

APLICAÇÃO INFORMÁTICA PARA GESTÃO DE MANUTENÇÕES EM EMPRESA DE CABLAGENS

Luís Miguel Felício Baptista Alves

Dissertação apresentada ao Instituto Politécnico de Castelo Branco para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Desenvolvimento de Software e Sistemas Interactivos, realizada sob a orientação científica do Professor José Carlos Meireles Monteiro Metrôlo, Professor Adjunto da Unidade Técnico-Científica de Informática do Instituto Politécnico de Castelo Branco.

Agradecimentos

A presente dissertação é o culminar de um esforço na realização do Mestrado em Desenvolvimento de Software e Sistemas Interactivos, na Escola Superior de Tecnologia de Castelo Branco. Vários foram aqueles que contribuíram, directa ou indirectamente, para a sua concretização e sem o contributo dos quais não seriam possível atingir os resultados propostos.

Neste projecto de desenvolvimento, apesar de muitos terem contribuído, devo agradecer a algumas pessoas em especial, pelo acompanhamento mais directo e constante que prestaram ao longo deste período.

Agradeço particularmente ao Professor Doutor José Carlos Meireles Monteiro Metrôlho, que prontamente aceitou orientar este projecto de mestrado, pela sua disponibilidade até ao último momento, pelo encorajamento e motivação que sempre transcenderam em muito o papel de orientador.

Ao Engenheiro António Barata, orientador externo do projecto de mestrado da Delphi - Automotive Systems. Pela amizade, apoio, disponibilidade e por muitas outras razões, é enorme a gratidão para com ele, pois sem ele e a empresa que representa, este projecto não seria possível.

À Escola Superior de Tecnologia pelas facilidades concedidas.

A todos os familiares e amigos por todo o apoio e interesse manifestado.

APLICAÇÃO INFORMÁTICA PARA GESTÃO DE MANUTENÇÕES EM EMPRESA DE CABLAGENS

Palavras-chave

PGM, Aplicação Web, Dispositivos Móveis, Gestão de Equipamentos

Resumo

Este projecto insere-se no âmbito do mestrado em Desenvolvimento de Software e Sistemas Interactivos que tem por objectivo a criação de uma aplicação para a empresa Delphi - Automotive Systems de Castelo Branco.

A aplicação destina-se à substituição do sistema actualmente em utilização na empresa, sistema esse que segundo os responsáveis, não cumpre a totalidade dos requisitos.

Por forma a satisfazer todas as necessidades da empresa, a aplicação será inteiramente desenvolvida conjuntamente com os intervenientes directos e indirectos de modo a conseguir fazer um levantamento exaustivo dos requisitos pretendidos, tendo por objectivo a completa satisfação do cliente.

O objectivo deste projecto consiste na criação de uma aplicação que permita reunir um conjunto de informação relativa aos vários tipos de equipamentos utilizados na manufactura de cablagens para automóveis (produto desenvolvido pela empresa).

Visto que a maioria destes equipamentos são críticos no processo, e que a sua paragem implica custos, é necessário que estes sejam devidamente controlados a nível de manutenções preventivas para que a sua disponibilidade seja (idealmente) de 100%.

As funções da aplicação passam por agrupar e processar a informação relativa aos números de ciclos efectuados e tempos desde a última manutenção. O software irá gerar avisos relativamente às manutenções a efectuar, considerando a sua prioridade e a carga de trabalho dos operadores.

Para reunir a informação do trabalho efectuado por cada equipamento existirá uma ligação ao software de controlo e gestão de produção existente, que irá disponibilizar a informação relativa à produção, que posteriormente será convertida em ciclos de trabalho de cada equipamento.

Outra das características do software será a de gerar os relatórios que serão preenchidos pelos operadores e que servirão de comprovativo da manutenção, bem como relatórios de apoio à gestão.

Com vista a reduzir as paragens será também realizado um tratamento estatístico dos principais problemas de modo a que se possa ajustar os intervalos entre manutenções dos equipamentos críticos e problemáticos.

A aplicação disponibilizará dois módulos distintos, um para computadores convencionais e outro para dispositivos móveis.

APLICAÇÃO INFORMÁTICA PARA GESTÃO DE MANUTENÇÕES EM EMPRESA DE CABLAGENS

Keywords

PGM, Web application, Mobile Device, Asset Management

Abstract

This project falls within the Masters' Degree in Software Development and Interactive Systems that aims to create an application to Delphi - Automotive Systems of Castelo Branco.

The application is intended to replace the system currently in use in the company, which according to the administrators, do not reach the current requirements.

In order to satisfy all the needs of the company, the application will be fully developed with the direct and indirect stakeholders in order to be able to do an exhaustive survey of the desired requirements, aiming to the complete customer satisfaction.

The aim of this project is to create an application that has information on various types of equipment used in the manufacture of automotive wiring (product developed by the company).

As the majority of these devices are critical in the process and, that its arrest involves costs, it is necessary that these devices should be properly controlled at the level of preventive maintenance to ensure that its availability is 100%.

Its functions are to collate and process the information on the numbers of cycles and times made since the last maintenance. The software will generate warnings for the maintenance, considering its priority and the workload of operators.

To gather the information of the work performed by each equipment, there is a link to the control and manage software, that will be later converted into work cycles.

Another feature of the software is the ability of generating reports that will be filled by operators and saved as proof of maintenance.

In order to reduce downtime it will also be carried out a statistical analysis of the main problems so that it can adjust the intervals between maintenance of critical equipment and problematic ones.

The application will provide two distinct modules, one for conventional computers and other to mobile devices.

Índice geral

| | | |
|--------|--|----|
| 1. | Introdução | 1 |
| 1.1. | Motivação..... | 1 |
| 1.2. | Enquadramento | 1 |
| 1.3. | Solução..... | 3 |
| 1.4. | Estrutura do relatório..... | 4 |
| 2. | Planificação..... | 6 |
| 2.1. | Metodologias do projecto..... | 6 |
| 2.2. | Estado da arte | 6 |
| 2.3. | Escolha da arquitectura | 10 |
| 2.4. | Especificação de Requisitos da Aplicação | 13 |
| 2.4.1. | Requisitos Não Funcionais..... | 13 |
| 2.4.2. | Requisitos Funcionais | 14 |
| 2.5. | Modelação do sistema..... | 15 |
| 2.5.1. | Diagramas de Casos de uso..... | 15 |
| 2.5.2. | Diagramas de <i>Casos de Utilização</i> da aplicação PGM | 17 |
| 2.5.3. | Diagramas de <i>Casos de Uso</i> da aplicação PGMMobile | 18 |
| 2.6. | Cronograma e metas do projecto..... | 19 |
| 3. | Funcionamento..... | 21 |
| 3.1. | Funcionamento geral do sistema | 21 |
| 3.1.1. | Arquitectura da solução..... | 21 |
| 3.1.1. | Ferramentas usadas na construção da solução..... | 22 |
| 4. | Implementação | 29 |
| 4.1. | Estrutura da base de dados | 29 |
| 4.2. | Estrutura geral da aplicação PGM..... | 29 |
| 4.2.1. | Operador..... | 30 |
| 4.2.2. | Gestor | 34 |
| 4.2.3. | Administrador | 40 |
| 4.3. | PGMMobile | 40 |
| 4.4. | Funcionamento de componentes específicas..... | 43 |
| 4.4.1. | Ajax na conclusão automática..... | 43 |
| 4.4.2. | Ajax na apresentação e gravação de formulários | 44 |
| 4.4.3. | Criação de <i>ficheiros PDF</i> | 44 |
| 4.4.1. | Criação de Ficheiros .xls | 45 |
| 4.5. | Instalação e testes finais | 45 |
| 5. | Análise dos resultados..... | 47 |
| 5.1. | O antes e o depois | 47 |
| 6. | Considerações finais | 49 |
| 6.1. | Principais dificuldades encontradas | 49 |
| 6.2. | Desenvolvimentos futuros..... | 49 |
| 6.2.1. | Distribuição automática de tarefas aos operadores | 49 |
| 6.2.2. | Cálculo da carga de trabalho dos operadores | 50 |

| | |
|---|----|
| 6.2.3. Ligação permanente entre dispositivos móveis e servidor..... | 50 |
| 6.2.4. Registo automático dos ciclos de operação de todos os equipamentos | 50 |
| 7. Conclusão | 52 |
| Referências | 53 |
| Anexos..... | 56 |

Índice de figuras

| | |
|---|----|
| Figura 1 - Representação do actor em UML [25] | 15 |
| Figura 2 - Representação de caso de uso em UML [26] | 16 |
| Figura 3 - Representação de uma relação de extend [27]..... | 16 |
| Figura 4 - Representação de uma relação de <i>include</i> [28]..... | 16 |
| Figura 5 – Diagrama <i>Casos de uso</i> da aplicação PGM | 17 |
| Figura 6 – Diagrama “Casos de uso” da aplicação - PGMMobile | 18 |
| Figura 7 – Cronograma do projecto de Mestrado..... | 19 |
| Figura 8 – Exemplo de utilização do Greybox na aplicação | 26 |
| Figura 9 – Exemplo de utilização do TinyMCE na aplicação | 27 |
| Figura 10 -Diagrama Relacional da base de dados | 29 |
| Figura 11 – Autenticação na aplicação | 30 |
| Figura 12 – Actualização de contadores - Operador | 31 |
| Figura 13 – Registo ficha manutenção - Operador..... | 32 |
| Figura 14 – Calendário - Operador | 33 |
| Figura 15 – Alterar dados pessoais - Operador | 33 |
| Figura 16 – Editar dados do programa, Listar – Gestor | 34 |
| Figura 17 – Editar dados do programa, Editar – Gestor..... | 35 |
| Figura 18 – Actualização automática de contadores – Gestor..... | 37 |
| Figura 19 – Visualizar histórico dos contadores – Gestor..... | 37 |
| Figura 20 – Visualizar das ordens de trabalho – Gestor..... | 38 |
| Figura 21 – Visualização do calendário – Gestor..... | 39 |
| Figura 22 – Autenticação - PGMMobile..... | 41 |
| Figura 23 – Actualização dos contadores - PGMMobile | 41 |
| Figura 24 – Visualização do calendário - PGMMobile..... | 42 |
| Figura 25 – Emissão ficha de reparação - PGMMobile | 42 |
| Figura 26 – Conclusão automática com Ajax | 44 |

Lista de abreviaturas

WAMP *Windows Apache MySql and PHP*

EAM *Enterprise Asset Management*

CMMS *Computerized Maintenance Management System*

IIS *Internet Information Services*

PHP *Hypertext Preprocessor*

AJAX *Asynchronous Javascript And XML*

PGM *Plano Geral de Manutenção*

PGMMobile *Plano Geral de Manutenção para Dispositivos Moveis*

GUI *Graphical User Interface*

SGBD *Sistemas de Gestão de Bases de Dados*

HTML *HyperText Markup Language*

PDA *Personal Digital Assistant*

DOM *Document Object Model*

CSS *Cascading Style Sheets*

GNU *General Public License*

MIT Licence *Massachusetts Institute of Technology Licence*

BD *Base de Dados*

WYSIWYG *What You See Is What You Get*

HPC *Hourly Production Control*

PN *Part Number*

PDF *Portable Document Format*

1. Introdução

1.1. Motivação

Este trabalho realizou-se no âmbito do projecto de mestrado em Desenvolvimento de Software e Sistemas Interactivos na Escola Superior de Tecnologia de Castelo Branco (ESTCB) e da necessidade da implementação de uma solução de software para um problema específico da empresa Delphi - Automotive Systems, SA de Castelo Branco.

Para satisfazer as necessidades de ambas as entidades, foi mais uma vez criada uma “ponte” entre o sistema de ensino superior e o mundo empresarial da região, neste caso através de uma das maiores empregadoras do distrito de Castelo Branco.

Tanto a Escola como a empresa disponibilizam todo o apoio necessário para tornar este projecto aplicado numa realidade.

A nível pessoal este trabalho ajudará a incrementar a experiência já de alguns anos na realização de soluções de informática industrial, área na qual desenvolvi um gosto especial.

Outra fonte de motivação pessoal para este trabalho é o facto de a aplicação ser destinada a uso concreto numa empresa de prestígio e contribuir para a melhoria dos processos.

1.2. Enquadramento

A empresa em questão dedica-se à produção de cablagens para automóveis, e apesar de ser uma indústria que recorre maioritariamente a trabalho manual, existem inúmeros processos que são automatizados e dependem de determinados equipamentos de produção. A avaria de um destes equipamentos pode implicar a paragem da produção e conseqüentemente dos diversos operadores, causando prejuízos elevados.

A aplicação em uso actualmente não preenche todos os requisitos da empresa e apresenta alguns problemas que podem directa ou indirectamente afectar o bom funcionamento do processo. Os principais problemas são os seguintes:

1º Problema

Para minimizar os tempos de paragens dos equipamentos é necessário que sejam respeitados os intervalos entre as manutenções. Estes intervalos podem ser temporais ou depender do número de utilizações.

O problema surge devido à dificuldade de registar eficazmente o número de utilizações (ciclos) dos equipamentos, pois actualmente requerem leitura manual.

Actualmente, um operador percorre todos os equipamentos registando o número de ciclos em papel para posteriormente introduzir no sistema informático.

Este processo, além de demorado, está sujeito a atrasos e a erros de registo, o que poderá implicar retardamento nas manutenções dos equipamentos e consequentes avarias.

2º Problema

Mesmo que todos os intervalos entre manutenções sejam respeitados, existem sempre alguns equipamentos mais problemáticos que, apesar de ter a manutenção em dia, por vezes falham. Quando se opera diariamente com mais de um milhar destes equipamentos, é difícil identificar quais os mais problemáticos e quais necessitam de ajustes nos intervalos entre manutenções.

O sistema actual não possui qualquer ferramenta de tratamento estatístico que permita identificar quais os equipamentos mais críticos, quais implicam maiores custos, etc.

3º Problema

O departamento da manutenção da empresa conta com várias dezenas de colaboradores em que a maior parte realiza as manutenções preventivas e curativas (caso aconteça uma paragem imprevista). Cada uma dessas manutenções implica o preenchimento de um relatório de intervenção. Após o seu preenchimento, esses relatórios são introduzidos manualmente no sistema informático por um outro colaborador.

Este processo dá muitas vezes origem a perdas de informação e erros de interpretação ou introdução, causando o surgimento de dados inválidos na aplicação e consequentemente erros no sistema.

4º Problema

Apesar de todos os equipamentos serem importantes para o processo produtivo, nem todos possuem a mesma relevância. Alguns equipamentos podem ser facilmente substituídos durante a produção, outros podem obrigar a longas paragens.

A aplicação actual não possui qualquer sistema de identificação e priorização de manutenções baseada na importância dos equipamentos no processo produtivo.

Outros problemas

Segundo relatos dos intervenientes, a aplicação actual depende de uma base de dados remota, e por causas aparentemente não identificadas, fica indisponível com alguma frequência.

Segundo os responsáveis da empresa, o suporte à aplicação é bastante lento e implica custos elevados.

Na empresa foram realizadas várias reuniões e entrevistas para que fosse possível identificar inequivocamente os requisitos da nova aplicação, de modo a torná-la a mais adequada possível ao problema e uma resposta eficaz aos problemas reportados anteriormente.

1.3. Solução

Após as primeiras avaliações da situação foi proposto aos responsáveis da empresa uma solução que respondia aos principais problemas. Essa passa pelo desenvolvimento de uma aplicação informática com as seguintes características:

Registar os diversos equipamentos:

A aplicação permitirá registar as diversas características dos equipamentos, com funcionalidades para inserir, editar, eliminar e pesquisar equipamentos, fornecendo adicionalmente outras classificações como: equipamento crítico, equipamento problemático, etc. Este módulo permitirá também exibir e imprimir listagens de equipamentos, tratamento estatístico das avarias, localização dos equipamentos, bem como ferramentas de exportação e importação de informação de e para outros formatos.

Registo automático e semi-automático do número de ciclos por equipamento:

Sendo o registo do número de ciclos dos equipamentos um factor crítico no processo, a solução encontrada passa pela importação automática desta informação. Revelou-se possível importar grande parte desta informação através do software de controlo de produção existente na empresa. Este processo automático alivia em muito o trabalho de recolha de dados e minimiza os erros de introdução. Apesar de este processo permitir recolher a grande maioria da informação, existem alguns equipamentos em que não é possível importar, então, será também criado um módulo para dispositivos móveis para esse efeito. Usando o dispositivo móvel é possível recolher a informação do número de ciclos junto do equipamento, para posterior sincronização automática com a base de dados principal. A utilização deste módulo elimina o passo intermédio de registo em papel, eliminando também um tempo considerável no processo, bem como a possibilidade de erros.

A importação automática consiste na ligação a uma aplicação externa que regista o número de peças produzidas, usando um factor de cálculo para cada peça, e associando também a sua localização. Assim, é possível identificar com precisão o número de ciclos de cada equipamento num dado período de tempo.

Este processo só se tornou possível devido à possibilidade de ligação à aplicação de gestão e controlo de produção existente, em que o administrador da aplicação permitiu acesso à informação.

Registo das fichas de manutenção

Este é outro dos pontos do processo que implicava o registo de informação em papel para posterior introdução no sistema. Com a implementação da nova aplicação, o processo passa a usar o módulo para dispositivos móveis. Além de ser apenas necessário registar uma única vez no sistema, a introdução de dados será mais fiável e rápida devido aos campos pré-preenchidos. A validação dos dados permitirá também reduzir o número de erros nos registos.

Outros

Além destes factores, pretende-se que o sistema seja baseado numa arquitectura estável que permita apresentar uma disponibilidade muito superior à actual aplicação.

Uma das arquitecturas possíveis, *Windows Apache MySql e PHP* (WAMP [1]) possui bastantes provas dadas nomeadamente nos servidores da própria empresa, onde está instalada outra aplicação sob o mesmo suporte com uma disponibilidade muito próxima dos 100% nos últimos 5 anos.

Em suma, pretende-se criar uma aplicação informática que responda aos critérios supra mencionados, que será intitulada de PGM (Plano Geral de Manutenção), adicionalmente será também disponibilizado um módulo para dispositivos móveis de nome PGMMobile (Plano Geral de Manutenção para dispositivos móveis).

1.4. Estrutura do relatório

Esta secção apresenta sucintamente a organização e o conteúdo do presente trabalho. Os capítulos estão organizados de modo a permitir uma compreensão progressiva, seguindo uma linha orientadora desde um conjunto de reflexões iniciais até uma série de expectativas futuras e considerações finais. De notar que, não obstante esta exposição progressiva, os capítulos são relativamente independentes entre si, podendo ser consultados sem afectar significativamente os seus objectivos de apresentação.

O primeiro capítulo, introdutório, começa por procurar descrever a motivação, enquadramento e a solução diagnosticada para o problema apresentado. Ainda sem entrar em pormenores técnicos, pretende descrever inequivocamente o “porquê” da aplicação e o “como” chegar à solução.

O segundo capítulo, de nome Planificação, tal como o nome indica, pretende descrever exhaustivamente os passos iniciais necessários para uma boa análise e planeamento do projecto. Esta fase é de primordial importância para o sucesso do projecto, pois serve como alicerce para todo o trabalho seguinte. Se nesta fase ocorrerem erros, estes podem causar sérios problemas no futuro.

Pretende-se analisar o estado da arte, analisar as soluções tecnológicas disponíveis actualmente e escolher a solução mais indicada para o problema. Adicionalmente são definidos os vários requisitos do projecto bem como a sua calendarização.

O terceiro capítulo poderia ter sido evitado caso se tratasse de uma aplicação *desktop* convencional. No entanto, devido à variedade de ferramentas utilizadas no desenvolvimento e implementação do sistema, este capítulo existe para melhor abordar e descrever as ferramentas e qual o seu papel no funcionamento geral do sistema.

Apesar de serem aqui mencionadas várias ferramentas, na maioria dos casos é realizada uma descrição muito sucinta, estando no entanto disponível a sua referência para uma análise mais aprofundada. Esta característica está presente não só neste capítulo, mas em todo o relatório, pois não seria viável, nem é esse o objectivo, de descrever exaustivamente todas as ferramentas utilizadas e analisadas, mas sim o porquê da sua utilização e conseqüente resultado final.

No quarto capítulo, a Implementação, pretende-se descrever eficientemente as características e funcionalidades da aplicação. Este capítulo é, sem dúvida, o mais técnico de todos.

Na análise de resultados e considerações finais, são evidenciados os resultados deste projecto. Esta fase acontece depois de concluído o desenvolvimento e a aplicação validada na fase de testes. São também evidenciados os principais obstáculos e desenvolvimentos futuros.

A informação contida neste capítulo foi, em grande parte, fornecida pelos intervenientes directos da empresa, pois são eles os maiores conhecedores da lógica do negócio e os mais indicados para avaliar o correcto funcionamento da aplicação.

Como é comum neste tipo de obra, finaliza-se com a conclusão, onde se avalia de uma forma geral o decorrer de todo este processo.

Adicionalmente podemos encontrar as referências necessárias para uma análise cuidada do conteúdo do relatório, devidamente indicadas no corpo deste documento.

2. Planificação

2.1. Metodologias do projecto

Durante o desenvolvimento deste trabalho foi seguida uma metodologia Waterfall [2] segundo a qual o desenvolvimento da aplicação envolve as seguintes etapas:

Descrição do Problema - esta é a etapa inicial, a qual visa a identificação do espaço do problema, e dos factores que possam influenciar os dados a analisar e a tratar. É nesta fase que é delineado o contexto da aplicação;

Análise de Requisitos - nesta fase é feita toda a recolha de informação necessária à definição, clarificação e classificação dos requisitos funcionais e não funcionais do problema;

Arquitectura da Solução - é caracterizado o espaço da solução do problema, sendo identificados e definidos os blocos funcionais e de interacção entre eles;

Implementação - nesta fase é implementada a arquitectura da solução numa linguagem de programação. É elaborado um protótipo de interface onde é avaliada a sua usabilidade;

Plano de Testes - esta etapa encontra-se dividida em duas fases: testes de execução e testes de integração dos vários componentes;

Considerações/Alterações Finais - fase que contempla modificações que, eventualmente, se considerem necessárias após a fase de testes.

Durante o período de realização do trabalho foi esta a metodologia seguida.

2.2. Estado da arte

Os softwares de gestão de manutenções e activos são direccionados para a indústria e serviços. Apesar de não serem tão divulgados como outros tipos de software de uso doméstico, estes representam uma tarefa importante em determinadas áreas de negócio.

Na área da saúde, por exemplo, existem vários softwares de gestão de manutenções e activos que desempenham eficientemente a tarefa, pois foram desenvolvidos especialmente para esse fim; na área da indústria, nomeadamente em fábricas, apesar de existirem várias soluções, é mais difícil encontrar uma que se adapte na perfeição à lógica do negócio em causa. Isto deve-se principalmente à diversidade e especificidade de cada negócio.

Em muitas áreas de negócio um software de uso genérico poderá adaptar-se bem à realidade em que está inserido. No entanto, haverá sempre situações específicas para as quais o software não estará preparado.

Um software que seja desenvolvido especificamente para uma dada empresa e adaptado à sua lógica do negócio, apresenta claras vantagens quando comparado com um de uso genérico.

Seguidamente, são apresentados alguns softwares de gestão de manutenções e activos e suas principais características:

Coswin do grupo Siveco

Actualmente na versão 7i, o software Coswin [3] é uma das referências no sector que conta com cerca de 25 anos de existência.

Este é o software utilizado actualmente na empresa numa versão mais antiga, mas como já foi referenciado, apresenta alguns problemas, maioritariamente devido ao facto de possuir uma estrutura fixa e de não se adaptar às características específicas do negócio.

As características chave anunciadas deste software são:

- Sistema de gestão de base de dados rápido e seguro;
- Utilização “*out of the box*”, permitindo uma rápida e fácil instalação;
- Ligação a sistemas de gestão de compras para uma rápida aquisição de peças de substituição;
- Agendamento de manutenções através da criação de ordens de trabalho.

Em suma, este software contém inúmeras funcionalidades destinadas a facilitar o processo de gestão de manutenções e activos actualmente em uso em área de negócio como aeronáutica, construção, electrónica, saúde, transportes, etc. A sua utilização na empresa conta já com vários anos de existência e, apesar de cumprir parte dos requisitos, apresenta alguns problemas. Assim, revelou-se o interesse de a substituir por um sistema mais adaptado à realidade da empresa.

ARCHIBUS Preventive Maintenance da empresa ARCHIBUS

A ARCHIBUS [4], empresa especializada no desenvolvimento de software para a indústria, apresenta algumas soluções informáticas para a gestão de manutenção de equipamentos de produção. O módulo aqui apresentado (*Preventive Maintenance*) destina-se a realizar a gestão dos equipamentos críticos com o objectivo de reduzir os tempos de paragem e custos de reparação, através de uma eficiente previsão da necessidade de manutenção.

As principais funcionalidades apresentadas são:

- Minimizar custos de reparação e paragens dos equipamentos;
- Alargar o tempo de vida dos equipamentos através de uma boa definição das necessidades de manutenção;
- Criação de históricos de manutenções para análise de performance;

- Detecção de padrões de avarias;
- Classificação de equipamentos segundo a sua prioridade no processo produtivo;
- Funcionamento em ambiente Web.

Este produto apresenta algumas vantagens quando comparado com o actual sistema, nomeadamente a classificação dos equipamentos, criação de relatórios com o histórico de avarias, etc. Ainda assim, o módulo aqui apresentado faz parte de um pacote de software mais complexo, que só será totalmente aproveitado quando em funcionamento no seu todo, e não apenas o módulo individualmente.

Infor EAM (Enterprise Asset Management) da empresa Infor

Aplicação Infor EAM [5] (Enterprise Asset Management) à semelhança das restantes aqui apresentadas, possui vantagens e desvantagens.

Segundo os fabricantes [6] as características-chave da aplicação são:

- Manter todos os equipamentos e infra-estruturas fiáveis e seguras;
- Manter os equipamentos aptos e funcionais;
- Reduzir custos de inventário;
- Aumentar a produtividade dos colaboradores;
- Auxiliar a tomada de decisões, no que respeita a reparações e aquisições de equipamentos.

À semelhança de outras aplicações aqui apresentadas, esta possui funcionalidades que vão além da simples gestão e manutenção de activos. Este factor não é propriamente uma mais-valia para a empresa, pois todo o restante processo (aquisição, controlo de inventário, gestão de equipas, etc.) já se encontra totalmente implementado e sem necessidade de mudança.

No que respeita à gestão e manutenção de activos, mantêm-se o mesmo problema de muitas outras: não se adaptar integralmente às necessidades da empresa.

CMMS da empresa NetFacilities

Este *Computerized Maintenance Management System*[7] (CMMS) tem como principal vantagem apresentada, o facto de se basear integralmente numa arquitectura Web. Segundo os fabricantes, este software abrange todas as necessidades do dia-a-dia laboral, por exemplo:

- Ordens de trabalho: Criar, aprovar, receber, acompanhar ordens de trabalho;
- Recursos: Acompanhar e gerir a utilização de recursos em tempo-real;
- Fornecedores: Gerir a interacção entre fornecedores e a empresa com vista à optimização do fluxo da distribuição;

- Colaboração em tempo-real: Devido a funcionar num ambiente Web é fácil a sua utilização em diversos locais simultaneamente, disponibilizando assim a informação de forma centralizada.

Outra grande vantagem anunciada pelo fabricante é a disponibilização de um módulo para dispositivos móveis, melhorando assim a consulta e actualização da informação.

É importante salientar que este tipo de aplicações não estão disponíveis em versões de demonstração, logo não foi possível testar o seu funcionamento para avaliação. A análise aqui apresentada foi totalmente baseada em informações dos fabricantes, com excepção da aplicação *Coswin* que, por estar em funcionamento na empresa, a apreciação foi baseada maioritariamente na opinião e vasta experiência dos utilizadores.

Resumindo, apesar de existirem no mercado soluções destinadas a este fim, nenhuma delas satisfaz plenamente as necessidades da empresa. Muitas destas aplicações contêm funcionalidades que, apesar de enriquecerem a aplicação, não terão utilidade prática, pois a empresa já possui outras ferramentas dedicadas a esses fins.

O custo elevado de aquisição, instalação e manutenção são também factores que não favorecem as soluções aqui apresentadas.

Para melhor demonstrar as diferentes características das diversas aplicações será apresentada a seguinte tabela:

| | Coswin | ARCHIBUS | Infor EAM | NetFacilities |
|--|---------------|-----------------|------------------|----------------------|
| Gerir Ordens de trabalho | Sim | Sim | Sim | Sim |
| Histórico de Manutenções | Sim | Sim | Sim | Sim |
| Detecção de padrões de avaria | | Sim | | |
| Classificação da prioridade dos equipamentos | | Sim | | |
| Funcionamento em ambiente Web | | Sim | | Sim |
| Módulo para dispositivos móveis | Sim | | | Sim |

Tabela 1 - Comparação das diferentes soluções

2.3. Escolha da arquitectura

Existem actualmente diversas formas de implementação de aplicações e todas elas apresentam prós e contras. Não existindo qualquer tipo de requisito por parte do cliente, a escolha poderia abranger todas as opções existentes no mercado, no entanto, estes requisitos existiram e obrigaram a uma filtragem inicial das opções. Considerando também as infra-estruturas informáticas do local, as opções de escolha foram mais uma vez reduzidas.

É importante salientar mais uma vez que um dos critérios de sucesso da aplicação é o custo inerente ao seu desenvolvimento e utilização, usando unicamente aplicações gratuitas e não requerendo muito investimento na aquisição de novos recursos no sistema informático, torna-se uma escolha atractiva. Desta forma, os custos podem ser consideravelmente reduzidos.

Escolha da arquitectura

O local de implementação conta com uma infra-estrutura informática baseada em máquinas Windows [8] ligadas através da rede interna. Existem também diversos servidores de rede para o desempenho de diversas tarefas inerentes ao processo produtivo.

Apesar da rigidez das normas a que o sistema informático está sujeito, foi disponibilizado um servidor para apoio à aplicação, que apesar de não ser totalmente dedicado a esta função, dispõe de bastantes recursos disponíveis.

Um dos principais requisitos do cliente é a facilidade de instalação e utilização, bem como a disponibilidade e facilidade de acesso aos dados para consulta e actualização, criação de relatórios, etc.

Com base nestes factores, as escolhas possíveis foram divididas em dois grandes grupos, aplicação *Web* e aplicação *Desktop*. No caso da aplicação *Web*, apesar de se destinar apenas ao uso na rede da empresa, esta assume o âmbito mundial, estando presente em dezenas de países e centenas de localizações.

Com a crescente importância da *Web* como plataforma, massificou-se a implementação de aplicações *Web*, e muitas vezes deu-se a migração de aplicações *Desktop*. Estas foram desenvolvidas inicialmente para serem inteiramente executadas localmente, e passaram para um ambiente partilhado. Esta implementação trouxe novos desafios mas também inúmeras vantagens, muito pelo facto de poderem ser partilhadas por inúmeros utilizadores. Estes desafios traduzem-se não só numa alteração da arquitectura da aplicação, pois novas questões têm de ser ponderadas, como por exemplo a nível de armazenamento de dados, comunicação cliente/servidor, segurança da aplicação, como também a nível de interacção com o utilizador, utilizando novas técnicas de interacção que foram desenvolvidas na procura de melhorar a experiência de utilização.

Vantagens e desvantagens das aplicações Web

Embora as aplicações *Web* possuam muitas características semelhantes às aplicações *Desktop*, devido ao diferente ambiente de execução, existem também características muito distintas. As aplicações *Web* possuem algumas desvantagens, sendo as mais importantes:

Segurança: Uma vez que as aplicações Web são executadas em um ambiente partilhado, os dados do utilizador estão mais vulneráveis a ataques, pelo que surge a necessidade de criar mecanismos de segurança mais fiáveis. Nas aplicações *Desktop*, uma vez que os dados são mantidos localmente, são menos susceptíveis de ataque.

O ambiente em questão apesar de não estar aberto para a Web é utilizado por algumas centenas de utilizadores.

Performance: Como na base das aplicações Web encontra-se uma arquitectura cliente/servidor em que a comunicação é feita através da rede, a performance da aplicação irá depender da largura de banda disponível ao utilizador. No caso das aplicações *Desktop*, uma vez que o *front-end* e o *back-end* da aplicação se encontram no mesmo local físico, a performance será mais eficiente.

Em relação às vantagens em comparação com as aplicações desktop, temos:

Acessibilidade: As aplicações *Desktop* necessitam de ser instaladas individualmente em cada computador, sendo esta tarefa da responsabilidade do administrador, enquanto uma aplicação Web ao ser instalada num servidor, fica automaticamente disponível para toda a rede a que está ligado, podendo ser acedida através de um *browser*.

Multi-Plataforma: Uma vez que as aplicações Web são executadas no browser, e atendendo a que este já é software integrante de qualquer sistema operativo, significa que as aplicações Web podem ser facilmente acedidas independentemente do sistema operativo em uso. No caso das aplicações *Desktop*, apesar de existirem também soluções multi-plataforma, estão condicionadas pela adição de *software* e pela arquitectura em questão, não podendo ser, por exemplo, facilmente acedida através de um dispositivo móvel.

Fácil manutenção: Como as aplicações Web estão alojadas num servidor, todas as actualizações/modificações são feitas apenas neste servidor, pelo que os utilizadores irão ter sempre acesso à versão mais actualizada da aplicação. No caso das aplicações *Desktop*, é da responsabilidade do administrador realizar a manutenção e actualização do mesmo. Mais uma vez, apesar de existirem métodos de actualização automática, nunca são de tão fácil utilização como a arquitectura Web.

Recursos utilizados: estando as aplicações Web instaladas num servidor remoto, estas não consomem recursos significativos no computador do utilizador, enquanto as aplicações *Desktop*

exigem espaço e processamento disponível localmente. No entanto, o desempenho desta arquitectura depende da capacidade de processamento do servidor. No cenário actual, o servidor disponibilizado possui recursos mais que suficientes para a tarefa em questão, considerando o baixo número de utilizadores (aproximadamente trinta).

Deste modo, tendo em conta a perspectiva do administrador do sistema informático, este não necessita de:

- Instalar a aplicação;
- Realizar as actualizações quando necessárias;
- Ou ainda realizar qualquer tipo de configurações.

Necessita apenas de possuir um browser e acesso à rede.

Com base na informação analisada, a aplicação *Web* configura-se como a melhor opção. Dentro desta, existe ainda um grande número de possibilidades de escolha que serão analisadas individualmente dependendo da sua função.

Servidor Web

Conforme foi referido anteriormente neste capítulo, um dos principais factores de sucesso é o baixo custo de implementação. Nesse sentido, foi dada grande importância a este factor.

Foram analisados os principais servidores Web disponíveis no mercado e, tendo em conta como principais factores o custo, performance e a estabilidade do sistema, a escolha resumiu-se ao servidor *Apache* [9] e *Internet Information Service* da *Microsoft* [10], principalmente devido ao facto de ambos terem provas dadas no mercado, e por possuírem as maiores cotas de mercado actualmente.

Após a análise de muita documentação, testes comparativos e também a experiência própria, foi decidida a utilização do Servidor Web *Apache*.

É importante salientar também que o facto deste Servidor Web já se encontrar em funcionamento no servidor destinado a esta aplicação, favoreceu também a sua escolha, pois se no mesmo equipamento físico fossem usados mais que um serviço para o mesmo fim, seriam necessárias configurações adicionais e conseqüentemente maiores dificuldades e constrangimentos.

Sistema de gestão de bases de dados

Se até a este ponto a decisão de escolha das tecnologias a utilizar não foi propriamente fácil, devido ao seu elevado número, no que respeita à escolha do Sistema de Gestão de Base de Dados (SGBD), a escolha também se revelou difícil.

Na decisão da escolha desta tecnologia foram considerados como principais factores a robustez, segurança, velocidade e custo.

Para este serviço foram analisadas várias possibilidades, sendo as principais: Oracle [11], MSSQL[12], MySQL[13], FireBird[14], Postgress[15] e Interbase[16].

Mais uma vez considerando os factores referidos anteriormente, a escolha recaiu no MySQL devido a aspectos como o custo, ser distribuído sob licença GPL (GNU General Public License) [17], ser muito rápido, multi-processos, multi-utilizador e robusto.

Linguagem de programação

Para o desenvolvimento do projecto, a linguagem *Hypertext Preprocessor (PHP)* [18] foi seleccionada devido às inúmeras vantagens apresentadas pela mesma, tais como: a sua vasta documentação, facilidade de uso e utilização, bem como a sua imensa biblioteca de funções.

Outras ferramentas

Com vista ao melhoramento da interface com o utilizador em factores como a usabilidade, precisão, desempenho, segurança, fiabilidade, robustez, restrições, entre outros; foram também analisadas outras ferramentas com o objectivo de melhorar a experiência do utilizador. Ferramentas como *Asynchronous Javascript and XML (Ajax)*[19], JQuery[20], GreyBox[21], etc permitem melhorar esta interacção com o utilizador.

2.4. Especificação de Requisitos da Aplicação

O objectivo deste projecto é a implementação de uma aplicação com as seguintes características:

- Facilidade de acesso e distribuição por vários pontos da fábrica;
- Minimização de introdução de dados manuais pelos operadores;
- Infra-estrutura estável idealmente com uma disponibilidade de 100%;
- Permitir analisar grandes quantidades de informação de forma fácil e produzir relatórios /resumos;
- Facilidade de utilização.

2.4.1. Requisitos Não Funcionais

Estes são os requisitos globais dos custos de desenvolvimento e dos custos operacionais. Abrangem as características de qualidade (requisitos não funcionais) "do que é" e "do que faz", em relação a: usabilidade, precisão, desempenho, segurança, fiabilidade, portabilidade, robustez, restrições, entre outros.

Usabilidade

Conforme já foi referido, a aplicação destina-se à empresa *Delphi* - Automotive Systems, e pretende ser uma aplicação para auxiliar os operadores do departamento de manutenção, entre outros, que actualmente estão familiarizados com um processo de inserção de dados manual. Assim, a interface da aplicação deve ser simples, intuitiva e flexível, de modo a fomentar um maior desempenho por parte dos utilizadores. É uma aplicação que vai trabalhar sobre ambiente Web, devendo por isso, ser compatível com a maioria dos *browsers* disponíveis no mercado.

Fiabilidade

No que diz respeito à fiabilidade, a aplicação deve estar preparada para ocorrências não documentadas, bem como apresentar o mínimo possível de falhas no sistema. As eventuais lacunas devem ser previsíveis e de fácil recuperação, de modo a não comprometer o bom funcionamento da aplicação.

Desempenho

Em termos de desempenho, a aplicação está dependente, principalmente, da taxa de ocupação de rede *Ethernet* existente na fábrica. Sendo esta uma aplicação cliente-servidor, as transferências de informação estão sempre dependentes de vários troços de rede. Deste modo, não é possível prever com exactidão qual a carga a que o sistema estará sujeito.

Os acessos concorrentes podem ser outro aspecto a ter em conta, uma vez que se um grande número de utilizadores tentarem aceder ao mesmo tipo de informação, poderá originar alguma perda de desempenho da aplicação.

Open-Source

A utilização de uma tecnologia deste tipo foi um dos requisitos inicialmente considerados, pelo que toda a aplicação irá ser implementada recorrendo a tecnologia *open-source*, nomeadamente: servidor HTTP Apache, linguagem programação PHP, e sistema de base de dados MySQL.

2.4.2. Requisitos Funcionais

A complexidade de um projecto de software é determinada, parcialmente, pelas suas funcionalidades (requisitos funcionais), ou seja, "o que é que" o sistema faz. Assim, a aplicação deve responder aos seguintes requisitos funcionais:

- Inserir, Alterar, Remover, Consultar Equipamentos;
- Inserir, Alterar, Remover, Consultar Centros de Custo;
- Inserir, Alterar, Remover, Consultar Tipos de Equipamentos;
- Inserir, Alterar, Remover, Consultar Intervenções;
- Inserir, Alterar, Remover, Consultar Operadores;
- Inserir, Alterar, Remover, Consultar Tarifas;
- Inserir, Alterar, Remover, Consultar Localizações;
- Actualizar e consultar histórico dos Contadores;
- Emitir, Alterar e Fechar Ordens de Trabalho;
- Exportar relatórios e análise estatística da informação.

2.5. Modelação do sistema

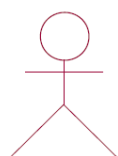
Seguidamente será apresentada a modelação do sistema que teve por base a análise de requisitos. Neste caso a modelação será realizada recorrendo ao UML2.0 [22] pois trata-se de uma linguagem universal que permite especificar, construir e visualizar um sistema informático.

Actualmente na versão 2, esta corresponde a uma evolução significativa relativamente à versão anterior (UML1.5), em que apesar das alterações mais significativas se reflectirem em aspectos arquitecturais internos, existem também aspectos visíveis ao utilizador do UML [23].

2.5.1. Diagramas de Casos de uso

Os diagramas de *casos de uso* [24] têm como objectivo mostrar para que serve o sistema, isto sem mostrar a sua organização interna, com base na identificação e descrição de *actores* e *casos de uso*, ou seja, mostrar o que o sistema deve efectuar e não como o vai fazer.

Estes diagramas são concebidos na fase de estudo e especificação de requisitos. O requisito é uma funcionalidade, ou característica, considerada de extrema relevância na óptica do utilizador, pois normalmente representa o comportamento esperado do sistema. Os diagramas de *casos de uso* utilizam os seguintes elementos de modelação:



<Actor>

Figura 1 - Representação do actor em UML [25]

Actor (Figura 1) - é o primeiro a ser identificado na construção do diagrama. Representa uma entidade externa que interage com o sistema. Deve ser caracterizado através de uma

pequena descrição de forma a assegurar a correcta compreensão do significado do *actor* por todos os elementos da equipa envolvida na análise.



Figura 2 - Representação de caso de uso em UML [26]

Caso de uso (Figura 2) - é uma descrição de um conjunto de sequências de acções, incluindo variantes, que um sistema realiza para produzir um resultado observável com valor para um *actor*. Um *caso de uso* é uma funcionalidade, é ainda uma situação de utilização do sistema com um determinado objectivo. Por outro lado, um tipo de interacção entre um ou mais *actores* e o sistema com um determinado objectivo.

Os casos podem estar relacionados entre si, para isso, utiliza-se frequentemente as relações «*include*», «*extend*» e «*generalização*».

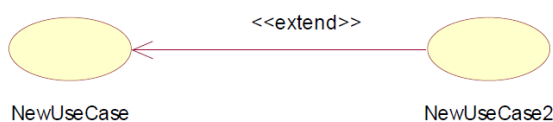


Figura 3 - Representação de uma relação de extend [27]

Relações de *Extend* (Figura 3) - a relação *extend* ocorre quando existe um comportamento opcional, que deve ser incluído num *caso de uso*. Este comportamento é definido num segmento *caso de uso* e invocado pelo *caso de uso* base.

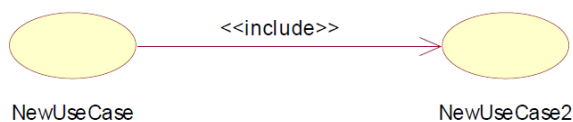


Figura 4 - Representação de uma relação de include [28]

- Relação de *Include* (Figura 4) - quando vários *casos de uso* têm um fluxo de eventos comum, é conveniente separar essa parte comum para um novo *caso de uso* que é incluído pelos primeiros.

Visto que este trabalho incide sobre duas aplicações, Plano Geral de Manutenção (PGM) e Plano Geral de Manutenção para Dispositivos Móveis (PGMMobile), serão apresentados dois diagramas, um para cada aplicação.

2.5.2. Diagramas de *Casos de Utilização* da aplicação PGM

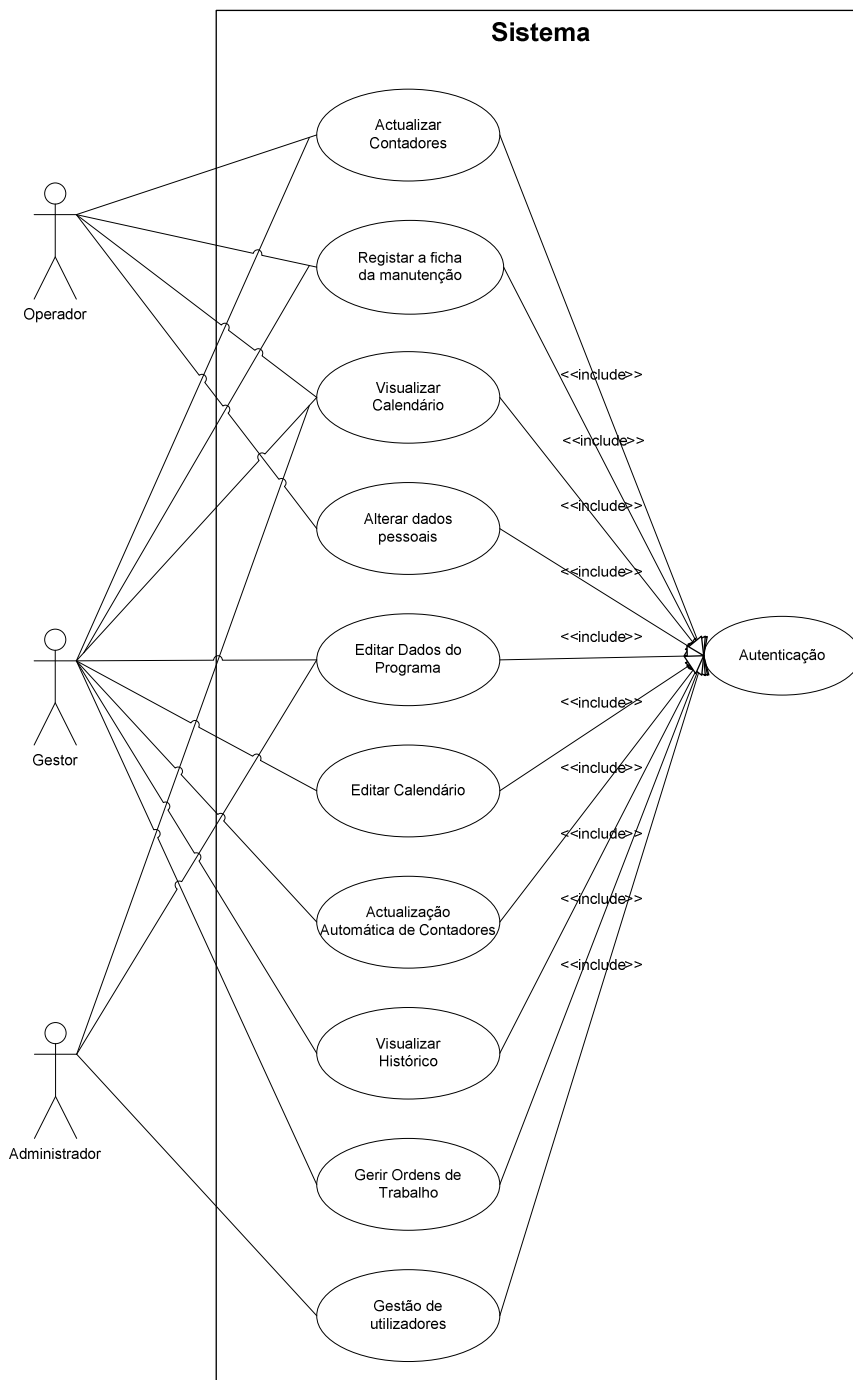


Figura 5 - Diagrama *Casos de uso* da aplicação PGM

Como se pode verificar na Figura 5, estão identificados três *actores* do sistema; o operador, o gestor e o administrador em que cada um deles possui *casos de uso* distintos. Como pode ser facilmente verificado, existem alguns *casos de uso* que são partilhados pelos três *actores*, como é o caso da visualização do calendário do sistema, outros *casos de uso* como a

gestão de utilizadores ou a gestão de ordens de trabalho são disponibilizados a apenas um *actor*, neste caso ao administrador e ao gestor respectivamente.

Outra característica importante do sistema é que todos os *casos de uso* dependem da autenticação do sistema.

2.5.3. Diagramas de *Casos de Uso* da aplicação PGMMobile

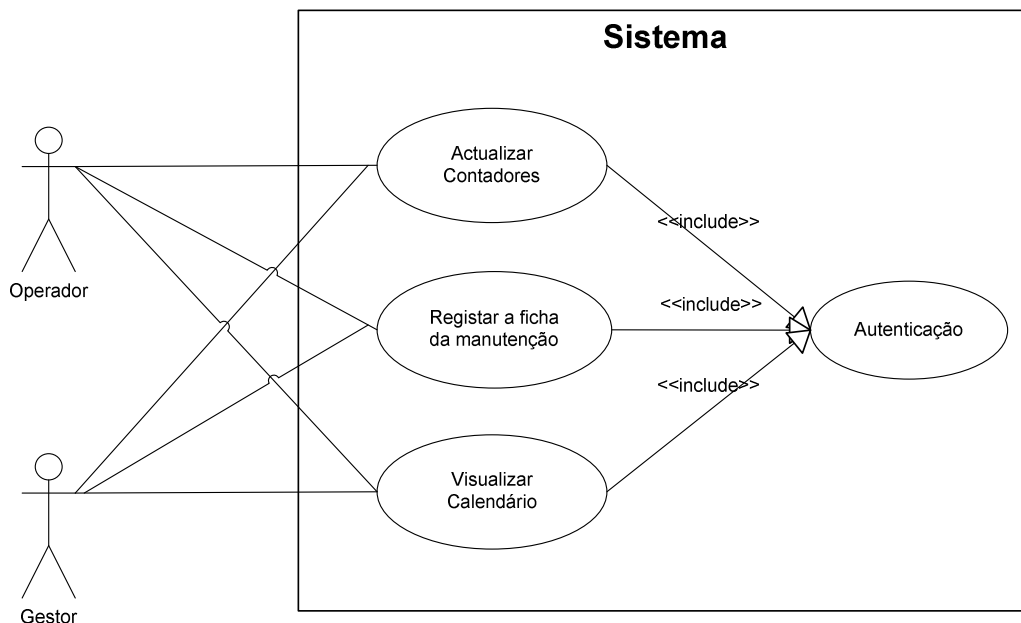


Figura 6 - Diagrama "Casos de uso" da aplicação - PGMMobile

Como pode ser verificado no diagrama anterior, a aplicação PGMMobile possui uma complexidade muito inferior à aplicação principal (PGM). Neste caso, existem apenas dois *actores*: o operador e o gestor.

Devido às reduzidas características desta aplicação, os três *casos de uso* disponíveis são partilhados pelos dois *actores* do sistema, possuindo, no entanto, algumas características específicas dependendo do tipo de utilizador. À semelhança da aplicação principal, todos os *casos de uso* exigem a autenticação no sistema.

2.6. Cronograma e metas do projecto

| ID | Task Name | Start | Finish | Duration | Q1 10 | Q2 10 | | | | Q3 10 | | | | Q4 10 | | | Q1 11 | |
|----|---------------------------------|------------|------------|----------|-------------------------------------|-------|-----|-----|-----|-------|-----|-----|-----|-------|-----|-----|-------|--|
| | | | | | Mar | Abr | Mai | Jun | Jul | Ago | Set | Out | Nov | Dez | Jan | Fev | | |
| 1 | Descrição do Problema | 04-03-2010 | 18-03-2010 | 10d | [Gantt bar from Mar to mid-Mar] | | | | | | | | | | | | | |
| 2 | Análise de Requisitos | 18-03-2010 | 29-04-2010 | 30d | [Gantt bar from mid-Mar to mid-Apr] | | | | | | | | | | | | | |
| 3 | Arquitectura da Solução | 29-04-2010 | 10-06-2010 | 30d | [Gantt bar from mid-Apr to mid-May] | | | | | | | | | | | | | |
| 4 | Implementação | 10-06-2010 | 30-12-2010 | 145d | [Gantt bar from mid-May to mid-Dec] | | | | | | | | | | | | | |
| 5 | Plano de Testes | 30-12-2010 | 03-02-2011 | 25d | [Gantt bar from mid-Dec to mid-Jan] | | | | | | | | | | | | | |
| 6 | Considerações/Alterações Finais | 03-02-2011 | 24-02-2011 | 15d | [Gantt bar from mid-Jan to mid-Feb] | | | | | | | | | | | | | |
| 7 | Elaboração da Tese | 25-02-2010 | 24-02-2011 | 260d | [Gantt bar from mid-Feb to mid-Feb] | | | | | | | | | | | | | |

Figura 7 - Cronograma do projecto de Mestrado

A realização do trabalho de mestrado possui a duração de um ano. Neste período muitas foram as etapas realizadas, mas para melhor as descrever, foram agrupadas em sete grupos principais.

Descrição do problema - Através de reuniões com os responsáveis de empresa e com os intervenientes mais directos, os operadores, procurou-se nesta fase identificar inequivocamente o problema. Este período consistiu em reuniões e compilação/análise da informação recolhida.

Análise de requisitos - Mais uma vez foi necessário recolher informação dos intervenientes directos do processo. Apesar de muita da informação ter sido já recolhida na fase de descrição do problema, foi necessário o esclarecimento de situações específicas para melhor identificação dos requisitos. Esta fase é de primordial importância, pois é com base na informação aqui recolhida que será desenvolvido o trabalho futuro.

Definição da arquitectura da solução - Após a identificação dos problemas e requisitos, foi necessário idealizar a solução e respectiva arquitectura. Nesta fase, os responsáveis da empresa foram novamente consultados de forma a que a solução escolhida fosse devidamente aprovada pelas políticas internas da empresa.

Implementação - Esta fase consistiu no desenvolvimento de toda a aplicação e dos módulos que a constituem.

Plano de testes - Depois de concluído o desenvolvimento passou-se à fase de testes para validação da aplicação. Estes testes foram divididos em dois tipos diferentes: testes em laboratório e testes em contexto real de trabalho. Só após a verificação exaustiva de todas as funções do programa em laboratório, foi posto em funcionamento no ambiente real. Nesta fase os intervenientes directos usaram os dois sistemas em paralelo por forma a validar o novo sistema, e não colocar em causa o funcionamento de todo o processo.

Considerações/Alterações finais - Tal como inicialmente previsto e como é expectável na criação de uma aplicação desta complexidade e especificidade, surgiram algumas situações a corrigir ou melhorar. Mesmo com uma análise de requisitos bastante aprofundada, existiram pormenores que não foram previstos. As alterações e correcções efectuadas não colocaram em

causa nem o prazo de entrega do trabalho nem os dados até aí recolhidos, sendo realizadas rápida e directamente actualizações no sistema em funcionamento.

Escrita do relatório - Esta etapa foi realizada em paralelo com todas as outras e teve a duração total do projecto. Nela foram registadas as evoluções, análises, decisões, etc. que foram ocorrendo no decorrer das outras tarefas.

3. Funcionamento

No desenvolvimento desta aplicação, esteve sempre em mente a modularidade e o reaproveitamento de código. Sendo esta aplicação desenvolvida maioritariamente usando linguagens de programação gratuitas e amplamente conhecidas, foram aproveitadas muitas ferramentas já disponíveis em repositórios de software. Todas as licenças de utilização das ferramentas foram devidamente respeitadas.

3.1. Funcionamento geral do sistema

O sistema está dividido em dois módulos principais: o módulo *PGM* (Plano Geral de Manutenção), destinado à utilização em computadores convencionais; e o módulo *PGMMobile* (Plano Geral de Manutenção para Dispositivos Móveis).

Em ambos os módulos, o utilizador necessita de se autenticar no sistema, sendo depois “encaminhado” para as opções disponíveis no seu nível de permissões. Visto que existem vários tipos de utilizadores, cada um destes possui opções e permissões diferentes.

As funcionalidades específicas de cada módulo, bem como de cada nível de utilizador, serão apresentadas mais à frente neste relatório.

3.1.1. Arquitectura da solução

A aplicação baseia-se numa arquitectura *3-Tier* [29], que tem como principal característica encapsular os servidores de aplicações que correm e executam a lógica do negócio, colocando no servidor toda a lógica de negócio da aplicação. A principal vantagem deste tipo de arquitectura é que a máquina cliente, além de não ter necessidade de nenhum tipo de instalação de software, não necessita de grande poder de processamento. Desta forma, a arquitectura *3-Tier* é decomposta nas seguintes três componentes (camadas):

- Os clientes que correm a lógica no interface gráfico com o utilizador (*GUI*);
- Os servidores de aplicações que correm a lógica de negócio;
- As bases de dados.

Cada uma das camadas assume diferentes responsabilidades, ou seja:

A Camada **Interface Gráfico com o Utilizador (*GUI*)**, é responsável pela apresentação dos dados. Recebe todos os eventos e pedidos do utilizador e controla toda a interface do utilizador. A actual lógica de negócio é passada para o servidor.

A camada **Servidores de Aplicações** é onde residem os objectos de negócio que vão executar as regras do negócio, estando disponíveis para os clientes. O Servidor de Aplicações é ainda responsável pela segurança dos dados, isto é, não permite um acesso directo aos dados por parte dos clientes.

A camada **SGBDs - Sistema de Gestão de Base de Dados** é responsável pelo armazenamento dos dados. Os limites entre as camadas são lógicos, ou seja, é possível funcionar com esta arquitectura numa só máquina. A principal vantagem é que o sistema está estruturado ordenadamente, e existe uma função bem delineada do planeamento dos limites do software entre as diferentes camadas. Em suma, uma arquitectura de três camadas é uma implementação elaborada do ambiente cliente servidor, na medida em que as regras do negócio são controladas por um agente, que tem como principal função servir de intérprete entre os dois extremos da arquitectura. Estes podem desempenhar muitas tarefas, entre as quais, as solicitações dos clientes e o mapeamento das várias solicitações feitas a múltiplos servidores.

Para melhor integrar a aplicação com o meio envolvente, foi desenvolvido um método de importação e exportação de dados, tendo como suporte entre outros, o Microsoft Excel® [30], ferramenta bastante utilizada na empresa alvo da aplicação desenvolvida, e onde se encontram as principais fontes e destinos dos dados.

3.1.1. Ferramentas usadas na construção da solução

APACHE

O Apache [9] é o resultado de um esforço colectivo de um grupo de voluntários para o desenvolvimento de um software robusto, gratuito e com qualidade para a implementação de um servidor HTTP.

Há vários anos que não lhe são detectados nenhuns problemas de segurança graves [31], e o seu uso está tão divulgado que é pouco provável que isso venha a acontecer no futuro.

PHP

O código *PHP*[18] é embebido no documento *HyperText Markup Language (HTML)* [32], de modo a que o servidor *Web*, a que foi acrescentado um módulo *PHP*, consiga interpretar os comandos aí inseridos, e assim gerar o código *HTML*. O mais interessante do *HTML* que afecta o *PHP* é, sem sombra de dúvida, os formulários (*forms*), sendo estes a chave capaz de transmitir os dados inseridos pelo utilizador para um *script PHP*, seguindo-se o seu processamento. Deste modo, a interacção com os “*inputs*” do utilizador torna-se de primordial importância, já que, com esta possibilidade o *PHP* permite gerar código dinâmico de acordo com o pedido efectuado.

MySQL

O MySQL[13] é o programa responsável pelo armazenamento dos dados. É também de uso livre o que não implica custos de aquisição.

As suas principais vantagens passam pela maior velocidade de execução das tarefas, no facto de consumir poucos recursos computacionais e uma grande facilidade de integração com o PHP.

Javascript

O *JavaScript*[33] é uma linguagem de programação que pode ser facilmente integrada com o código HTML.

Esta é uma linguagem “*client-side script*”, que significa que é processada do lado do cliente, e não no servidor à semelhança do código PHP. Estas características permitem efectuar um grande conjunto de funções, com o objectivo de otimizar o funcionamento do sistema.

.Net Compact Framework 3.5

O Microsoft *.NET Compact Framework* (.NET CF)[34] é uma versão do *.NET Framework* para dispositivos móveis baseados em Windows CE [35], tais como *Personal Digital Assistant* (PDA) e telemóveis. A sua distribuição é gratuita e autorizada pela Microsoft. Esta *Framework* utiliza algumas das bibliotecas da *.NET Framework*, bem como algumas específicas para dispositivos móveis.

A sua utilização neste projecto centrou-se na criação do módulo para dispositivos móveis.

Microsoft Excel®

O Microsoft Excel® [30] é a única ferramenta não gratuita, contudo, não constitui qualquer necessidade de custos de aquisição, pois esta ferramenta já se encontra actualmente na empresa.

A sua escolha advém, principalmente, do facto de que todos os documentos a exportar, se encontram actualmente neste suporte, assim como, os funcionários envolvidos já se encontrarem familiarizados com a ferramenta.

jQuery

jQuery é uma *framework* que tem como um dos seus principais objectivos ajudar os programadores a concentrarem-se na lógica dos sistemas da web e não nos problemas de incompatibilidade dos navegadores actuais, bem como fornecer um conjunto de funcionalidades pré-programadas para funções que anteriormente implicavam repetição de código.

Desde o seu surgimento em início de 2006, tem sofrido constantes melhorias, tanto através de novas versões da plataforma base, como do aumento exponencial de extensões (*plugins*) que lhe adicionam funcionalidades.

Estão disponíveis *plugins* adicionais que vão desde meros efeitos visuais até à comunicação de dados entre cliente/servidor via AJAX.

Muitas operações realizadas anteriormente em Javascript puro são notavelmente reduzidas quando reescritas com esta *Framework*.

Como principais funcionalidades apresenta [20]:

- Resolução da incompatibilidade entre os *browsers*;
- Redução de código;
- Reutilização do código através de *plugins*;
- Utilização de uma vasta quantidade de *plugins* criados por outros programadores e de uso livre;
- Trabalha com AJAX e *Document Object Model* (DOM);
- Eventos;
- Efeitos e animações;
- Implementação segura de recursos do *Cascading Style Sheets* (CSS).

O uso desta *Framework* é livre pois é licenciado pelas normas *General Public License* (GNU) e *Massachusetts Institute of Technology* (MIT) Licence [36]. Este factor torna-o apelativo para todos os públicos. Recentemente a Microsoft e a Nokia começaram a introduzir esta tecnologia em alguns dos seus produtos.

Classe Listagem

A classe listagem consiste num projecto próprio do candidato, que consiste numa mini-*framework* que abstrai a camada da base de dados e da apresentação em formato HTML. Esta ferramenta consiste mais especificamente numa classe em PHP, que permite simplificar a codificação de operações com a base de dados e criação de formulários para o utilizador.

Esta ferramenta foi iniciada em alguns trabalhos académicos, nomeadamente no projecto final de curso. Nessa altura consistia apenas em alguns métodos simples que permitiam poupar algumas linhas de código repetitivo, por forma a facilitar a programação e tornando o processo de desenvolvimento mais rápido. Com a realização deste trabalho, esta ferramenta foi reaproveitada e melhorada, adicionando funcionalidades, de modo a ser uma peça fundamental deste projecto.

Uma das grandes mais-valias desta ferramenta é a facilidade de criação de formulários de manipulação de dados, quer sejam de inserir, alterar ou eliminar registo.

Esta classe utiliza um ficheiro de configuração também escrito na linguagem principal (PHP) que descreve a forma como as opções são apresentadas.

Funcionamento

Na criação da classe apenas será referenciado o nome da tabela e da operação. A classe encarrega-se de consultar a estrutura da tabela, seus relacionamentos e o seu conteúdo. Posteriormente, outras funções com base na informação da Base de Dados (BD), criam o código HTML necessário para apresentar a informação.

Num formulário de inserção, os tipos de dados dos campos das tabelas irão definir qual o controlo a apresentar. Por exemplo, um campo de texto será apresentado numa Caixa de Texto simples, enquanto um campo booleano deverá ser representado por uma caixa de selecção.

É também possível personalizar a forma de apresentação de um determinado tipo de campo, existindo várias configurações possíveis.

Dentro das novas actualizações destacam-se as seguintes:

Imagens - Campos definidos como imagem serão automaticamente apresentados com pré-visualização (nos formulários de consulta), e nos formulários de inserção todo o processo de importação e armazenamento de referência da imagem é feito automaticamente.

Pesquisa - Nos formulários de consulta, foi adicionado o componente da consulta (opcional) que apresenta os campos indicados para a filtragem.

Validação de formulários - Antes de ser submetida qualquer inserção ou alteração no sistema, o formulário será validado segundo as configurações fornecidas. Estas validações funcionam do lado do cliente, utilizando a tecnologia *Javascript*. Estas validações são um bom exemplo das mais-valias desta ferramenta. Anteriormente, cada formulário de inserção de dados necessitava de possuir código específico para validar os campos dos formulários. Esta classe cria esse código dinamicamente segundo os critérios da base de dados (ex. campos de preenchimento obrigatório) ou segundo critérios específicos, tudo isto sem escrever qualquer código Javascript.

Como dito anteriormente, esta ferramenta evoluiu bastante desde a última actualização, tornando-se numa ferramenta importante no desenvolvimento deste e de outros projectos. Estou convicto que no futuro, esta continue em utilização e seja alvo de evolução.

Outras

Foram usadas ainda outras ferramentas que permitiram efectuar variadas funções:

Greybox

O *GreyBox* é uma ferramenta *open-souce* orientada para a forma de apresentar a informação no *browser*. Permite apresentar as tradicionais “Janelas *popup*” de uma forma mais

simples e agradável aos utilizadores, sem recorrer à abertura de novas janelas. Permite ainda apresentar páginas HTML, imagens ou outros tipos de conteúdo.

As suas principais características são:

- Não entra em conflito com bloqueadores de janelas nem qualquer tipo de antivírus ou afins;
- Simples de utilizar e configurar;
- O seu estilo pode ser personalizado para melhor se enquadrar na aplicação;
- Tamanho dos ficheiros muito reduzido;
- Melhora substancialmente a qualidade gráfica da aplicação.

Na esmagadora maioria das vezes, a sua utilização nesta aplicação está encapsulada dentro da classe listagem.

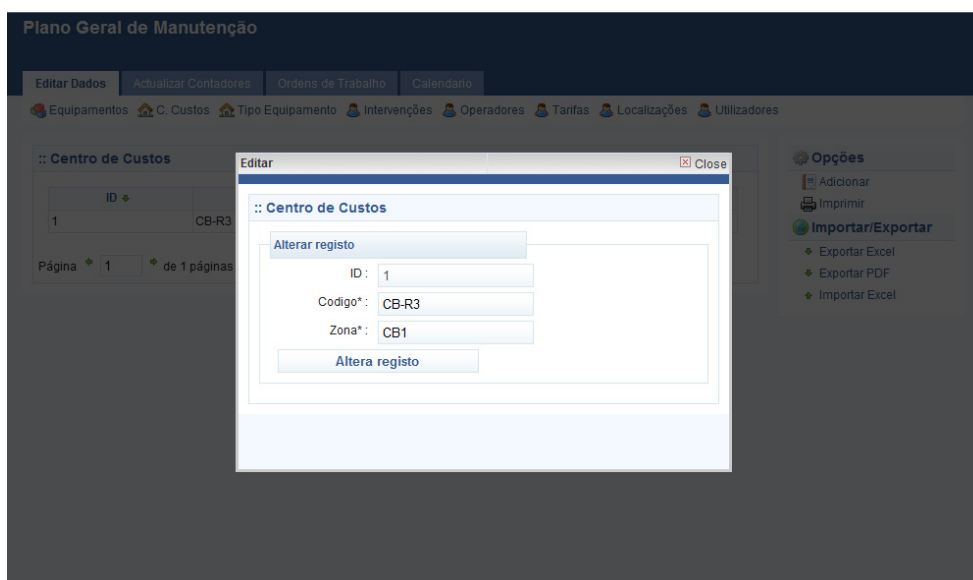


Figura 8 - Exemplo de utilização do Greybox na aplicação

Tinymce

O TinyMCE [37], também conhecido como o *Tiny Moxiecode Content Editor*, é um editor WYSIWYG (What You See Is What You Get) HTML, criado em *Javascript* lançado pela empresa *MoxiecodeSystems AB* sob a licença LGPL. Tem a vantagem de ser independente do tipo de *browser* em utilização, e a capacidade de converter campos de texto “*Textarea*” em editores de texto com variadas funções de formatação.

Esta ferramenta pode ser facilmente personalizada para se adaptar à realidade em que está inserida. Neste caso específico, visto que não são necessárias funções avançadas de formatação, a sua configuração apresenta apenas as opções mínimas.

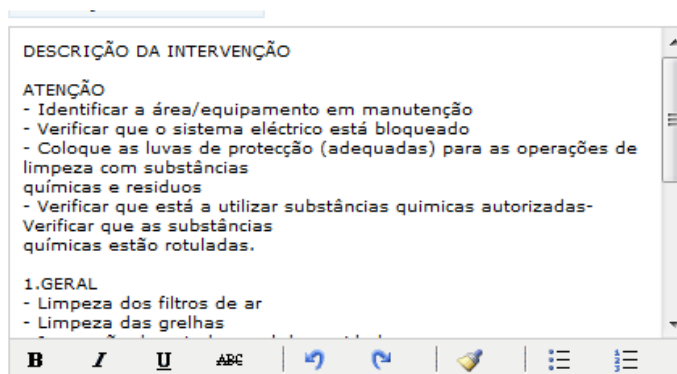


Figura 9 - Exemplo de utilização do TinyMCE na aplicação

ADODB for PHP

As livrarias ADODB for PHP [38] permitem uma abstracção da camada da base de dados. Facilita a programação em termos de interacção com a base de dados, abstraindo totalmente a sua estrutura e funcionamento interno.

As suas principais vantagens são:

- Optimizado para a velocidade. É provavelmente a livraria de abstracção de base de dados mais rápida actualmente [39];
- Portabilidade entre SGBD's realizada muito facilmente. A qualquer altura a migração para outro SGBD necessita apenas de ligeiras alterações no ficheiro de configuração.
- Fácil de utilizar, apresenta um funcionamento muito semelhante à sua homóloga da Microsoft ADO.NET[40];
- Maturidade, desde a sua criação no ano 2000 tem sido utilizada e melhorada amplamente por uma grande comunidade de utilizadores e programadores, e usada em muitas das mais conhecidas aplicações Web.
- Licenciamento BSD [41], ou seja, pode ser usada e incorporada noutras aplicações de forma totalmente livre.

Neste trabalho especificamente, a sua utilização foi na maioria das situações usada de forma encapsulada dentro da classe listagem.

VCalendar

A ferramenta VCalendar [42] consiste numa aplicação *de código aberto* para gestão de calendários para múltiplos utilizadores.

A sua incorporação na aplicação PGM permite a criação de eventos específicos para utilizadores específicos, grupos de utilizadores, gerir eventos individuais ou recursivos, etc.

Apesar de esta ferramenta estar preparada para funcionar de forma isolada, a sua incorporação numa outra aplicação permite criar valor adicional, tirando partido das suas poderosas funcionalidades.

Várias foram as ferramentas usadas neste projecto, umas mais *standard* que outras, mas todas elas devidamente testadas antes da sua inclusão. Algumas das ferramentas menos comuns aqui utilizadas foram, após análise exaustiva, as mais adequadas para o fim a que se destinam.

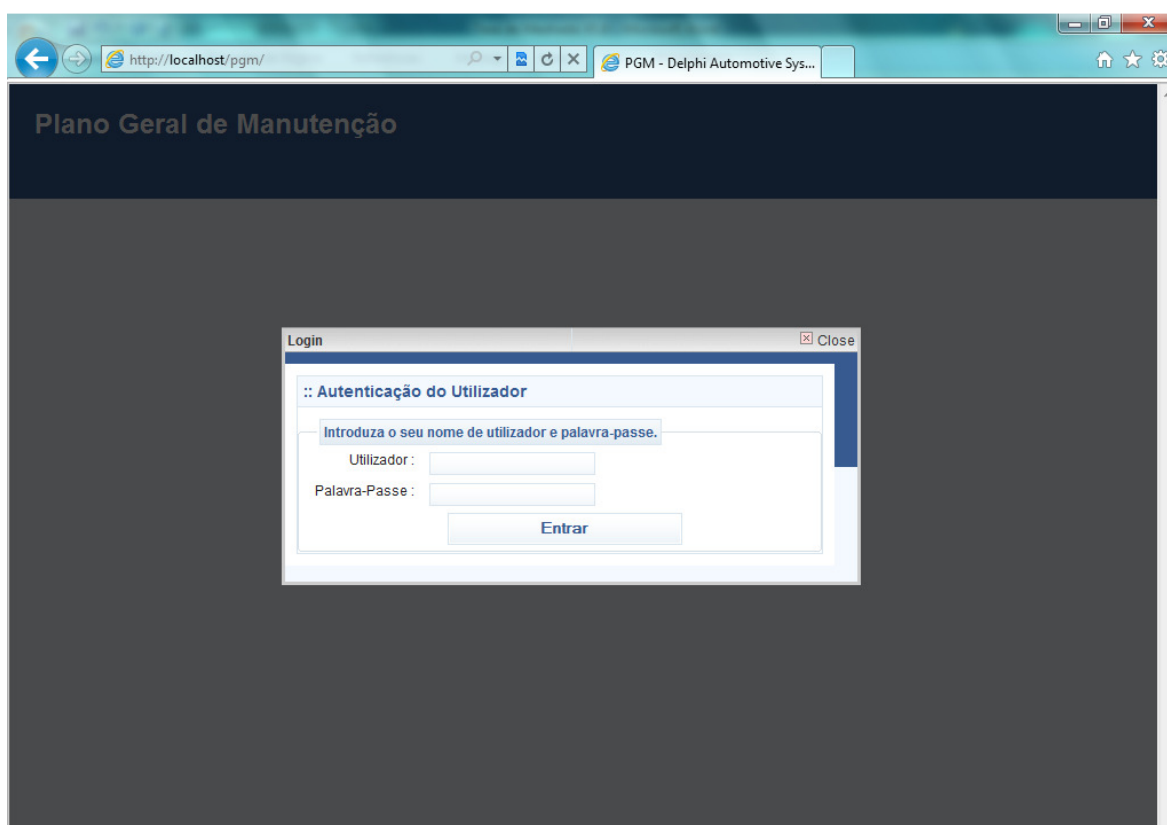


Figura 11 - Autenticação na aplicação

Nesta fase, a totalidade dos utilizadores são conhecedores do idioma português, e por esse motivo todos os menus se encontram neste idioma, no entanto ambas as aplicações (PGM e PGMMobile) se encontram preparadas para receber outros idiomas, bastando para isso a criação do respectivo ficheiro das linguagens. Depois disso, a aplicação estará preparada para funcionar noutros países, pois esta hipótese foi considerada aquando da análise de requisitos com os intervenientes da empresa.

4.2.1. Operador

Este nível de utilizador destina-se aos operadores da manutenção, que no sistema actual não tinham qualquer contacto com o registo digital da informação. As suas tarefas no sistema passam por registar as fichas de manutenção e actualizar o número de ciclos.

Para minimizar a complexidade do sistema e reduzir as possibilidades de erros, este utilizador não possui acesso a nenhuma outra função.

Depois de autenticado, as opções disponíveis são:

- **Actualização de contadores:**

A imagem seguinte ilustra o formulário de actualização de ciclos. Em primeiro lugar, o operador deverá escolher o equipamento a actualizar através de uma lista predefinida. O método de escolha do equipamento foi desenvolvido usando AJAX [19] para facilitar a sua pesquisa e selecção.

Após a escolha do equipamento, serão exibidos os campos para registo da data (pré-preenchida com a data actual) e número de ciclos. Para informação do operador é também exibido os dados da última leitura.

The screenshot shows a web application interface titled "Plano Geral de Manutenção". It features a navigation bar with tabs: "Editar Dados", "Actualizar Contadores" (selected), "Ordens de Trabalho", and "Calendario". Below the navigation bar are three icons: "Actualização Manual", "Actualização Automática", and "Ver Histórico". The main content area is a form titled "Escolha o equipamento a actualizar". The form contains the following fields and values:

- Equip: ROB2
- Leitura Anterior: 12377 em: 2011-12-11
- Nº Ciclos: (empty field)
- Data: 2011-02-04
- Previsão Dia: 2,8

At the bottom of the form are two buttons: "Guardar" and "Voltar".

Figura 12 - Actualização de contadores - Operador

- **Registar a ficha da manutenção:**

Como o nome indica, esta função serve para o operador registar a ficha da manutenção correspondente a uma ordem de trabalho, em que se registará algumas informações como a data, tempo gasto, operações realizadas, etc.

Para a realização desta tarefa, o operador terá de escolher através de uma lista qual a ordem de trabalho a que se refere a manutenção, lista esta que foi previamente criada pelo gestor da aplicação.

The screenshot displays the 'Plano Geral de Manutenção' interface. At the top, there are navigation tabs: 'Editar Dados', 'Actualizar Contadores', 'Ordens de Trabalho', and 'Calendario'. Below these, there are icons for 'Ordens em Aberto', 'Todas as Ordens', and 'Criar OT'. The main form is titled 'Criar nova Ordem de Trabalho' and contains the following fields:

- Equip: ROB2
- Nome: ROB2
- Data Prevista*: [Empty field]
- Intervenção*: AAP XX P 01
- Tipo*: Curativa
- Centro Custos*: CB-R3
- Supervisor*: António Barata

Below these fields is a section titled 'Detalhes' with a sub-tab 'Empregado'. This section contains three input fields:

- D. Plan.*: [Empty field]
- D. Inicio*: [Empty field]
- D. Fim*: [Empty field]

At the bottom of the form, there is a note: '* - Campos de preenchimento obrigatório'. Below the note are two buttons: 'Guardar' and 'Voltar'.

Figura 13 - Registo ficha manutenção - Operador

- **Calendário e alertas**

Este componente da aplicação foi outra das características diferenciadoras em relação à concorrência. É mais um exemplo da personalização da aplicação para se adaptar convenientemente aos requisitos do cliente.

A secção do calendário e alertas permite informar os utilizadores de diversas situações, mediante o seu nível de permissões.

Para os utilizadores Operadores, é possível visualizar as suas manutenções/reparações passadas, abertas e futuras. As passadas apenas servem de histórico, sendo apenas de carácter informativo. As ordens em aberto permitem ao utilizador gerir o seu trabalho, garantindo que nada é esquecido. É comum um operador possuir várias ordens em aberto simultaneamente, pois, por diversos factores, nem sempre é possível iniciar e terminar a operação sem interrupções.

A visualização de ordens futuras permite ao utilizador antever os seus próximos dias de trabalho. Apesar de todos os operadores se reunirem frequentemente com as suas chefias directas, esta funcionalidade permite também melhorar o fluxo de informação, pois a informação das ordens futuras ficarão automaticamente disponíveis no equipamento após o utilizador *Gestor* as atribuir ao operador.

Os alertas são outro componente importante do sistema, informando o utilizador sempre que existam situações anormais como: ordens em aberto em que o tempo previsto de conclusão seja ultrapassado, ou ordens que já deveriam ter sido iniciadas.



Figura 14 - Calendário - Operador

- **Alterar dados pessoais:**

Esta opção permite ao utilizador alterar os dados da sua conta de utilizador, nomeadamente o nome de utilizador, palavra passe e turno.



Figura 15 - Alterar dados pessoais - Operador

4.2.2. Gestor

O gestor representa o nível intermédio de permissões. Este tem todas as funcionalidades disponíveis para os operadores e muitas outras:

- **Edição de dados do programa (equipamentos, localizações, intervenções, etc.).**

Esta funcionalidade é integralmente gerada pela classe listagem e é usada para vários tipos de informação, tais como: equipamentos, centro de custos, tipo de equipamento, intervenções, operadores, tarifas e localizações. Cada um deles divide-se em Inserir, Alterar, Eliminar e Listar. Todos têm um funcionamento semelhante, para simplificar a descrição. Não serão abordados individualmente mas sim de uma forma genérica.

- **Listar**

Esta opção permite listar todos os registos de um determinado tipo (Ex. Equipamentos). É possível escolher o número de resultados a exibir por página, ordenar por qualquer um dos campos, pesquisar, visualizar os resultados, etc.

Para cada registo são apresentadas as opções de visualização detalhada, alterar e eliminar. Estas são as opções originais, podendo a classe listagem ser configurada para qualquer outra apresentação.

Por defeito, a utilização da classe listagem irá exibir todas as colunas, com o nome existente na base de dados, no entanto, é possível alterar esta disposição, indicando para isso quais os campos a ignorar, bem como os nomes relativos a cada uma das colunas.

Na figura seguinte será exibido a título de exemplo a listagem de intervenções.

| ID | COD | T.Equipamento | Desc. | Acções |
|-----|-------------|-------------------------------------|--|--------|
| 461 | AAP XX P 01 | Aparelhos de Ar Condicionado | Intervenção Electromecânica - Mensal | |
| 463 | ABR YY P 01 | Abre Bagues | Intervenção Electromecânica - Semestral | |
| 464 | ALE TR P 01 | Alicates de Estripar Térmicos Aston | Intervenção Electromecânica - Trimestral | |
| 465 | AMP XX P 01 | Máquina de Cravar MT-E2 | Intervenção Electromecânica Quadrimestral | |
| 466 | APA XX P 01 | Aparafusadora Pneumática | Interv. Electromecânica - Mensal | |
| 467 | APL YY P 01 | Rede de Águas Pluviais | Intervenção Electromecânica Semestral | |
| 468 | ARA YY P01 | Máquina de corte ARA | Interv. Electromecânica - Semanal ou 105 000 ciclo | |
| 469 | ARA YY P02 | Máquina de corte ARA | Interv. Electromecânica - Mensal ou 5 000 000 cicl | |
| 470 | ARA YY P03 | Máquina de corte ARA | Interv. Electromecânica - Anual | |
| 471 | ART DL P 01 | Desenrolador de fio Artos | Interv. Electromecânica - 1.500.000 ciclos | |
| 472 | ART YY P 01 | Prensa Artos | Interv. Electromecânica - 1.500.000 ciclos | |
| 473 | ASP YY P 01 | Separadores de água-óleo | Intervenção Electromecânica - Trimestral | |
| 474 | ATB YY P 01 | Aparelho de Teste Eléctrico ATBU | Intervenção Electromecânica - Semestral | |
| 475 | BAL YY P 01 | Balizas | Intervenção | |
| 476 | BAN DM P 01 | Painel dimensional | Intervenção Electromecânica 10.000 C ou 12 Meses | |
| 477 | BOM YY P 01 | Bombas de Água | Intervenção Electromecânica Trimestral | |
| 478 | BRA YY P01 | Máquina de Braiding | Intervenção Electromecânica Semanal | |
| 479 | BRA YY P02 | Máquina de Braiding | Intervenção Electromecânica Mensal | |
| 480 | BRA YY P03 | Máquina de Braiding | Intervenção Electromecânica Semianual | |
| 481 | CAI DE P 01 | Cais de Carga e Descarga | Intervenção Electromecânica Semestral | |

Figura 16 - Editar dados do programa, Listar - Gestor

- **Inserir, Alterar e Eliminar**

Mais uma vez recorrendo à classe listagem, este formulário é exibido dinamicamente, sendo semelhante para as três opções, à excepção da acção do botão (Inserir, Alterar e Eliminar) e forma como o formulário é apresentado, em branco na opção de inserir e preenchido nas opções de alterar e eliminar.

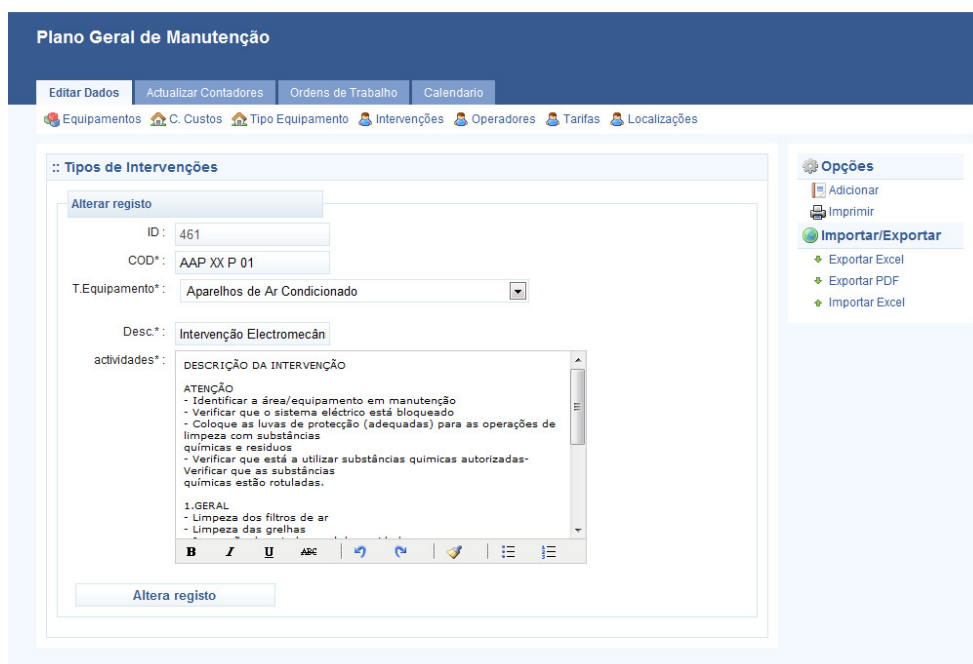


Figura 17 - Editar dados do programa, Editar - Gestor

- **Registo e actualização de ciclos dos equipamentos**

Esta secção permite ao Gestor registar leituras de ciclos dos equipamentos, consultar históricos e médias de utilização, bem como efectuar a importação de informação do sistema HPC [44].

- **Registo de leituras de contadores**

Para registar o valor da leitura, o utilizador terá primeiro de escolher o equipamento. Para facilitar a escolha, pois a lista contém milhares de entradas, foi usado AJAX, parte integrante da *framework* jQuery [20].

Quando o utilizador inicia a escrita do nome do equipamento, o jQuery encarrega-se de, a cada tecla pressionada, consultar a base de dados e apresentar os resultados (função *autocomplete*). O utilizador pode optar por completar a escrita ou escolher uma opção da lista. Este método, apesar de conter alguma complexidade, é realizado de forma muito rápida, pois as consultas à base de dados bem como a comunicação cliente/servidor estão perfeitamente optimizadas.

Após a escolha de equipamento é exibida a informação da última leitura registada, assim como os campos para introdução dos novos valores. Ao premir o botão inserir, o utilizador é “encaminhado” para o início, a escolha de um novo equipamento.

Mais uma vez a ferramenta jQuery é usada para que todas as comunicações com o servidor sejam realizadas em *background*, sem uma única actualização da página. Este método permite otimizar a velocidade de cada operação e enriquecer a experiência de utilização da aplicação.

Estes métodos de comunicação com o servidor são cada vez mais usados nos ambientes Web, pois permitem reduzir o tráfego de rede necessário e consequentemente o aumento da performance.

Neste caso o aspecto da aplicação é em tudo semelhante à apresentada na Figura 12.

- **Actualização automática dos contadores**

Esta funcionalidade constitui uma das principais mais-valias desta aplicação, visto que neste passo é reduzido em muito o tempo dispendido pelos operadores na recolha de dados manualmente.

Através de uma conta de utilizador aberta na base de dados do software HPC, com acesso a determinadas tabelas e acesso de leitura, é possível recolher uma estimativa muito aproximada do número de ciclos de cada equipamento.

O cálculo dos ciclos funciona da seguinte forma:

1. É recolhido do HPC o número de peças (Cabos) produzidas;
2. Cruzando a informação do número de série da peça (*Part Number*) com a tabela de localizações, é possível saber quais os equipamentos utilizados na produção;
3. Como um equipamento pode ser utilizado mais que uma vez na produção de uma peça, existe também uma tabela de utilizações, convertendo assim o número de peças em ciclos dos equipamentos.

Tal como descrito anteriormente, esta operação depende da existência das tabelas das localizações e ciclos por *Partnumber*. Apesar desta informação já existir na empresa, encontrava-se descentralizada e em formatos diferentes, o que obrigou a um esforço adicional para reunir e inserir nos formatos indicados.

Após estas configurações iniciais, a aplicação permite também criar uma centralização da informação, podendo esta ser acedida e modificada de forma mais fácil e consistente.



Figura 18 - Atualização automática de contadores - Gestor

- **Ver histórico**

O histórico dos contadores é uma peça importante no funcionamento desta aplicação. A sua existência permite analisar o desgaste dos equipamentos, prever a data da próxima intervenção, etc.

Esta opção permite visualizar as leituras de cada equipamento, corrigir valores inválidos, ter uma previsão do número de utilização por intervalo de datas, etc.

Mais uma vez é utilizada a *framework* jQuery de forma semelhante à opção de registo de leituras.



Figura 19 - Visualizar histórico dos contadores - Gestor

- **Ordens de trabalho**

A secção correspondente às ordens de trabalho para o utilizador divide-se em três partes: visualizar ordens em aberto, visualizar todas as ordens e criar nova ordem, cada uma delas com diversas opções.

Internamente, esta secção é composta apenas por três partes principais: a listagem, a edição e a impressão de ordens. A listagem é usada tanto para visualizar as ordens em aberto como para todas as ordens, mudando apenas a configuração da listagem. Para visualizar as ordens em aberto é aplicado um filtro ao campo estado (invisível ao utilizador), e na segunda opção, são listadas todas as ordens. Adicionalmente, em ambas as listagens é apresentada a opção de pesquisar, em que o utilizador tem a possibilidade de filtrar a pesquisa por diversos critérios como operador, tipo de intervenção, tipo de equipamento, centro de custos, etc.

A edição de ordens, apesar de para o utilizador ter uma dupla função (criar nova e editar existente), internamente é apenas um módulo, apresentando diferentes configurações. Para criar uma nova ordem, o formulário é apresentado em branco. Depois de preenchido pelo utilizador e validado pelo sistema, é inserido na base de dados. No segundo caso (edição), o mesmo formulário é apresentado com os campos já preenchidos, podendo o utilizador alterá-los. Após a alteração e validação dos dados, são actualizados na base de dados. Existem ainda algumas restrições neste processo, como a impossibilidade de alteração de ordens fechadas ou arquivadas. Por motivos de segurança e integridade dos dados, só um utilizador administrador poderá reabrir ou restaurar ordens. Estas e outras condicionantes funcionais foram requeridas pela empresa.

| ID | Equip. | Inicio Previsto | Intervenção | Tipo | D. Inicio | Operador | Estado | Acções |
|----|--------|-----------------|-------------|-----------|------------|----------|--------|--------|
| 22 | Equip2 | 1999-01-01 | AAP XX P 01 | Curativa | 1999-10-23 | Manuel | Aberta | |
| 9 | ROB2 | 2010-12-01 | BAL YY P 01 | Preditiva | 2010-12-01 | Manuel | Aberta | |

Figura 20 - Visualizar das ordens de trabalho - Gestor

- **Calendário e alertas**

Para os utilizadores gestores, este módulo permite acompanhar a evolução do trabalho em curso, visualizar e escalonar trabalhos futuros, bem como ser alertado de situações anormais.

Uma das mais-valias deste sistema passa pela possibilidade de visualizar as futuras necessidades de manutenções até um mês de antecedência. Como descrito anteriormente, o intervalo entre manutenções pode ser definido em tempo ou em ciclos. Neste caso, com base no histórico, é possível prever com bastante exactidão qual a data da próxima intervenção.

O gestor poderá então distribuir as ordens de trabalho pelos diversos operadores. Esta tarefa é facilitada, pois está também disponível a opção de visualizar individualmente os seus calendários.

Com a implementação desta funcionalidade, surgiu a ideia de apresentar também a carga de trabalho (*Workload*) de cada operador, bem como a possibilidade de atribuição automática de

ordens. No entanto, devido à falta de alguma informação importante, não foi ainda possível implementá-la nesta fase o seu desenvolvimento, ficando previsto e registado para desenvolvimentos futuros.

Visto que os calendários dos vários operadores estão disponíveis para o gestor, este pode também visualizar o estado dos trabalhos em curso, facilitando assim a tomada de decisões preventivas ou curativas relativamente a atrasos.

Os alertas funcionam de forma semelhante à dos operadores (alertando situações anormais), mas neste caso a informação refere-se a todos os operadores.



Figura 21 - Visualização do calendário - Gestor

- **Fichas reparação**

Este módulo assume diferentes configurações, pois está presente na aplicação principal e no dispositivo móvel.

Na versão principal, está disponível para os utilizadores Operadores e Gestores, apresentando o mesmo formato e configurações, com a única diferença de que para o operador apenas estão disponíveis aquelas que lhe foram atribuídas, enquanto o gestor tem acesso a todas.

À semelhança da versão móvel, o formato do formulário foi criado de forma a ser o mais parecido possível ao formato em papel, para que a adaptação seja o mais simples possível.

- **Gestão de utilizadores:**

Esta opção permite ao gestor a criação, edição e eliminação de contas de utilizadores operadores. Para resolução de problemas, como senhas de acesso perdidas, este tipo de utilizador possui permissões necessárias para solucionar estes problemas.

Este modelo de acesso não permite a criação de utilizadores gestores nem administradores.

4.2.3. Administrador

- **Edição de dados do programa**

Esta opção é usada para vários tipos de informação tal como nos gestores. No entanto, devido ao superior nível de permissões, este tipo de utilizadores tem acesso a toda a informação, tanto a disponível para os gestores como outras.

Para a implementação deste sistema noutras localizações, ao administrador basta apenas criar as novas localizações, centros de custo e respectivos utilizadores.

O funcionamento interno desta secção é em tudo semelhante à disponível para os gestores, com a diferença de este possuir mais informação disponível.

- **Calendário e alertas**

Para este tipo de utilizador, apenas está disponível a informação das ordens passadas, actuais e futuras, e, visto que não é da sua responsabilidade realizar a distribuição de trabalho, esta opção não está disponível.

- **Gestão de utilizadores:**

Esta opção permite ao administrador a criação, edição e eliminação de contas de todo o tipo de utilizadores. Este tipo de acesso permite realizar a gestão de todos os utilizadores. No entanto, destina-se principalmente à criação de novas contas para gestores. Apesar de possuir permissões para a criação de operadores, esta tarefa é da responsabilidade dos gestores.

4.3. PGMMobile

A aplicação móvel destina-se aos utilizadores operadores e gestores. As funcionalidades de ambos são idênticas, visto que esta aplicação apenas contempla duas funcionalidades: actualização de ciclos dos equipamentos e introdução da ficha de reparação.

A utilização de dispositivos móveis por estes utilizadores é novidade para eles, portanto, será necessário realizar uma formação para ambientação a estes equipamentos, além da formação específica da aplicação.

Tal como qualquer outra aplicação, esta requer a autenticação no sistema, não só por questões de segurança, mas também para facilmente identificar o utilizador, visto que cada equipamento poderá ser usado por diferentes operadores.



Figura 22 - Autenticação - PGM Mobile

- **Actualização de ciclos dos equipamentos**

O processo de actualização de ciclos funciona de forma semelhante à da aplicação principal. O utilizador escolhe o equipamento pretendido e, seguidamente, o número de ciclos. Para tornar o processo menos susceptível a erros, o nome do equipamento será escolhido através de uma lista predefinida, e também apresentado o valor da última leitura e uma sigestão da leitura actual baseada na data, também para despistagem de erros.

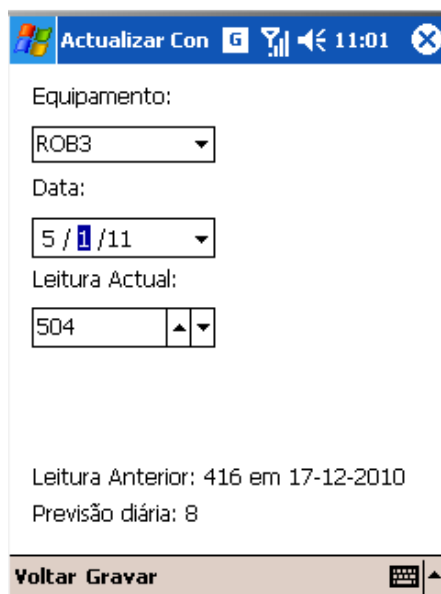


Figura 23 - Actualização dos contadores - PGM Mobile

- **Visualização do calendário**

À semelhança da aplicação principal, o calendário apresenta ao utilizador os eventos passados, presentes e futuros. Os utilizadores conseguem assim saber eficazmente que trabalhos têm em aberto e quais os próximos.

À medida que os trabalhos vão sendo concluídos, também o seu estado na aplicação é alterado, sem necessidade de sincronização com o servidor.

Sempre que existam trabalhos em atraso, é também criado um alerta na aplicação até que este seja concluído.



Figura 24 - Visualização do calendário - PGMMobile

- **Ficha de reparação**

O preenchimento da ficha de reparação em formato digital é um processo que não existia, pois o seu preenchimento era em papel. O desenho do formulário foi realizado para o tornar o mais parecido possível com o formato original, facilitando assim a habituação ao sistema.



Figura 25 - Emissão ficha de reparação - PGMMobile

- **Sincronização da aplicação**

O cenário ideal de utilização desta aplicação seria num formato “sempre ligado”, tendo ligação permanente ao servidor, não necessitando assim de armazenar qualquer tipo de informação, consultando e actualizando directamente a base de dados principal.

Visto que actualmente na empresa não existe cobertura de rede sem fios, nem está prevista a sua implementação, a aplicação funcionará num formato “desligado”, sem ligação permanente. Será necessário que regularmente o equipamento seja sincronizado com o servidor, existindo para isso algumas localizações específicas para realizar essa operação. Todo o processo é realizado automaticamente caso não ocorram conflitos de informação na base de dados. Se isso acontecer, é necessária a intervenção humana para resolver os conflitos.

A nível de sincronização com o sistema principal, foi necessário ter em conta as limitações inerentes aos sistemas móveis, nomeadamente a performance e capacidade de armazenamento.

Alguns dados da informação terá de estar sempre presente no dispositivo, como nomes dos equipamentos, localizações e últimas leituras. A restante informação é sincronizada em apenas um sentido, no do servidor, não sendo copiada qualquer informação adicional para o dispositivo e a desnecessária será apagada.

4.4. Funcionamento de componentes específicas

4.4.1. Ajax na conclusão automática

Em inúmeras situações, um utilizador de sistemas informáticos necessita de escolher um valor de uma lista predefinida. Esta situação é muito frequente tanto em aplicações Web como Desktop, para tornar a aplicação o mais funcional possível. O método de preenchimento foi tido em conta para que permitisse simultaneamente ser rápido e à prova de erros.

O método utilizado provém de um *plugin* da Framework jQuery, designado de *autocomplete*, permitindo que, à medida que o utilizador introduz os caracteres, vão sendo apresentados os registos que lhes correspondam. O utilizador pode optar por escolher a opção desejada através de um clique com o dispositivo apontador ou finalizar a escrita. Os resultados provêm da base de dados, o que garante que estejam devidamente actualizados, permitindo ainda eliminar a introdução de valores inválidos.

Este processo é realizado recorrendo à tecnologia AJAX, a qual permite realizar consultas ao servidor e à base de dados, de forma totalmente transparente ao utilizador.



Figura 26 - Conclusão automática com Ajax

4.4.2. Ajax na apresentação e gravação de formulários

A apresentação e gravação de formulários, sendo um processo presente na maioria das aplicações, constitui um processo que deverá estar o mais otimizado possível. Nesse sentido, o método utilizado conta com o apoio da Framework jQuery, a qual permite que este processo seja mais rápido e mais agradável ao utilizador.

Nesta aplicação, em algumas situações, é necessário escolher um elemento de uma lista, como explicado no ponto anterior. Após essa escolha, em vez de a página ser submetida e recarregada, o formulário surge naturalmente, sem ser necessário clicar em qualquer botão ou submissão da página. Esta funcionalidade depende, como já foi dito, do jQuery e recorre à função *innerHTML*[45] do Javascript, possibilitando alterar o conteúdo de qualquer página HTML sem submissão de formulários.

Também na gravação dos dados, a informação é enviada para o servidor em *background*, e o resultado da operação apresentado ao utilizador.

Todo este processo é realizado sem um único recarregamento da página. Pelos métodos tradicionais, seriam necessários três, revelando-se num atraso substancial, considerando que muitas destas operações são realizadas com grande frequência.

4.4.3. Criação de ficheiros PDF

Um dos formatos mais usados nos nossos dias para criação e partilha de documentos é, sem dúvida, o PDF (*Portable Document Format*) [46]. Nesse sentido, vários dos relatórios criados pela aplicação são neste formato. A sua escolha deveu-se principalmente ao facto de muitos dos relatórios criados, serem posteriormente enviados via correio electrónico e o seu reduzido tamanho facilita as comunicações.

O processo de criação é realizado através das funções presentes no PHP e dependem também de consultas à base de dados.

Os ficheiros criados serão mantidos no servidor por forma a estarem disponíveis no histórico da aplicação. Ainda assim, poderão sempre ser gerados novamente a qualquer altura. Está também presente na aplicação um sistema de eliminação automática de ficheiros antigos, que apaga todos os anteriores a um ano.

4.4.1. Criação de Ficheiros .xls

Muitos dos ficheiros gerados pela aplicação estão em formato PDF. No entanto, outro tipo de relatório poderá ser alvo de diferentes tipos de análise. Nestes, será criado o ficheiro no formato .xls para posterior tratamento no Microsoft Excel. Esta funcionalidade foi um requisito da empresa, pois o Microsoft Excel faz parte do pacote de software *standard* da empresa, estando disponível em todas as estações de trabalho e é muito usado para análise e compilação de informação.

Estes ficheiros são criados via PHP e sem recurso a nenhum outro tipo de software de terceiros.

4.5. Instalação e testes finais

A empresa disponibilizou um servidor para a aplicação, e a nível de estações de trabalho, não foi contemplada qualquer aquisição de novo hardware. Visto que a aplicação é executada no cliente através de um browser e o consumo de recursos é muito baixo, serão usadas as actuais estações de trabalho afectas ao departamento em questão, neste caso, o departamento de manutenção.

Nas estações destinadas a realizar a sincronização entre o dispositivo móvel e o servidor, foi necessário instalar o respectivo software de sincronização.

Os dispositivos móveis serão disponibilizados aos utilizadores já com a aplicação instalada. Ainda assim, as estações de trabalho destinadas à sincronização poderão também instalar a aplicação móvel nos dispositivos. Este processo poderá ser usado para actualização de novas versões e também, no caso de aquisição de novos equipamentos, instalar a aplicação.

A operação mais trabalhosa neste processo foi, sem dúvida, a instalação no servidor. Apesar de possuir muitos recursos disponíveis, está também a ser usado por outras aplicações, sendo por isso necessário verificar a existência de incompatibilidades entre os diversos serviços em execução.

Antes de instalar a aplicação foi necessário preparar o servidor: verificar a compatibilidade do servidor Web, o sistema de gestão de base de dados e o interpretador PHP.

Após a verificação da não existência de conflitos ou incompatibilidade entre os vários serviços e softwares, foi então instalada a aplicação. O processo dividiu-se nas seguintes etapas:

Instalação da base de dados: No desenvolvimento foi usado um servidor de teste, que serviu também de estação de desenvolvimento. A instalação da BD no servidor final consistiu em exportar a estrutura das tabelas e relacionamentos.

Instalação da aplicação: Visto que o PHP é uma linguagem interpretada, a instalação da aplicação propriamente dita, consiste apenas em copiar os ficheiros para a localização correcta.

Carregamento da informação inicial: Este passo representou o maior consumo de tempo de todo o processo de instalação, pois foi necessário introduzir bastante informação na base de dados, informação essa que é imprescindível para o correcto funcionamento da aplicação. Antes

dos utilizadores poderem, por exemplo, criar uma ordem de trabalho, é necessário que exista na BD a listagem dos equipamentos, dos tipos de intervenções, etc.

Muitas vezes, neste tipo de aplicação, os utilizadores ou administradores ficam responsáveis pelo carregamento inicial da informação. Neste caso, para facilitar o processo e aumentar a satisfação da empresa, a informação foi carregada *à priori*. O processo de inserção da informação consistiu na recolha de vários ficheiros de Excel presentes na empresa, e, depois de compilados, foram importados usando um processo semi-automático.

Configuração das estações de trabalho: Neste tipo de arquitectura, os clientes apenas necessitam de possuir uma ligação à rede e um browser. Assim sendo, a única necessidade de configuração nos clientes foi a criação de um atalho para a aplicação para um mais rápido acesso.

Depois do processo de instalação concluído foram realizadas várias sessões de formação aos diversos utilizadores e tipos de utilizadores. Esta formação foi realizada já com o sistema em funcionamento com dados reais, por forma a que fosse o mais eficaz possível.

Para facilitar o processo de familiarização com a aplicação dos utilizadores actuais, bem como próximos que venham a utilizar a aplicação, foram criados manuais de utilização da aplicação.

Para a aplicação PGM existem três tipos diferentes de manuais, dependendo do tipo de utilizador: Administrador, Gestor e Operador.

Para a aplicação PGMMobile, visto que as suas funcionalidades são comum aos dois tipos de utilizador em questão (Gestor e Operador), foi apenas criado um único manual.

5. Análise dos resultados

5.1. O antes e o depois

Nesta secção do relatório, serão descritas as principais evoluções manifestadas no decorrer deste projecto. Para melhor identificá-las, serão caracterizadas em tópicos, acompanhadas da sua situação antes e depois.

Registo do número de ciclos dos equipamentos

Neste aspecto, a redução de tempo foi muito significativa. No cenário anterior um operador precisava de cerca de um dia por semana para a recolha e registo do número de ciclos. Actualmente, esse tempo foi reduzido para menos de metade. Apesar da instalação do novo sistema ser muito recente e não ser possível tirar conclusões de médio e longo prazo, prevê-se também uma redução considerável no número de erros de introdução.

Criação de Ordens de Trabalho e relatório de avarias

Apesar de anteriormente o processo de emissão e registo de ordens de trabalho ser através de ferramentas electrónicas, todo o processo envolvia a utilização de papel. Com esta nova ferramenta, foi possível reduzir consideravelmente a sua utilização. Esta redução deve-se maioritariamente ao uso do módulo PGMMobile, em que a informação é registada directamente no sistema, minimizando assim o tempo gasto e os erros que possam ocorrer. Neste novo processo, a informação é preenchida apenas uma vez e não em duplicado como anteriormente.

A redução de papel não foi de 100%, pois o procedimento interno da empresa contempla a existência de histórico dos documentos em formato papel. Se assim não fosse, o papel poderia ser eliminado por completo deste processo.

Criação de relatórios de avarias

Outro dos casos de franca evolução consistiu na criação de vários tipos de relatórios relacionados com as avarias e consequentes paragens nos equipamentos. Anteriormente, criar este relatório consistia num trabalho moroso e baseado muitas vezes no conhecimento pessoal dos operadores. Com o novo sistema, é possível, em poucos segundos, gerar o relatório preciso com os parâmetros pretendidos. Desta forma, é possível analisar convenientemente os problemas para um maior apoio na tomada de decisões sobre, por exemplo, substituição de equipamentos.

Análise da informação

Os relatórios indicados no ponto anterior são apenas uma parte da informação que pode ser extraída da aplicação. Sendo um dos problemas da aplicação anterior, a falta de documentação de apoio e análise, este ponto teve primordial importância na fase de análise de requisitos e, consequentemente, no desenvolvimento. Com a implementação desta aplicação,

tanto os operadores como os decisores possuem um maior conjunto de informação útil, tais como: frequências de utilização por tipo de equipamento, tarefas realizadas por operador, etc.

Para melhor ilustrar as características desta aplicação face às soluções concorrentes referenciadas no estado da arte, será novamente apresentada a tabela comparativa das diferentes soluções, desta vez incluindo a aplicação PGM e uma nova funcionalidade que não se encontra presente em nenhuma outra solução.

| | Coswin | ARCHIBUS | Infor EAM | NetFacilities | PGM |
|--|---------------|-----------------|------------------|----------------------|------------|
| Gerir Ordens de trabalho | Sim | Sim | Sim | Sim | Sim |
| Histórico de Manutenções | Sim | Sim | Sim | Sim | Sim |
| Deteção de padrões de avaria | | Sim | | | Sim |
| Classificação da prioridade dos equipamentos | | Sim | | | Sim |
| Funcionamento em ambiente Web | | Sim | | Sim | Sim |
| Módulo para dispositivos móveis | Sim | | | Sim | Sim |
| Contabilização automática de utilizações | | | | | Sim |

Tabela 2 - Comparação das diferentes soluções com PGM

6. Considerações finais

Os projectos para as empresas por vezes sofrem profundas alterações no seu desenvolvimento, sendo quase impossível realizar uma análise de requisitos completa na fase inicial. Esta situação deve-se maioritariamente ao facto de o analista/programador não estar familiarizado com a lógica do negócio da empresa. Mesmo com uma análise exaustiva, durante a implementação, podem surgir obstáculos imprevistos, situações específicas, etc.

Neste projecto, a única alteração que ocorreu no desenvolvimento foi a criação do módulo para dispositivos móveis, que, apesar de no início não ser contemplado, os responsáveis da empresa consideraram de grande utilidade e rentabilidade económica, mesmo requerendo a aquisição de vários dispositivos.

6.1. Principais dificuldades encontradas

Numa empresa com a dimensão da Delphi - Automotive Systems, as alterações de processos dificilmente são de fácil implementação. A empresa conta com várias centenas de funcionários e duas décadas de existência. Qualquer alteração implica que ocorra com várias pessoas e modificação de processos antigos, em que os utilizadores já estão perfeitamente familiarizados. Apesar do esforço da empresa em manter os seus funcionários actualizados para as mais recentes tecnologias de informação através de formações regulares, existem ainda alguns funcionários que não estão totalmente abertos para elas.

Devido a recentes factores externos, o funcionamento interno da empresa sofreu algumas alterações durante o período de realização deste trabalho, o que implicou um esforço adicional de todos os funcionários para se adaptarem à nova realidade. Esse esforço limitou em diversas alturas a disponibilidade dos intervenientes da empresa para o apoio ao desenvolvimento, tanto na fase de análise de requisitos como na de instalação e testes. No entanto, todas essas dificuldades foram ultrapassadas e os objectivos atingidos.

6.2. Desenvolvimentos futuros

6.2.1. Distribuição automática de tarefas aos operadores

Durante o desenvolvimento desta aplicação, várias foram as ideias que surgiram, as quais poderiam melhorar o funcionamento geral, através de novas funcionalidades ou melhoramentos às actuais. Uma das possíveis melhorias apontadas pelos responsáveis da empresa é a distribuição automática de tarefas aos operadores, uma vez que todos eles estão criados no sistema, e este consegue prever as futuras ordens de trabalho.

Esta funcionalidade, tal como outras, necessita de informação adicional, informação essa que não existe actualmente. Para este caso específico seria necessário conhecer os tempos necessários para realizar cada uma das manutenções, pois estes podem variar substancialmente. Ficou acordado com os responsáveis da empresa, que estes iriam iniciar o levantamento destes tempos para que no futuro esta funcionalidade possa ser implementada.

6.2.2. Cálculo da carga de trabalho dos operadores

Tal como referido no ponto anterior, alguma da informação não se encontra ainda disponível actualmente. O cálculo da carga de trabalho dos operadores seria uma informação útil para realizar a distribuição de tarefas, e para medir a eficiência dos operadores. Para tal, é necessário que exista uma tarifa associada a cada intervenção, pois estas variam substancialmente.

Tal como para a funcionalidade descrita no ponto anterior, esta só será possível caso seja feito um levantamento destes tempos. Assim que essa informação exista, esta funcionalidade poderá ser facilmente implementada.

6.2.3. Ligação permanente entre dispositivos móveis e servidor

Os dispositivos móveis necessitam de ser regularmente sincronizados com o servidor, tanto para actualizar a informação por eles recolhida, como para actualizar alguma informação proveniente do servidor.

Apesar do sistema funcionar convenientemente com o processo de sincronização regular através das estações de trabalho, seria uma mais-valia para o sistema se existisse uma ligação permanente, pois eliminaria a necessidade do operador se dirigir à estação de trabalho, e garantia que a informação estivesse permanentemente actualizada, tanto no servidor como no dispositivo móvel.

Para esta situação ser possível, é necessária uma ligação sem fios entre o servidor e os dispositivos móveis. Apesar de actualmente a empresa não ter prevista a implementação deste sistema, caso isso aconteça no futuro, o sistema este perfeitamente preparado para essa situação.

6.2.4. Registo automático dos ciclos de operação de todos os equipamentos

O processo de importação automática dos ciclos de operação dos equipamentos é um dos principais trunfos desta aplicação face à concorrência, pois permite reduzir consideravelmente o tempo dispendido pelos operadores nesta recolha.

Apesar de actualmente ser possível recolher automaticamente a informação de cerca de 80% dos equipamentos, ainda não é possível para a totalidade. Neste ponto, a aplicação está perfeitamente preparada para a recolha de toda a informação, e só não acontece devido à inexistência do factor de conversão entre peças produzidas e ciclos dos equipamentos para alguns destes equipamentos. Os responsáveis da empresa referiram que, devido a algumas situações particulares, nomeadamente o arranque de novos projectos, nunca será possível

abranger a totalidade dos equipamentos, mas esperam no curto/médio prazo, aumentar a abrangência de equipamentos para valores perto dos 95%.

7. Conclusão

Este trabalho de mestrado, tendo como objectivo a implementação de uma aplicação para uma grande empresa, possuiu um duplo objectivo: o escolar e o profissional.

Por um lado, sendo este um mestrado em desenvolvimento de software e sistemas interactivos, deverão ser analisados e investigados os métodos mais eficientes de resolução de problemas, utilizando para isso técnicas e procedimentos inovadores, e, ao mesmo tempo, respeitando as regras e boas práticas da engenharia de software.

Por outro lado, para a empresa, não é tão relevante a forma como a aplicação foi desenvolvida, mas sim o seu funcionamento e facilidade de utilização.

Apesar de no final do desenvolvimento os dois grandes objectivos convergirem para um único, durante o processo ocorreram situações que tornaram a tarefa mais difícil. O facto da arquitectura a utilizar estar condicionada, por diversos factores impostos pela empresa, impossibilitou a utilização de algumas ferramentas e técnicas mais avançadas. Também o facto de não estar familiarizado com a lógica do negócio, constituiu um obstáculo na fase inicial. Ainda assim, com o grande apoio prestado pela empresa e Escola, este obstáculo foi superado com sucesso.

Convém referir que este tipo de empresa possui normas rígidas relativamente à confidencialidade da informação, por esse motivo os materiais utilizados neste trabalho, nomeadamente nas imagens da aplicação, utilizam dados fictícios. Em muito poucas situações foram usados dados reais (ex. Ordem de trabalho em anexo) e nestes casos a sua utilização foi devidamente autorizada pelos responsáveis da empresa.

Como autor de uma aplicação de software, um projecto real em contexto de trabalho, constitui uma motivação extra, pois existe a garantia de que o trabalho realizado terá utilidade no dia-a-dia. E sendo submetido a uma utilização diária por vários utilizadores, permitirá receber comentários, sejam eles críticas, sugestões ou elogios. Esta contínua avaliação do meu trabalho e os comentários que daí advenham, ajudar-me-ão a crescer como profissional.

Este tipo de projecto traz também o benefício de conhecer de perto o ambiente e modo de funcionamento de uma empresa que serve de exemplo para muitas outras. A experiência adquirida constitui uma mais-valia considerável a nível profissional e pessoal.

Outro aspecto que colocou sobre mim uma maior pressão, foi o facto deste projecto consistir numa parceria entre a Escola Superior de Tecnologia e a Empresa Delphi de Castelo Branco, e que o trabalho por mim realizado será sempre um reflexo da própria Escola. Ainda assim, estou convencido que, na realidade, consistiu também numa motivação extra para melhor desempenhar a minha função.

Referências

- [1] - Wikipédia. Definição de WAMP [online] <http://en.wikipedia.org/wiki/WAMP> [cited 2010-10-01].
- [2] - Wikipédia. Definição de Waterfall model [online] http://en.wikipedia.org/wiki/Waterfall_model. [cited 2010-10-01]
- [3] - Sítio Oficial Siveco. Produto Coswin 7i [online] <http://www.siveco.com/pt/produit/coswin-7i> [cited 2010-10-01].
- [4] - Sítio Oficial Archibus. Produto Preventive Maintenance [online] - http://www.archibus.com/index.cfm/pages.content_application/template_id/845/section/Preventive%20Maintenance/path/1.3.29.104/menuid/104 [cited 2010-10-01].
- [5] - Sítio Oficial InforTM. Produto Enterprise Asset Management [online] <http://www.infor.com/solutions/eam/> [cited 2010-10-01].
- [6] - Sítio Oficial InforTM. Características do produto Enterprise Asset Management [online] http://www.infor.com/product_summary/eam/enterpriseedition/ [cited 2010-10-01].
- [7] - Sítio Oficial NetFacilities. Produto Computerized Maintenance Management Software [online] <http://www.netfacilities.com/index.php?program=articles§ion=36> [cited 2010-10-01].
- [8] - Sítio Oficial Microsoft. Produto Windows XP [online] <http://www.microsoft.com/windows/windows-xp/default.aspx> [cited 2010-10-01].
- [9] - Sítio Oficial Apache Software Foundation. Produto Apache HTTP Server Project [online] <http://httpd.apache.org/> [cited 2010-10-01].
- [10] - Sítio Oficial Microsoft Internet Information Services (IIS) for Windows® Server. [online] <http://www.microsoft.com/windowsserver2008/en/us/internet-information-services.aspx> [cited 2010-10-01].
- [11] - Sítio Oficial Oracle. [online] <http://www.oracle.com/index.html> [cited 2010-10-01].
- [12] - Sítio Oficial Microsoft SQL Server [online] <http://www.mysql.com/products/community/> [cited 2010-10-01].
- [13] - Sítio Oficial MySQL. Produto MySQL Community Edition [online] <http://www.microsoft.com/sqlserver/en/us/default.aspx> [cited 2010-10-01].
- [14] - Sítio Oficial Firebird. [online] <http://www.firebirdsql.org/> [cited 2010-10-01].
- [15] - Sítio Oficial PostGreSQL. [online] <http://www.postgresql.org/> [cited 2010-10-01].
- [16] - Sítio Oficial Embarcadero. Produto InterBase XE [online] <http://www.embarcadero.com/products/interbase> [cited 2010-10-01].
- [17] - Sítio Oficial GNU. GNU General Public License [online] <http://www.gnu.org/licenses/gpl.html> [cited 2010-10-01].
- [18] - Serrão, C. e Marques, J. Programação com PHP 5.3 FCA - Editora de Informática, 2010

- [19] - Brinzarea, Bogdan e Darie, Cristian - AJAX and PHP: Building Modern Web Applications, Packt Publishing, 2ª Edição, 2010
- [20] - Cody Lindley, jQuery Cookbook: Solutions & Examples for jQuery Developers, O'Reilly Media, 2009
- [21] - Sítio Oficial Orango Labs. Produto GreyBox Smart Window [online] <http://orango.com/labs/GreyBox/> [cited 2010-10-01].
- [22] - Silva, A. e Videira, C. UML Metodologias e Ferramentas CASE (2ª Edição), Centro Atlântico, 2005
- [23] - Silva, A. e Videira, C. UML Metodologias e Ferramentas CASE (2ª Edição) - Visão Histórica, Centro Atlântico, PP103-106, 2005
- [24] - Silva, A. e Videira, C. UML Metodologias e Ferramentas CASE (2ª Edição) - UML Casos de Utilização, Centro Atlântico, PP131, 2005
- [25] - Silva, A. e Videira, C. UML Metodologias e Ferramentas CASE (2ª Edição) - UML Casos de Utilização, Centro Atlântico, PP143, 2005
- [26] - Silva, A. e Videira, C. UML Metodologias e Ferramentas CASE (2ª Edição) - UML Casos de Utilização, Centro Atlântico, PP134, 2005
- [27] - Silva, A. e Videira, C. UML Metodologias e Ferramentas CASE (2ª Edição) - UML Casos de Utilização, Centro Atlântico, PP140, 2005
- [28] - Silva, A. e Videira, C. UML Metodologias e Ferramentas CASE (2ª Edição) - UML Casos de Utilização, Centro Atlântico, PP138, 2005
- [29] - Wikipédia. Definição de 3-Tier [online] http://en.wikipedia.org/wiki/Multitier_architecture [cited 2010-10-01].
- [30] - Loureiro, H Excel 2007 Macros & VBA - Curso Completo FCA - Editora de Informática, 2007.
- [31] - Oliveira, R e Fernandes, N Apache Instalação, Configuração Gestão Servidores Web FCA - Editora de Informática, PP 5-8, 2007.
- [32] - Coelho, P HTML 4 & XHTML - Curso Completo FCA - Editora de Informática, PP 65-97, 2008.
- [33] - Coelho, P. Javascript - Animação e Programação em Páginas Web FCA - Editora de Informática, 2008
- [34] - Sítio Oficial .NET Framework Developer Center. Produto .NET Compact Framework [online] <http://msdn.microsoft.com/en-us/netframework/aa497273> [cited 2010-10-01].
- [35] - Queirós, R Programação para Dispositivos Móveis em Windows Mobile 6 - Curso Completo FCA - Editora de Informática, PP 7-34, 2008.
- [36] - Sítio Oficial Open Source Initiative. MIT License [online] <http://opensource.org/licenses/mit-license.php> [cited 2010-10-01].
- [37] - Sítio Oficial Moxiecode. Produto TinyMCE [online] <http://tinymce.moxiecode.com/> [cited 2010-10-01].
- [38] - Serrão, C. e Marques, J. Programação com PHP 5.3 FCA - Editora de Informática, PP209-224 2010

[39] - Benchmark comparativo das principais livrarias de abstracção de base de dados [online] <http://phplens.com/lens/adodb/> [cited 2010-10-01].

[40] - Sitio oficial Microsoft MSDN. Produto ADO.NET [online] [http://msdn.microsoft.com/en-us/library/h43ks021\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/h43ks021(v=vs.71).aspx) [cited 2010-10-01].

[41] - Sitio Open Source Initiative. BSD License. [online] <http://www.opensource.org/licenses/bsd-license.php> [cited 2010-10-01].

[42] - Sitio oficial UltraApps. Produto VCalendar (Virtual Web Calendar) [online] http://www.ultraapps.com/app_overview.php?app_id=19 [cited 2010-10-01].

[43] - Damas, L SQL - Structured Query Language - 6ª Edição Actualizada e Aumentada FCA – Editora de Informática, PP 50-57, 2008.

[44] - Alves, L - Relatório do Projecto Final de Curso - Hourly Production Control, ESTCB, 2005

[45] - Coelho, P. Javascript - Animação e Programação em Páginas Web FCA – Editora de Informática, PP 244-249, 2008

[46] - Serrão, C. e Marques, J. Programação com PHP 5.3 FCA - Editora de Informática, PP427-444 2010

Anexos

ANEXO 2 - Exemplo de utilização da classe listagem

```
include_once "./includes/listagem.class.php";

// recebe o nome da tabela pela URL
$list = new listagem($_GET["table"]);

//define nº de resultados a mostrar por página (por defeito)
$list->$resultados_pagina = 20;

// define o nome da coluna a exibir
// nome_na_bd, nome_a_apresentar
$list->define_nome_campo("cod_supervisor","Cod. Supervisor");
$list->define_nome_campo("tipo_equipamento","Tipo Equip.");
//ignora o campo, não o exibe na listagem
$list->ignora_campos("PASSW");

// Mostra o comentario da tabela (descrição)
$list->devolve_comentario();

// recebe da URL a opção: inserir,alterar,eliminar ou listar
switch(@$_GET["opt"]){
    case 'table_ins':{
        $list->mostra_form_inserir();
        break;
    }
    case 'table_alt':{
        $list->mostra_form_alterar();
        break;
    }
    case 'table_eli':{
        $list->mostra_form_eliminar();
        break;
    }
    case 'table_list':{
        $list->mostra_form_list();
        break;
    }
}
```