

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Desenvolvimento de Sistemas de Controlo de Equipamento de Produção de TV

Rui Jorge Canelhas Bastos Neves

VERSÃO DEFINITIVA

Relatório de Projecto/Dissertação
Mestrado Integrado em Engenharia Informática e Computação

Orientador: Professor Artur Pimenta Alves (Prof. Catedrático)

Julho de 2009

Desenvolvimento de Sistemas de Controlo de Equipamento de Produção de TV

Rui Jorge Canelhas Bastos Neves

Relatório de Projecto/Dissertação
Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Nome do presidente do júri (Título)

Arguente: Nome do arguente do júri (Título)

Vogal: Nome do vogal do júri (Título)

31 de Julho de 2009

Resumo

Na era digital, uma das primeiras fases para a produção de conteúdos multimédia, quer para televisão como para outros meios emergentes, como a internet, passa pela captura e gravação do sinal original de vídeo e áudio, armazenando-o num qualquer suporte digital, para que possa em seguida ser editado ou emitido imediatamente, conforme a necessidade. Estas capturas são feitas actualmente por servidores dedicados, que recebem o sinal e o convertem para o formato desejado, colocando depois o(s) ficheiro(s) resultante(s) à disposição do produtor ou editor.

É possível ter múltiplas destas estações de captura, em qualquer momento, cada uma a receber o seu sinal de entrada, sendo o seu operador responsável por capturar o sinal nas alturas pretendidas, operação particularmente crítica para transmissões em directo. A grande limitação deste sistema prende-se com o facto do operador ter que controlar cada estação individualmente, mesmo que o possa fazer remotamente.

Este projecto, descrito no presente documento, pretende apresentar uma solução para o problema, na forma de uma aplicação que permite o controlo centralizado de todas estas estações, automatizando até algumas das tarefas. Esta aplicação irá interagir, através dos seus *web services*, com um dos modelos de estações de captura existentes na MOG, o *mxSPEEDRAIL S1000*.

O sistema desenvolvido permite não só controlar ou monitorizar remotamente estações de captura individuais, como oferecer controlo simultâneo para grupos de unidades. Como forma de automatizar alguns mecanismos, possui também um módulo que possibilita o agendamento de capturas, tanto para as estações individuais como para os grupos descritos.

A solução resultante do projecto contribui assim para a redução e facilitação de diversas tarefas repetitivas que terão, invariavelmente, que fazer parte do processo. Se bem que a premissa original de tornar toda a operação possível de gerir por apenas um operador foi concluída com sucesso, será também relevante notar as possibilidades de automatização extensíveis ao projecto, desde introdução de metadados à geração de *scripts* para gestão automática de capturas, que auguram um futuro com um nível de intervenção humana em constante redução.

Abstract

In the digital age, one of the first stages for the production of multimedia content, for television as well as other emergent media, such as the internet, lies in the capture and recording of the original video and audio signal, storing it on some digital support, so it can immediately be edited or transmitted, depending on the need. These captures are currently made by dedicated servers, who receive the signal and convert it to the desired format, quickly making the resulting file(s) available to the producer or editor.

It is possible to have multiple of said capture stations, at any time, each receiving it's own source signal, while its operator is responsible for capturing said signal at the appropriate times, a particularly critical operation for live transmissions. The major limitation of this system lies in the need to control each station individually, even if it is possible to do it remotely.

This project, described in the current document, will present a solution for this problem, shaped as an application who allows centralized control for all capture stations, enabling some task automation. The application will interact, through web services, with one model of capture station developed at MOG, the mxfsPEEDRAIL S1000.

The developed system allows, not only the remote control and monitoring of individual capture stations, but also offers simultaneous control for groups of those units. As a way of automating some mechanisms, it contains a module to allow scheduling captures, both for individual stations as well as for the described groups of units.

The project's resulting system contributes for reducing and facilitating several repetitive tasks that must, unavoidably, take part in the process. While the original premise of making the entire operation manageable by a single operator was successfully concluded, it also leaves an open path of relevant possibilities for future development. From automatic metadata insertion to the usage of scripts capture management, one can expect future developments could keep reducing the level of human intervention required by said processes.

Agradecimentos

Ao Eng^o Luis Miguel Sampaio, responsável da MOG por este projecto, pela total disponibilidade e apoio e conhecimentos prestados no decurso do mesmo.

Ao Prof. Artur Pimenta Alves, orientador da FEUP, por todos os esclarecimentos e assistência prestada durante o projecto.

A todos os meus colegas na MOG, que sempre me esclareceram todas as dúvidas e auxiliaram com qualquer tipo de dificuldade.

E por último, mas não com menos importância, à minha família e amigos, pelo contínuo incentivo e apoio, que sem dúvida contribuiu decisivamente para o sucesso deste projecto.

Rui Neves

Índice

1	Introdução.....	1
1.1	Enquadramento.....	1
1.2	O Projecto.....	2
1.3	Motivação e Objectivos.....	2
1.4	Estrutura do Documento.....	4
2	Revisão Tecnológica.....	5
2.1	Ambientes de produção multimédia.....	5
2.2	Estações de Captura.....	6
2.3	Soluções Existentes.....	9
2.4	Tecnologias Utilizadas.....	11
2.4.1	Adobe <i>Flex</i>	11
2.4.2	<i>Python</i>	14
2.4.3	SOAP.....	14
2.4.4	<i>Suds</i>	15
2.4.5	<i>TurboGears WebServices</i>	15
2.4.6	<i>Bonjour</i>	16
2.4.7	<i>Flex Scheduling Framework</i>	16
2.4.8	<i>Mate Flex Framework</i>	17
2.4.9	Conclusões.....	18
3	Sistema de Controlo de Equipamento de Produção de TV.....	20
3.1	Objectivos.....	20
3.2	A Solução.....	21
3.3	Resumo e Conclusões.....	23
4	Especificação de Requisitos.....	24
4.1	Requisitos Funcionais e Casos de Uso.....	24
4.1.1	Autenticação.....	27
4.1.2	Alterar Password.....	27
4.1.3	Adicionar um S1000/ <i>Gang</i> ao sistema.....	28
4.1.4	Efectuar o <i>Logout</i>	28
4.1.5	Obter ajuda.....	29
4.1.6	Aceder à agenda.....	29
4.1.7	Navegar na agenda.....	30
4.1.8	Alterar modo de visualização da agenda.....	30
4.1.9	Adicionar um novo evento à agenda.....	31
4.1.10	Remover um evento da agenda.....	31
4.1.11	Editar um evento.....	32
4.1.12	Renomear <i>Gang</i>	32
4.1.13	Remover um ou mais <i>Gangs</i>	33
4.1.14	Juntar <i>Gangs</i>	33
4.1.15	Separar <i>Gangs</i>	34

4.1.16	Ver Estado.....	34
4.1.17	Iniciar capturas dum <i>gang</i>	35
4.1.18	Parar capturas num <i>gang</i>	35
4.1.19	Ver informação sobre um S1000.....	36
4.1.20	Restabelecer ligação a um S1000.....	36
4.1.21	Iniciar uma Captura.....	37
4.1.22	Parar uma Captura.....	37
4.1.23	Renomear S1000.....	38
4.1.24	Mover S1000.....	38
4.1.25	Remover S1000.....	39
4.1.26	Configurar opções de captura.....	39
4.1.27	Configurar estratégia de <i>filenaming</i>	40
4.2	Requisitos Não Funcionais.....	40
4.2.1	Desempenho.....	40
4.2.2	Robustez.....	41
4.2.3	Fiabilidade.....	41
4.2.4	Interoperabilidade.....	41
4.2.5	Usabilidade.....	42
4.2.6	Modularização.....	42
4.2.7	Internacionalização.....	42
4.2.8	Interface.....	43
4.3	Conclusões.....	43
5	Arquitectura da Solução.....	44
5.1	Arquitectura Física.....	44
5.2	Arquitectura Lógica.....	45
5.2.1	Decomposição horizontal.....	46
5.2.2	Decomposição vertical.....	47
5.3	Conclusões.....	51
6	Desenvolvimento do Protótipo.....	52
6.1	Visão Geral da Aplicação.....	52
6.2	Gestão de <i>Gangs</i>	55
6.3	Visualização por <i>Gang</i>	59
6.4	Gestão dos dispositivos S1000.....	60
6.5	Gestão da Agenda.....	64
6.6	Testes e Simulações.....	70
6.7	Conclusões.....	73
7	Conclusões.....	74
7.1	Sumário.....	74
7.2	Resultados.....	74
7.3	Trabalho Futuro.....	75
	Referências.....	77

Lista de Figuras

Figura 1.1: Ciclos de desenvolvimento com a utilização de Programação Extrema.....	3
Figura 2.1: Etapas de um processo de produção de televisão.....	5
Figura 2.2: Interação entre <i>Flex</i> , Flash e sistemas externos.....	12
Figura 2.3: O modelo de arquitectura MVC.....	13
Figura 2.4: O modelo MVC no contexto de uma aplicação web.....	13
Figura 2.5: O modelo MVC num contexto de uma aplicação <i>Flex</i>	13
Figura 2.6: Arquitectura das mensagens SOAP.....	15
Figura 2.7: Exemplo das potencialidades da <i>Flex Scheduling Framework</i>	16
Figura 2.8: Arquitectura de uma aplicação utilizando a <i>framework Mate</i>	17
Figura 2.9: Tecnologias utilizadas por camada.....	18
Figura 3.1: Exemplo de um cenário de utilização de estações de captura.....	20
Figura 3.2: Exemplo de um cenário de utilização de estações de captura utilizando o sistema proposto.....	22
Figura 4.1: Casos de uso do sistema.....	26
Figura 5.1: Arquitectura Física do Sistema.....	45
Figura 5.2: Arquitectura Lógica do Sistema (decomposição horizontal).....	46
Figura 5.3: Arquitectura Lógica do Sistema (decomposição vertical).....	47
Figura 5.4: Decomposição vertical da arquitectura: sub-sistema de gestão de agenda.....	48
Figura 5.5: Decomposição vertical da arquitectura: sub-sistema de gestão de gangs.....	49
Figura 5.6: Decomposição vertical da arquitectura: sub-sistema de gestão interna.....	50
Figura 6.1: Ecrã de autenticação do protótipo.....	52
Figura 6.2: Barra de acesso às funcionalidades gerais da aplicação.....	53
Figura 6.3: Menu para alteração de passwords para acesso ao sistema.....	53
Figura 6.4: Menu para alteração de passwords para acesso ao sistema.....	54
Figura 6.5: Ecrã principal, evidenciando as janelas representativas de <i>gangs</i> no sistema.....	54
Figura 6.6: Selecção de grupos de <i>gangs</i> , obtida arrastando o rato no ecrã.....	55
Figura 6.7: Exemplo de menu contextual de relativo a um, ou vários, <i>gangs</i>	56
Figura 6.8: Menu de adição de novos dispositivos <i>mxSPEEDRAIL S1000</i>	56
Figura 6.9: Exemplo do ficheiro XML de configuração de <i>gangs</i> da aplicação.....	57
Figura 6.10: Lista de possíveis estados, e respectiva visualização, de um <i>gang</i>	58
Figura 6.11: Exemplo da visualização de estados dos servidores S1000.....	59
Figura 6.12: Exemplo da interface de visualização de um único <i>gang</i> , composto por dois servidores S1000.....	59
Figura 6.13: Janela para controlo remoto de um servidor S1000.....	60
Figura 6.14: Opções de configuração da estratégia de <i>filenaming</i> de um S1000.....	62
Figura 6.15: Opções de configuração de capturas para um S1000.....	63
Figura 6.16: Opções de configuração de capturas para um S1000.....	63
Figura 6.17: Gestão de perfis de metadados.....	64
Figura 6.18: Opções de visualização da agenda.....	64
Figura 6.19: Vista mensal (parcial) da agenda.....	65
Figura 6.20: Vista diária (parcial) da agenda.....	66

Figura 6.21: Vista semanal (parcial) da agenda.....	66
Figura 6.22: Menu de navegação na agenda.....	67
Figura 6.23: Menu de controlo de eventos.....	67
Figura 6.24: Janela para criação de um novo evento na agenda.....	68
Figura 6.25: Janela para edição de um evento existente na agenda.....	68
Figura 6.26: Exemplo de uma situação de <i>drag'n'drop</i> na agenda.....	69
Figura 6.27: Formato do ficheiro XML onde é guardada a configuração da agenda.....	69
Figura 6.28: Estado da aplicação com uma captura parcial em curso.....	71
Figura 6.29: GUI com capturas a decorrer em doze máquinas simultaneamente.....	72

Lista de Tabelas

Tabela 2.1: Visão global das funcionalidades das estações de captura individuais existentes...	10
Tabela 4.1: Especificação do caso de uso “Autenticação”	27
Tabela 4.2: Especificação do caso de uso “Alterar Password”	27
Tabela 4.3: Especificação do caso de uso “Adicionar <i>Gang/S1000</i> ”	28
Tabela 4.4: Especificação do caso de uso “Fazer <i>Logout</i> ”	28
Tabela 4.5: Especificação do caso de uso “Obter Ajuda”	29
Tabela 4.6: Especificação do caso de uso “Aceder à agenda”	29
Tabela 4.7: Especificação do caso de uso “Navegar na agenda”	30
Tabela 4.8: Especificação do caso de uso “Alterar modo de visualização da agenda”	30
Tabela 4.9: Especificação do caso de uso “Adicionar evento”	31
Tabela 4.10: Especificação do caso de uso “Remover evento”	31
Tabela 4.11: Especificação do caso de uso “Editar evento”	32
Tabela 4.12: Especificação do caso de uso “Renomear <i>Gang</i> ”	32
Tabela 4.13: Especificação do caso de uso “Remover <i>Gang(s)</i> ”	33
Tabela 4.14: Especificação do caso de uso “Juntar <i>Gangs</i> ”	33
Tabela 4.15: Especificação do caso de uso “Separar <i>Gangs</i> ”	34
Tabela 4.16: Especificação do caso de uso “Ver Estado”	34
Tabela 4.17: Especificação do caso de uso “Iniciar capturas dum <i>gang</i> ”	35
Tabela 4.18: Especificação do caso de uso “Parar capturas num <i>gang</i> ”	35
Tabela 4.19: Especificação do caso de uso “Ver informação sobre um <i>S1000</i> ”	36
Tabela 4.20: Especificação do caso de uso “Restabelecer ligação a um <i>S1000</i> ”	36
Tabela 4.21: Especificação do caso de uso “Iniciar uma captura”	37
Tabela 4.22: Especificação do caso de uso “Parar uma captura”	37
Tabela 4.23: Especificação do caso de uso “Renomear <i>S1000</i> ”	38
Tabela 4.24: Especificação do caso de uso “Mover <i>S1000</i> ”	38
Tabela 4.25: Especificação do caso de uso “Remover <i>S1000</i> ”	39
Tabela 4.26: Especificação do caso de uso “Configurar opções de captura”	39
Tabela 4.27: Especificação do caso de uso “Configurar opções de <i>filenaming</i> ”	40
Tabela 6.1: Tempo de resposta da ferramenta nas operações mais relevantes	71
Tabela 6.2: Sincronização entre máquinas e operações de um <i>gang</i> (3 servidores)	72

Abreviaturas e Símbolos

MXF	Material eXchange Format
SDI	Serial Digital Interface
SMPTE	Society of Motion Pictures and Television Engineers
RIA	Rich Internet Application
MVC	Model View Controller
AIR	Adobe Integrated Runtime
SOAP	Simple Object Access Protocol
HTTP	HyperText Transfer Protocol
XML	Extended Markup Language
SMTP	Simple Mail Transfer Protocol
WSDL	Web Service Definition Language
API	Application Programming Interface
GUI	Graphic User Interface

1 Introdução

1.1 Enquadramento

No mundo actual, estações de televisão, produtoras e casas de edição, recebem os seus conteúdos numa imensa variedade de formatos, desde cassetes a ficheiros ou "*live-feeds*". Recebem também, dos estúdios, diversos canais de vídeo, provenientes de múltiplas câmaras, recorrentemente em simultâneo.

Independentemente do canal de entrada, torna-se necessária a sua captura, catalogação e disponibilização a operadores e editores, prosseguindo assim o processo de produção, culminando na disponibilização desse conteúdo, entretanto editado, ao espectador, através dos múltiplos canais de saída actualmente disponíveis, como o sejam a televisão, cinema ou até a internet.

O projecto descrito neste documento assenta precisamente na fase inicial do processo de produção, conhecida como a fase de *Ingest*. Devido à referida multiplicidade de canais de entrada, muitas das vezes a chegarem em simultâneo, pode tornar-se complicado gerir a captura de todos esses sinais. Esta dificuldade agrava-se, no caso, muito frequente, de transmissões em directo, no qual será ainda mais relevante a necessidade de gerir e controlar estas capturas em tempo real

O objectivo é, portanto, o de facilitar este tipo de operações, estudando as soluções existentes e desenvolvendo um protótipo que permita, de um modo o mais simples possível, a monitorização e controlo centralizado, através de uma interface web intuitiva, de múltiplos canais de entrada simultaneamente, por forma a permitir que um único operador consiga coordenar a fase de *Ingest* de todos esses canais, automatizando, dentro do possível, todo o processo.

1.2 O Projecto

A MOG (*Media Objects and Gadgets*), empresa onde foi realizado este projecto, possui já soluções especializadas na fase de *Ingest*. O produto em questão, denominado *mxfsPEEDRAIL S1000*, actua como um servidor dedicado, permitindo capturar um sinal de entrada *SDI/HDSDI* e transcodificá-lo para um de entre uma multiplicidade de formatos suportados, para que possa ser posteriormente trabalhado.

Este projecto consiste, resumidamente, no desenvolvimento e prototipagem de software, denominado *mxfsPEEDRAIL C1000*, que monitorize e controle, através de *web services*, múltiplos destes dispositivos, permitindo assim que, um único utilizador, consiga coordenar e controlar os diversos canais de entrada em simultâneo.

Como resultado final, pretende obter-se um protótipo demonstrador do sistema de controlo de equipamento de produção de televisão, que interaja com múltiplos servidores do tipo *mxfsPEEDRAIL S1000*, desenvolvidos na MOG, e que permita, de forma intuitiva e eficaz, a captura simultânea de múltiplos sinais provenientes de um qualquer estúdio onde decorra uma gravação multi-câmara, bem como a calendarização dessas mesmas capturas, que chegam a uma estação de TV vindas de outras estações ou distribuidores.

O protótipo a desenvolver, requer não só uma interface *web*, de fácil utilização para o utilizador, que permita o controlo dos diversos servidores de captura, ou mesmo de grupos de servidores em simultâneo, como se de uma única máquina se tratasse, mas também a implementação de todo o *back-end* necessário a estas operações, assentando a aplicação numa estrutura cliente-servidor, que comunique através de *web services* com qualquer conjunto de estações de *Ingest* previamente configuradas.

1.3 Motivação e Objectivos

Existem já múltiplas soluções, das quais faz parte o *mxfsPEEDRAIL S1000*, mencionado anteriormente, que permitem capturar, anotar e disponibilizar conteúdos provenientes de um qualquer *input*. No entanto, para suportar diversos canais em simultâneo, é recorrentemente necessária a troca do sinal de entrada ou a utilização de múltiplos operadores nas estações, o que não será certamente a forma mais eficaz de gerir a situação, levando frequentemente a uma perda de produtividade desnecessária.

A motivação e foco principal deste projecto, é o de aumentar a eficiência e produtividade neste tipo de operação, diminuindo ou eliminando o tempo despendido nas tarefas, nas situações em que existam múltiplos dos referidos canais de entrada, libertando assim o operador de uma grande percentagem de tarefas repetitivas associadas ao processo.

Estes objectivos são exequíveis através da centralização do controlo relativo a todos os canais, num único local, controlável por um único operador. Deste modo, é possível ter múltiplos servidores de *Ingest mxfsPEEDRAIL S1000* em funcionamento, um para cada canal de entrada pretendido, em qualquer local desejado, tendo constante acesso remoto a qualquer das suas funcionalidades, a partir de uma interface *web* central.

Introdução

É também objectivo a implementação da funcionalidade de calendarização de capturas, automatizando assim grande parte deste processo, e dando um grande passo na direcção final, que será a de obter uma intervenção humana mínima na operação de *Ingest*, libertando editores e operadores de tarefas de mais baixo nível, para que o seu foco esteja na totalidade virado para o processo criativo de todo o processo.

Além disto, é também necessário assegurar a compatibilidade total com os servidores *mxfsPEEDRAIL S1000* que se pretendem controlar, bem como garantir a fiabilidade e *performance* necessárias neste género específico de aplicação.

Múltiplos factores entraram na consideração da metodologia de desenvolvimento a utilizar mas, devido em grande parte, à componente de inovação associada ao projecto, que garantidamente levaria os requisitos a mudarem frequentemente, optou-se pela utilização de metodologias ágeis para o seu desenvolvimento.

Tendo também em conta a ainda considerável dimensão do projecto e o facto de que seria desenvolvido maioritariamente a nível individual, o modelo escolhido foi o de Programação Extrema, convenientemente ajustado às particularidades inerentes ao projecto.

Esta metodologia permite reduzir significativamente o custo de alterações nos requisitos do sistema a desenvolver, situação que se previa frequente no decorrer do projecto. Para atingir este objectivo, utilizam-se ciclos rápidos de desenvolvimento, com *feedback* frequente, começando por implementar as funcionalidades mais importantes e, a cada iteração, acrescentando novas ou alterando as existentes, permitindo assim uma constante adaptação a requisitos variáveis. Este estilo de desenvolvimento funciona particularmente bem em ambientes com equipas pequenas e projectos com considerável componente de inovação, como no caso presente.

Pode ser visto na Figura 1.1 um diagrama ilustrativo destes ciclos de desenvolvimento.

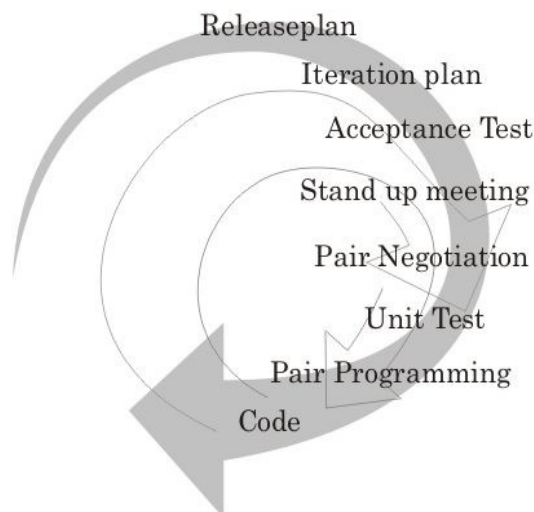


Figura 1.1: Ciclos de desenvolvimento com a utilização de Programação Extrema.

1.4 Estrutura do Documento

Quanto à sua estrutura, este documento encontra-se dividido em sete capítulos, sendo que no primeiro teremos uma breve introdução sobre o tema e objectivos do projecto.

Após uma breve introdução e análise dos objectivos no primeiro capítulo, segue-se, no capítulo dois, o estado da arte e conseqüente comparação de soluções, onde são apresentados alguns projectos relacionados. No capítulo três, aprofunda-se a descrição do projecto e da solução encontrada. No quarto capítulo, serão enunciados os requisitos do mesmo, incluindo os casos de uso a implementar. Posteriormente, no capítulo cinco, encontra-se a arquitectura da aplicação, tanto lógica como física, sendo que o capítulo seis está reservado para a descrição mais pormenorizada da implementação e desenvolvimento do protótipo resultante, bem como alguns testes e simulações efectuadas. Por último, no sétimo capítulo, serão apresentados os resultados e conclusões retiradas, bem como algum possível trabalho futuro.

2 Revisão Tecnológica

Este capítulo abordará o estudo realizado acerca do estado da arte na área proposta, incluindo uma análise sobre produtos semelhantes e tecnologias e ferramentas utilizadas no âmbito do projecto.

2.1 Ambientes de produção multimédia

Um ambiente profissional de produção de qualquer conteúdo multimédia, no caso particular, de produção de televisão, possui um *workflow* associado dividido em diversas fases, como se pode ver na representação do processo, esquematizado na Figura 2.1.



Figura 2.1: Etapas de um processo de produção de televisão

Numa primeira fase, os conteúdos são adquiridos, seja por via de uma câmara, ou qualquer outro dispositivo semelhante que permita obter sinal vídeo (e áudio).

Seguidamente entra-se na fase de *ingest*, a que mais respeita a este projecto, onde os conteúdos vindos de uma dessas fontes são então capturados para um qualquer sistema de armazenamento. Nesta fase é também usual transcodificar o sinal, convertendo-o para outros formatos e adicionar-lhe alguns metadados que permitam facilitar a sua identificação em fases posteriores.

Nas duas últimas fases, esse conteúdo armazenado é editado, produzindo-se assim o resultado final, e finalmente emitido. No caso de transmissões de televisão em directo, todas estas fases acontecem em tempo real (ou muito próximo), acompanhando sempre o evento que se pretende transmitir.

Para que melhor se perceba como o processo de *ingest* funciona no mundo real, será então exposto um exemplo de um cenário comum de utilização, no qual se espera obter vantagens através da utilização do sistema descrito neste documento.

Suponha-se uma transmissão televisiva em directo de um qualquer noticiário. Existem diversas câmaras a filmar o mesmo cenário, embora de ângulos diferentes. A qualquer momento, o conteúdo emitido será o adquirido por uma destas câmaras. Cada uma delas estará então ligada a uma estação de *ingest*, que capturará o conteúdo da mesma, convertendo e armazenando o mesmo de forma a poder ser editado (obviamente uma edição rápida, como sobreposição de logótipos ou rodapés) e transmitido de imediato.

Obviamente, nem todas as estações de *ingest* se encontrarão a capturar o vídeo em simultâneo, pois apenas uma das filmagens será transmitida a cada momento. Esse controlo é efectuado por um operador humano, que indicará ao longo da transmissão qual a máquina que deverá capturar o vídeo em cada momento.

Este tipo de controlo, embora aparentemente simples, pode tornar-se algo complexo e pouco intuitivo, uma vez que é pedido ao operador que aceda e controle cada estação individualmente, mantendo sempre uma ideia do estado geral da situação. Seria então de grande utilidade para este, manter o estado de cada máquina, bem como o respectivo controlo das mesmas numa aplicação central, objectivo a que este projecto se propõe. Como passo seguinte, seria também interessante poder automatizar, dentro do possível, estas operações, programando-as com antecedência, ou de forma a seguirem, tal como o próprio conteúdo que se encontram a capturar, algum tipo de guião.

2.2 Estações de Captura

Actualmente, as estações de *ingest*, ou captura, oferecem um conjunto de funcionalidades, com algumas já consideradas essenciais, que permitem grande controlo sobre o processo efectuado no mesma, desde a configurações de formatos de entrada e saída, até a estratégias de nomenclatura dos ficheiros gerados pelas próprias.

A primeira característica de uma estação deste tipo passará por permitir um ou mais tipos de formatos de entrada. Neste campo, pode aceitar sinais de entrada baseados já em

ficheiros digitais, servindo neste caso mais como conversor, transcodificando o ficheiro para o formato desejado como *output*. A outra possibilidade, é este *input* vir através de SDI (*Serial Digital Interface*). O SDI [Poy03] é um standard para transmissão de vídeo digital através de cabo coaxial, utilizado na grande maioria dos equipamentos profissionais, permitindo assim obter o sinal directamente de câmaras de filmar e similares. Este sinal carrega então o vídeo, mas também o sinal de áudio e o *timecode* da transmissão, que é, na sua essência, uma sequência de códigos numéricos gerados a intervalos regulares por um sistema de temporização, e utilizado primariamente para registo e sincronização do conteúdo capturado. O standard da SMPTE para o *timecode* especifica um formato de *hora:minuto:segundo:frame*, que permite assim não só sincronizar como facilmente identificar com base no mesmo qualquer bloco de conteúdo capturado.

É então praticamente um requisito, para uma estação de captura, suportar a entrada, no mínimo, de um sinal SDI. Opcionalmente poderão também ter outras entradas, para outro tipo de sinais. Se bem que as estações poderiam suportar simultaneamente mais que um input, devido ao desempenho exigido do sistema para esses casos, particularmente em formatos de alta definição, cada vez mais *standard* na indústria, esta será uma funcionalidade disponibilizada apenas em casos pontuais.

Uma outra funcionalidade, quase obrigatória, de qualquer estação de captura actual será a monitorização em tempo real do sinal de entrada. Como qualquer destes dispositivos estará a receber um, ou mais, sinais de vídeo na sua entrada. A visualização do sinal possibilita ao operador iniciar ou parar a captura do mesmo de acordo com o que observa, algo muito útil para capturas que não se baseiem em marcações entre dois *timecodes*. Isto possibilita também a fácil verificação de alguma falha no sinal, sem a necessidade de movimentação contínua entre a estação de captura e a fonte real do vídeo. Como será facilmente perceptível, a funcionalidade será útil apenas caso o input seja verdadeiramente acompanhado em tempo real, pois qualquer dessincronização poderá rapidamente influenciar negativamente todo o processo.

Para uma adaptação a qualquer ambiente, torna-se bastante útil a disponibilização de múltiplas resoluções ou formatos de *output*. Actualmente, muitas das soluções existentes disponibilizam apenas algumas escassas opções, em termos de formato, para a geração dos ficheiros de *output*. Para contornar esta limitação, existem depois alguns produtos derivados, dedicados exclusivamente à transcodificação entre os múltiplos formatos existentes.

O *output* é gerado usualmente em alta definição, por forma a obter a melhor qualidade de imagem possível. No entanto, existem situações em que será útil gerar um ficheiro de resolução mais baixa, possivelmente para disponibilização noutros meios, como a internet ou telefones móveis, que cada vez mais suportam esse tipo de conteúdos. Será por este facto que algumas das estações de captura recentes disponibilizam a possibilidade de gerar um *proxy*, o dito ficheiro de baixa resolução, em simultâneo com o *output* regular. Estes dois ficheiros poderão, em alguns casos, conter uma ligação entre ambos, de modo a que qualquer edição feita a um deles se reflecta nos dois. Isto torna-se útil visto que o conteúdo será igual em ambos, possibilitando assim uma edição e tratamento mais célere do *output*. Uma vez que o ficheiro *proxy* é gerado num intervalo de tempo significativamente mais reduzido em relação ao seu par de alta resolução, com a óbvia excepção de sinais cuja recepção esteja a ser efectuada em tempo real ou directo, pode também ser útil para poder iniciar o trabalho no primeiro, enquanto o seu par está ainda em geração.

Como modo de acelerar o processo de produção do vídeo, uma funcionalidade de algumas estações de captura é a de permitir edição em *frame chase*, que se prende com a possibilidade de editar o vídeo enquanto a captura ainda decorre. A geração do ficheiro pode assim continuar enquanto o editor inicia imediatamente o trabalho nos blocos já capturados.

Este tipo de funcionalidade é particularmente útil para capturas de considerável duração, ou mesmo para efectuar algum tratamento rápido em transmissões em tempo muito próximo do real.

Uma das grandes dificuldades relacionadas com o processo de produção de conteúdos multimédia, é a procura e identificação de um determinado ficheiro, e respectivo conteúdo. Esta procura pode estender-se a inúmeros ficheiros, possivelmente até em diferentes locais de armazenamento. Para facilitar esta operação, é característica de algumas estações de captura possibilitarem a definição de uma estratégia de nomenclatura de ficheiros. A estratégia pode ser tão básica como números sequenciais identificadores da captura, ou algo mais complexo e distintivo como a utilização de rótulos específicos, a data e hora de captura ou qualquer tipo de combinação entre as possibilidades disponibilizadas.

Outra funcionalidade avançada para este tipo de dispositivos é a de inserção de metadados, automática ou manual. Estes metadados são, resumidamente, informação sobre o tipo de conteúdo capturado. Qualquer tipo de informação pode ser adicionada como metadados, funcionando numa base de *chave:valor*, podendo então registar-se informação simples como o nome do vídeo ou duração do mesmo, ou enriquecendo a mesma com nomes de actores, realizador, género, ou qualquer outra de inúmeras possibilidades. Estes metadados providenciarão, após a captura, a procura e identificação dos conteúdos gravados, algo cada vez mais crítico, em sistemas que podem armazenar e disponibilizar até milhares de ficheiros provenientes de capturas, tornando essa tarefa cada vez mais complexa.

Este tipo de inserção de metadados só será possível caso o formato de output do ficheiro contenha o respectivo suporte em questão, como o recente MXF [WMD06].

A utilização do MXF como formato de output é também característica das estações de captura mais actuais. É um formato aberto, direccionado à troca de conteúdo audiovisual, com metadados associados. Foi criado com o objectivo de melhorar a interoperabilidade entre servidores e outros dispositivos de criação de conteúdo multimédia. Funciona como um contentor para o conteúdo audiovisual, bem como os seus metadados, sendo por isso independente de formatos de compressão, não estando assim preso a equipamento ou ferramentas específicas. Além de oferecer uma melhor interoperabilidade, funcionando em diferentes aplicações ou equipamento, o suporte aos metadados permite a criação de novas poderosas ferramentas de gestão deste conteúdo, bem como melhorar os *workflows* actuais de criação de conteúdo, ao eliminar a re-inserção repetitiva dos metadados.

As tarefas efectuadas numa estação de captura podem tornar-se monótonas ou repetitivas para um operador humano, facto que contribui para que tentem automatizar algumas destas operações. Esta automatização pode ter variadas vertentes, como a programação antecipada de capturas. Algumas soluções existentes disponibilizam então a possibilidade de agendamento de capturas, que serão depois efectuadas automaticamente na altura correcta. Este tipo de funcionalidade pode ser tão básico ou avançado como se pretenda, desde uma simples indicação à máquina para iniciar e parar uma captura, até à especificação de configurações detalhadas para cada uma delas, ou inserção automática de metadados.

Uma outra possibilidade de automatização prende-se com a captura imediata de qualquer *input* detectado. Esta funcionalidade pode existir na forma de detecção de ficheiros adicionados a algum directório associado ao servidor de captura, ou aquando da introdução de algum dispositivo externo de armazenamento. Tal como em algumas das funcionalidades descritas anteriormente, esta também só fará sentido para estações direccionadas para a utilização de ficheiros como *input*.

Como se pôde verificar, a maior limitação encontra-se ainda na falta de aplicações de permitam centralizar o controlo de múltiplas estações de captura, cenário que ocorre frequentemente em produtoras e estações televisão. Seria então extremamente útil a existência de ferramentas que permitissem o controlo e monitorização de todas estes dispositivos, possibilitando não só o acesso a todas as suas funcionalidades, remotamente, e permitindo a um único operador a gestão de todo o sistema.

Iremos agora analisar algumas das soluções já existentes, encontradas actualmente no mercado, de modo a obter um melhor entendimento das suas reais limitações e das áreas onde a ferramenta desenvolvida no decurso do projecto poderá preencher algumas lacunas.

2.3 Soluções Existentes

No seguimento do capítulo anterior, agora com uma melhor compreensão das funcionalidades e objectivos de uma estação de captura, será oportuno verificar o estado do mercado actual, na forma de uma comparação entre os produtos disponibilizados, as suas vantagens e limitações.

O *MediaHUB* [MED09], da *AdTec Digital*, é uma estação de *ingest* com suporte vários formatos de entrada (SDI, *S-Video*, *Firewire*, entre outros), capturando em tempo real e permitindo visualizar esse conteúdo no decurso do processo. Possui algum suporte básico para metadados, inserindo automaticamente a duração do conteúdo, tempo de captura ou operador respectivo.

Outra solução existente é o *adCode* [ADC09], da *Strong Digital*, que suporta, tal como a anterior, vários formatos de entrada, permitindo a visualização em tempo real do conteúdo capturado, suportando diversas resoluções de *output*, embora apenas para formatos MPEG 2. Quanto à funcionalidade de agendamento de captura, está limitada a permitir definir a duração da próxima captura, sendo que para operações mais avançadas, é comercializado um produto separado. Não tem nenhum suporte para metadados, até porque não utiliza o formato MXF para os seus *outputs*.

O *Vivesta Content Ingest Application Suite* [VIV09], da *Vivesta*, providencia suporte a diversos formatos de *input*, tal como as outras ferramentas descritas. Quanto à transcodificação para diferentes formatos de *output*, essa poderá ser feita num outro produto independente, e não na própria estação. Possui também a funcionalidade de agenda e programação de capturas e, devido a gerar o ficheiro de *output* em formatos MXF, suporte de metadados.

A *PX Ingest Station*, da *Focus Enhancements* [PXI09], é uma das poucas que permite, para os *inputs* suportados (SDI, *Firewire* ou *Composite*) capturar dois em simultâneo, ou até três no caso do *Firewire*. Um dos dois modelos disponíveis, apresenta a funcionalidade de agenda, possibilitando a marcação de capturas, com metadados associados. O formatos de saída limitam-se também a MPEG 2, tendo a transcodificação para outros, que ser efectuada num produto dedicado, estando também limitado a essa aplicação o suporte ao formato MXF.

À semelhança da estação anterior, a *GVS9000 VTR* [GVS09], da *GVS*, permite também a captura de dois *inputs* em simultâneo, embora o suporte esteja limitado a SDI. Quanto à saída, pode ser efectuada para uma considerável panóplia de formatos, sendo esta uma das vantagens da ferramenta. Possui também suporte considerável para agenda de capturas, bem como suporte para MXF, embora as funcionalidades de metadados sejam algo limitadas.

Será oportuno terminar esta lista com a estação de captura que será utilizada como base para a ferramenta a desenvolver, o *mxfsPEEDRAIL S1000* [MOG09], da MOG. Esta estação possui suporte para um canal SDI como *input*, fazendo a sua monitorização em tempo real. Permite também transcodificar o vídeo para uma considerável quantidade de formatos (MPEG2, DVCPRO, IMX, DNxHD, entre outros). Gera também um *proxy* de baixa resolução simultaneamente, opcionalmente, com o formato MPEG2. Os ficheiros são gerados à saída utilizando o formato MXF, que proporciona o já mencionado suporte a metadados, embora ainda algo limitado. Permite definir estratégias de nomenclatura de ficheiros e editar (*frame chase*) os ficheiros durante o decorrer da captura.

A tabela 2.1 permite uma visão global das funcionalidades oferecidas pelas soluções apresentadas. Pode ver-se como ainda há espaço para melhoramentos e inovação em qualquer uma delas, validando assim os objectivos do projecto.

Tabela 2.1: Visão global das funcionalidades das estações de captura individuais existentes

	mxfsPEEDRAIL S1000	MediaHUB	adCode	Vivesta	PX Ingest Station	GVS9000
Formatos Input	1 SDI	1 (Vários)	1 (Vários)	1 (Vários)	1 (Vários)	2 SDI
Formatos Output	Múltiplos	Limitados	Limitados	Limitados	Limitados	Múltiplos
Preview em tempo real	Sim	Sim	Sim	Sim	Sim	Sim
Agenda	Não	Não	Não	Sim	Sim	Sim
MXF	Sim	Sim	Não	Sim	Não	Sim
Metadados	Sim	Sim	Não	Sim	Sim	Sim
Geração de proxy	Sim	Não	Não	Não	Não	Não
Capturas simultâneas	Não	Não	Não	Não	Não	Sim

O cerne principal do projecto, que passa então pela especificação, desenho e desenvolvimento de uma ferramenta que permita automatizar capturas e respectivas configurações associadas, bem como gerir as múltiplas estações controladas, unificando assim o seu controlo numa interface intuitiva e passível de utilização por um único operador. O protótipo resultante fará então uso do modelo *mxfsPEEDRAIL S1000*, como estação a controlar, providenciando assim que se faça uso das mesmas como se de uma única máquina se tratasse. Como vantagem acrescida apresenta o facto de, devido à sua natureza distribuída, o sistema nunca apresentar problemas ou deficiências ao nível do desempenho, independentemente do número de capturas simultâneas que se pretendam realizar.

Pretende-se que a aplicação resultante do projecto possibilite todas as operações que um *mxfsPEEDRAIL S1000* individual, tomando também partido do facto de controlar vários para permitir a associar as máquinas a grupos, denominados também como *gangs*, que possibilitam essas mesmas operações simultaneamente para todos os seus membros. Será também

disponibilizada a funcionalidade de agenda de capturas, estendendo o conceito visto em algumas das aplicações existentes, para permitir num único interface agendar as capturas de todas as máquinas, ou grupos de máquinas, de forma eficiente e simples. Como último objectivo, pretende também adicionar-se algumas funcionalidades, como a criação de perfis de metadados, estratégias de nomenclatura ou outras configurações, permitindo assim facilmente alternar entre configurações ou associar perfis e eventos agendados específicos.

Concluindo, pode verificar-se então que é esperado que o projecto apresente algumas soluções inovadoras e colmate lacunas dos produtos existentes, ajudando assim na evolução e automatização deste género de ferramentas, e libertando os seus operadores de algumas tarefas até agora laboriosas.

2.4 Tecnologias Utilizadas

Nesta secção iremos analisar as tecnologias mais relevantes estudadas no decurso do projecto, providenciando uma resumida descrição das mesmas, bem como dos objectivos com que se pretendia a sua utilização.

2.4.1 Adobe Flex

O *Adobe Flex* [Ado08] é uma *framework* que permite o desenvolvimento de *Rich Internet Applications* (RIA). O facto destas aplicações correrem sobre o *Flash Player* é uma das suas vantagens, visto que está presente na grande maioria dos computadores por todo o globo.

A *framework* disponibiliza componentes visuais interactivos extensíveis e reutilizáveis, serviços de procura de dados e funcionalidades de gestão de eventos. Tal como o *Flash*, o *Flex* é também baseado no *ActionScript*, uma poderosa linguagem orientada a objectos, mas enquanto que o primeiro está mais orientado para designers, o *Flex* apresenta-se mais intuitivo para programadores.

Isto permite à ferramenta ser centrada no código, mas colher também os benefícios das aplicações *Flash*, como o sejam a habilidade de construir interfaces gráficas ricas sem preocupações com as limitações do *browser* utilizado ou a fácil integração de *streaming* de vídeo e áudio.

Podemos ver na Figura 2.2 um esquema da interacção entre o *Flex*, *Flash* e sistemas e serviços externos.

Outra vantagem interessante desta *framework* é a possibilidade das suas aplicações poderem ser corridas como uma qualquer aplicação de *desktop*, utilizando o *Adobe AIR* (*Adobe Integrated Runtime*). O AIR é um *runtime engine* que permite correr *Rich Internet Applications* directamente no *desktop* como se duma normal aplicação se tratasse. Estas aplicações são, portanto, independentes da plataforma onde correm e torna-se também desnecessário o recurso a um *browser*.

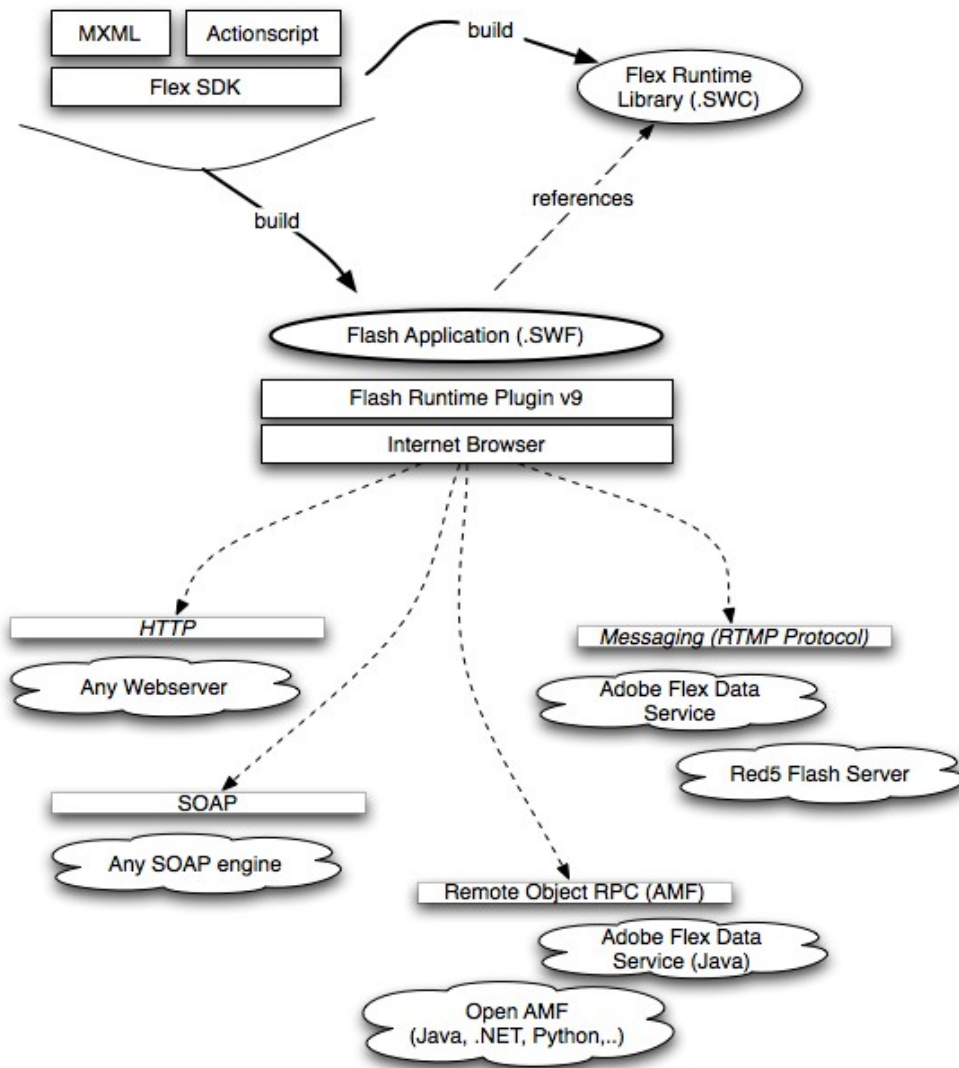


Figura 2.2: Interação entre Flex, Flash e sistemas externos

A arquitectura duma aplicação desenvolvida com o *Adobe Flex* [FAF09] pode ser vista como uma adaptação do modelo de arquitectura MVC (*Model View Controller*) onde o *controller* tomará a forma dum serviço ou objecto no servidor, que lerá e actualizará o modelo (possivelmente uma base de dados) ao receber pedidos da aplicação, que mostra essas actualizações ao utilizador. Para melhor se compreender o paralelo estabelecido, vejam-se as Figuras 2.3, 2.4 e 2.5, respectivamente referentes ao modelo MVC no geral, no contexto de uma aplicação *web* e, finalmente, no caso de uma RIA desenvolvida em *Flex*.

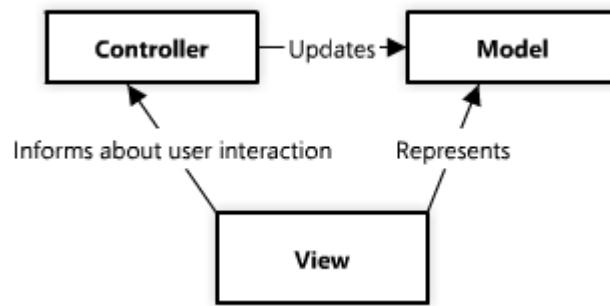


Figura 2.3: O modelo de arquitectura MVC

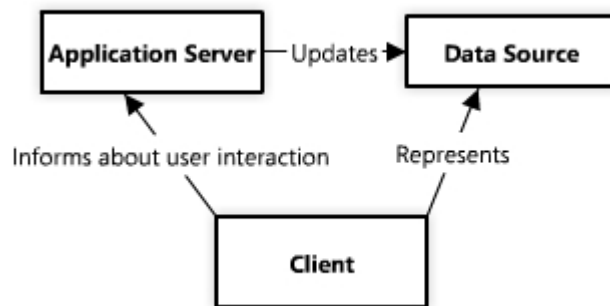


Figura 2.4: O modelo MVC no contexto de uma aplicação *web*

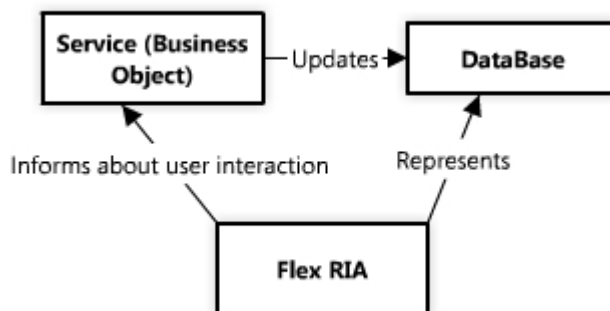


Figura 2.5: O modelo MVC num contexto de uma aplicação Flex

2.4.2 Python

O *Python* [Pyt09] é uma linguagem de programação orientada a objectos, que se caracteriza por não ser compilada, mas sim interpretada. Apesar deste facto poder preconizar níveis de *performance* mais baixos, quanto comparados com outras linguagens populares (C++, Java, etc), as suas outras vantagens levam o *python* a ser a escolha de muitos programadores.

Entre essas vantagens encontra-se a facilidade de aprendizagem, que leva a que as aplicações possam ser desenvolvidas mais rapidamente. Para além disso, o forte suporte no que diz respeito a integração com outras linguagens e ferramentas, faz com que facilmente interaja com estas. São também vantagens a sua sintaxe simples e elegante e tipagem dinâmica, bem como a extensa colecção de bibliotecas.

O facto da linguagem ser interpretada leva-a a poder também ser utilizada como linguagem de *scripting*, rivalizando em popularidade com outras linguagens populares neste campo, como o *Perl*. Para além disso, este facto permite-lhe ser uma linguagem multi-plataforma, visto existirem interpretadores para uma multitude de sistemas, entre os quais *Windows*, *Linux/Unix*, *Mac OS X*, *OS/2*, *Amiga*, *Palm Handhelds* e *Nokia Mobile*.

2.4.3 SOAP

O SOAP [SWS09] é um protocolo para a troca estruturada de informação para a implementação de *web services* em plataformas distribuídas [STK01]. Utiliza XML como formato das suas mensagens, que podem ser transmitidas sobre HTTP. Na Figura 2.6 pode ver-se a arquitectura de uma mensagem SOAP.

A utilização de SOAP facilita a comunicação através de *proxies* e *firewalls* e, embora normalmente seja utilizado sobre HTTP, é suficientemente versátil para utilizar diferentes protocolos de transporte (SMTP, por exemplo).

Torna-se também uma enorme vantagem o facto de ser independente da linguagem ou da plataforma. Desde que sigam o protocolo, qualquer aplicação em qualquer plataforma poderá comunicar com qualquer outra aplicação e plataforma do mesmo modo.

A utilização de SOAP serve maioritariamente para aceder a serviços ou invocar métodos em servidores. O facto de ser fortemente estruturada permite-lhe ser usada por qualquer aplicação, sem conhecimento do funcionamento interno do serviço ou método a que pretende aceder.

O servidor simplesmente disponibiliza o serviço, descrevendo o método e respectivos parâmetros necessários. Em seguida, o cliente envia uma mensagem SOAP (utilizando o formato XML) que o servidor consumirá, fazendo o *parsing* do XML recebido, retornando então o resultado numa nova mensagem SOAP.

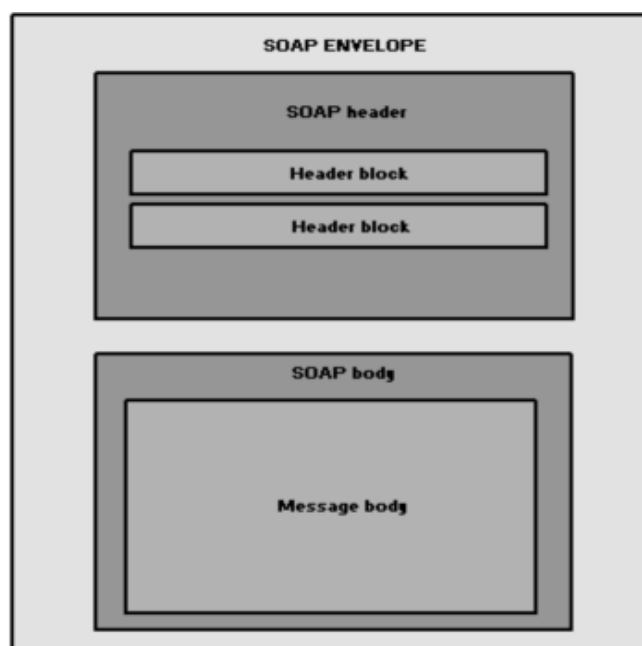


Figura 2.6: Arquitectura das mensagens SOAP

2.4.4 Suds

O *Suds* [Sud09] é um cliente de *web services* (baseado em SOAP) para *python*. Tem como objectivo apresentar um interface simples para interacção com os serviços. Isto significa que os utilizadores podem abstrair-se das complexidades inerentes da WSDL.

Ao criar um novo cliente, é automaticamente processado o *wSDL*, criando assim, durante a execução, os objectos que reflectem os tipos e métodos descritos no documento. Estes objectos podem então ser usados, como qualquer outro objecto no *python*, para invocar os métodos do serviço, que retornarão o resultado, também na forma de qualquer dos tipos gerados anteriormente.

2.4.5 TurboGears WebServices

O *TurboGears* é uma *framework* desenvolvida em *python* para agilizar o desenvolvimento de aplicações *web*. Essa *framework* providencia uma API simples para a criação de *web services* disponíveis via SOAP, o *TurboGears WebServices* [TGW09].

Esta API permite gerar automaticamente os WSDL a partir dos métodos especificados no *python*, que podem estar completamente abstraídos de tudo o que se relacione com *web services*, funcionando como métodos de uma qualquer outra aplicação *python*.

2.4.6 Bonjour

O *Bonjour*, previamente conhecido como *Rendezvous*, é um protocolo da *Apple* para descoberta de serviços. Pode localizar impressoras, computadores e os serviços que estes disponibilizam numa rede local.

O protocolo apenas anuncia os serviços existentes, sendo que o acesso a estes é feito de forma completamente independente. Isto significa que os serviços anunciados têm também que utilizar o *Bonjour* para o poderem fazer.

Este protocolo foi utilizado, através duma interface *python*, para facilitar a descoberta de servidores da captura na rede local, sendo que estes publicariam os seus serviços utilizando o protocolo, enquanto que a aplicação desenvolvida os procuraria baseando-se no mesmo.

2.4.7 Flex Scheduling Framework

A *Flex Scheduling Framework* é, como o nome indica, uma *framework* que permite criar e personalizar múltiplos componentes relacionados com a calendarização e agendamento de eventos.

Para atingir o seu objectivo de ser completamente extensível e facilitar a criação destes componentes, permite a visualização e manipulação de milhares de entradas agendadas, em tempo real. Pode ser usada em quer através de *MXML* ou *ActionScript* e é consistente com a abordagem utilizada nas aplicações *Flex*, ou seja, orientada a *data providers* (modelos com os dados visualizados/manipulados pela aplicação) e despacho de eventos.

Na Figura 2.7 pode ver-se um exemplo das funcionalidades que esta *framework* permite.



Figura 2.7: Exemplo das potencialidades da *Flex Scheduling Framework*

Apesar das suas potencialidades optou-se, após algum estudo da *framework*, prosseguir por outro caminho, uma vez que nem todas as funcionalidades requeridas poderiam ser implementadas intuitivamente utilizando esta *framework*, já bastante direccionada para um tipo muito específico de utilização.

2.4.8 Mate Flex Framework

O *Mate* [Mat09] é também uma *framework* para o *Flex*, que permite desenvolver aplicações com uma arquitectura orientada a eventos e baseada em *tags*.

Foi criada para facilitar a gestão de eventos criados pelo Flex. É assim possível definir facilmente onde serão tratados esses eventos, se existem dados que precisam de ser recuperados ou eventos a ser lançados.

Na Figura 2.8 está representada a arquitectura da comunicação de uma aplicação utilizando *Mate*.

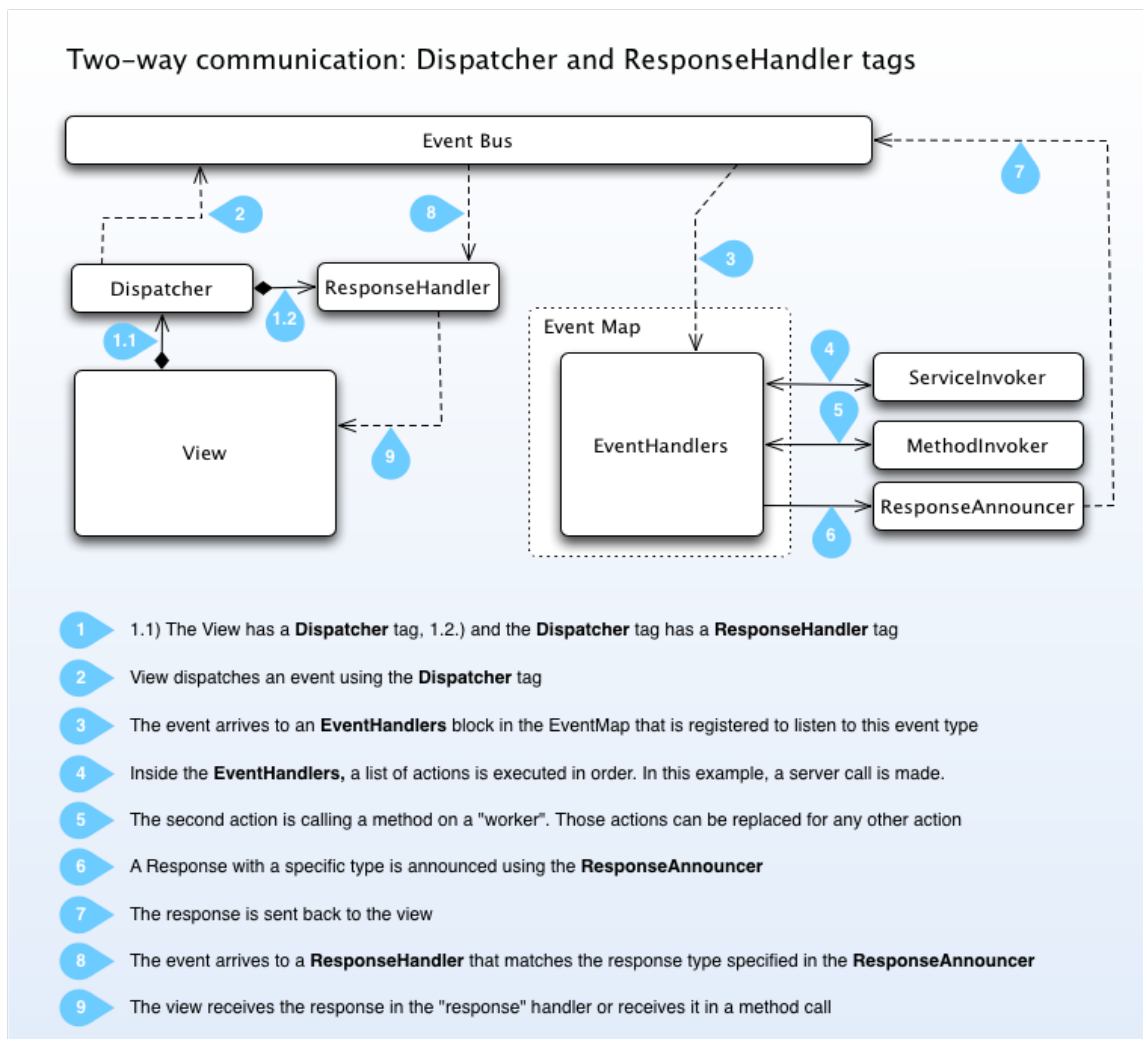


Figura 2.8: Arquitectura de uma aplicação utilizando a *framework* *Mate*.

Sumariamente, o *Mate* consiste numa arquitectura baseada em mapas de eventos, onde se define o que acontece quando certos eventos são lançados, que podem invocar métodos ou serviços e, no final, retornam esses resultados ao componente visual que iniciou o processo.

Apesar da simplicidade apresentada, optou-se por não utilizar a *framework*, devido à sua ainda falta de maturidade e a algumas limitações tecnológicas que não se coadunavam com os objectivos do projecto.

2.4.9 Conclusões

Após o estudo das tecnologias aqui apresentadas, iremos então resumir como foram efectivamente utilizadas no desenvolvimento do projecto.

Separando a aplicação por camadas, cujo diagrama pode ser visto na Figura 2.9, foi decidida a utilização de *Adobe Flex* para a camada de apresentação, por proporcionar a criação de interfaces visualmente apelativos, dinâmicos e intuitivos, pilares de qualquer aplicação RIA, mas também porque os dispositivos que a aplicação pretende controlar, possuíam já o seu GUI na mesma tecnologia, favorecendo a reutilização de componentes que se pretendiam idênticos em ambos os sistemas.

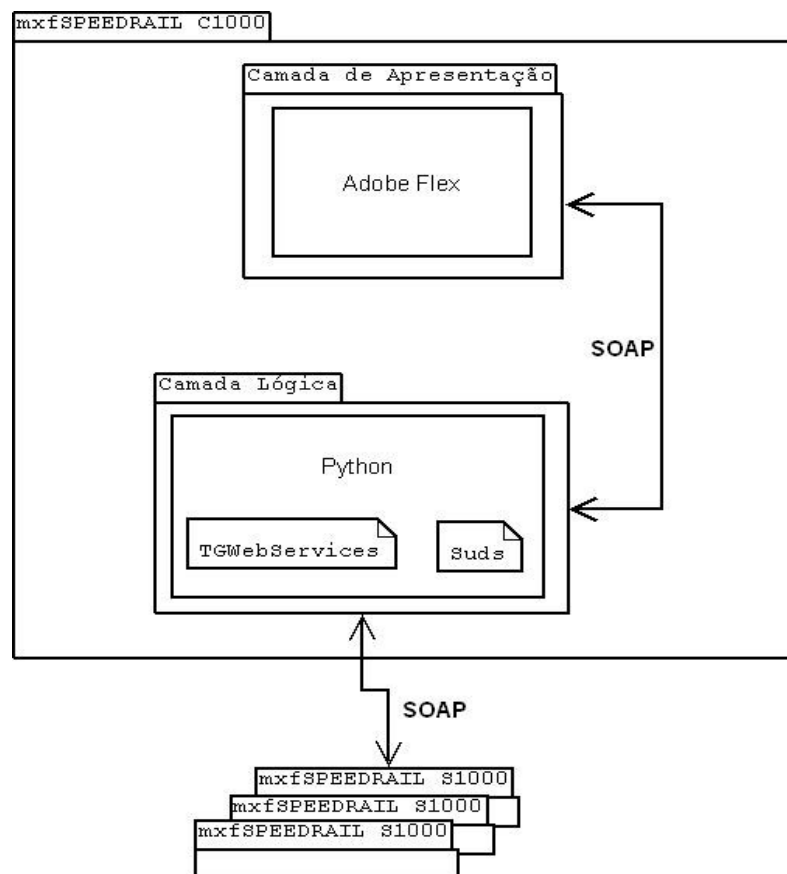


Figura 2.9: Tecnologias utilizadas por camada

Revisão Tecnológica

Para o desenvolvimento em *Flex* optou-se pela utilização do *Adobe Flex Builder*, um IDE em tudo semelhante ao Eclipse. Embora a ferramenta seja poderosa e intuitiva, a falta de opções não nos permitiu sequer avaliar alternativas.

A comunicação entre a camada lógica e a de apresentação seria implementada utilizando *web services* (com SOAP). Uma vez que a camada lógica teria que comunicar através de SOAP com os servidores S1000 que iriam ser controlados pela aplicação, nem sequer foram consideradas alternativas neste ponto. Desta forma, a integração e compatibilidade entre produtos seria em muito facilitada, o que era também um dos objectivos do projecto.

Finalmente, para a camada lógica utilizou-se *python*, tendo por base o desempenho dos produtos da mesma família, já existentes na empresa, desenvolvidos utilizando *python*. Como se verificou que estes corriam sem apresentar qualquer problema, quer técnico quer de *performance*, que justificasse alguma troca de linguagem.

Para o desenvolvimento do *python*, optou-se pela utilização do Eclipse, devido principalmente à sua extensa colecção de *plugins*. Entre estes encontra-se o *PyDev*, um *plugin* direccionado para programadores de *python*, que facilita de sobremaneira o desenvolvimento nesta linguagem.

3 Sistema de Controlo de Equipamento de Produção de TV

Neste capítulo será detalhado, com mais pormenor, o problema que se propunha resolver, sendo que em seguida se apresentará também a solução proposta para o resolver.

3.1 Objectivos

Tal como visto no capítulo anterior, qualquer processo de produção multimédia, no qual se inclui a produção de televisão, tem que passar por diversas fases. Entre essas fases encontra-se a fase de *Ingest*, que será a mais relevante para o projecto. Esta fase corresponde à captura e armazenamento dos conteúdos obtidos, quer de vídeo quer de áudio.

Tipicamente, esta captura dos conteúdos processa-se da seguinte forma: é adquirida a fonte do sinal que se pretende capturar, que passará pela estação de captura, sendo aí tratado e armazenado no formato desejado para a sua posterior edição ou transmissão imediata.

Actualmente, o método mais comum para a realização desta operação será a presença de uma estação de captura para cada sinal de entrada, como se pode ver no exemplo da Figura 3.1.



Figura 3.1: Exemplo de um cenário de utilização de estações de captura

Sistema de Controlo de Equipamento de Produção de TV

Como fontes de entrada, existirão diversos sinais SDI, podendo estes ter origem directa, através da obtenção do sinal de uma câmara em plena filmagem. Esta situação será usual para transmissões em directo, por exemplo. Podem também originar noutros dispositivos de armazenamento temporário, sendo depois retransmitidos para a estação de captura.

Após a sua passagem pela estação de captura, onde serão convertidos para qualquer formato desejado, posteriormente utilizado pelos responsáveis pela edição, são finalmente enviados para um servidor de armazenamento.

Isto coloca o problema de ter múltiplas estações de captura para múltiplos sinais de entrada, o que obriga ao controlo de cada uma individualmente. Embora se possam, com certeza, controlar todas a partir de uma mesma máquina, acedendo-lhes remotamente, continua a ser algo complexa a gestão de todas em simultâneo. Iniciar capturas ou configurar as estações tornam-se assim tarefas repetitivas e enfadonhas, obrigando a maior atenção do operador, ou mesmo obrigando a manter vários operadores para estações diferentes.

Outro cenário será o de manter apenas uma estação de captura, para a qual se pode alterar a fonte do sinal de entrada, através de algum dispositivo, como um *switch*. Nesta situação alguns dos problemas mantêm-se, enquanto outros são substituídos por novos.

Uma vez que se utiliza a mesma estação não será necessário configurá-la de novo para cada sinal, a menos que se pretenda realmente mudar algo que dependa do mesmo, caso em que o sistema não será o mais apropriado. No entanto, o problema mais significativo, prende-se com a necessidade de trocar o sinal, que pode originar alguma dessincronização ou atrasos, tornando este sistema pouco vantajoso para utilização em tempo real.

Segue-se então a conclusão lógica que a melhor maneira de gerir ambos os cenários passaria por manter as múltiplas estações de captura dedicadas, cada uma, a um sinal de entrada específico, embora disponibilizando mecanismos para controlo centralizado, simples e eficiente.

Pretende-se também ir um passo adiante, automatizando as tarefas possíveis, através do agendamento de capturas, permitindo assim que grande parte das operações de captura da estação respectiva funcionem sob supervisão.

A próxima secção serve então para descrever a solução proposta, na forma de um protótipo para uma aplicação que contempla a resolver todos estes problemas.

3.2 A Solução

Antes de passar à especificação da solução proposta, será relevante descrever resumidamente o ambiente em que se pretende que a mesma trabalhe e as estações com que deverá interoperar.

De modo a centralizar todas as tarefas anteriormente efectuadas em cada estação de captura, pretende desenvolver-se um sistema que permita monitorizar e controlar cada uma remotamente, num mesmo ambiente e respectiva interface.

Para os propósitos deste projecto, serão utilizadas as estações de captura já desenvolvidas na própria MOG, onde o mesmo decorre, designadas de *mxSPEEDRAIL S1000*. O protótipo desenvolvido será em diante designado como *mxSPEEDRAIL C1000*. Pode ver-se na Figura 3.2 o objectivo final pretendido, usando como base o mesmo cenário do exemplo anterior, visível na Figura 3.1.

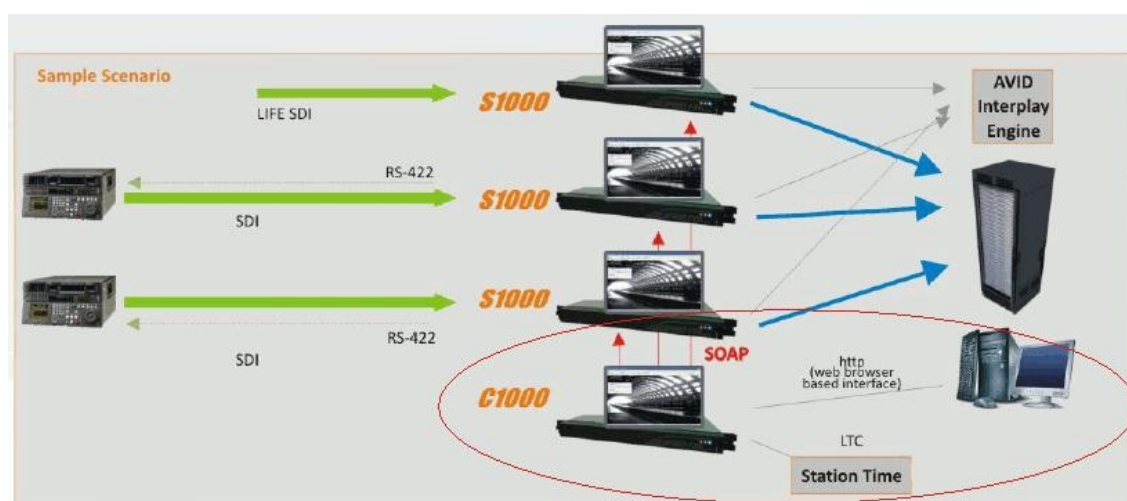


Figura 3.2: Exemplo de um cenário de utilização de estações de captura utilizando o sistema proposto

Resumidamente, o C1000, que será então um controlador para as estações existentes, comunicará via SOAP com as diversas estações de captura existentes no ambiente, permitindo controlá-las do mesmo modo que se procederia ao utilizar as próprias estações.

Esta solução passará então por permitir a configuração do acesso às máquinas que se pretendam controlar, providenciando assim acesso remoto às mesmas, e sendo também esta comunicação efectuada via SOAP, de modo a possibilitar este controlo através da internet.

Uma vez que as interfaces das próprias estações já utilizam SOAP para comunicar com o servidor respectivo, já se poderia tirar partido de algumas dessas mesmas funcionalidades no novo sistema, bastando replicar esses pedidos na respectiva máquina central, associados à estação correspondente.

Levando o conceito de centralização ainda mais longe, para além de uma interface agregadora de múltiplos servidores de captura, pretende-se também dar a possibilidade de organizar e agrupar estas máquinas de acordo com as preferências do operador. Isto implica poder controlar cada máquina individualmente, mas também controlar várias simultaneamente. Esta funcionalidade poderosa fará todo o sentido, em particular para qualquer cenário de gravação multi-câmara de uma mesma cena. Neste caso seria de extrema utilidade a possibilidade de automaticamente iniciar capturas para todas as câmaras seleccionadas, por exemplo.

Estes grupos de máquinas, denominados *gangs*, permitiriam simultaneamente reconfigurar máquinas com um propósito comum, para além das já mencionadas capturas simultâneas. A solução proposta contempla portanto, não só o controlo de múltiplos S1000, como também a configuração de toda esta estrutura de controlo, de uma forma suficientemente intuitiva, permitindo não só que um único operador possa gerir todo o sistema, como também aumentar substancialmente a sua eficiência.

Para além destas inovações, é também objectivo, permitir automatizar muitas das tarefas repetitivas ou previstas a longo prazo, permitindo ao operador programar, antecipadamente, todas as capturas passíveis de agendamento, nas quais se incluem gravações contendo algum padrão de periodicidade, por exemplo.

Sistema de Controlo de Equipamento de Produção de TV

Esta funcionalidade será disponibilizada na forma de um calendário interactivo, tornando a experiência de utilização mais fluída e agradável, permitindo assim manter as máquinas em funcionamento, com uma componente reduzida de intervenção humana.

3.3 Resumo e Conclusões

Este capítulo descreveu então o problema, ou limitação, nos sistemas actuais para ambientes de captura com a utilização de múltiplas fontes ou câmaras, que passa pela dificuldade de controlo central e automatizado. Foi também descrita a solução proposta, que permite anular ou minimizar a grande maioria destes problemas ou dificuldades, e que resultou no desenvolvimento de um protótipo já direccionado para aplicação no mundo real.

Nos capítulos seguintes será detalhada com maior profundidade esta solução, desde os seus requisitos principais, passando pelo desenho da arquitectura e implementação efectiva do sistema em foco.

4 Especificação de Requisitos

No presente capítulo iremos proceder à descrição dos requisitos exigidos para a aplicação desenvolvida. Para esse efeito serão especificados primeiro os requisitos funcionais, sob a forma de casos de uso, identificados na fase inicial do projecto, bem como os requisitos não funcionais do mesmo.

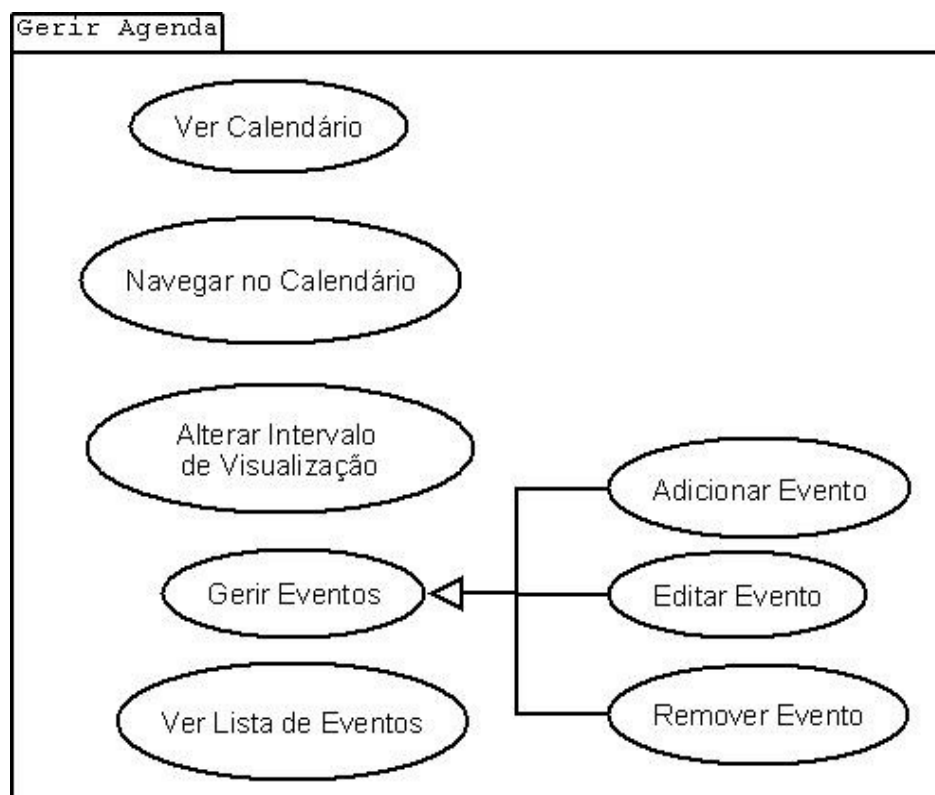
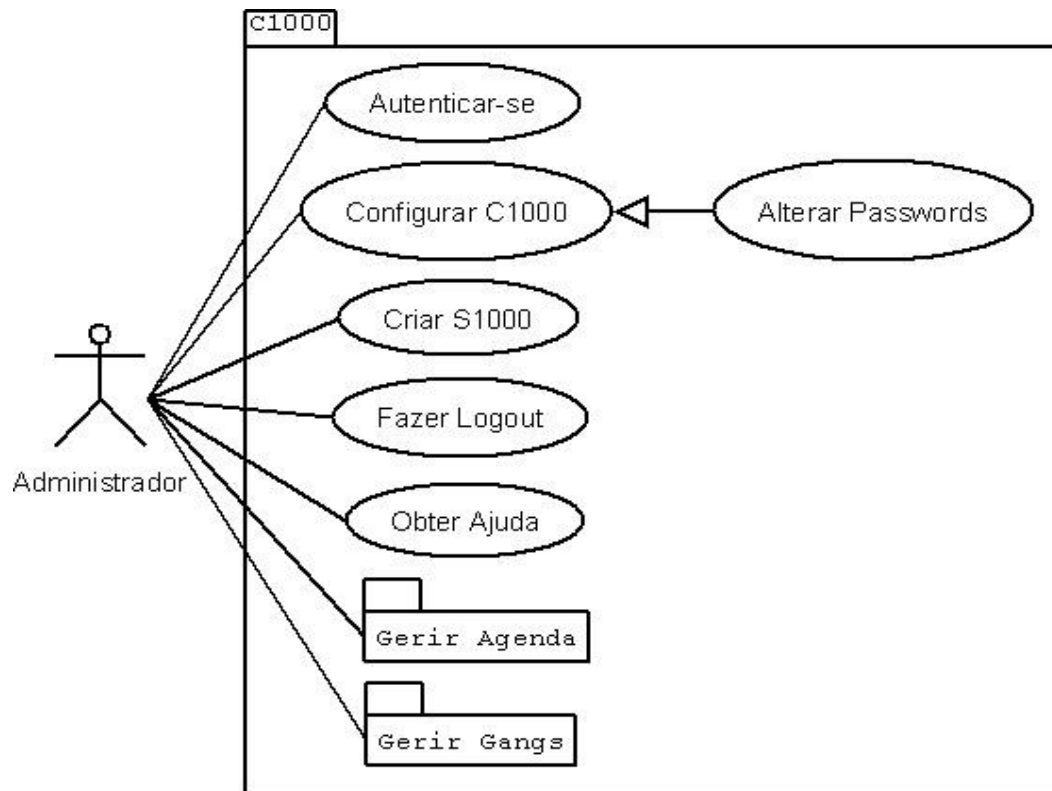
4.1 Requisitos Funcionais e Casos de Uso

Os requisitos funcionais de um projecto pretendem descrever o comportamento do sistema, ou seja, as suas acções para cada *input* permitido pela aplicação, e o respectivo *output* obtido. Uma das formas mais eficazes de fazer o levantamento destes requisitos é através da utilização de casos de uso [LW03].

Os casos de uso representam as possíveis interacções entre os utilizadores de um sistema com o próprio. São um excelente ponto de partida na construção de qualquer *software* uma vez que, não especificando a forma como deverá ser construído, especificam como deve o sistema comportar-se no final.

Com este objectivo em mente, foram definidos à partida os casos de uso que o software a desenvolver deveria suportar. O diagrama obtido pode ser visto na Figura 4.1, enquanto que cada caso representado na imagem será, nas próximas secções, descrito isoladamente e com maior detalhe.

Especificação de Requisitos



Especificação de Requisitos

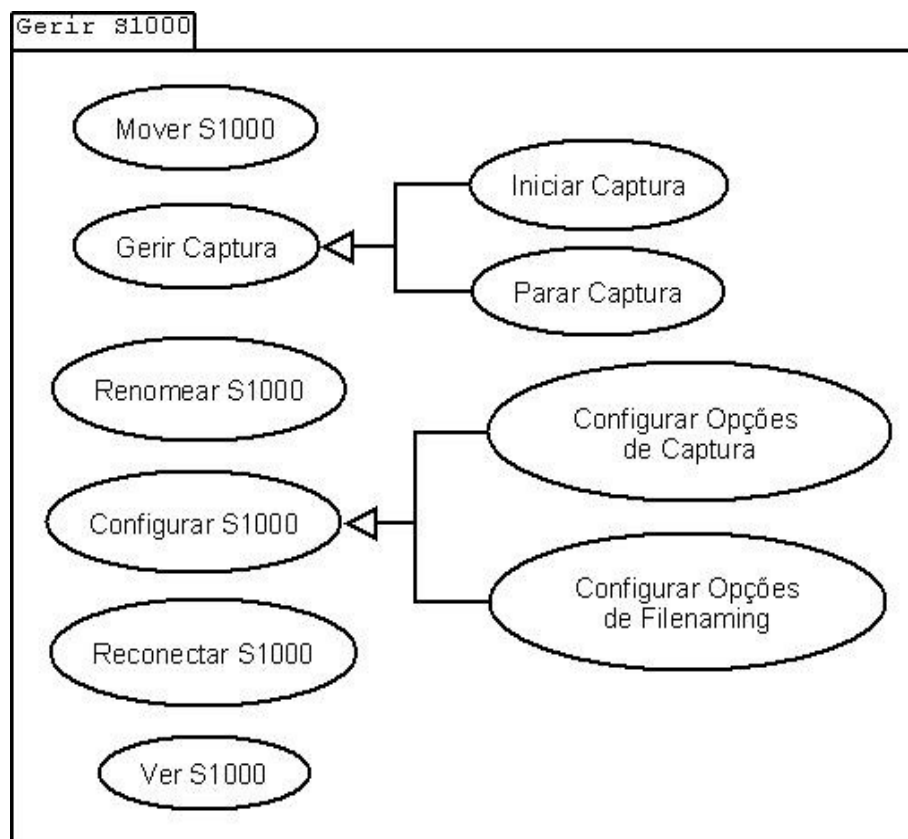
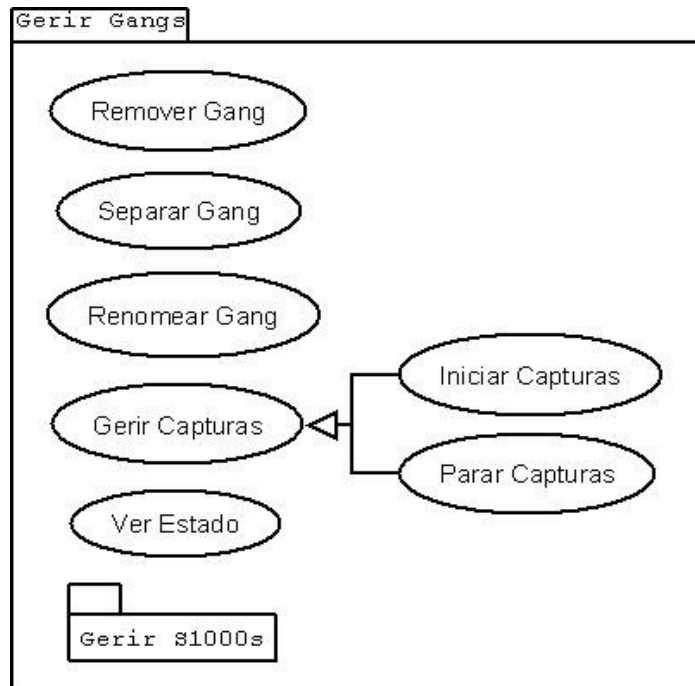


Figura 4.1: Casos de uso do sistema

4.1.1 Autenticação

Este caso de uso descreve o passo inicial pelo qual terá que passar qualquer utilizador, de modo a autenticar-se perante o sistema, obtendo assim acesso às restantes funcionalidades.

Tabela 4.1: Especificação do caso de uso “Autenticação”

Resumo	Autentica um utilizador perante o sistema
Actores	Todos
Pré-Condições	Nenhuma
Procedimento	Introduzir um <i>username</i> e uma <i>password</i> ; Pressionar OK.
Resposta do Sistema	Permite acesso às restantes funcionalidades do <i>software</i> ou apresenta um erro, caso o processo não possa ser concluído.

4.1.2 Alterar Password

Existem dois tipos de utilizadores que o sistema distingue: *administrator* e *ingester*. Para que se efectue a autenticação, é necessário introduzir um nome de utilizador e uma *password*, correspondente a um desses tipos. Essa *password* poderá ser alterada para qualquer outra através da acção de um utilizador, desde que autenticado como *administrator*.

Tabela 4.2: Especificação do caso de uso “Alterar Password”

Resumo	Altera a password correspondente a um dos tipos de utilizador existentes (<i>Administrator</i> ou <i>Ingester</i>)
Actores	<i>Administrator</i>
Pré-Condições	Estar autenticado
Procedimento	Aceder às configurações gerais do C1000; Introduzir a <i>password</i> de <i>administrator</i> ; Seleccionar o tipo de utilizador respeitante à alteração; Introduzir a nova <i>password</i> e respectiva confirmação.
Resposta do Sistema	Altera a password respectiva, substituindo-a pela nova ou apresenta um erro, caso o processo não possa ser concluído.

4.1.3 Adicionar um S1000/*Gang* ao sistema

De modo a poder controlar os diversos servidores S1000, o utilizador terá que se conectar remotamente aos respectivos. Este caso de uso trata dessa operação, que permitirá adicionar um desses servidores ao sistema, criando, se necessário, um novo *gang* para o efeito.

Tabela 4.3: Especificação do caso de uso “Adicionar *Gang*/S1000”

Resumo	Acrescenta um novo S1000 ao conjunto monitorizado, criando um novo <i>gang</i> se necessário
Actores	<i>Administrator</i>
Pré-Condições	Estar autenticado
Procedimento	Seleccionar o endereço do novo S1000; Introduzir um nome para o S1000; Seleccionar o nome do <i>gang</i> , ou introduzir um novo, onde será agrupado o S1000; Introduzir o nome de utilizador e password associados à nova máquina.
Resposta do Sistema	Acrescenta o S1000 ao <i>gang</i> pretendido, ou cria um novo <i>gang</i> para o servidor. Apresenta um erro, caso o processo não possa ser concluído.

4.1.4 Efectuar o *Logout*

Após uma sessão de utilização do equipamento, o utilizador pode efectuar o *logout* da aplicação, deixando assim o sistema protegido contra utilizações abusivas na sua ausência.

Tabela 4.4: Especificação do caso de uso “Fazer *Logout*”

Resumo	Efectua o <i>logout</i> da aplicação
Actores	Todos
Pré-Condições	Estar autenticado
Procedimento	Seleccionar a opção “ <i>logout</i> ” do menu.
Resposta do Sistema	Efectua o <i>logout</i> , retornando o sistema a um estado não autenticado.

4.1.5 Obter ajuda

Durante a utilização, e apesar do esforço para tornar o ambiente intuitivo e de fácil compreensão, o utilizador pode, ocasionalmente, possuir necessidade de assistência mais detalhada no cumprimento de alguma operação. É então possível visualizar um ecrã de ajuda que facilite essa mesma compreensão dos processos associados a qualquer tipo de operação.

Tabela 4.5: Especificação do caso de uso “Obter Ajuda”

Resumo	Permite a visualização de informação detalhada sobre o funcionamento da aplicação
Actores	Todos
Pré-Condições	Estar autenticado
Procedimento	Seleccionar a opção correspondente no menu principal.
Resposta do Sistema	Apresenta um ecrã com informação detalhada sobre a aplicação

4.1.6 Aceder à agenda

Uma das funcionalidades mais significativas da aplicação é a possibilidade de agendar capturas de vídeo em determinados servidores para datas futuras. Este caso de uso diz respeito precisamente a como aceder a essa agenda.

Tabela 4.6: Especificação do caso de uso “Aceder à agenda”

Resumo	Permite aceder à agenda da aplicação, dando acesso às funcionalidades que a ela dizem respeito
Actores	Todos
Pré-Condições	Estar autenticado
Procedimento	Seleccionar a opção correspondente no menu principal.
Resposta do Sistema	Apresenta um ecrã com toda a informação respeitante à agenda e capturas agendadas

4.1.7 Navegar na agenda

O caso de uso presente refere-se à navegação que um utilizador do sistema poderá efectuar no calendário disponibilizado pela aplicação. Esta navegação realiza-se de modo intuitivo, permitindo avançar ou recuar a data unidade a unidade, ou saltar directamente para alguma data específica.

Tabela 4.7: Especificação do caso de uso “Navegar na agenda”

Resumo	Permite ao utilizador navegar no calendário da aplicação, alterando a data do dia ou conjunto de dias visíveis no mesmo.
Actores	Todos
Pré-Condições	Estar autenticado
Procedimento	Seleccionar uma data específica ou avançar/recuar a data mostrada utilizando os botões de navegação disponibilizados pela interface.
Resposta do Sistema	Altera o conjunto de dias visíveis no calendário da aplicação

4.1.8 Alterar modo de visualização da agenda

Para que a experiência de utilização se torne agradável e simples, é também permitido ao utilizador alterar o modo de visualização do calendário/agenda da aplicação. Existem portanto 3 modos diferentes, visão de mês, semana e dia, que permitem a visualização da agenda com um maior ou menor detalhe.

Tabela 4.8: Especificação do caso de uso “Alterar modo de visualização da agenda”

Resumo	Permite ao utilizador alterar o modo de visualização da agenda.
Actores	Todos
Pré-Condições	Estar autenticado
Procedimento	Seleccionar o modo, de entre os apresentados na interface (mês, semana ou dia)
Resposta do Sistema	Altera o conjunto de dias visíveis no calendário, de acordo com o modo seleccionado

4.1.9 Adicionar um novo evento à agenda

Este caso de uso diz respeito à adição de um nova captura futura à agenda. É permitido ao utilizador seleccionar o dia e hora de início e fim da captura, bem como o *gang* de S1000s que estarão encarregados da mesma. Este poderá também estabelecer a periodicidade da captura, de entre as opções: diária, semanal, mensal, anual ou única.

Tabela 4.9: Especificação do caso de uso “Adicionar evento”

Resumo	Adiciona um novo evento à agenda
Actores	Todos
Pré-Condições	Estar autenticado; A existência de, pelo menos, um S1000 no sistema.
Procedimento	Seleccionar a data de início e fim da captura; Seleccionar o <i>gang</i> encarregado da mesma.; Seleccionar a periodicidade do evento.
Resposta do Sistema	Adiciona o novo evento à agenda, ou várias instâncias do mesmo, caso se pretenda repetir o evento segundo alguma periodicidade previamente definida. Apresenta um erro, caso o processo não possa ser concluído.

4.1.10 Remover um evento da agenda

Num paralelo com o caso de uso anterior, o caso de uso descrito prende-se com a remoção de uma captura agendada. O utilizador pode remover não só a captura

Tabela 4.10: Especificação do caso de uso “Remover evento”

Resumo	Remove um evento da agenda
Actores	Todos
Pré-Condições	Estar autenticado; Existência de eventos na agenda.
Procedimento	Seleccionar o evento na interface; Optar pela remoção isolada do evento seleccionado ou de todas as repetições do mesmo.
Resposta do Sistema	Remove o evento da agenda, ou as várias instâncias do mesmo. Apresenta um erro, caso o processo não possa ser concluído.

4.1.11 Editar um evento

Na sequência dos dois últimos casos de uso, o presente refere-se à possibilidade de edição de uma captura agendada. O utilizador pode editar não só a captura

Tabela 4.11: Especificação do caso de uso “Editar evento”

Resumo	Edita um evento da agenda
Actores	Todos
Pré-Condições	Estar autenticado; Existência de eventos na agenda
Procedimento	Seleccionar o evento na interface; Existem duas opções de edição de eventos: 1- Arrastar o evento, com o rato, para a data pretendida; 2- Arrastar a zona inferior do evento, alterando a sua duração; 3- Abrir a janela de edição detalhada, seleccionando a nova data de início e fim da captura, bem como o gang que a assegurará. Esta opção permite seleccionar a edição isolada do evento ou de todas as suas repetições.
Resposta do Sistema	Remove o evento da agenda, ou as várias instâncias do mesmo. Apresenta um erro, caso o processo não possa ser concluído.

4.1.12 Renomear Gang

O primeiro caso de uso relativo à gestão de *gangs* relaciona-se com o seu nome. Para facilitar a identificação do conjunto de S1000s controlado por um *gang* é possível dar-lhe, em qualquer altura, um outro nome, desde que não exista nenhum *gang* a utilizar o mesmo.

Tabela 4.12: Especificação do caso de uso “Renomear Gang”

Resumo	Altera o nome do <i>gang</i> seleccionado
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema
Procedimento	Seleccionar o <i>gang</i> pretendido; Seleccionar a opção correspondente num dos menus disponíveis (menu do <i>gang</i> ou contextual); Introduzir o novo nome pretendido para o <i>gang</i> .
Resposta do Sistema	Altera o nome do <i>gang</i> , ou apresenta um erro caso não consiga concluir o processo.

4.1.13 Remover um ou mais *Gangs*

Este caso de uso prende-se com a remoção de um *gang* do sistema, deixando o utilizador, efectivamente, de ter controlo dos servidores S1000 que lhe estavam associados.

Tabela 4.13: Especificação do caso de uso “Remover *Gang(s)*”

Resumo	Remove o agrupamento de servidores de captura do sistema
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema
Procedimento	Seleccionar um ou mais <i>gangs</i> ; Seleccionar a opção correspondente num dos menus disponíveis (menu do <i>gang</i> ou contextual).
Resposta do Sistema	Remove o <i>gang</i> , ou apresenta um erro caso não consiga concluir o processo.

4.1.14 Juntar *Gangs*

Caso o utilizador necessite de agrupar os servidores S1000 correspondentes a dois *gangs* distintos, é-lhe também possível proceder a essa operação, o que essencialmente se traduz numa movimentação de todos os servidores dos *gangs* seleccionados para o primeiro desses.

Tabela 4.14: Especificação do caso de uso “Juntar *Gangs*”

Resumo	Agrupa dois ou mais <i>gangs</i> , movendo todos os servidores controlados pelos <i>gangs</i> seleccionados para um único destes.
Actores	Todos
Pré-Condições	Estar autenticado; A existência de dois ou mais <i>gangs</i> no sistema.
Procedimento	Seleccionar dois ou mais <i>gangs</i> ; Seleccionar a opção correspondente no menu contextual.
Resposta do Sistema	Agrupa os <i>gangs</i> seleccionados num único, ou apresenta um erro caso não consiga concluir o processo.

4.1.15 Separar Gangs

Outro caso de uso implementado é o que diz respeito ao processo de separar um *gang*. Esta funcionalidade refere-se ao processo de mover todos os servidores S1000 para novos *gangs*, criados especificamente para albergar cada um destes.

Tabela 4.15: Especificação do caso de uso “Separar Gangs”

Resumo	Separa o(s) <i>gang(s)</i> seleccionados em vários, um por cada servidor S1000 que o mesmo agrupava.
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Seleccionar um ou mais <i>gangs</i> ; Seleccionar a opção correspondente no menu contextual ou no menu do próprio <i>gang</i> .
Resposta do Sistema	Cria um novo <i>gang</i> para cada S1000 pertencente ao <i>gang</i> seleccionado, removendo-o no final, ou apresenta um erro caso não consiga concluir o processo.

4.1.16 Ver Estado

É de extrema utilidade para o utilizador poder obter uma rápida panorâmica geral do estado dos servidores associados a um *gang*, sem precisar de navegar pela informação detalhada de cada S1000. É-lhe por isso facultada a funcionalidade de visualização do estado actual de cada servidor respeitante ao *gang* seleccionado. Estes estado indica se existe algum problema com o servidor, ou se este se encontra a capturar actualmente.

Tabela 4.16: Especificação do caso de uso “Ver Estado”

Resumo	Apresenta uma lista com o estado geral de cada servidor S1000. Este estado pode ser: “ <i>recording</i> ”, “ <i>idle</i> ”, “ <i>error</i> ” ou “ <i>disconnected</i> ”
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Seleccionar o <i>gang</i> ; Seleccionar a opção correspondente no menu do próprio.
Resposta do Sistema	Apresenta uma lista com o estado de cada S1000

4.1.17 Iniciar capturas dum gang

O objectivo de controlar remotamente múltiplos servidores de captura tem como uma das suas vantagens mais significativas a possibilidade de poder iniciar e parar essas capturas em qualquer momento. Se a isso se juntar o benefício de poder enviar comandos a vários servidores em simultâneo, utilizando o conceito de *gangs*, obtém-se uma das mais poderosas funcionalidades da aplicação, como o seja a gestão de capturas por *gang*.

Tabela 4.17: Especificação do caso de uso “Iniciar capturas dum gang”

Resumo	Inicia a captura, simultaneamente, em todos os S1000 pertencentes ao <i>gang</i> seleccionado.
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Seleccionar o <i>gang</i> ; Seleccionar a opção correspondente no menu do próprio.
Resposta do Sistema	Inicia a captura em todos os servidores pertencentes a esse <i>gang</i> , ou apresenta um erro caso não o consiga.

4.1.18 Parar capturas num gang

Como se viu no caso de uso anterior, uma das opções mais significativas é a de iniciar capturas para todo um *gang*. Obviamente, isto não faria sentido sem a possibilidade de as parar a qualquer momento.

Tabela 4.18: Especificação do caso de uso “Parar capturas num gang”

Resumo	Finaliza a captura, simultaneamente, em todos os S1000 pertencentes ao <i>gang</i> seleccionado.
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Seleccionar o <i>gang</i> ; Seleccionar a opção correspondente no menu do próprio.
Resposta do Sistema	Finaliza a captura em todos os servidores pertencentes ao <i>gang</i> , ou apresenta um erro caso não o consiga.

4.1.19 Ver informação sobre um S1000

Apesar de poderem ser controlados em conjunto, é de enorme relevância a possibilidade de se poder aceder isoladamente aos servidores de cada um desses conjuntos. Isto traduz-se na necessidade inicial de visualizar informação sobre o estado actual de cada S1000 monitorizado.

Tabela 4.19: Especificação do caso de uso “Ver informação sobre um S1000”

Resumo	Apresenta informação sobre o S1000 seleccionado, na forma de um GUI, que também possui menus para o controlo individual do respectivo S1000.
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Seleccionar o <i>gang</i> ; Seleccionar a opção correspondente no menu do próprio; Seleccionar o servidor do qual se pretende obter informação.
Resposta do Sistema	Apresenta informação detalhada sobre o estado de cada servidor S1000 presente num dado <i>gang</i>

4.1.20 Restabelecer ligação a um S1000

É possível que, em alguns casos, a ligação a um servidor S1000 seja perdida, quer por falha da própria máquina ou mesmo eléctrica. Neste caso, é permitido ao utilizador, após a regularização da situação localmente, repor a ligação remotamente. Optou-se por não automatizar este processo, perspectivando situações em que o servidor esteja desligado conscientemente, independentemente do motivo para tal.

Tabela 4.20: Especificação do caso de uso “Restabelecer ligação a um S1000”

Resumo	Tenta restabelecer a ligação a um servidor S1000 no qual esta se tenha perdido
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um S1000 desconectado.
Procedimento	Aceder à informação detalhada do S1000 (no <i>gang</i> respectivo); Seleccionar o servidor pretendido; Pressionar o botão correspondente..
Resposta do Sistema	Repõe a ligação ao S1000 (autenticando-se novamente e iniciando uma nova sessão) ou apresenta um erro caso não consiga concluir o processo

4.1.21 Iniciar uma Captura

Isoladamente, é também possível enviar comandos para iniciar e parar uma captura relativa a um determinado servidor S1000.

Tabela 4.21: Especificação do caso de uso “Iniciar uma captura”

Resumo	Inicia uma captura no S1000 seleccionado.
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Acéder à informação detalhada do S1000 (no <i>gang</i> respectivo); Seleccionar o servidor pretendido; Seleccionar a opção correspondente no menu do próprio.
Resposta do Sistema	Inicia uma captura no servidor em questão, ou apresenta um erro caso não o consiga.

4.1.22 Parar uma Captura

No lado oposto ao do caso de uso anterior virá o seguinte, que permite ao utilizador a não menos importante acção de parar uma captura anteriormente iniciada.

Tabela 4.22: Especificação do caso de uso “Parar uma captura”

Resumo	Finaliza a captura no S1000 seleccionado
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Acéder à informação detalhada do S1000 (no <i>gang</i> respectivo); Seleccionar o servidor pretendido; <i>Seleccionar a opção correspondente no menu do próprio.</i>
Resposta do Sistema	Finaliza uma captura no servidor em questão, ou apresenta um erro.

4.1.23 Renomear S1000

Tal como para um *gang*, torna-se necessário identificar facilmente e inequivocamente cada um dos servidores S1000 controlados pela aplicação. Para este efeito é-lhes dado um nome único aquando da sua adição ao sistema. Em qualquer momento pode o utilizador alterar esse nome, desde que a nova designação proposta não possua nenhuma incompatibilidade com os já existentes.

Tabela 4.23: Especificação do caso de uso “Renomear S1000”

Resumo	Altera o nome identificativo do S1000 em questão
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Aceder à informação detalhada dos S1000 (no <i>gang</i> respectivo); Seleccionar o servidor pretendido; Seleccionar a opção correspondente no menu do próprio; Introduzir o novo nome que irá identificar o S1000.
Resposta do Sistema	Altera o nome identificativo do S1000, ou apresenta um erro caso não consiga concluir o processo.

4.1.24 Mover S1000

Para facilitar a gestão de conjuntos de servidores, implementou-se a funcionalidade especificada neste caso de uso, que diz respeito à possibilidade de mover qualquer S1000 entre *gangs*, permitindo assim o rearranjo simples da organização dos conjuntos. Este procedimento será sem dúvida útil caso se pretenda reutilizar uma máquina associada a um determinado *gang* para propósitos que melhor se identificam com outro destes. Caso o *gang* de destino não exista no sistema, este será criado, contendo unicamente o servidor em questão.

Tabela 4.24: Especificação do caso de uso “Mover S1000”

Resumo	Move um S1000 para o <i>gang</i> seleccionado, criando-o se necessário
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Aceder à informação detalhada dos S1000 (no <i>gang</i> respectivo); Seleccionar o servidor pretendido; Seleccionar a opção correspondente no menu do próprio; Seleccionar o nome do <i>gang</i> para onde se pretende mover o servidor.
Resposta do Sistema	Move o S1000 para o <i>gang</i> seleccionado, criando-o se não existir, ou apresenta um erro caso não consiga concluir o processo.

4.1.25 Remover S1000

Tal como no caso dos gangs, também é possível a remoção de um S1000 individual, efectivamente apagando-o do seu grupo, e cessando assim a sua monitorização.

Tabela 4.25: Especificação do caso de uso “Remover S1000”

Resumo	Remove um S1000 do seu <i>gang</i> .
Actores	Todos
Pré-Condições	Estar autenticado; A existência, no sistema, de um <i>gang</i> que controle, pelo menos, dois servidores de captura.
Procedimento	Acéder à informação detalhada dos S1000 (no <i>gang</i> respectivo); Seleccionar a opção correspondente no menu do próprio; Confirmar a operação.
Resposta do Sistema	Remove o S1000 seleccionado, ou apresenta um erro caso não consiga concluir o processo.

4.1.26 Configurar opções de captura

Tal como explicado anteriormente, é permitido ao utilizador controlar cada S1000 individualmente, tendo assim acesso a todas as opções relevantes do mesmo. Este caso de uso refere portanto o acesso e alteração das configurações de um S1000; no presente caso opções relacionadas com capturas. Entre estas configurações encontram-se a escolha de formatos de *input* e *output*, ou o local de gravação dos ficheiros.

Tabela 4.26: Especificação do caso de uso “Configurar opções de captura”

Resumo	Configura as opções relacionadas com formatos de input/output e localização para os ficheiros gerados aquando das capturas
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Acéder à informação detalhada dos S1000 (no <i>gang</i> respectivo); Seleccionar o servidor pretendido; Seleccionar a opção correspondente no menu do próprio; No painel das configurações, seleccionar a <i>tab</i> correspondente; Configurar as opções de acordo com o pretendido.
Resposta do Sistema	Altera as configurações (caso existam alterações) das opções relacionadas com capturas de um servidor S1000

4.1.27 Configurar estratégia de *filenaming*

Devido à quantidade de ficheiros capturados por uma qualquer produtora de televisão, torna-se difícil identificá-los posteriormente. É por essa razão que, na linha do caso de uso anterior, é também possível configurar opções relativas à nomenclatura dos ficheiros gerados durante uma captura. Isto permitirá ao utilizador decidir a estratégia de nomeação dos ficheiros, utilizando combinações de opções como data/hora, rótulos personalizados, identificadores ou sequências de letras ou números.

Tabela 4.27: Especificação do caso de uso “Configurar opções de *filenaming*”

Resumo	Configura as opções relacionadas com a estratégia de nomenclatura dos ficheiros gerados na captura
Actores	Todos
Pré-Condições	Estar autenticado; A existência de um <i>gang</i> no sistema.
Procedimento	Aceder à informação detalhada dos S1000 (no <i>gang</i> respectivo); Seleccionar o servidor pretendido; Seleccionar a opção correspondente no menu do próprio; No painel das configurações, seleccionar a <i>tab</i> correspondente; Configurar as opções de acordo com o pretendido.
Resposta do Sistema	Altera as configurações (caso existam alterações) relativas a opções relacionadas com estratégias de <i>filenaming</i> de um servidor S1000

4.2 Requisitos Não Funcionais

Na secção anterior foram especificados os requisitos funcionais do sistema a desenvolver, sob a forma de casos de uso. Iremos agora proceder à descrição dos requisitos não funcionais do projecto, que não sendo tão imediatamente visíveis, as importantes mais-valias que adicionam ao sistema não são de menosprezar.

Os requisitos não funcionais de um projecto são aqueles que especificam os critérios com que se irá avaliar a operacionalidade de um sistema, em lugar de comportamentos específicos deste. Sumariamente, definem o que o sistema deve ser, e não o que deve fazer. Podem comparar-se estes requisitos a qualidades que o *software* deve apresentar.

Passamos então de seguida a apresentar os requisitos não funcionais de maior importância que o projecto deve ter em conta.

4.2.1 Desempenho

Como primeiro requisito não funcional e, sem dúvida, um dos mais importantes, encontra-se o desempenho da aplicação. Por desempenho entenda-se tanto o tempo de resposta a comandos do utilizador, como a fluidez de actualização da informação disponibilizada acerca das máquinas em localizações remotas, que terá que se manter coerente independente do número de servidores monitorizados.

Prevê-se que o sistema possa ser usado para controlar remotamente diversos servidores simultaneamente, pelo que o seu tempo de resposta deverá ser curto. Serão de evitar situações

Especificação de Requisitos

em que o utilizador precise de aguardar mais do que alguns segundos pela conclusão de qualquer operação. Tanto actualizações que resultam de operações efectuadas pelo utilizador sobre os servidores remotos (S1000), como outras derivadas de acções da própria máquina ou de outros utilizadores deverão ser reflectidas na interface em tempo muito próximo do real, de modo a evitar conflitos de utilização, principalmente informação relativa a capturas em curso.

4.2.2 Robustez

A robustez de um sistema prende-se com a capacidade da aplicação suportar, preferencialmente de um modo elegante, *inputs* errados ou inesperados, independentemente da sua proveniência, continuando com um funcionamento dito normal, durante e após um tal evento.

Esta qualidade é de superior importância neste projecto, visto que qualquer tipo de falha inesperada relativamente a qualquer das máquinas controladas não deverá ter influência no resto do sistema, sendo que é de prever que a aplicação seja utilizada em no tipo de cenário em que estará continuamente activa. Como uma das suas importantes funcionalidades se refere à automatização de certos processos, através da calendarização de capturas, é expectável que para além do seu funcionamento continuado, seja também utilizada, na maioria do tempo, sem qualquer tipo de interacção humana, razão pela qual se torna de extrema importância a sua capacidade de resistir a falhas inesperadas.

4.2.3 Fiabilidade

Na mesma linha do requisito anterior encontra-se a fiabilidade. Certamente se compreenderá que melhor do que recuperar e resistir a falhas inesperadas será que estas nunca aconteçam. A fiabilidade diz respeito à capacidade da aplicação de se manter em execução durante longos períodos de tempo continuando sempre a funcionar como originalmente pretendido.

Por razões em tudo idênticas às apresentadas na secção anterior, esta é uma característica relevante que o sistema deverá apresentar. Devido ao estado de execução permanente em que se pretende que seja utilizado, do qual dependerá o funcionamento correcto e de acordo com o esperado dos vários servidores controlados pelo sistema, este é um requisito fundamental no desenvolvimento do projecto.

4.2.4 Interoperabilidade

A interoperabilidade é uma propriedade referente à capacidade de diversos sistemas funcionarem em conjunto. Isto implica a capacidade de comunicarem e trocarem informação de forma transparente, quer através da utilização de protocolos compatíveis ou até a escrita e leitura de ficheiros com o mesmo formato.

A sua relevância para o projecto deverá apresentar-se óbvia, visto que a principal funcionalidade do sistema prende-se com a sua capacidade de controlar e monitorizar diversos servidores de captura. É portanto essencial que exista a capacidade de comunicar e compreender

Especificação de Requisitos

esses mesmos servidores, bem como de que os servidores compreendam e reajam correctamente aos comandos enviados remotamente. Apesar dos objectivos do projecto se referirem apenas ao controlo de um tipo de servidor (o *mxSPEEDRAIL S1000*), é de ressaltar que futuramente poderia apresentar-se o caso de pretender controlar-se diferentes tipos de servidor, pelo que era importante desenvolver o *software* partindo de mecanismos que permitissem esta interoperabilidade e compatibilidade com qualquer tipo de servidor minimamente preparado para este tipo de utilização.

4.2.5 Usabilidade

Com este requisito pretende inferir-se a facilidade com que os utilizadores conseguem compreender e operar um sistema, de modo a realizarem as tarefas pretendidas, atingindo determinado objectivo. São ramificações desta capacidade a elegância e clareza do *design* da interface que permite interagir com a aplicação, bem como a sua eficiência.

Independentemente do nível de automatização da aplicação, é esperado que um utilizador, minimamente familiarizado com este tipo de produto consiga, não só, rapidamente compreender e utilizar o *software*, mas também controlá-la e visualizar a informação pretendida de forma o mais fluida e simples possível.

4.2.6 Modularização

A modularidade de um sistema implica que este seja constituído por várias partes, ou módulos, separados. Desta forma o sistema pode ser dividido em vários subsistemas, os quais podem ser trocados facilmente, sem interferir com o funcionamento do resto do *software*.

Uma vez que a própria aplicação está dividida em dois grande blocos. O primeiro é o servidor que gere todo o sistema, desde a monitorização e controlo dos servidores remotos, como a própria informação relativa ao próprio sistema, como a agenda e listas de *gangs* e os seus membros. É este servidor que reencaminhará os pedidos da interface referentes a comandos para os servidores de captura monitorizados. O segundo é o GUI, que será o cliente, que disponibiliza a informação ao utilizador, permitindo o envio de todo o tipo de comandos ao servidor central, mas separado da camada lógica da aplicação. Isto permite facilmente trocar a interface sem afectar o funcionamento interno do sistema, o que é sem dúvida uma vantagem, conhecendo a velocidade a que a tecnologia evolui actualmente.

Internamente, os próprios módulos principais apresentam-se modularizados eles próprios, permitindo não só a sua fácil substituição como também a reutilização em projectos futuros. Como exemplo real, temos a interface, que partilha alguns componentes com a dos servidores S1000, nomeadamente no que diz respeito a ecrãs de configuração, facilitando a transição de utilizadores de um para o outro sistema.

4.2.7 Internacionalização

Este requisito refere-se à facilidade com que a aplicação é compreendida, e por conseguinte, utilizada, independentemente da sua localização geográfica.

Especificação de Requisitos

No âmbito deste projecto, foi considerado relevante facilitar a utilização do produto com uma mínima dependência da linguagem utilizada. Para este efeito optou-se pela utilização, sempre que possível, de ícones, símbolos ou imagens, promovendo assim uma compreensão tendencialmente mais intuitiva e universal do sistema, por utilizadores dos diversos pontos do globo.

4.2.8 Interface

Terminámos com um requisito que poderá não parecer óbvio numa primeira análise, mas nem por isso menos relevante. O aspecto geral da aplicação, na forma da sua interface gráfica, tem também que partir de alguns princípios importantes, que tornarão a experiência do utilizador mais agradável.

Entre esses princípios, destaca-se o de manter um estilo gráfico, ou de *design*, semelhante ao utilizado nos servidores que se pretende controlar. Esta propriedade torna-se mais importante quando se compreende que muitos dos futuros utilizadores da aplicação serão os mesmos que anteriormente controlavam os servidores S1000 isoladamente. Retira-se daqui que um utilizador já familiarizado com o *design* e organização do GUI terá assim uma maior facilidade de adaptação ao sistema.

É também importante, em termos de identidade, manter a coerência entre produtos e projectos da mesma família, sendo assim de mais fácil identificação, razão que também leva a adoptar este requisito como importante.

4.3 Conclusões

Sendo que os requisitos apresentados neste capítulo se apresentam como de prioridade mais elevada no desenvolvimento do projecto, é de igual relevância compreender que a natureza aberta e inovadora do projecto terão que manter algum espaço para a alteração ou introdução de novos requisitos funcionais, embora sem nunca perder de vista o objectivo primordial.

Em termos de requisitos não funcionais, e tendo em que conta que a prioridade mais elevada será para com os já descritos não é de alguma importância não descurar, sempre que se perspectivar possível, outros requisitos como segurança, escalabilidade ou testabilidade, que trarão mais-valias acrescidas ao projecto.

5 Arquitectura da Solução

Neste capítulo será descrita a arquitectura da aplicação desenvolvida. Para uma melhor compreensão da mesma, será evidenciada tanto a arquitectura física como lógica, providenciando assim uma compreensão mais detalhada do projecto.

5.1 Arquitectura Física

A arquitectura física documenta a estrutura física de alto nível do sistema, evidenciando a organização em camadas “físicas” (máquinas cliente e servidores, entre outros). É bastante útil para ter uma compreensão geral do funcionamento do sistema, precisamente de um ponto de vista “físico”, detalhando ligações entre as diversas camadas.

Neste projecto, inicia-se com o utilizador, que interage com um qualquer *browser* de navegação na rede, e podendo desde aí aceder ao GUI da aplicação, construído em *Flex*. Esta interface gráfica permite então comunicar com o servidor central do sistema, através de mensagens SOAP sobre HTTP, enviando comandos ou pedidos e recebendo, assincronamente, as respectivas respostas.

O servidor, desenvolvido em *python*, irá gerir todo o ambiente do C1000, desde iniciar e parar capturas agendadas a fazer pedidos individuais a cada S1000 controlado. O próprio tratará de reenviar os pedidos específicos para cada servidor de captura, ou para grupos destes, em simultâneo, no caso de operações sobre *gangs*.

Como foi visto, o C1000 comunica também utilizando SOAP com os diversos S1000, sendo que toda esta troca de mensagens decorre de forma assíncrona, de forma a não perturbar a fluidez do sistema, nem afectar o seu desempenho.

Embora um pouco fora do âmbito do projecto, é também de interesse perceber que cada S1000 está, por seu turno, a receber um sinal SDI de entrada, enquanto que as capturas efectuadas por cada um poderão ser gravadas num servidor *Avid*, localizado também na mesma rede.

O diagrama da arquitectura descrita pode ser visto na Figura 5.1.

Arquitectura da Solução

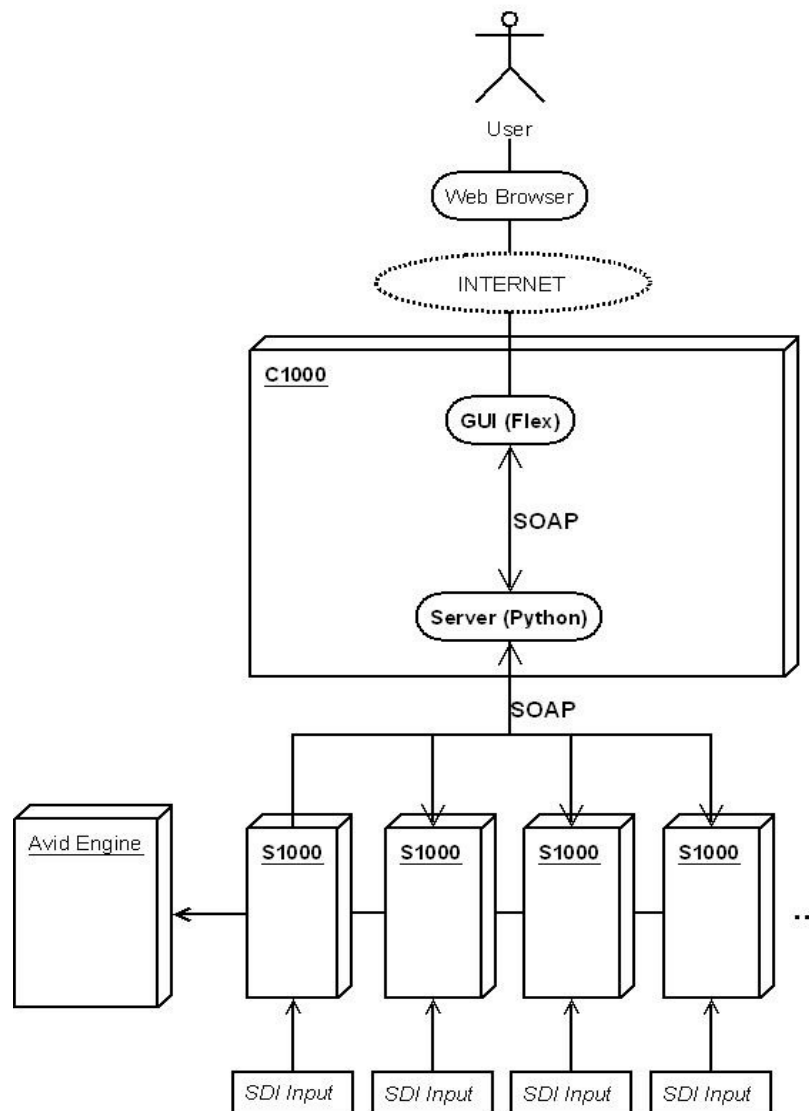


Figura 5.1: Arquitectura Física do Sistema

Optou-se, de forma a modularizar o mais possível a aplicação, por separar, dentro da possibilidade, toda a lógica de negócio da interface gráfica, sendo que toda a informação disponibilizada ao utilizador é enviada pelo servidor para a GUI através de mensagens SOAP.

5.2 Arquitectura Lógica

A arquitectura lógica mostra a organização lógica do sistema, especificando as relações entre os seus componentes. De forma a providenciar maior detalhe, a arquitectura será decomposta tanto horizontal como verticalmente.

Na decomposição horizontal é feita por camadas ou conceitos de implementação, de forma relativamente independente das funcionalidades do sistema enquanto que a decomposição vertical corresponde à divisão e organização por sub-sistemas, de forma hierárquica, em que cada um deles se refere a um grupo de funcionalidades, abrangendo todas as camadas de implementação.

5.2.1 Decomposição horizontal

O diagrama da Figura 5.2 apresenta a decomposição horizontal da arquitectura lógica do sistema.

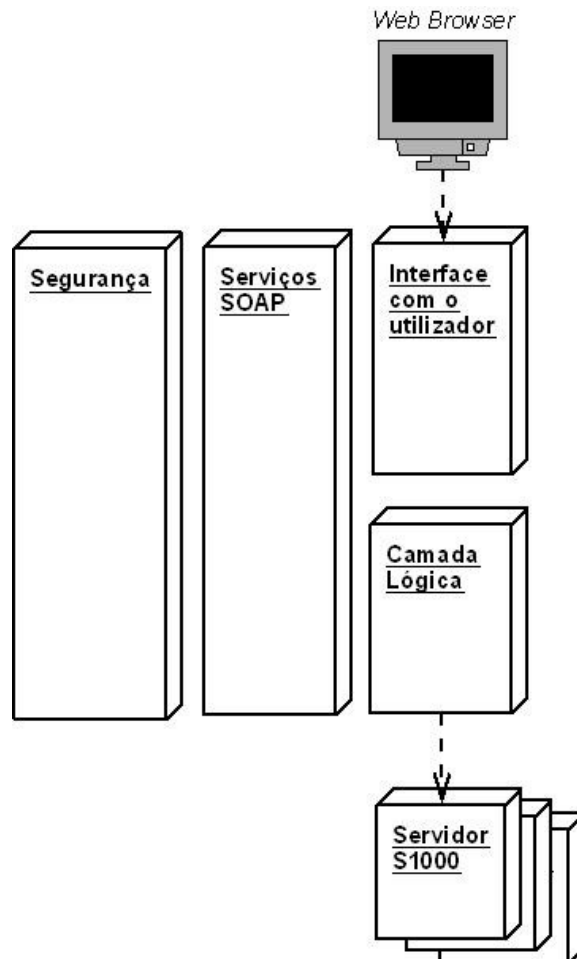


Figura 5.2: Arquitectura Lógica do Sistema (decomposição horizontal)

É facilmente perceptível que o sistema assenta sobre uma arquitectura cliente-servidor, sendo que a comum divisão em três camadas, interface, lógica e dados, apresenta neste caso uma variação, que se prende com a inexistência da camada de dados. Pode-se argumentar que na realidade a camada de dados estaria ligada aos servidores S1000, pois são eles que efectivamente irão tratar de alojar as capturas, comunicando com discos e bases de dados. Este tipo de arquitectura prevalece uma vez que o conjunto de dados pertinentes e tratados pela aplicação é, na sua maioria, obtido dos diversos servidores de captura, como o sejam configurações e informação sobre o estado de cada um.

A camada de interface representa todas as interfaces com o utilizador, que permitem o envio de pedidos à camada lógica, que tratará de os processar, reflectindo possíveis alterações novamente na interface, mas também nos servidores S1000, que representa a verdadeira camada de dados da aplicação.

É visível uma camada de segurança ao longo de toda a aplicação. Embora não seja efectivamente uma camada isolada do resto do sistema, ilustra os mecanismos de segurança

Arquitectura da Solução

implementados ao longo de todas as outras camadas, nomeadamente em relação a autenticações, permissões e validações, contribuindo assim para a fiabilidade e robustez originalmente pretendidas.

A camada de serviços SOAP apresenta-se também como parte do sistema, visto que todas as outras camadas a utilizam para comunicação. A interface comunica com a camada lógica através de mensagens SOAP, que comunica com os servidores S1000 da mesma forma. Isto permite não só manter a coerência através de todo o sistema, como também facilita a possibilidade de rápida substituição dos mecanismos de comunicação sem afectar o resto da aplicação, promovendo assim, mais uma vez, o requisito de modularidade.

5.2.2 Decomposição vertical

Outro tipo de decomposição arquitectural é a vertical, detalhada em seguida, no diagrama da Figura 5.3.

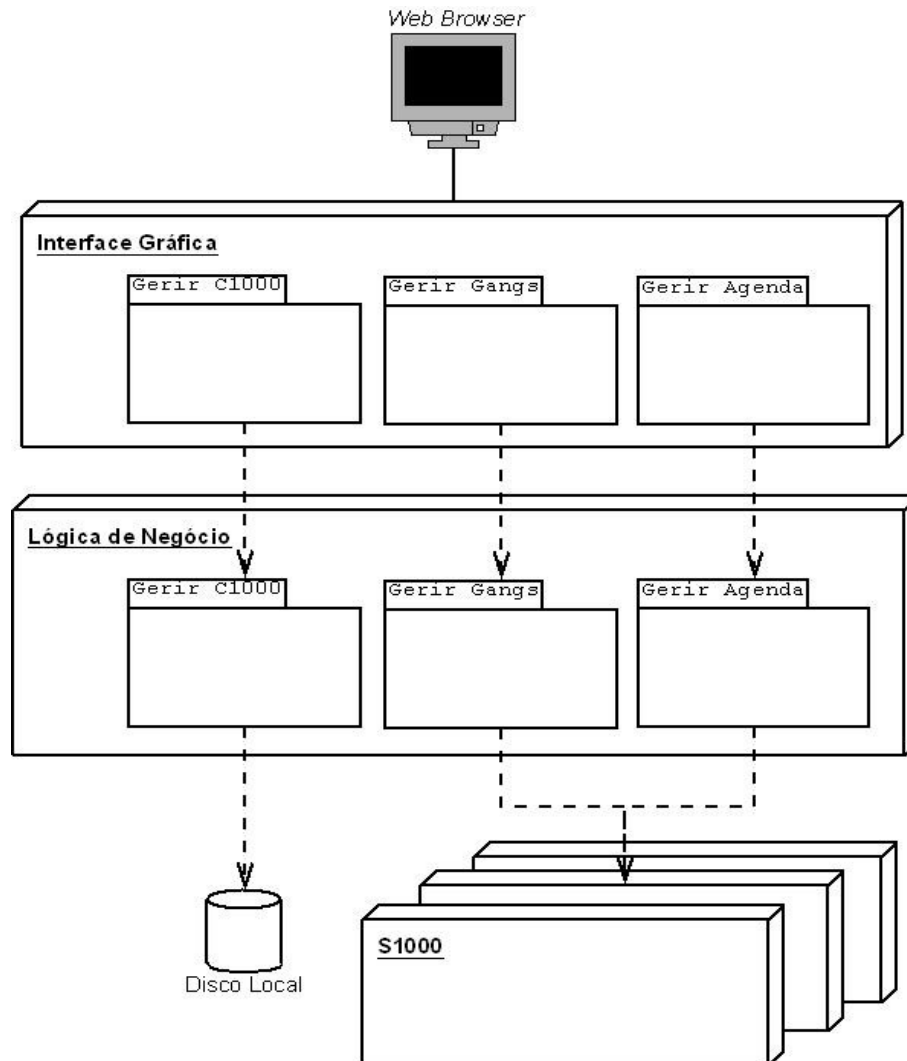


Figura 5.3: Arquitectura Lógica do Sistema (decomposição vertical)

Arquitectura da Solução

No nível mais elevado de uma decomposição vertical do sistema, pode verificar-se a separação da aplicação por processos, como o são a gestão da agenda, dos dispositivos e seus agrupamentos e, de forma mais ténue, de alguns processos internos da própria aplicação.

A gestão da agenda inclui todos os processos relativos a gestão de eventos, bem como monitorização dos mesmos, confirmando que são iniciados e finalizados de acordo com o programado anteriormente. Na Figura 5.4 é detalhada a arquitectura deste sub-sistema, ao longo de todas as camadas da aplicação.

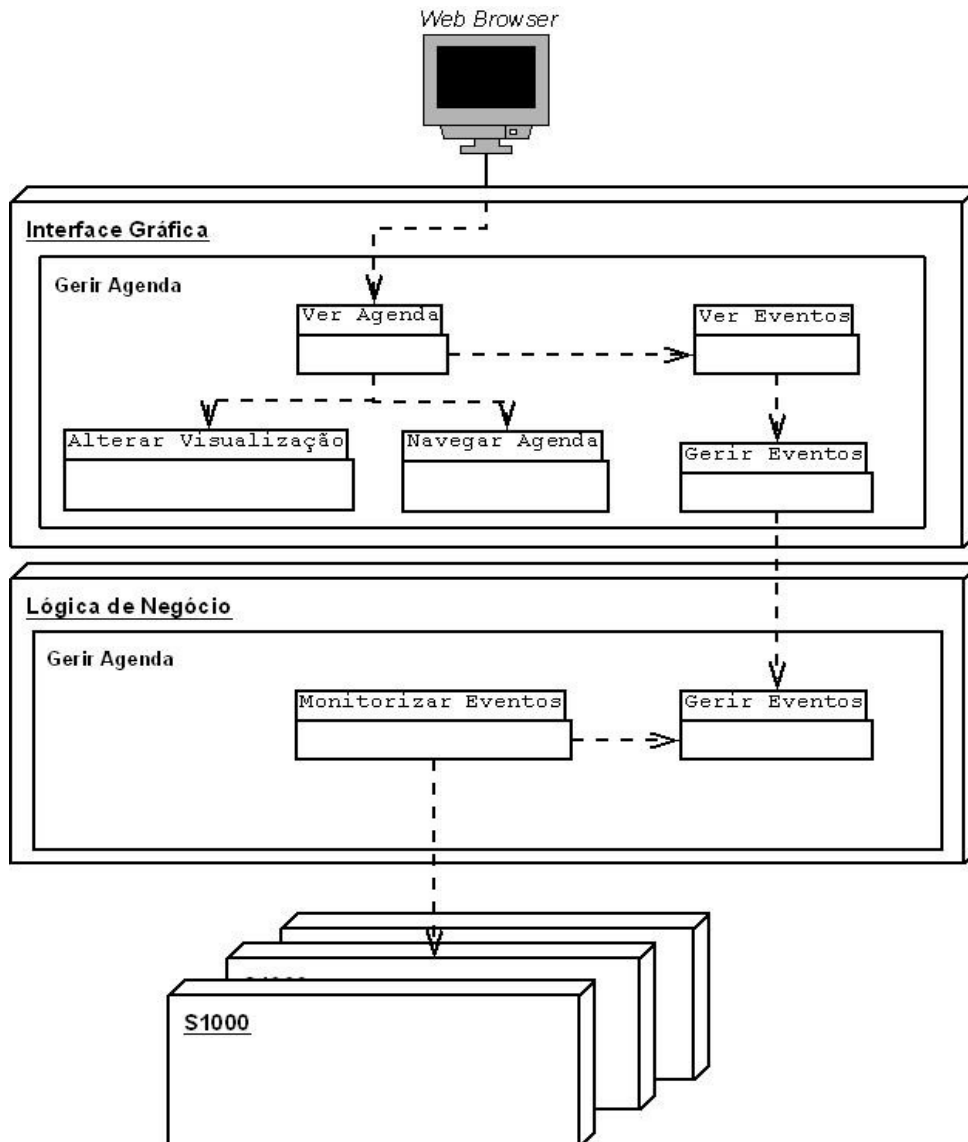


Figura 5.4: Decomposição vertical da arquitectura: sub-sistema de gestão de agenda

Pode ver-se que existe uma parte puramente visual, de visualização e navegação da agenda. Em paralelo, persiste o processo de gestão de eventos, que atravessa para a camada lógica. Nesta camada são também monitorizados estes eventos, de modo a proceder ao envio

Arquitectura da Solução

dos comandos de início ou paragem de capturas, nos dispositivos correspondentes, atempadamente.

A gestão de *gangs* refere-se ao tipo de operações efectuadas sobre os servidores S1000, desde configurações individuais e de grupo, à criação, remoção e alteração desses grupos ou dispositivos individuais. A arquitectura deste sub-sistema encontra-se detalhada na Figura 5.5.

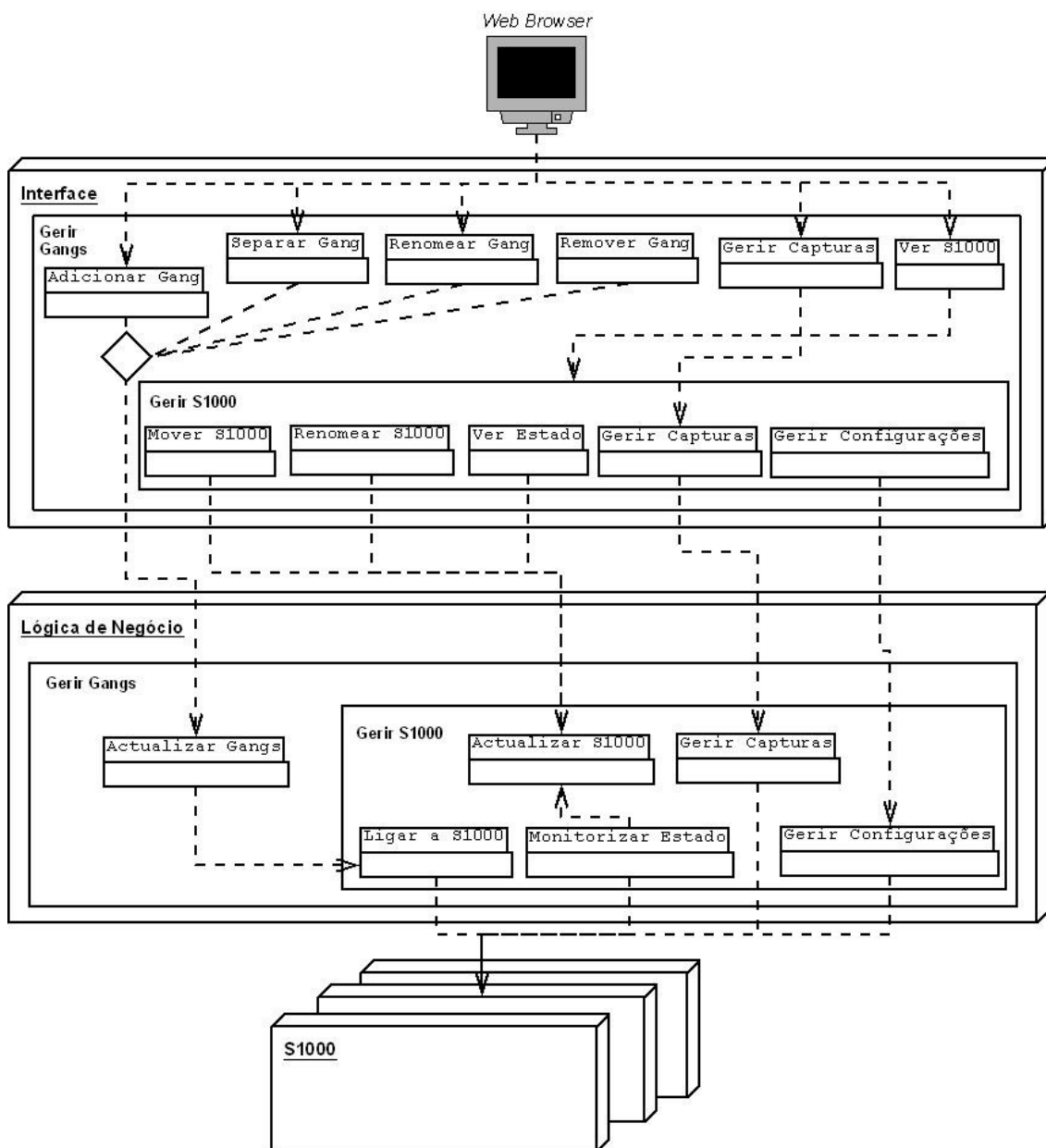


Figura 5.5: Decomposição vertical da arquitectura: sub-sistema de gestão de *gangs*

As funcionalidades relativas a gestão de *gangs* e S1000s são acedidas na interface, através do recurso a um *browser*. Desde adicionar e remover *gangs* a iniciar e parar capturas, individuais ou de grupo, tudo se inicia nesta camada, passando depois para a camada lógica.

Na camada lógica, os *gangs* e dispositivos individuais são actualizados conforme as alterações pedidas pela interface. Existe também um processo que monitoriza o estado de cada

Arquitectura da Solução

S1000, que obviamente será parte relevante do estado de cada *gang*, actualizando a interface com essa informação. Para além deste, todos os outros processos relacionadas especificamente com o controlo dos servidores de captura, comunicam com o respectivo S1000, como visto anteriormente, através de mensagens SOAP.

Finalmente, a gestão do C1000, ou seja, da própria aplicação, relaciona-se apenas com configurações internas específicas do sistema, como autenticação e gestão de *passwords* de utilizadores, ou mesmo a frequência de actualizações de estado e número máximo de unidades S1000 suportadas. Todas estas configurações são guardadas e carregadas localmente de ficheiros XML.

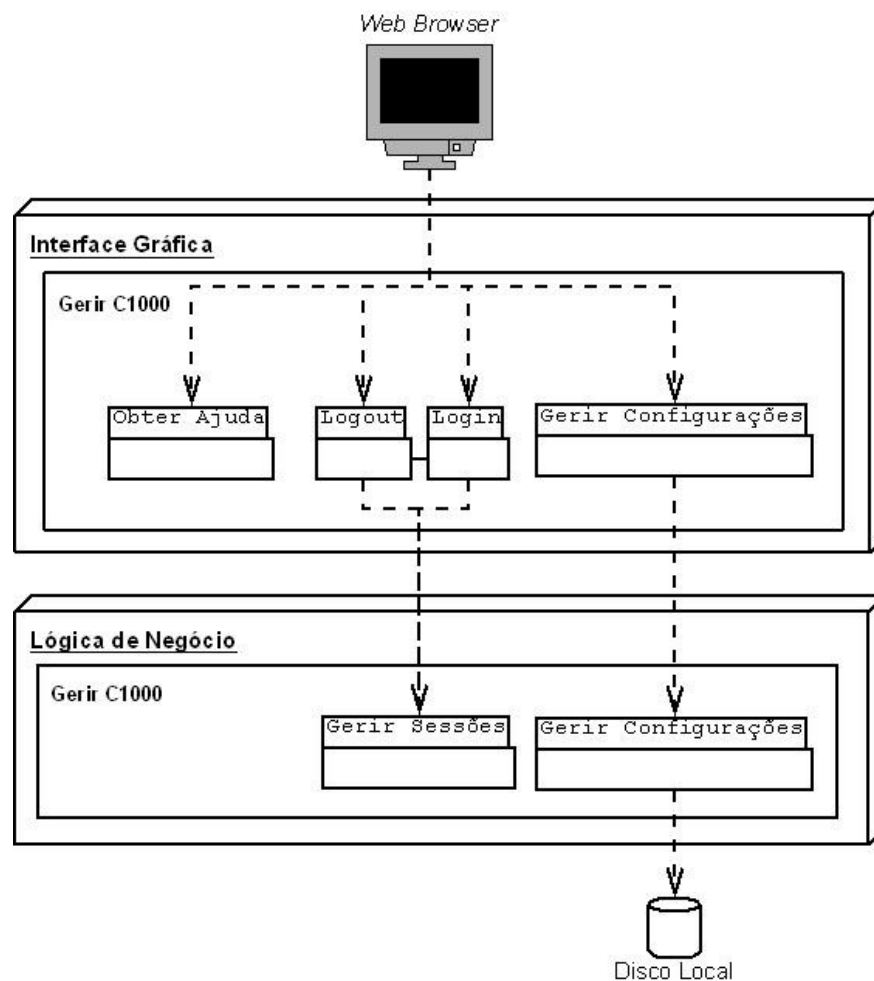


Figura 5.6: Decomposição vertical da arquitectura: sub-sistema de gestão interna

O sub-sistema relacionado com a gestão interna de configurações do próprio sistema é, de alguma forma, mais simples que os anteriores, como aparenta a sua arquitectura. Mais uma vez, a camada de interface permite o acesso inicial aos processos, como gestão de configurações e autenticação, que atravessarão para a lógica de negócio. As configurações são lidas e guardadas no disco local ao servidor, sob a forma de ficheiros XML.

5.3 Conclusões

Este capítulo detalhou a arquitectura da aplicação, esclarecendo assim de uma melhor forma, a implementação da separação por camadas.

Verifica-se assim o isolamento total entre as camadas lógica e de interface, bem como a inexistência de uma verdadeira camada de dados, atendendo às particularidades do sistema, como o seja o facto da grande maioria dos dados que pelo mesmo transitam terem a sua proveniência de sistemas externos, como o *mxSPEEDRAIL S1000*.

6 Desenvolvimento do Protótipo

Este capítulo tem como objectivo descrever de um modo mais aprofundado os aspectos mais relevantes relacionados com a implementação e desenvolvimento do projecto.

Sem entrar em temas demasiado específicos ou de programação, será aqui apresentada a implementação da solução que permitiu a criação de um produto compatível com os requisitos pretendidos, e seguindo a arquitectura especificada no capítulo anterior.

Para além dos detalhes de implementação mais relevantes apresenta-se também o resultado final obtido, sendo para este efeito utilizada uma perspectiva de utilizador para uma melhor compreensão destes resultados. Após os capítulos anteriores, em que se discutiam aspectos mais abstractos do projecto, pretende-se agora possibilitar uma noção mais prática e objectiva do teor do mesmo e do produto que originou.

O protótipo resultante do projecto abarca todos os requisitos funcionais especificados inicialmente, bem como os não funcionais, se bem que neste aspecto a avaliação e medição seja necessariamente mais abstracta.

6.1 Visão Geral da Aplicação

Para aceder à aplicação é necessário que o sistema autentique o utilizador, associando-lhe uma sessão. Tal como para a grande maioria das acções, o utilizador introduz um *username* e uma *password* na interface, vista na Figura 6.1, que enviará a informação via SOAP, para que o servidor possa criar a nova sessão, ou recusar o pedido em caso de incorrecção.



Figura 6.1: Ecrã de autenticação do protótipo

Desenvolvimento do Protótipo

Estas sessões têm a particularidade de expirar ao fim de determinado período de tempo, caso não existam pedidos ou acções a elas associadas, impedindo assim o seu avolumar ao longo do tempo, visto que se supõe um servidor em funcionamento contínuo.

Embora não exista a noção de diferentes utilizadores, existe sim o conceito de perfis de utilizador. Neste caso apenas dois, o perfil de *Administrator* ou *Ingestor*. Isto significa que todos os utilizadores terão que se autenticar com um dos dois perfis.

A única diferença efectiva entre os dois perfis é a de alteração de configurações. Enquanto que um *Ingestor* pode apenas gerir capturas, um *Administrator* pode também alterar todo o tipo de configurações assim como editar a agenda.

Uma vez autenticado, o utilizador tem acesso ao ecrã principal da aplicação, a partir do qual se pode aceder a todas as funcionalidades disponibilizadas, ou pelo menos às que o seu perfil lhe permita. Este ecrã divide-se em duas áreas principais. Uma barra de opções no canto superior direito, evidenciada na Figura 6.2, onde é possível aceder a funcionalidades gerais do sistema, como ligação a novos servidores, visualização da agenda ou fazer o *logout* do sistema.



Figura 6.2: Barra de acesso às funcionalidades gerais da aplicação

Uma das funcionalidades obtida através deste menu é a de alteração de *passwords*, acessível a partir das configurações gerais, e que permite alterar a password de acesso ao sistema para qualquer dos perfis já descritos. Este menu é visível na Figura 6.3, e permite alterar qualquer password de acesso ao sistema, apenas a administradores, obviamente, aumentando assim a segurança do mesmo.

A imagem mostra uma janela de configuração intitulada 'C1000 Settings'. Ela contém campos de texto para 'Admin Password', 'New password' e 'Repeat Password'. Abaixo de 'Admin Password', há uma opção 'Password to change' com dois botões de rádio: 'Ingestor' (selecionado) e 'Administrator'. Na base da janela, há dois botões: um verde com um símbolo de confirmação e um cinza com um símbolo de cancelamento.

Figura 6.3: Menu para alteração de *passwords* para acesso ao sistema

Desenvolvimento do Protótipo

Outra opção disponível no menu é a de obter ajuda, que disponibiliza uma janela com a descrição das principais funcionalidades do sistema e o seu modo de funcionamento, por forma a quando a interface não se apresentar suficientemente intuitiva, o utilizador possa recorrer à mesma como forma de ultrapassar possíveis dúvidas. Esta janela pode ver-se na Figura 6.4.

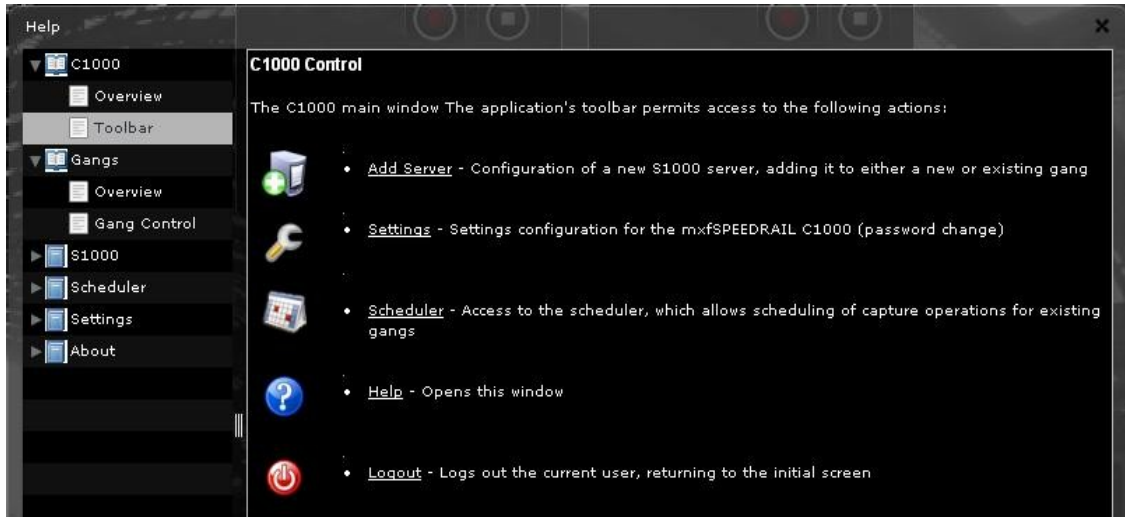


Figura 6.4: Menu para alteração de *passwords* para acesso ao sistema

As funcionalidades mais relevantes da aplicação prendem-se, no entanto, com a visualização dos grupos de S1000 actualmente controlados pelo *software*, os *gangs*. Na Figura 6.5 pode ver-se um exemplo de um cenário em que existem dois *gangs* no sistema.



Figura 6.5: Ecrã principal, evidenciando as janelas representativas de *gangs* no sistema

6.2 Gestão de Gangs

No cenário patente na Figura 6.5, verifica-se que existem dois *gangs*, denominados “Gang 1” e “Gang 2”, sendo que o primeiro apresenta algum problema, evidenciado pelo ícone de estado, enquanto que o segundo se apresenta operacional.

Para além do estado, cada representação de um *gang* contém na sua interface botões relacionados com a gestão de capturas, do próprio *gang* e dos seus membros (servidores S1000).

É então possível iniciar e parar capturas para todo o grupo de S1000 respeitante ao *gang*. Estas operações passam por mandar ao servidor do C1000 um pedido de início de captura que será reencaminhado para cada S1000 pertencente ao *gang* respectivo, de forma simultânea, sendo todo o processo executado através de mensagens SOAP, tal como nos casos anteriores.

Relativamente a opções sobre *gangs*, é possível efectuar diversas operações sobre os mesmos. Estas operações afectam muitas vezes, simultaneamente, mais do que um *gang*, mesmo que de forma opcional. Para além desse facto, é permitido ao utilizador aceder às funcionalidades a partir de um menu contextual ou de teclas de atalho. De forma a possibilitar à interface o conhecimento de quais os *gangs* afectados pela operação seleccionada, é necessário permitir ao utilizador seleccionar os mesmos. Para este efeito, foram aplicados mecanismos para que essa experiência seja o mais intuitiva possível, permitindo uma selecção de qualquer aplicação que trabalhe com o conceito de janelas. Pode-se então seleccionar os *gangs* com o rato, de forma independente ou arrastando-o sobre o grupo pretendido, como visto na Figura 6.6.



Figura 6.6: Selecção de grupos de *gangs*, obtida arrastando o rato no ecrã

Todas as funcionalidades relacionadas com gestão de *gangs*, apresentam-se disponíveis apenas quando se encontra seleccionado o número correcto dos mesmos, sendo o facto

Desenvolvimento do Protótipo

particularmente verdade no que se refere às opções acessíveis ao menu contextual, do qual pode ser visto um exemplo na Figura 6.7.



Figura 6.7: Exemplo de menu contextual de relativo a um, ou vários, *gangs*

Todas as operações sobre *gangs* tratam-se, em termos gerais, de mecanismos que facilitam o agrupamento de servidores S1000. As funcionalidades disponibilizadas facilitam a gestão destes dispositivos, que se apresenta como o verdadeiro cerne da aplicação.

Uma das opções essenciais do sistema, particularmente nas fases iniciais da sua utilização, é a de adicionar novos servidores de captura, para que possam ser controlados e monitorizados remotamente, o principal objectivo do projecto. Embora não seja possível criar um novo *gang* directamente, uma vez que a existência dos mesmos implica que contenham pelo menos um S1000, é sim possível fazê-lo na ligação a um novo destes dispositivos.

A screenshot of a dialog box titled 'Add New S1000'. The dialog box has a dark background and contains several input fields and buttons. The fields are: 'S1000 Alias' with the value 's1000_1', 'S1000 IP' with the value '192.168.1.126', 'username' with the value 'username', 'password' with the value '*****', and 'Gang' with the value 'Gang 2'. At the bottom of the dialog box, there are two buttons: a green checkmark button and a grey 'X' button.

Figura 6.8: Menu de adição de novos dispositivos *mxSPEEDRAIL* S1000

Desenvolvimento do Protótipo

Acedendo ao menu de adição de servidores S1000, reproduzido na Figura 6.8, é possível introduzir um nome para a nova ligação, bem como um *username* e *password* para acesso ao dito servidor que, recorde-se, é um equipamento independente e que funciona isoladamente. Para além disto, é também permitida a escolha entre juntar o novo servidor a um *gang* já existente ou criar um novo que o contenha. O campo restante, e essencial à conclusão da operação, é o do seu endereço de rede. Este endereço pode ser introduzido manualmente, caso seja conhecido, ou seleccionado de uma lista contendo os servidores S1000 detectados.

Esta detecção é efectuada utilizando o protocolo de descoberta de serviços descrito no capítulo 2, o *Bonjour*. Os servidores de captura anunciam o serviço disponibilizado, neste caso a sua própria existência, utilizando o protocolo, que especifica o endereço, nome e porto onde se encontram, enquanto que o sistema também fará uso do mesmo, para os descobrir, preenchendo assim a lista de endereços de S1000 encontrados. Esta lista é editável uma vez que podem existir versões do *mxfSPEEDRAIL* S1000 sem esta funcionalidade, que foi implementada nos mesmos em paralelo com este projecto.

A aplicação tentará, assim que todos os campos tenham sido preenchidos, conectar-se ao S1000, utilizando as configurações dadas. Caso seja bem sucedida no contacto e autenticação com o servidor em questão, o novo S1000 será adicionado à interface, quer na forma de um novo *gang* ou na adição de um novo membro a um *gang* existente. A partir deste momento, o seu estado será continuamente monitorizado, podendo ser controlado a partir do sistema.

Tal como foi explicado, existem diversas operações que podem ser efectuadas sobre os *gangs*. É possível renomear um *gang*, uma forma simples de reorganizar a estrutura de grupos utilizada. Remover um *gang* corta efectivamente o acesso a todos os S1000 em si contidos, deixando de os monitorizar. Estas funcionalidade alteram reflectem-se também na agenda, de modo a garantir a sua fiabilidade. As duas operações restantes são, de alguma forma, opostas. É então possível juntar dois *gangs* (ou até mais), seleccionando-os primeiro, o que fará com que os S1000 contidos em ambos sejam agrupados num mesmo *gang*. A operação oposta será então a de separar *gangs*, que faz com que todos os servidores de captura de determinado grupo se movam para o seu próprio *gang* individual, permitindo novamente o seu controlo isolado.

Toda esta configuração estrutural e organizacional, de grupos e seus membros, é guardada num ficheiro XML, permitindo assim ao utilizador recuperá-la em qualquer altura, ao restabelecer a ligação ao servidor. Um exemplo de um destes ficheiros, e respectiva configuração, pode ser visto na Figura 6.9.

```
- <settings>
- <ganglist>
- <gang name="verdadeiro">
  <s1000 alias="S1000 remoto" password="21232f297a57a5a743894a0e4a801fc3"
    url="http://192.168.1.216/ws/soap/api.wsdl" username="admin" />
  </gang>
- <gang name="falso">
  <s1000 alias="S1000 local" password="21232f297a57a5a743894a0e4a801fc3"
    url="http://192.168.1.126/ws/soap/api.wsdl" username="admin" />
  </gang>
</ganglist>
- <users>
  <admin>21232f297a57a5a743894a0e4a801fc3</admin>
  <ingester>d4408643ccbd7e83d0c54f42e405d618</ingester>
</users>
<customvalues />
</settings>
```

Figura 6.9: Exemplo do ficheiro XML de configuração de *gangs* da aplicação

Desenvolvimento do Protótipo

Uma vez que cada *gang* contém, possivelmente, diversos servidores S1000, cujo comportamento é individual e independente dos restantes, poderia tornar-se difícil apresentar um estado geral para o mesmo, com vista a uma rápida monitorização da situação. A aplicação apresenta assim, no canto superior direito de cada grupo, um estado que pretende representar, dentro do possível, a situação conjunta dos seus membros. São então possíveis 4 tipos diferentes de estados, vistos na figura 6.10, que representam as quatro combinações possíveis da situação em que se pode encontrar um *gang*:

- em captura (*all recording*), caso todos os servidores estejam a capturar em simultâneo;
- em espera (*all waiting*), caso todos os servidores estejam, no momento, a aguardar;
- em captura parcial (*some recording*), caso por algum motivo, nem todos se encontrem a capturar;
- com problemas (*problems*), caso algum dos servidores apresente algum erro, de ligação ou qualquer outro tipo.

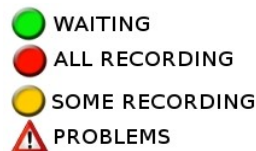


Figura 6.10: Lista de possíveis estados, e respectiva visualização, de um gang

Como se poderá verificar, existem casos em que a simples visualização destes estados não permitirá esclarecer por completo a situação de cada S1000 envolvido. Para estes casos, existe a possibilidade, na interface de cada *gang*, de visualizar o estado de cada um deles individualmente, permitindo assim a fácil identificação de possíveis problemas ou casos específicos. Esta listagem, que pode ser vista na Figura 6.11, apresenta o estado específico de cada S1000, que neste caso pode também ser um de quatro:

- em captura (*recording*), caso esteja a capturar;
- em espera (*waiting*), caso esteja a aguardar;
- desligado (*disconnected*), caso não seja possível aceder-lhe;
- com problemas (*problems*), caso apresente algum outro tipo de erro.

No caso apresentado no exemplo da Figura 6.11 podemos ver que um dos servidores se encontra em espera enquanto que o outro está desligado. A situação reflecte-se no *gang*, que apresenta o ícone correspondente à existência de problemas.

Após a apresentação das funcionalidades relativas a *gangs*, o próximo passo será a descrição do controlo e monitorização efectivos de cada S1000 individualmente.



Figura 6.11: Exemplo da visualização de estados dos servidores S1000

6.3 Visualização por Gang

Embora a gestão de *gangs* tenha a sua utilidade, na maioria dos casos será mais interessante para um operador do sistema, a possibilidade de gerir um determinado *gang*, que controlará diversos servidores S1000. Esta situação exige um interface que permita rapidamente providenciar acesso à informação básica de cada servidor, bem como do estado geral do grupo. Na Figura 6.12 podemos ver o aspecto deste nível de visualização.



Figura 6.12: Exemplo da interface de visualização de um único *gang*, composto por dois servidores S1000

Com este modo de visualização, é possível ter acesso à informação de cada S1000, ou apenas dos desejados, sendo possível mostrar apenas as janelas pretendidas, adaptando assim a interface ao gosto do utilizador. Para além de janelas correspondentes a cada um dos servidores controlados, existem também outras que permitem a gestão do gang, possibilitando algumas definições que serão aplicadas a cada um deles, como etiquetas para nomenclatura de ficheiros ou selecção das directorias de destino dos ficheiros criados.

6.4 Gestão dos dispositivos S1000

A interface individual de cada S1000 permite, basicamente, o mesmo tipo de funcionalidades que se poderiam obter através do GUI do mesmo. Está dividida em duas secções, como se pode ver através da análise da Figura 6.13.



Figura 6.13: Janela para controlo remoto de um servidor S1000

A informação e detalhes exibidos na janela passa por, além do estado básico, como visto anteriormente para cada *gang*, poder consultar o *timecode* actual, que será um código temporal associado ao sinal SDI de entrada do S1000.

Caso o servidor se encontre a capturar vídeo, serão visíveis também o *timecode* em que a captura se iniciou e a duração actual da mesma, bem como o nome do ficheiro que se encontra em geração, que será útil para facilitar a posterior identificação do mesmo.

Desenvolvimento do Protótipo

No local oposto a esta informação, encontra-se o menu do dispositivo. Aqui é possível invocar comandos remotamente, cujas funcionalidades apresentam diversas semelhanças com as disponibilizadas no menu relativo aos *gangs*.

É então possível renomear ou remover o dispositivo ou até movê-lo para um outro grupo, novo ou já existente, separando-o do *gang* corrente. Também será possível, através desta interface, iniciar e parar capturas no servidor, independentemente do estado do geral do *gang*. Será, no entanto, necessária alguma atenção de modo a não influenciar comandos anteriores dados a todo o agrupamento, ou mesmo capturas automáticas iniciadas via agenda.

No caso da ligação ao dispositivo ter sido previamente interrompida, como o exemplo da Figura 6.10 mostra, é também possível tentar restabelecê-la, através do GUI.

Outra funcionalidade interessante é a que possibilita bloquear ou desbloquear o servidor de captura, em relação ao seu *gang*. Qualquer ordem dada a um *gang* completo (iniciar/parar capturas, configurações, etc) será apenas enviada para os servidores bloqueados, sendo assim extremamente rápido e simples excluir qualquer servidor de alguma ordem em particular, sem ter de o remover do grupo.

É também permitida a configuração remota do servidor de captura. Existem dois grupos de configurações a que se pode aceder remotamente, que passam pela definição de opções de *filenaming* ou formatos e localizações para captura.

A estratégia de *filenaming* prende-se com a forma como é escolhida a nomenclatura dos ficheiros gerados. Este passo torna-se bastante importante, se pensarmos que num dado ambiente de produção poderão existir milhares de ficheiros gerados pelo sistema, cuja identificação deverá ser o mais eficiente e simples possível. As opções de configuração disponibilizadas, apresentadas na Figura 6.14, dão conta da variedade de estratégias possíveis.

Esta configuração iniciará com a escolha de que partes entrarão na composição do nome de futuros ficheiros. Pode optar-se por utilizar qualquer combinação destas, nas quais se encontram: um número identificador, uma sequência de caracteres, um rótulo personalizado ou a data (e hora) actuais.

Após a escolha das partes a utilizar, é ainda possível configurar como serão utilizadas, seja a especificação do rótulo ou sequência de letras a adicionar ao nome, ou mesmo o formato e campos da data utilizáveis.

O ecrã de configuração mostra também alguma informação útil na escolha desta estratégia. No topo da janela pode ver-se um exemplo do tipo de nomes de ficheiros que serão gerados com a estratégia corrente. Este exemplo é requisitado ao servidor quando existe alguma alteração nesta configuração. A outra informação é o número de nomes distintos que podem ser gerados com a estratégia definida. Isto permite ao utilizador antecipar repetições no caso de utilizar a mesma estratégia por longos períodos e, conseqüentemente, em múltiplas capturas distintas.

É também possível adicionar um prefixo a esta configuração de nomenclatura, específico para cada *gang*, tornando a procura dos ficheiros a gerar significativamente mais simples.

Desenvolvimento do Protótipo



Figura 6.14: Opções de configuração da estratégia de *filenaming* de um S1000

O outro agrupamento de configurações relaciona-se com os formatos e localizações para as capturas a efectuar no futuro. Tal como no exemplo anterior, também estas configurações podem ser alteradas na janela respectiva do menu de qualquer dispositivo S1000, que pode ser vista na Figura 6.15.

Neste menu podem configurar-se os critérios de paragem de captura para o servidor, como o sejam erros de sinal ou perda de *frames*, ou o número de canais de áudio. No entanto, as configurações mais relevantes prendem-se com os formatos de *input* e de *output*, cuja correcção é essencial a uma captura bem sucedida. Pode escolher-se estes formatos de entre uma lista pedida ao servidor, pois cada S1000 poderá ter opções distintas devido a factores que este documento não explorará, por serem fora do âmbito do projecto. Para além de formatos de *input* e *output*, pode optar-se por gerar um *proxy* de baixa resolução da captura, em simultâneo com a geração do ficheiro de alta resolução, seleccionando também o seu formato da lista disponibilizada.

A segunda parte desta configuração prende-se com os locais de armazenamento dos ficheiros. É possível escolher o tipo de armazenamento, entre alguns populares no mundo da produção multimédia. Para o tipo de armazenamento seleccionado, pode depois definir-se o local do armazenamento, bem como possíveis parâmetros de autenticação do mesmo.

Desenvolvimento do Protótipo



Figura 6.15: Opções de configuração de capturas para um S1000

Embora a maioria destas configurações não seja algo recorrentemente utilizado, o caminho específico onde irão ser guardados os ficheiros já o pode ser. Por este facto, é também possível, no próprio *gang*, seleccionar o caminho ou directório (numa estrutura em árvore) para onde se pretende capturar o sinal de vídeo, aplicando a configuração a todos os elementos do grupo, como pode ser visto na Figura 6.16.



Figura 6.16: Opções de configuração de capturas para um S1000

No âmbito deste projecto, foi adicionada às próprias estações S1000, a possibilidade de adicionar metadados a cada captura. Este componente, visível na Figura 6.17, permite criar e gerir perfis de metadados, cada um com os campos que o utilizador pretenda, associando-os depois ao ficheiro MXF gerado.



Figura 6.17: Gestão de perfis de metadados

6.5 Gestão da Agenda

Para além da configuração e controlo directo de *gangs* e servidores de captura, a outra vertente com maior relevância da aplicação prende-se com a agenda que disponibiliza. Esta agenda permite agendar capturas automáticas em determinada data para *gangs* específicos, permitindo à aplicação libertar a atenção do utilizador para outras tarefas.

Nesta agenda é disponibilizada a possibilidade de organizar a visibilidade temporal de acordo com as suas preferências. No menu apresentado na Figura 6.18, pode seleccionar-se o tipo de vista pretendida entre as opções de visualização disponíveis: mensal, semanal ou diária.



Figura 6.18: Opções de visualização da agenda

Cada opção não mais fará que redefinir a data inicial e final apresentadas ao utilizador, alterando a interface de acordo com o especificado. A vista utilizada por defeito será a mensal, visível na Figura 6.19.

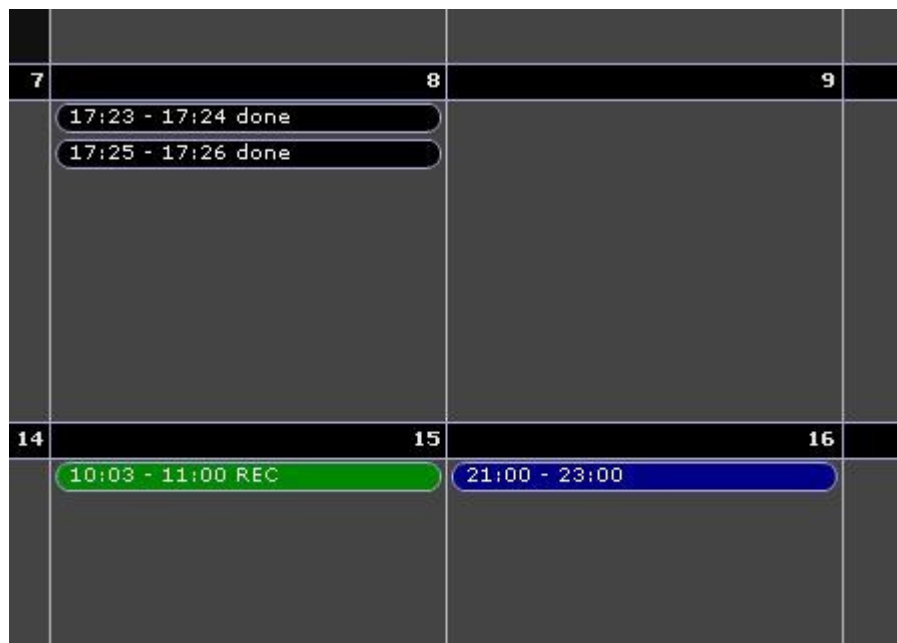


Figura 6.19: Vista mensal (parcial) da agenda

Este tipo de vista apresenta todos os dias de um determinado mês, mostrando ao utilizador a distribuição de eventos ao longo do mesmo, providenciando-lhe assim uma panorâmica geral da situação. Estes eventos aparecerão, ordenados temporalmente, informando qual a hora programada para o seu início e o *gang* a que dizem respeito.

Se for necessário utilizar um tipo de visualização mais aproximada e detalhada, pode ser aplicada a vista diária, evidenciada na Figura 6.20, onde já é possível visualizar a estrutura de eventos agendados ao longo do dia, correlacionando-os com a coluna onde são mostradas as horas.

É também mais fácil verificar, nesta vista, a existência de eventos paralelos e horas livres para cada *gang*, ou dia.

Finalmente, para um equilíbrio entre a panorâmica geral da visão mensal e a detalhe de uma visão diária, existe a visão semanal, que mostra vários dias numa perspectiva similar à da visão diária, obtendo-se assim algumas das vantagens de ambos os métodos. Um exemplo desta vista é apresentado na Figura 6.21.

Desenvolvimento do Protótipo

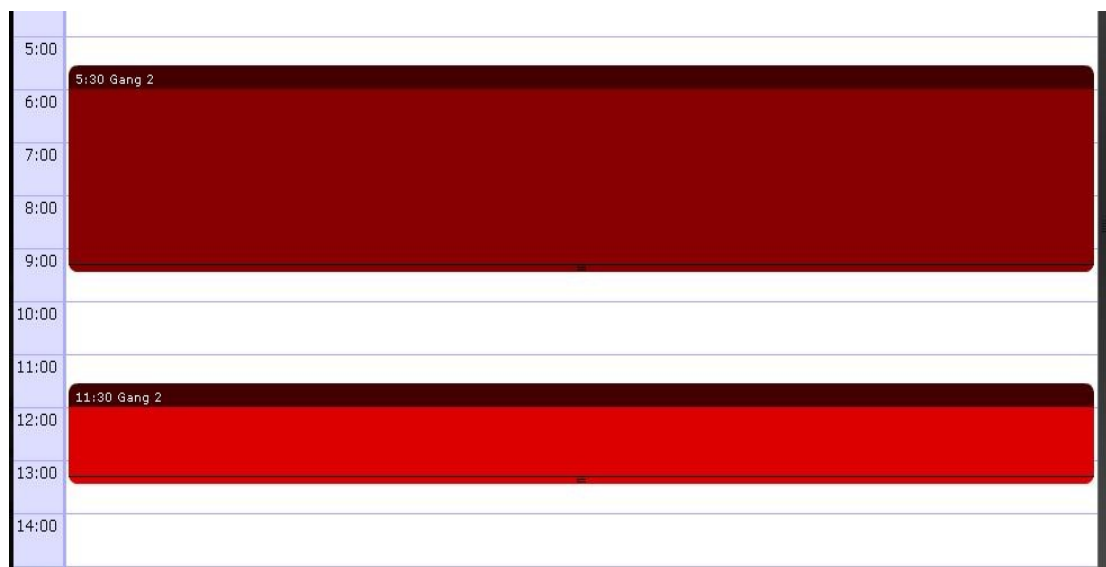


Figura 6.20: Vista diária (parcial) da agenda

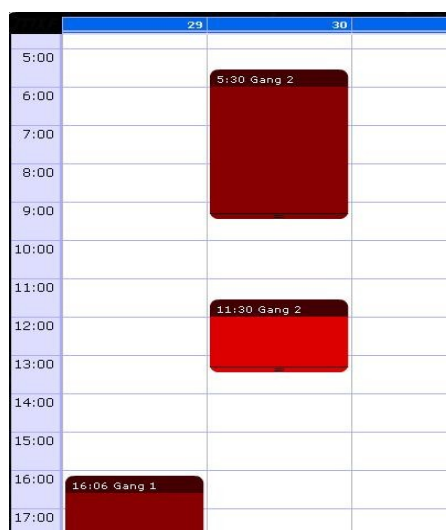


Figura 6.21: Vista semanal (parcial) da agenda

Obviamente que, por si só, estas visualizações seriam ineficazes. Também existe um outro menu que permite ao utilizador navegarem entre datas, quer se trate de meses, semanas ou dias, dependendo da vista seleccionada. Este menu pode ser visto na Figura 6.22 e possibilita avançar, recuar ou ir directamente para a data actual ou algum dia específico, e até seleccionar conjuntos de dias para visualização.



Figura 6.22: Menu de navegação na agenda

Uma vez descrita a navegação na agenda, bem como os seus modos de visualização, resta detalhar o funcionamento dos eventos, que é de facto, a parte mais relevante do processo.

Como seria de esperar, é permitida a criação, edição e remoção de eventos na agenda, a qualquer momento. Tal como para outras funcionalidades da aplicação, estas acções podem ser acedidas através de teclas de atalho, do menu contextual, ou de um segundo menu, visto na Figura 6.23, inserido na interface da agenda. Qualquer destas opções permite acesso às três funcionalidades básicas, que serão em seguida descritas.



Figura 6.23: Menu de controlo de eventos

Para criar um novo evento, é requisitado ao utilizador o preenchimento da sua data e hora de início e fim, bem como o *gang* a que diz respeito e a sua periodicidade.

Só poderão ser adicionados eventos cuja data e hora de início ainda não tenha passado e que, obviamente, tenham uma data e hora de fim superior a essa. Quanto ao *gang*, poderá ser seleccionado de uma lista contendo todos os existentes no sistema.

Finalmente, a periodicidade do evento, permite adicionar simultaneamente, repetições de eventos segundo determinado padrão. Pode adicionar-se o evento apenas na data seleccionada, mas também indicar a sua repetição todos os dias, semanas, meses ou anos (na hora introduzida) até uma outra determinada data. Caso todos os campos sejam válidos, é então adicionado o evento à agenda. A janela que permite esta criação encontra-se na Figura 6.24.



Figura 6.24: Janela para criação de um novo evento na agenda

Para além da criação de novos eventos, é também necessário providenciar acesso à sua remoção ou edição.

Editar um evento já existente pode ser conseguido de formas diferentes. Por um lado, é possível abrir uma janela para a edição do evento, semelhante à de criação de um novo, seleccionando aí a nova data de início e fim, que serão novamente validadas, bem como o *gang* a que o evento se referirá. É depois possível seleccionar entre aplicar a edição a todos os eventos dessa série ou unicamente ao evento seleccionado. Esta janela de edição pode ser vista na Figura 6.25.



Figura 6.25: Janela para edição de um evento existente na agenda

Convém também referir que qualquer edição ou criação de eventos só será bem sucedida caso as alterações não tenham como resultado uma agenda com eventos sobrepostos. Sobreposição de eventos refere-se apenas a eventos com alguma fracção da data e hora em comum, e referentes ao mesmo *gang*. Caso dois eventos estejam sobrepostos mas sejam referentes a *gangs* distintos, a alteração será permitida.

O outro método existente para editar um evento, mais directo embora não tão preciso, é o de arrastá-lo, na própria interface, para a data desejada. Um exemplo desta situação pode ser visto na Figura 6.26. Esta acção é possível em qualquer das vistas disponíveis, sendo que utilizando a diária ou semanal é também permitido aumentar ou diminuir a duração do evento, tornando assim a experiência do utilizador mais intuitiva e fluída.

Desenvolvimento do Protótipo

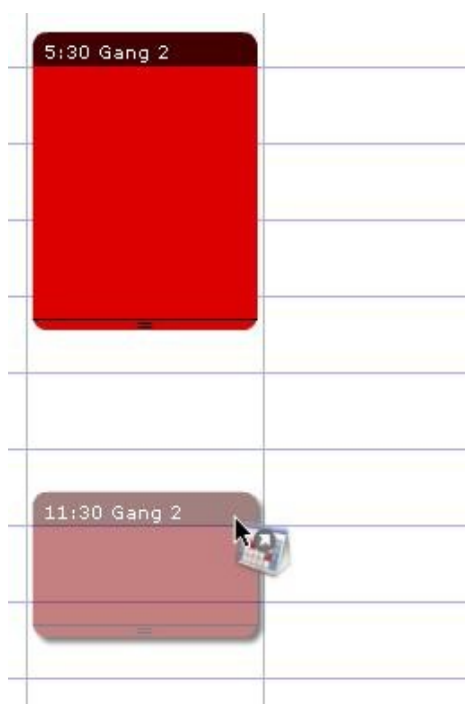


Figura 6.26: Exemplo de uma situação de *drag'n'drop* na agenda

Finalmente, é também permitido ao utilizador remover eventos, exceptuando durante a sua execução, sendo também possível seleccionar a pretensão de eliminar o evento seleccionado especificamente, ou todos os eventos da série. É de relembrar que para que um evento faça parte de uma série é necessário que tenha sido originalmente criado com uma opção de periodicidade não única.

Todas estas operações sobre eventos reflectem-se depois na agenda, que será transmitida ao servidor, para que este a guarde sob a forma de um ficheiro XML, tal como nos outros casos anteriormente descritos.

```
- <schedule>
- <event group_id="1241705402000" id="1241708400000verdadeiro">
  <start day="7" hour="16" minute="0" month="5" year="2009" />
  <end day="7" hour="18" minute="0" month="5" year="2009" />
  <periodicity period="2" />
  <gang name="verdadeiro" />
</event>
- <event group_id="1241705402000" id="1241794800000verdadeiro">
  <start day="8" hour="11" minute="0" month="5" year="2009" />
  <end day="8" hour="16" minute="30" month="5" year="2009" />
  <periodicity period="2" />
  <gang name="verdadeiro" />
</event>
```

Figura 6.27: Formato do ficheiro XML onde é guardada a configuração da agenda

Como pode ser visto na Figura 6.27, o formato deste ficheiro permite manter informação sobre as datas de início e fim do evento, a sua periodicidade, o nome do *gang* a que diz respeito e um número identificador tanto para o próprio evento como para a série a que se refere.

Resta referir que a monitorização destes eventos está a cargo do servidor (*python*), sendo que, ao ser iniciado, verifica também a existência de eventos em atraso, dando início imediato a essas capturas. Este tipo de monitorização implica que caso algum evento não possa ser iniciado na altura pretendida, por falha da própria máquina ou outra, o sistema irá tentar novamente iniciar a captura até que seja bem sucedido, ou que a data de paragem da captura chegue (ou passe).

6.6 Testes e Simulações

De maneira a garantir que todos os requisitos e funcionalidades esperadas da aplicação desenvolvida estavam correctamente implementados, e que a ferramenta como um todo era funcional e capaz de operar num cenário real, foram efectuados alguns testes e simulações sobre a mesma.

Devido a alguma escassez de estações de captura reais do modelo mxfsPEEDRAIL S1000, que serviriam de suporte aos testes, foram utilizados apenas dois cenários para simular a operação do protótipo resultante deste projecto.

O primeiro cenário tinha como objectivo garantir o correcto funcionamento do sistema e a sua interoperabilidade com modelos reais dos servidores de captura. Foram utilizadas três máquinas reais, cada uma a receber o seu próprio sinal de vídeo (e áudio).

Para cada uma destas máquinas foram efectuados diversos testes, sendo os seguintes os mais relevantes:

- Iniciar e parar captura, verificando a criação bem sucedida do ficheiro, com o nome correcto e armazenado na localização correcta.
- Junção das duas máquinas num só grupo, ou *gang*, efectuando algumas capturas para o mesmo, e verificando assim que todas procediam à captura através de um único comando.
- Acesso e alteração de configurações para cada uma das estações.
- Verificação do estado de cada máquina, em cada momento, de forma a garantir a sua coincidência com o estado apresentado pelo protótipo em teste
- Calendarização de capturas, verificando o seu início e paragem atempada
- Verificação do comportamento do protótipo aquando da falha de alguma das máquinas controladas pelo próprio

Todos os testes obtiveram resultados bastante satisfatórios, sendo que mesmo nas ocasiões em que alguma das estações envia valores respeitantes ao seu estado com algum atraso, o próprio sistema se encarrega de tentar simulá-los, fazendo uma estimativa aproximada com base no último estado recebido, notificando o utilizador apenas em caso de falhas graves ou mais

Desenvolvimento do Protótipo

relevantes. Pode ver-se na Figura 6.28 o estado de uma destas estações, no decurso de um teste de captura.



Figura 6.28: Estado da aplicação com uma captura parcial em curso

Para o segundo cenário simulado, foram utilizados servidores de captura virtuais, devido à já mencionada escassez de aparelhos reais, de modo a efectuar alguns testes de carga e desempenho do protótipo. Estes servidores virtuais assemelham-se em tudo aos reais, embora o sinal recebido, bem como as capturas que efectuam, sejam simuladas. Controlando inicialmente uma única estação de captura virtual, este número foi incrementado até se obter o controlo de doze servidores, um valor já significativamente superior ao proposto originalmente como máximo permitido (oito servidores) por aparelho. Neste cenário, os testes efectuados sobre as máquinas foram, na sua maioria, idênticos aos apresentados para o primeiro cenário, sendo que se mediu primariamente a diferença, ou não, no comportamento do protótipo, quando em comparação com o cenário anterior.

Os resultados obtidos neste último cenário apresentaram-se também bastante promissores, sofrendo apenas uma perda de desempenho mínima na passagem do controlo de duas máquinas para o de uma dúzia. Ainda neste campo, a única alteração sentida foi na actualização do estado das máquinas, que por vezes era recebido com um atraso, embora mínimo. Na tabela 6.1 podem verificar-se alguns valores referentes ao tempo de resposta da aplicação.

Tabela 6.1: Tempo de resposta da ferramenta nas operações mais relevantes

	Cenário 1	Cenário 2
Iniciar/Parar Capturas Individuais	< 1 seg	< 1 seg
Iniciar/Parar Capturas para Grupos	< 1 seg.	< 2 seg
Actualização do estado	< 1 seg	< 1 seg
Alteração de Configurações nas Estações de Captura	< 2 seg	< 4 seg

Quanto aos testes de captura, as máquinas virtuais responderam sempre de forma aceitável, com os comandos relativos a capturas a obterem tempos de resposta menores que dois segundos, como estipulado nos requisitos. Todos os outros comandos obtiveram o mesmo tipo

Desenvolvimento do Protótipo

de resposta, embora não se apresentassem tão críticos para o correcto funcionamento do sistema. Na Figura 6.29 podemos ver a interface da aplicação, num momento em que as doze estações virtuais se encontravam no decurso de uma captura.



Figura 6.29: GUI com capturas a decorrer em doze máquinas simultaneamente

Quando se efectuaram alterações ao estado das estações de captura por meios externos, ou seja, na própria estação, a actualização figurou também imediatamente na interface do protótipo, mostrando que também aí o seu funcionamento se encontrava de acordo com o esperado.

Outra medida de desempenho com bastante interesse, dada a utilização real que se pretende do sistema, prende-se com a sincronização entre máquinas, nomeadamente ao iniciar capturas para um grupo inteiro, ou na apresentação do *timecode* do sinal recebido, que se pretende bastante fiável. Este tipo de testes foi obviamente feito apenas no cenário 1, visto que requer a utilização de máquinas reais com sinais de vídeo reais. A tabela 6.2 exhibe os resultados dos testes, obtidos através de cerca de 30-40 medições.

Tabela 6.2: Sincronização entre máquinas e operações de um gang (3 servidores)

	Desfasamento Mínimo (frames)	Desfasamento Máximo (frames)	Desfasamento Médio (frames)
Iniciar/Parar Capturas	1	7	3
Timecode apresentado / efectivo	0	3	1

Pode verificar-se que a diferença entre o timecode efectivo e o apresentado pela aplicação nunca ultrapassa um desfasamento de 3 *frames*, valor que nos parece francamente positivo, tendo em conta que na maioria das situações a diferença é de 1 *frame*, ou até nenhum, não tendo assim nenhuma influência aquando da monitorização do servidor de forma remota. Os resultados para o início de uma captura “simultânea” foram também positivos, sendo que entre

Desenvolvimento do Protótipo

as 3 máquinas nunca existiu um desfasamento superior a 7 frames, sendo a média entre 3 a 4, valores perfeitamente aceitáveis para o tipo de aplicação.

Quanto à robustez, e uma vez que o sistema não poderia nunca estar dependente do bom funcionamento das máquinas que controla, foram aleatoriamente desconectadas algumas das máquinas, verificando-se que o resto do sistema continuou operacional, procedendo apenas à actualização do estado da máquina correspondente, para informação do utilizador. No caso de capturas programadas, caso as estações associadas à captura não estivessem disponíveis na altura correcta, o sistema efectuou novas tentativas para iniciar essas capturas até o conseguir, ou até que a captura agendada expirasse. Para evitar atrasos no envio de algum comando para início de captura, que poderiam levar a perdas de informação, a agenda envia então esse aviso, de início de captura, alguns segundos antes do início exacto do evento, facto também verificado durante os testes.

Por último, foi também testada, de certo modo, a fiabilidade do sistema, permitindo ao servidor continuar em execução durante várias horas, verificando-se assim que mantinha, ao fim desse considerável tempo de execução, um funcionamento em tudo idêntico ao apresentado inicialmente, podendo concluir-se que, apesar das falhas provocadas propositadamente nas máquinas controladas pelo próprio, nunca a sua correcta execução foi afectada.

Apesar dos resultados bastante positivos obtidos nos dois tipos de cenários testados, seria sempre relevante, no futuro, efectuar testes mais exaustivos sobre o protótipo, nomeadamente através da utilização de múltiplas estações reais, executando assim o cenário 2, mas sem recurso à utilização de servidores virtuais, em que cada um receberia o seu próprio sinal, de modo a reforçar a confiança que o protótipo actualmente já proporciona.

6.7 Conclusões

Pode concluir-se deste capítulo que todos os requisitos foram, de facto, satisfeitos, e que a aplicação tenta permitir uma utilização com elevado nível de intuitividade. As funcionalidades foram agrupadas coerentemente encaminhando o utilizador para uma experiência simples e fluída.

Quanto aos requisitos não funcionais, o desempenho geral da aplicação apresenta-se sempre em níveis satisfatórios, mesmo quando testada com cerca de uma dezena de dispositivos controlados, o que estará perfeitamente dentro dos limites médios para uma utilização dita normal. A sua robustez é assegurada certificando que em caso de falha alguma das máquinas, o resto da aplicação não é comprometida nem nenhum processo bloqueará o sistema.

Uma última palavra para reiterar a arquitectura modular implementada, que permite a separação total da interface, visível nos exemplos do capítulo presente, do servidor, que trata efectivamente dos pedidos e sua execução. Aquando da necessidade torna-se efectivamente simples a troca de componentes visuais sem afectar a funcionalidade e estabilidade do sistema.

7 Conclusões

Este capítulo sumaria as principais conclusões e resultados do trabalho desenvolvido no decurso do projecto, avaliando o cumprimento e satisfação dos objectivos propostos. Apresenta também algumas considerações sobre a sua contribuição no campo da produção de multimédia profissional, bem como algum possível trabalho futuro que poderá desenvolver-se partindo do conseguido com o projecto.

7.1 Sumário

Após o final do projecto, pode concluir-se que os objectivos foram cumpridos na íntegra, sendo que todos os requisitos especificados no seu início foram satisfeitos, com maior ou menor dificuldade.

O sucesso obtido deve-se também à investigação e planeamento efectuados previamente, que permitiram que o desenvolvimento não fosse, significativamente, afectado por factores externos, como limitações tecnológicas.

De acordo com o especificado, o protótipo desenvolvido é então capaz de controlar múltiplos servidores de captura em simultâneo, tanto individualmente como agrupados, o que permite capturas simultâneas de vários dispositivos. Outra funcionalidade com grande relevância é a de agendamento de capturas para estes dispositivos ou agrupamentos, que também é suportada pela aplicação, permitindo configurar com total liberdade uma agenda de capturas.

O protótipo desenvolvido permite assim, com grande confiança, e particularmente devido à alargada utilização de ambientes multi-câmara, antever que o futuro da produção multimédia profissional passará certamente por sistemas como o descrito neste documento, replicando ou aproveitando muitas das funcionalidades aqui apresentadas.

Como nota final resta referir que todas as partes envolvidas evidenciaram a sua satisfação pelo cumprimento das metas propostas.

7.2 Resultados

De forma mais prática, podemos agora analisar os principais resultados obtidos a partir do projecto.

Conclusões

O principal objectivo a atingir com o projecto era sem dúvida o controlo centralizado de múltiplas estações de captura. No que se relaciona com este objectivo, sem dúvida que pode ser inferido que o protótipo desenvolvido cumpriu com o proposto, aumentando assim a fasquia para este tipo de sistemas, no leque de funcionalidades que poderão, a breve prazo, ser disponibilizadas no mundo real.

Quanto aos requisitos não funcionais, embora de mais difícil avaliação, pode considerar-se que foram também cumpridos com sucesso.

A nível de desempenho, o protótipo não foi submetido a cargas demasiado exigentes, mas em todas as simulações feitas, que incluíram o controlo de até uma dezena de dispositivos, valor já bastante razoável e até superior ao limite que se prevê para numa utilização real do sistema, o seu comportamento foi sempre regular, sem sinais de sobrecarga, quer medida em atrasos a comandos do utilizador, como em utilização dos recursos da máquina onde corre. A sincronização entre as máquinas controladas apresentou também resultados significativamente positivos, como pôde ser visto no capítulo anterior.

Quanto à fiabilidade e robustez, inúmeros cenários de utilização abusiva ou incoerente foram também testados, permitindo um nível de confiança elevado quanto à estabilidade do sistema. Para os casos de falhas impossíveis de prever, o seu funcionamento no geral não é afectado, sendo que o utilizador ficará devidamente informado acerca do acontecimento inesperado.

Outros requisitos também relevantes, como a interoperabilidade e modularidade, ficam também assegurados, sendo possível trocar qualquer dos principais módulos da aplicação eficientemente, bem como integrá-la com novas funcionalidades que surjam nos dispositivos que controla, algumas das quais foram também acrescentadas no decurso do projecto, muitas das vezes derivando da investigação efectuada para o mesmo, reflectindo assim a importância das mesmas.

Finalmente, também se pode considerar atingido o objectivo de assegurar uma interface gráfica intuitiva, eficaz e coerente com a utilizada nos dispositivos cujo propósito era controlar remotamente.

Atendendo à completude dos objectivos, ao desenvolvimento do protótipo e inovação que o mesmo trouxe, ao nível de automatização e facilitação do controlo de estações de captura, conclui-se então que o projecto terminou de forma bem sucedida, deixando agora porta aberta para futuros desenvolvimentos sobre o mesmo.

7.3 Trabalho Futuro

Apesar do cumprimento de todos os objectivos, existiriam sempre inúmeras outras funcionalidades sem dúvida bem-vindas e apreciadas pelos utilizadores do tipo de sistemas em questão. Devido ao limitado tempo disponível, optou-se por investigar, desenhar e implementar as mais relevantes, não esquecendo no entanto, as portas que o projecto iria abrir em termos de possibilidades e novas inovações no campo.

Possíveis desenvolvimentos futuros, que providenciariam ainda um maior controlo sobre as operações de captura, tirando partido da nova estrutura organizacional dos dispositivos controlados, com especial relevância para o seu agrupamento em *gangs*, podem ser tão ambiciosos como o controlo de diferentes tipos de estações de captura, com diferentes características, aumentando assim exponencialmente a possibilidade de controlo central, ou mais simples mas igualmente úteis como a criação de um sistema automático de introdução de metadados, que existe de momento numa versão com algumas limitações.

Conclusões

Outras funcionalidades interessantes a desenvolver no futuro poderão passar pela possibilidade de planeamento de produção, permitindo mapear capturas à cena ou *take* correspondente, incluindo também os metadados associados. Este sistema poderia até permitir carregar algum tipo de ficheiro representando o guião para determinada produção, sendo as capturas das cenas e informação correspondente geradas automaticamente a partir do mesmo.

Como prognóstico final relembra-se que o mundo da produção de televisão profissional, bem como a tecnologia utilizada no mesmo, encontra-se em constante evolução, o que permite antever que as possibilidades de inovação na área serão sempre ilimitadas, sendo que a meta passará pela libertação total dos intervenientes humanos de tarefas automatizáveis, para que possam concentrar-se na parte criativa, realmente a essencial, de todo este processo. Espera-se que o contributo dado com este projecto permita, de certa forma, mais uma aproximação a este objectivo final.

Referências

- [ADC09] adCode Media Ingest Station, Strong Digital. Disponível em <http://www.strongdigital.com/products/adcode/index.html>, acessado a última vez em 26 de Junho de 2009.
- [Ado08] Adobe. Adobe FLEX 3 Developer Guide. Adobe, 2008.
- [FAF09] Flex Architecture Fundamentals. Disponível em <http://www.dehats.com/drupal/>, acessado a última vez em 26 de Junho de 2009.
- [FSF09] Flex Scheduling Framework. Adobe Labs. Disponível em <http://labs.adobe.com/>, acessado a última vez em 26 de Junho de 2009.
- [GSV09] GSV9000 VTR Ingest Station, GVS. Disponível em <http://www.gvs9000.com/gvs90001u-is.html>, acessado a última vez em 26 de Junho de 2009.
- [Jak03] Jakob Nielsen. Usability Engineering. Morgan Kaufmann, 2003.
- [Kauf09] Sam Kauffmann. Avid Editing: A Guide for Beginning and Intermediate Users, Focal Press, 2009.
- [LW03] Dean Leffingwell and Don Widrig. Managing Software Requirements, A Use Case Approach. Addison-Wesley, 2003.
- [Mag05] Magnus Lie Hetland. Beginning Python, From Novice to Professional. Apress, 2005.
- [Mat09] Mate Flex Framework. Disponível em <http://mate.asfusion.com/>, acessado a última vez em 26 de Junho de 2009.
- [MED09] Media HUB Media Ingest Station, AdTec Digital. Disponível em http://www.digitalsignagetoday.com/products.php?prod_id=1110, acessado a última vez em 26 de Junho de 2009.
- [MOG09] mxSPEEDRAIL S1000, MOG. Disponível em <http://www.mog-solutions.com/produtos.php?ID=128>, acessado a última vez em 26 de Junho de 2009.
- [Phi07] Phillip A. Laplante. What Every Engineer Should Know about Software Engineering. CRC Press, 2007.
- [Poy03] Charles A. Poynton. Digital Video and HDTV: algorithms and interfaces, Morgan Kaufmann, 2003.
- [PXI09] PX Ingest Station, Focus Enhancements. Disponível em <http://www.videonics.com/solutions/px-ingest.asp>, acessado a última vez em 26 de Junho de 2009.
- [Pyt09] Python Software Foundation. Python v2.6.2 documentation. Disponível em <http://docs.python.org/>, acessado a última vez em 26 de Junho de 2009.

Conclusões

- [SMP04a] SMPTE. EG41: Material Exchange Format – SMPTE Engineering Guideline. SMPTE, 2004.
- [SMP04b] SMPTE. EG42: MXF Descriptive Metadata – SMPTE Engineering Guideline. SMPTE, 2004.
- [SMP04c] SMPTE. S377M: MXF File Format Specification – SMPTE Standard. SMPTE, 2004.
- [SMP04d] SMPTE. S380M: MXF Descriptive Metadata Scheme-1 – SMPTE Standard. SMPTE 2004.
- [STK01] James Snell, Doug Tidwell, Pavel Kulchenko. Programming Web Services with SOAP, O'Reilly, 2001.
- [Sud09] Suds. Disponível em <https://fedorahosted.org/suds/wiki/Documentation>, acessado a última vez em 26 de Junho de 2009.
- [SWS09] W3Schools. SOAP Introduction. Disponível em <http://www.w3schools.com/SOAP>, acessado a última vez em 26 de Junho de 2009.
- [TGW09] TGWebServices 1.2.2. Disponível em <http://pypi.python.org/pypi/TGWebServices>, acessado a última vez em 26 de Junho de 2009.
- [VIV09] Vivesta Content Ingest Application Suite, Vivesta. Disponível em <http://www.vivesta.com/index.php?cid=2&pid=510>, acessado a última vez em 26 de Junho de 2009.
- [WMD06] Nick Wells, Oliver Morgan, Jim Wilkinson, Bruce Devlin. The MXF Book – Introduction to the Material eXchange Format, Focal Press, 2006.

