3 to 4 variables, works with both methods Sum of Products and Product of Sums, supports terms with indifferent value and also has the functionality of filling in the map's content automatically, an appreciated feature when you just wish to test your knowledge of solving Karnaugh maps.

At the time of this paper's writing this application is at version 3.0.

2.3. Karma

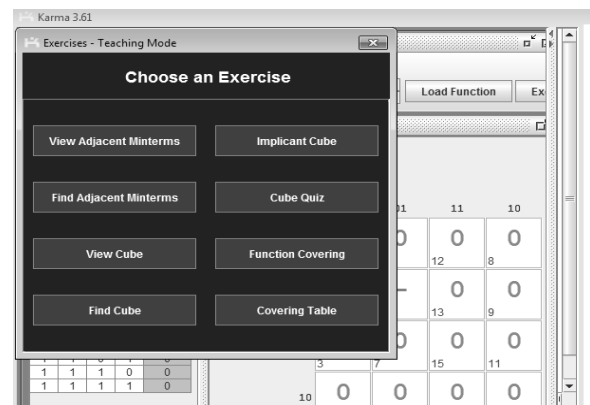


Fig. 3. Karma (Reis *et* Ribas, 2011).

The Karma application developed by (Reis *et* Ribas, 2011), is a powerful application that includes among its features a component devoted to Karnaugh maps. This component supplies the user with the Karnaugh Map's solution with the usual data like the solution's boolean expression, the properly filled map, but it also provides other detailed data as the Quine

McCluskey process steps and its Prime Implicants Table.

Within all its features the most relevant to this study consists of a series of exercises that this application provides to its user. This feature offers a considerable variety of good exercises for the user to test his knowledge. This feature provides eight different types of exercises as seen in Fig. 3

At the time of writing this article this application was in its 3.61 version.

3. CHOOSING TECHNOLOGIES

Considering that the outcome of this work is intended to be used as an auxiliary tool for learning the choice of technologies to use focused on the following criteria:

- The technologies used shall be part of the intended community's everyday life and the choice should focus on the one that is available to the highest possible proportion of individuals in this community;
- Acknowledging that this is an academic community, the chosen technologies should be based on the principle of Free Software. Thus the developed applications should not require pre-installation of any commercial software.

In order to meet the above criteria the chosen technologies were as follows:

- **Computer Application** – For this platform, the application was developed in Java. Currently, this technology is found in Release 6. The choice fell on Java because it offers versatility, efficiency and portability. In order to run *KarnUMa* on a computer it only requires an installed Java virtual machine, known as Java Runtime Environment (JRE). Note that this technology is free (ORACLE, 2012).
- **Mobile Application** – For this platform the choice fell on the Java MicroEdition (ME). This (ORACLE, 2012) technology is currently on its 3.0.5 version. The main factor for choosing this technology lies in the fact that currently, “around 80% of all mobile handsets worldwide support the mobile Java standard (Java ME formerly known as J2ME)” (Enough Software, 2011). Yet in order to ensure greater accessibility of the application, it was adapted to Android due to the high growth this technology has recently sustained. Current Android version is 4.0 (Google, 2012).

The development of the *KarnUMa* applications in both PC and handset version required the usage of source code obtained from (LiteratePrograms, 2010) to implement the method of Quine Mckluskey, method which is used as background for the Karnaugh map's resolution.

4. KARNUMA APPLICATION (PC VERSION)

This section contains a description of how *KarnUMa* application works.

4.1. New Map

When starting the application, a form is available to the user where he can choose the Karnaugh map's specifications. Fig. 4 shows the available options on this form:

- **Number of variables** – The user can choose from 3 to 6 variables;
- **Method of resolution** – The user can choose between Sum of Products and Product of Sums;
- **Order of input variables** – Lets the user choose the map's orientation;
- **Filling method** – To choose between introducing the Karnaugh map's content by the map itself or by a Truth table.

Once the Karnaugh map's specifications have been chosen, the user can proceed to the next screen by pressing "Begin" at the bottom of the screen. Note that the user could simply leave the default specifications.

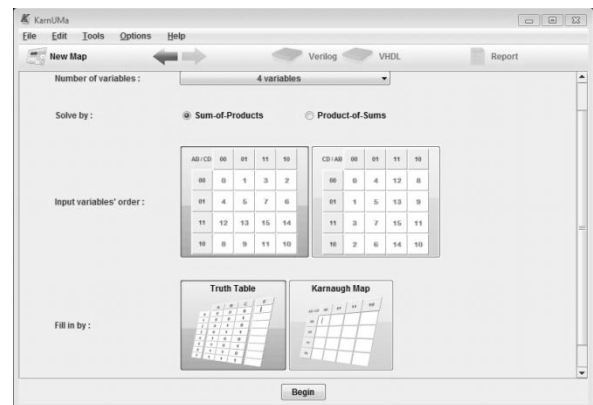


Fig. 4. *KarnUMa*'s new map screen.

4.2. Filling the map's content

After choosing the specifications in the previous screen (Fig. 4) the user will insert the Karnaugh map's content. This filling can be done in the map itself or through a Truth table as shown in Fig. 5.

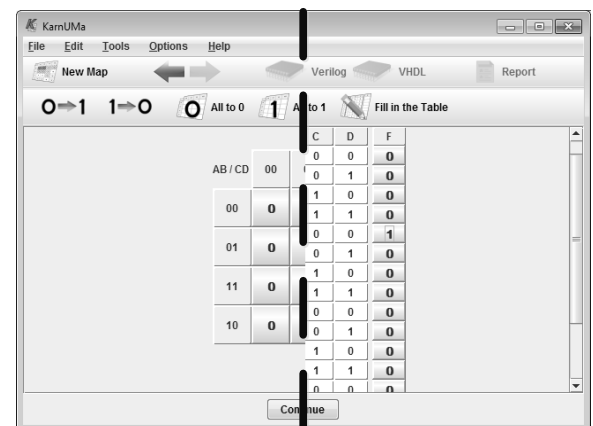


Fig. 5. Filling the Karnaugh map either by the map itself (right) or by Truth table (left).

Regardless of the chosen input method the procedure is the same, the user just needs to press the map or

table cells he wishes to change the value, or choose any of the available options to quickly fill the map or table:

- **0 → 1** – This option allows the user to modify all terms of value 0 to 1;
- **1 → 0** – Makes exactly the opposite of the previous option, i.e. modifies all terms of value 1 to 0;
- **All the 0** – Sets all terms to 0;
- **Everything for 1** – Sets all terms to one;
- **Automatically Fill** – Fill the entire map or table randomly.

4.3. Choose between learning and testing knowledge

In this section the screen is divided, on the left there's the Truth table, and on the right side the matching Karnaugh map. Here the user can choose from four available options described in the following paragraphs.

4.3.1. View Solution



Fig. 6. View solution.

Once the user chooses this option the application presents him with the Karnaugh map's solution as seen in Fig. 6.

When viewing the solution a set of new features is available. On the map's toolbar the user can choose the following features:

- When a map has more than one solution the user may watch one of these solutions at a time simply by pressing the desired solution ("Solution 1" or "Solution 2");
- The solution can be viewed by the Sum of Products or by the Product of Sums by pressing buttons "1" and "0";
- The user can also change the map orientation (this option is always available for as long as a Karnaugh map is shown).

In this application's section there are yet another set of important features available on the top toolbar:

- From the buttons "Verilog" and "VHDL" the user can get the solution's source code in the respective Hardware Description Language;
- By pressing "Report" the user may also get a full report with all Karnaugh map's relevant information. Finally it remains to be said that below the Karnaugh

map is the solution's boolean expression in which there is a correspondence between the Implicant's color in the expression and its representation on the map. The user can click on each Implicant to see it exclusively represented on the map. To redisplay the representation of all Implicants simply re-press the Implicant that is selected or press "All".

4.3.2. View Solution (in detail)

This option allows a detailed view of how to get the map's solution. It consists of several steps where each step corresponds to one of the needed stages to reach the Karnaugh map's solution. As can be seen in Fig. 7, this option displays the screen organized as follows, on the right side is the description of the current stage of the resolution process. At the bottom of the description is usually a button that allows for the definition of the main element of this stage of resolution. On the left side is the map depicting the current step of the resolution.

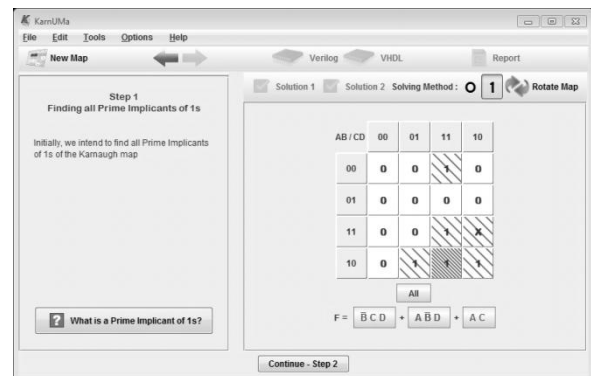


Fig. 7. First stage from "View Solution (in detail)".

This option typically comprises the following steps:

- 1st – Presentation of all Prime Implicants;
- 2nd – Identification of which Prime Implicants are Essential. When the Karnaugh map's solution lacks of Essential prime Implicants the user is properly informed and the application proceeds to the next step;
- 3rd – Of the remaining Prime Implicants (those that aren't Essential), the ones necessary to obtain the final solution are selected. This step may also not be performed, in this case when the solution consists solely of Essential Prime Implicants. The user is informed and the final solution presented.

Once this step by step solution process is finished, the application ends on the same screen as described in section 4.3.1.

4.3.3. Solve Map

This option presented in Fig. 8, allows the user to solve the Karnaugh map quickly, i.e. in a single step. For this the user must form Implicants until he achieves the solution established by application or an equivalent one. To form an Implicant the user must select the map cells he wants to constitute the Implicant and then press "Validate Implicant" where

the application will validate it. If it isn't well-formed the application informs the user of the errors found. In this mode you should get exactly the Implicants needed to achieve a minimal solution, to help in this matter there's a feedback bar at the bottom of the screen that indicates how many Prime and Essential Prime Implicants are needed to achieve the minimal solution. This bar is filled in accordance with the user's progress, and once it is completely filled the "Check Solution" option is unlocked. The most important aspects in this section consist of the feedback bar mentioned above and in the fact that the application accepts any minimal solution, not being restricted to the solution previously established by it.

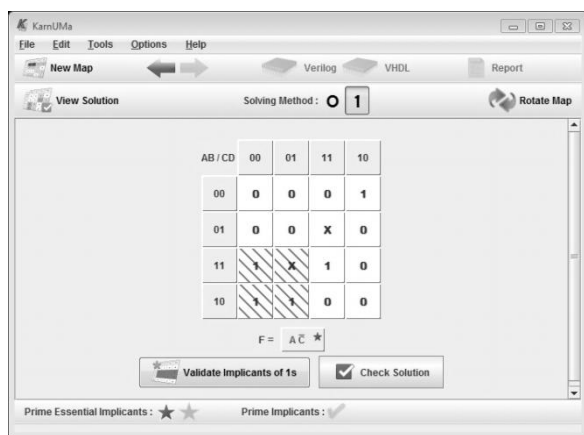


Fig. 8. Solve map.

4.3.4. Solve Map in detail

This user resolution mode follows exactly the same steps described in section 4.3.2, but this time it is the user that must complete each step. This allows him to test the knowledge acquired in 4.3.2, knowledge which is relevant for obtaining a correct Karnaugh map's Solution.

This mode also counts with the same feedback described at point 4.3.3 *Solve Map*, but this time with the proper elements of each resolution's step. Once finished every step, it becomes possible to validate the final solution.

4.4. Other important details

Other relevant features which the application provides are the availability of two languages (Portuguese and English) that can be accessed from the Options menu, and the ability to Redo and Undo, two features increasingly common these days but always important.

5. POCKET KARNUMA APPLICATION (HANDSET)

This version of the application is an adaptation of the PC application. As such it consists of the counterpart features that could be implemented and that are

relevant in this platform. Since these are the same features their description in this paper's section will be more succinct.

5.1. New Map

As can be seen in Fig. 9, this screen provides exactly the same Karnaugh map's specifications as the previously displayed version, with the only exception to the number of variables which in this version can only be 3 or 4. This difference is due to the fact that the screen dimensions don't allow a good representation of more than four variables in the majority of handset terminals.



Fig. 9. (Left) New Map (Pocket KarnUMA) Fig. 10. (Right) Solve map.

5.2. Filling the map's content

The screen shown in Fig. 11 covers exactly the same options described in 4.2 *Filling the map's content* of the PC application.

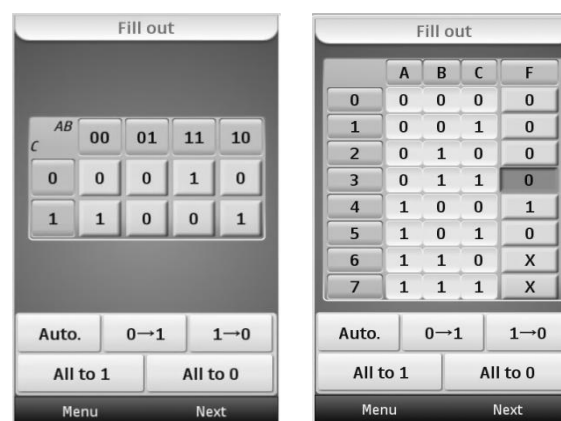


Fig. 11. Fill out the Karnaugh map.

5.3. Solve or get the solution

Unlike the PC version this one doesn't have a major educational component since this section only provides the option to view the solution and solve the map. Due to their complexity the options to see the detailed solution and solve in detail aren't included in this version.

5.3.1. View Solution

Again, this option has the same features described in section 4.3.1, except for the possibility of obtaining the HDL source code and get a Karnaugh map's report. These features were not included since they were not of importance in this version of the application.

5.3.2. Solve Map

This option operates in the same way as reported in 4.3.3 *Solve Map*. As can be seen in Fig. 10 there is a different organization of the display since the feedback bar now placed above the Karnaugh map and due to its small size each bar element's description can only be seen in the Help menu.

5.4. Other important details

Similarly to what happens with the PC version, this one has the same two languages and allows Undo and Redo actions.

In addition to the features not included in this release, the most important difference lies in the different organization of the screen due to its small size. The differences in the organization focused mainly on diverting some options to the menu and also removing some descriptions that now can only be viewed in the Help menu.

6. CONCLUSION

KarnUMa applications unlike another existing applications emphasizes the Essential Prime Implicants identifying them unequivocally. These Implicants are essential to the resolution of a Karnaugh map, because without them it is not possible to obtain the minimal solution of a Karnaugh map.

Another important feature of this application is that it provides all the steps constituting the Karnaugh Map solving process, adding even the opportunity for the user to solve it the same way, which reinforces the educational component.

The application *KarnUMa* is capable of delivering up to two map's solutions and check for any alternative solution, since it checks if the alternative solution is whether identical to the two solutions found by the application or equivalent.

Considering the other features that were already present in existing applications, the application tries to make these features more complete and intuitive, such as the resolution by the user in which was introduced a feedback bar, which allows the user to become aware of his performance in this process. Still in user's resolution mode, this application is able to identify the mistakes made by the user and provide tips in order to avoid making the same mistakes in the future attempts.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Portuguese Foundation for Science and Technology for their support for this work through project PEst-OE/EEI/LA0009/2011.

REFERENCES

- Bowles, M. (2004). *Relearning to E-Learn - Strategies for electronic learning and knowledge* (Vol. Cap. 1). Merbourne University Publishing Ltd, Polar Design Pty Ltd.
- Enough Software. (2011). *Don't Panic Mobile Developer's Guide to the Galaxy* (9th ed.). Sogestrass, Bremen, Germany: Enough Software.
- Google. (2012). *Android 4.0 Platform Highlights*. Retrieved February 16, 2012, from <http://developer.android.com/sdk/android-4.0-highlights.html>
- LiteratePrograms. (2010). *Quine-McCluskey algorithm (Java)*. Retrieved February 22, 2012, from [http://en.literateprograms.org/Quine-McCluskey_algorithm_\(Java\)](http://en.literateprograms.org/Quine-McCluskey_algorithm_(Java))
- ORACLE. (2012). *Java*. Retrieved February 16, 2012, from http://www.java.com/pt_BR/about/
- ORACLE. (2012). *JAVA FOR MOBILE DEVICES*. Retrieved February 16, 2012, from <http://www.oracle.com/technetwork/java/javame/javamobile/overview/getstarted/index.html>
- Reis, A. I., et Ribas, R. P. (2011). *Karma*, 3.61. (Logic Circuit Synthesis Labs, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Brazil) Retrieved October 2011, from http://www.inf.ufrgs.br/logics/index.php?option=com_docman&task=doc_download&gid=202&Itemid=58
- Sena, A., et Torres, M. (2009). *EasyKarnaugh 3.0 - Uma ferramenta computacional para o auxílio no ensino de Mapas de Karnaugh em Lógica Digital*. Universidade Estadual de Santa Cruz, Departamento de Ciências Exatas e Tecnologias, Brazil.
- ShurikSoft. (2008). *Karnaugh Minimizer*, 2.0. (Shuriksoft) Retrieved November 2010, from <http://karnaugh.shuriksoft.com/>
- Traxler, J. (2005). *Defining Mobile Learning. International Conference Mobile Learning*. Wolverhampton, United Kingdom: IADIS.