

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Aplicação modular de tele-monitorização para o setor da hotelaria baseada em Redes de Sensores Sem Fios

Pedro Santos Moreira

VERSÃO DE TRABALHO

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Gil Manuel Gonçalves

19 de Fevereiro de 2014

Aplicação modular de tele-monitorização para o setor da hotelaria baseada em Redes de Sensores Sem Fios

Pedro Santos Moreira

Mestrado Integrado em Engenharia Informática e Computação

Resumo

As Redes de Sensores Sem Fios (RSSF) são cada vez mais um tópico de enorme actividade de investigação, principalmente devido aos avanços nas tecnologias de *microchips*. As RSSF estão a emergir em áreas de investigação tão distintas como a saúde, a psicologia, a prevenção de fogos, a segurança e as áreas militares. Este tipo de redes permite acompanhar, monitorizar, estudar, compreender e actuar sobre um determinado fenómeno ou acontecimento. [TC]

Segundo várias organizações Portugal é eleito um destino de eleição para o turismo, segundo a Globe Spots, Portugal é o melhor destino do mundo em 2013 [P313], segundo a Associação de Viagens britânica (ABTA) Portugal está nos melhores destinos do mundo para o ano de 2014. [Gv14].

A FreedomGrow é proprietária de uma Framework, de nome PlugSense, que tem como função a agilização do processo de criação de aplicações baseadas em RSSF. Este processo costuma ser bastante trabalhoso, pois a gestão e integração do hardware bem como a integração dos diferentes protocolos de comunicação é bastante complexa. A PlugSense é uma aplicação de software, que simplifica a implementação e gestão das RSSF, permitindo ao programador focar-se apenas nas especificações da aplicação e não em superar as dificuldades no seu desenvolvimento. As aplicações desenvolvidas sobre a aplicação criada pela Framework, são aplicações verticais, que podem ser aplicadas em vários campos, desde a agricultura, saúde, hotelaria entre outros.

A FreedomGrow, tendo em conta estes factores, decidiu criar uma aplicação modular de telemonitorização para o setor da hotelaria baseada em Redes de Sensores Sem Fios. O desafio é utilizar da PlugSense para o desenvolvimento de uma aplicação agregadora de sistemas de monitorização, críticas para um serviço de hotelaria, de elevado valor acrescentado e obedecendo aos mais rigorosos padrões de qualidade, de forma a obter toda a informação recolhida pelos diversos sistemas de monitorização, numa única plataforma, acessível por qualquer dispositivo fixo ou móvel. A aplicação deverá ser modular e incluir tecnologias de videovigilância, eficiência energética, localização de pessoas e equipamentos, deteção de incêndios e inundações, controlo de acessos, segurança alimentar e controlo ambiental.

Tendo em conta que todos os sensores usados para a monitorização do hotel, são sensores sem fios, e que os sensores de videovigilância necessitam de uma grande largura de banda, torna-se extremamente importante ter uma ligação de rede robusta e fiável. Para resolver os problemas de largura de banda e espaço de armazenamento, implementou-se um sistema de videovigilância que apenas efectua gravação com ocorre um evento. Neste caso o sistema inicia gravação quando ocorre um alerta sensorial. A aplicação ao receber o alerta envia um comando para o sensor de videovigilância com a posição do alerta, este ao recebe-lo efectua a rotação para o local e inicia a gravação. Com esta implementação o sistema além de reduzir espaço de armazenamento e largura de banda, aumenta a qualidade da informação guardada.

No final da implementação realizaram-se testes à solução, estes mostraram que as alterações realizadas são bastante benéficas para o projecto, pois verifica-se uma redução de cerca de 80% na largura de banda consumida e cerca de 87% no espaço de armazenamento.

Abstract

The Wireless Sensor Networks (WSN) are a topic of enormous research activity, mainly due to advances in microchip technology. The WSN are emerging in research areas like psychology, fire prevention, security and military areas. This type of networking allows you to track, monitor, study, understand and act on a particular phenomenon or event. [TC]

According to several organizations, Portugal is elected a destination of choice for tourism. According to the Globe Spots Portugal was the best destination of the world in 2013 [P313] and according to the Association of British Travel (ABTA) Portugal is in the best destinations worldwide for 2014. [Gv14]

The FreedomGrow is owner of a Framework named PlugSense, whose function is to expedite the creation of process-based WSN applications. This process is usually very laborious because the management and integration of hardware as well as the integration of different communication protocols are quite complex. The PlugSense is a software application that simplifies the implementation and management of the WSN allowing the programmer to focus only on application requirements and not to overcome the difficulties in their development. Applications developed on the application created by the Framework, are vertical applications that can be applied in various fields like agriculture, healthcare, hospitality and more.

The FreedomGrow, taking into account these factors, decided to create a modular application of telemonitoring for the hospitality sector based in Wireless Sensor Networks. The challenge is to use the PlugSense for the development of an aggregating application of monitoring systems, critical for hotel service, high value added and satisfying the most stringent quality standards in order to obtain all the information collected by the various systems of monitoring in a single platform accessible by any device fixed or mobile. The application should be modular and should include video surveillance, energy efficiency, location of people and equipment, fire and flood detection, access control, food safety and environmental control technologies.

Considering that all sensors used for monitoring of the hotel are wireless sensors and sensors for video surveillance systems this require a large bandwidth becomes extremely important to have a robust network connection and reliable. To solve the problems of bandwidth and storage space was implemented a surveillance system that performs just recording when the event occurs. In this case the system starts recording when a sensory alert occurs. The application receives the alert and sends a command to the sensor video surveillance with the position of warning, this one that receives the command rotate to the local of the warning and starts recording. With this new system the storage space and bandwidth are reduced and the quality of the stored information is increased.

At the end of the deployment test solution were done. These showed that the changes were very beneficial because a reduction of approximately 80% in the consumption of bandwidth and about 87% of the storage space was observed.

Agradecimentos

Ao Professor Gil Gonçalves e ao Engenheiro Marco Alves pela orientação ao longo da dissertação.

À minha família por acreditarem sempre em mim, em especial aos meus pais e irmãos, pelo sacrifício e pelo apoio, pois sem eles não era possível.

Aos meus amigos que me acompanharam ao longo desta caminhada e que sem eles teria sido tudo mais difícil ou pelo menos tudo muito mais aborrecido.

Por fim, à minha namorada e pelo companheirismo e apoio incondicional.

Pedro Moreira

Conteúdo

1	Introdução	1
1.1	Contexto/Enquadramento	1
1.2	Projeto	2
1.3	Motivação e Objetivos	3
1.4	Estrutura da Dissertação	3
2	Revisão Bibliográfica	5
2.1	HotelSense	5
2.1.1	Arquitetura da Solução	6
2.2	Aplicações Concorrentes	7
2.2.1	Hotel Systems	8
2.2.2	Intelligent Hotel	9
2.3	Sensores de Videovigilância	9
2.3.1	AXIS M1034-W	9
2.3.2	AXIS M1033-W	10
2.3.3	AXIS M1054	10
2.3.4	AXIS M1011-W	10
2.3.5	Largura de Banda nos Sistemas de Videovigilância	10
2.4	Fusão de Dados	11
2.4.1	Dados Complementares	12
2.4.2	Dados Redundantes	12
2.4.3	Dados Cooperativos	13
2.5	Arquiteturas Básicas de Fusão de Dados	13
2.5.1	Arquitetura Centralizada	13
2.5.2	Arquitetura Descentralizada	14
2.5.3	Arquitetura Hierárquica	15
2.6	Resumo ou Conclusões	15
3	Problema	17
3.1	Descrição	17
3.2	Proposta de Valor	18
3.3	Solução	19
3.3.1	Optimização da Largura de Banda	19
3.3.2	Optimização da Qualidade da Informação	19
3.3.3	Posição Home	20
3.3.4	Sistema de Posicionamento Sensorial	20
3.3.5	Resumo ou Conclusões	20

CONTEÚDO

4	Conceção	21
4.1	Arquitectura da Solução	21
4.2	Casos de Uso	22
4.3	Requisitos	23
4.3.1	Integração dos Sensores de Videovigilância	23
4.3.2	Otimização da Qualidade da Informação	23
4.3.3	Otimização do Consumo da Largura de Banda e do Espaço de Armazena- mento	24
4.3.4	Visão Geral	26
4.4	Testes Simulados	27
4.4.1	Simulação de Monte Carlo	27
4.5	Resumo ou Conclusões	28
5	Implementação	29
5.1	Integração dos Sensores de Videovigilância	29
5.1.1	Axis M1034-W	30
5.1.2	Foscam	30
5.1.3	Base de Dados	31
5.2	Otimização da Largura de Banda/Qualidade da Informação	32
5.2.1	Fusão de Dados Sensoriais	32
5.2.2	Definição de Posições	33
5.2.3	Rotação dos Sensores para a Posição do Alerta e Tratamento de Múltiplos Alertas.	37
5.2.4	Gravação	41
5.3	Outras funcionalidades	43
5.3.1	Resumo e Conclusão	45
6	Testes Comparativo	47
6.1	Cenário	47
6.2	Teste de Pequena Duração	49
6.2.1	Solução Inicial	49
6.2.2	Solução Implementada	50
6.2.3	Análise dos Resultados e Conclusões	52
6.3	Teste de Maior Duração	53
6.3.1	Solução Inicial	53
6.3.2	Solução Implementada	54
6.3.3	Análise dos Resultados e Conclusões	56
6.4	Simulação de Monte Carlo	56
7	Conclusões e Trabalho Futuro	61
7.1	Cumprimento do Objetivo	61
7.2	Trabalho Futuro	62

Lista de Figuras

2.1	Mockup da interface do sistema	6
2.2	Mockup da interface do sistema com alertas	6
2.3	Arquitetura da Aplicação	7
2.4	Hotel Systems	8
2.5	Intelligent Hotel	9
2.6	Fusão Complementar,Redundante e cooperativa	12
2.7	Arquitetura Centralizada	14
2.8	Arquitetura Descentralizada	14
2.9	Arquitetura Hierárquica	15
4.1	Arquitetura da solução	21
4.2	Fluxo de Dados entre Componentes	22
4.3	Caso de uso - Administrador	23
4.4	Diagrama de Sequência -Adicionar Coordenadas	24
4.5	Diagrama de Sequência- Alerta	25
4.6	Diagrama de sequência- Arquivar Alerta	26
5.1	Interface do Projeto Inicial	29
5.2	Interface - Sensor Axis	30
5.3	Interface - Sensor Foscam	31
5.4	Tabelas Camera e CameraType	32
5.5	Interface -Adicionar Coordenadas	34
5.6	Tabela Coordenadas	34
5.7	Tabelas CameraUso e AlertaEspera	37
5.8	View de Histórico de Alertas	44
5.9	View de Visualização das Gravações	44
6.1	Cenário - sala1	47
6.2	Cenário - sala2	48
6.3	Grafico de Consumos- Camera Foscam	49
6.4	Grafico de Consumo- Camera Axis	50
6.5	Consumo de Largura de Banda da Aplicação Desenvolvida - Camera Foscam	51
6.6	Alertas detetados no teste- Axis	52
6.7	Consumo de Largura de Banda- Projecto inicial	54
6.8	Consumo de Largura de Banda- Projecto final	55
6.9	Monte Carlo- Sala1	58
6.10	Monte Carlo- Sala2	59
6.11	Monte Carlo- Total	59

LISTA DE FIGURAS

Lista de Tabelas

4.1	Relação de requiitos com usecases	26
6.1	Alertas detetados no teste- Foscam	51
6.2	Alertas detetados no teste- Axis.	52
6.3	Analise dos Resultados- Teste de Pequenas Dimensões.	53
6.4	Alertas detetados na Sala1.	54
6.5	Alertas detetados na Sala2.	55
6.6	Analise dos Resultados- Teste de Maior Dimensão.	56
6.7	Modelo de simulação Monte Carlo-Variáveis Dependentes	56
6.8	Modelo de simulação Monte Carlo-Variáveis Independentes	56
6.9	Modelo de simulação Monte Carlo-Valores finais	57

LISTA DE TABELAS

Abreviaturas

OMT	Organização Mundial do Turismo
RSSF	Redes de Sensores Sem Fios
FEUP	Faculdade de Engenharia da Universidade do Porto
LGP	Laboratório de Gestão de Projeto
CASE	Computer-Aided Software Engineering
RFID	Radio-Frequency IDentificationg

Capítulo 1

Introdução

O Turismo é um dos setores que mais cresce no mundo, fenómeno que é fomentado pela globalização e aumento das comunicações a nível mundial. Este setor tem sido fundamental para os países que apresentam potencial turístico, como é o caso de Portugal, que é altamente reconhecido pelas suas praias, gastronomia e herança cultural e patrimonial. [Fer11]

Segundo a Organização Mundial do Turismo (OMT), as receitas do turismo internacional à escala global aumentaram 4% em 2012, atingindo um novo recorde de 837 mil milhões de euros, segundo a mesma organização (OMT) Portugal em 2012 registou 8,6 mil milhões de euros em receitas turísticas, o que representou um crescimento de 5,6% face ao ano anterior. [dN13].

Com a entrada em força das companhias de aviação de baixo custo, são cada vez mais os turistas que chegam a Portugal. Essas chegadas mexem pela positiva com todo o mercado da hotelaria. Exemplo disso é o crescimento significativo de unidades hoteleiras no Porto, desde hotéis de luxo a hostels. [Pub10]

As Redes de Sensores Sem Fios (RSSF) nasceram nas aplicações militares, um início comum a algumas das tecnologias que hoje conhecemos. Estas redes são compostas por um número potencialmente elevado de nós sensores inteligentes e com um núcleo de processamento. As aplicações das RSSF estenderam-se para áreas de utilização civil, nomeadamente monitorização ambiental, monitorização de edifícios, aplicações industriais, saúde e espaços inteligentes, entre outras. [PACSN]

1.1 Contexto/Enquadramento

Esta dissertação enquadra-se na área de redes de sensores sem fios, e irá ser realizada com a colaboração da empresa FreedomGrow sob a orientação do Professor Gil Gonçalves e do Engenheiro Marco Alves.

A FreedomGrow trata-se de uma empresa de base tecnológica, fundada em 2010 e especializada em soluções de software para redes de sensores sem fios (RSSF). É detetora de uma Framework, de nome PlugSense, que tem como função a agilização do processo de criação de aplicações baseadas em RSSF. Este processo costuma ser bastante trabalhoso devido à dificuldade

Introdução

de gestão e integração de hardware e protocolos. A PlugSense é uma aplicação de software, desenvolvida na .NET Framework 4.0 da Microsoft, que simplifica a implementação e gestão das RSSF, permitindo ao programador focar-se apenas nas especificações da aplicação e não em superar as dificuldades no seu desenvolvimento, pois contempla características únicas tais como a receção, análise de processamento de dados de qualquer dispositivo sem fios existente no mercado, assim como a integração com qualquer sistema de gestão, protocolos de comunicação ou aplicações finais específicas. Atualmente, a PlugSense é capaz de criar aplicações Web de uma forma dinâmica, ajustando-as de acordo com os requisitos do utilizador, sendo estas aplicações utilizadas para monitorizar as RSSF através de um browser.

1.2 Projeto

Numa altura em que o turismo em Portugal tem vindo a crescer, sendo uma das áreas que contribui mais para o crescimento económico nacional e sendo os sensores dispositivos com um enorme potencial, a FreedomGrow, tendo em conta estes fatores, decidiu criar uma aplicação modular de tele-monitorização para o setor da hotelaria baseada em Redes de Sensores Sem Fios. O desafio é utilizar a PlugSense para o desenvolvimento de uma aplicação agregadora de sistemas de monitorização, críticas para um serviço de hotelaria, de elevado valor acrescentado e obedecendo aos mais rigorosos padrões de qualidade, de forma a obter toda a informação recolhida pelos diversos sistemas de monitorização, numa única plataforma, acessível por qualquer dispositivo fixo ou móvel.

A Freedomgrow decidiu investir no turismo low-cost, visto que é um nicho de mercado em expansão, sendo o desafio desenvolver uma aplicação low-cost agregadora de sistemas de monitorização, críticas para um serviço de hotelaria, de elevado valor acrescentado e obedecendo aos mais rigorosos padrões de qualidade, de forma a obter toda a informação recolhida pelos diversos sistemas de monitorização, numa única plataforma, acessível por qualquer dispositivo fixo ou móvel, para implementar em hotéis de pequenas dimensões, hostels, pousadas entre outras.

A aplicação deverá ser modular e incluir tecnologias de videovigilância, eficiência energética, localização de pessoas e equipamentos, deteção de incêndios e inundações, controlo de acessos, segurança alimentar e controlo ambiental. [Mar13]

Tendo em conta o facto dos sensores de videovigilância serem sensores wi-fi, ou seja, sensores sem fios, torna-se importante ter em atenção a largura de banda necessária para o seu bom funcionamento. Devido a isto, o projeto terá como principal foco a gestão e otimização da largura de banda e o espaço de armazenamento do sistema de videovigilância.

Pretende-se também no decorrer do projeto fazer um estudo sobre o impacto da otimização da largura de banda, de forma a verificar o verdadeiro impacto das otimizações realizadas durante o projeto.

1.3 Motivação e Objetivos

Os sensores são instrumentos fixos ou móveis que medem ou detetam características do mundo real, como movimento, calor, luz ou localização, e convertem essas medidas numa representação analógica ou digital. As redes baseadas em sensores são grupos de sensores com uma infraestrutura de comunicações destinadas a monitorizar e gravar cooperativamente medidas em localizações diversas. [UMI13]

Tendo em conta estas características e a as RSSFs são uma tecnologia emergente que promete uma funcionalidade sem precedentes para monitorar, instrumentar, e, possivelmente, controlar o mundo físico.

O objetivo do projeto é utilizar todo o potencial dos sensores e das suas redes numa área específica, neste caso a área hoteleira, tendo o principal foco a redução da largura de banda e espaço de armazenamento consumido pelos sensores de videovigilância.

Além da redução do consumo de largura de banda e espaço de armazenamento, pretende-se melhorar a qualidade da informação transmitida pelos sensores de videovigilância, para tal pretende-se reduzir/eliminar áreas desnecessárias e focar a imagem no que realmente é importante, como por exemplo, eliminar paredes brancas/sem informação.

1.4 Estrutura da Dissertação

Este documento está organizado em 7 capítulos e começa com a introdução na qual é feito um enquadramento do problema. No capítulo 2 é descrita a plataforma que está na base desta dissertação, é apresentado o estado da arte e são referenciados os trabalhos relacionados. No capítulo 3 é detalhado o problema e apresentada a proposta de solução. Os capítulos 4 e 5 dizem respeito à conceção e implementação da respetiva solução. No capítulo 6 são apresentados os testes efetuados e os resultados obtidos e, por fim, no capítulo 7 é onde constam as conclusões.

Introdução

Capítulo 2

Revisão Bibliográfica

Nesta seção importa explorar o estado da arte relacionado com o assunto proposto. É importante analisar que outras soluções já existem atualmente e que alternativas há para resolver o problema original. Primeiramente será feita uma breve descrição da plataforma que está na base desta dissertação e depois será descrito então de forma breve aquilo que foi encontrado através da pesquisa de artigos científicos em bases de dados bibliográficas e bibliotecas digitais.

2.1 HotelSense

No início do projecto de dissertação, o projecto já se encontrava parcialmente implementado, este foi desenvolvido por alunos da FEUP, na unidade curricular de Laboratório de Gestão de Projecto (LGP). Estes desenvolveram as seguintes funcionalidades e objectivos:

- Interfaces Apropriadas
- Lançamento de Alertas
- Integração de Sensores de RFID
- Integração básica de Sensores de Videovigilância (apenas *Streaming de vídeo*)

As imagens seguintes mostram a aparência das interfaces do projeto inicial. A Figura 2.1, mostra a interface básica, onde o administrador tem uma visão geral do hotel e dos valores associados a cada sensor. A figura 2.2 mostra como será a interface caso seja lançado algum alerta, ou seja, quando algum dos valores sensoriais está fora dos limites estipulados.

Revisão Bibliográfica

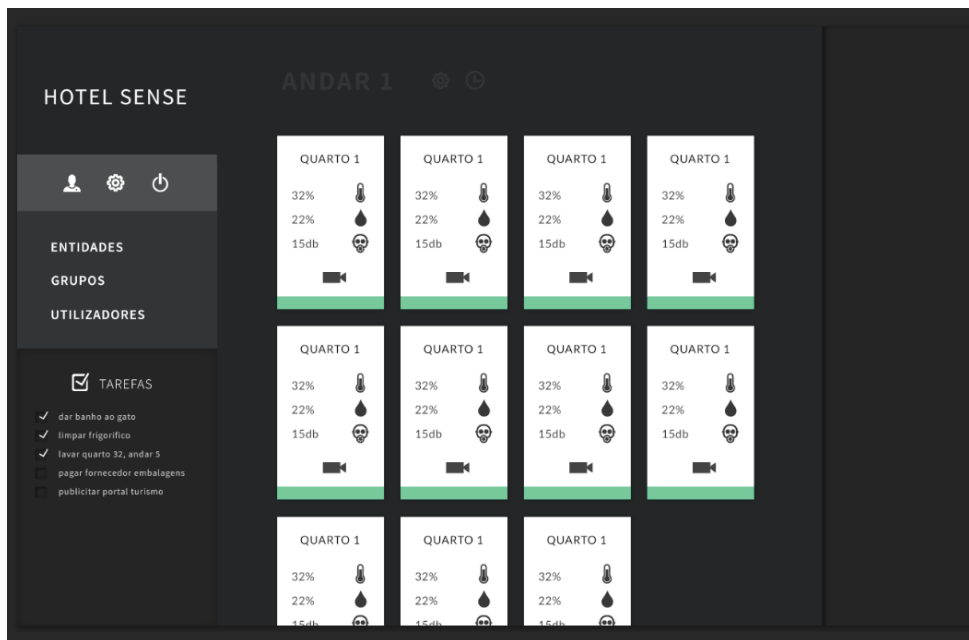


Figura 2.1: Mockup da interface do sistema

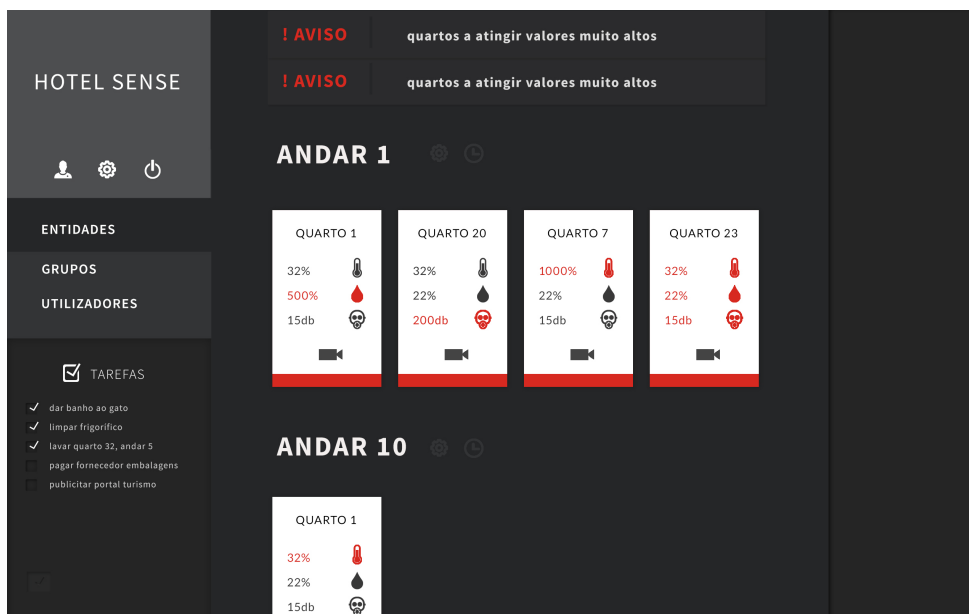


Figura 2.2: Mockup da interface do sistema com alertas

2.1.1 Arquitectura da Solução

Quanto à arquitetura da aplicação HotelSense, como mostra a figura 2.3, divide-se em 3 camadas principais:

Revisão Bibliográfica

- *Universal Gateway* -A *Universal Gateway* trata-se do dispositivo que funciona como ponte entre os sensores e o servidor. É responsável pela receção dos dados dos sensores e do seu envio ao servidor.
- *Servidor* -O *Servidor* é a máquina responsável pelo funcionamento da aplicação. O componente *Servidor de Dados* armazena todos os dados recebidos dos sensores através do *Universal Gateway* e dados dos Clientes (Agentes). O componente *Servidor Web* é responsável pelo processamento dos pedidos efectuados pelos Clientes e apresentação dos resultados através de um conjunto de páginas *Web*.
- *Cliente* - O cliente trata-se do computador pelo qual um utilizador visualiza e interage com a aplicação a correr no servidor através de um *Browser* compatível com *Javascript*, *CSS* e *XHTML*. As funcionalidades disponíveis são influenciadas pelas permissões do utilizador.

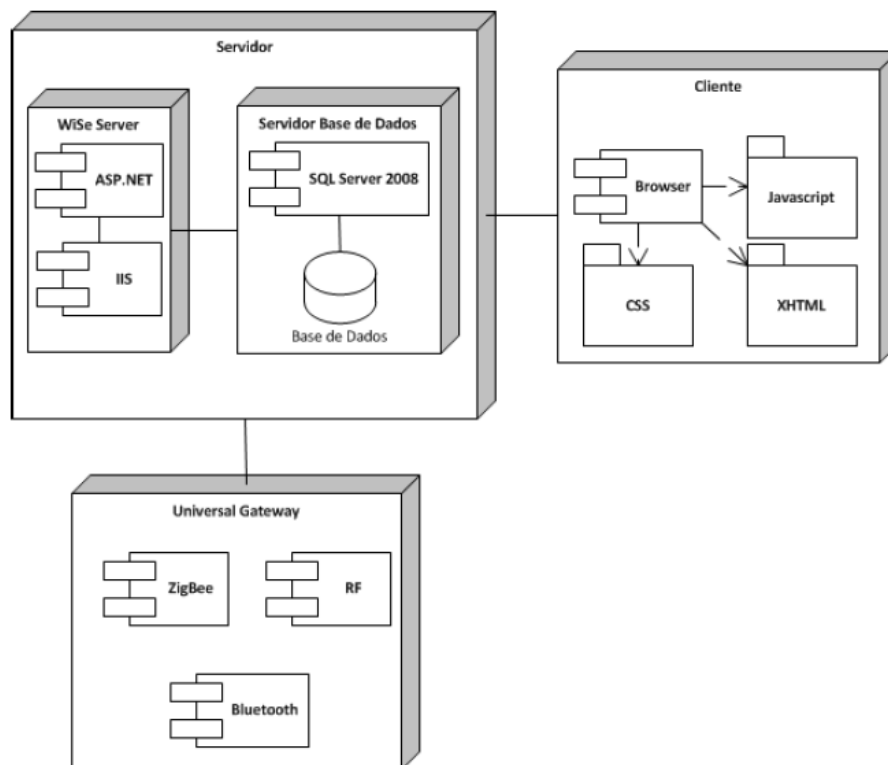


Figura 2.3: Arquitetura da Aplicação

2.2 Aplicações Concorrentes

Esta secção tem como objetivo mostrar os prós e contras das aplicações existentes neste momento no mercado capazes de competir com a HotelSense.

2.2.1 Hotel Systems



Figura 2.4: Hotel Systems

O Hotel Systems é uma aplicação desenvolvida pela Axxess industries, e tem como principal objectivo melhorar a performance dos funcionários dos hotéis, usando 4 sistemas integrados numa única plataforma, os 4 sistemas são:

- Sistema de Privacidade - Verifica qual o estado dos hóspedes (disponível, não incomodar), substituindo assim os tradicionais cartões pendurados à porta.
- Sistema de Presença - Verifica se os quartos estão ocupados, este sistema é útil para o processo de limpeza, evitando incomodar os hóspedes.
- Mini-bar- Verifica quais os quartos que usaram o Mini-bar, facilitando a reposição dos stock.
- Detenção de Tabuleiros- Verifica se os Tabuleiros já estão disponíveis para levantar, evitando a acumulação de tabuleiros nos corredores. [Axx13]

Esta aplicação apesar de ter sistemas que atualmente o HotelSense não possui, que é o caso do Minibar e detecção de tabuleiros, esta peca por defeito, comparativamente ao HotelSense, em número de funcionalidades e em número de sensores instalados.

2.2.2 Intelligent Hotel



Figura 2.5: Intelligent Hotel

O Intelligent Hotel é uma aplicação desenvolvida pela HDL, esta é direccionada mais para o conforto dos hóspedes. Esta aplicação proporciona enumeras funcionalidades aos seus clientes, tais como:

- Controlo do ambiente do quarto (Temperatura, luz, humidade)
- Sistema de privacidade-Verifica qual o estado dos hóspedes (disponível, não incomodar), substituindo assim os tradicionais cartões pendurados à porta. [HDL13]

2.3 Sensores de Videovigilância

Abaixo seram demonstrados os sensores de videovigilância wi-fi disponíveis, tendo em consideração o facto de este ser obrigatoriamente um sensor da marca Axis.

2.3.1 AXIS M1034-W

A câmara com a designação de AXIS M1034-W, é uma câmara pequena e com maior mobilidade, rodando nos dois eixos (horizontal e vertical). Esta câmara suporta Alta-Definição 720p / 1MP e tem uma grande capacidade de deteção (movimento e ruído), usando para tal sensor PIR, que permite também a deteção mesmo no escuro. Usa uma ligação de rede sem fios, um *Wi-Fi Protected Setup* que garante o fácil manuseio pelo utilizador e uma alta segurança de rede, o que permite que o vídeo *HDTV* seja distribuído de forma confiável pela rede sem fio. A sua instalação é fácil e flexível. Outros recursos detetáveis nesta câmara é a Segurança de rede usando proteção por senha de vários níveis, filtragem de endereços *IP* e criptografia *HTTPS*, suporte à Qualidade de Serviço (QoS), que ajuda a proteger a largura de banda necessária para o fluxo de vídeo e comandos de controle pela rede e uma potente Interface de programação de aplicativo (API) para a integração de software, que inclui *AXIS VAPIX* e o *AXIS Media Control SDK*. Memória *Flash* para o carregamento de aplicativos embutidos. [AXI13c]

2.3.2 AXIS M1033-W

A câmara com a designação de AXIS M1033-W é uma câmara pequena e com maior mobilidade, rodando nos dois eixos (horizontal e vertical). Esta câmara suporta Alta-Definição 720p / 1MP e tem uma grande capacidade de deteção (movimento e ruído), usando para tal sensor PIR, que permite também a deteção mesmo no escuro. Usa uma ligação de rede sem fios, um *Wi-Fi Protected Setup* que garante o fácil manuseio pelo usuário e uma alta segurança de rede, o que permite que o vídeo *HDTV* seja distribuído de forma confiável pela rede sem fio. Possui microfone e alto-falante, que permite tanto entrada como saída de som. A sua instalação é fácil e flexível. Outros recursos detetáveis nesta câmara é a Segurança de rede usando proteção por senha de vários níveis, filtragem de endereços *IP* e criptografia *HTTPS*, suporte à Qualidade de Serviço (QoS), que ajuda a proteger a largura de banda necessária para o fluxo de vídeo e comandos de controle pela rede e uma potente Interface de programação de aplicativo (API) para a integração de software, que inclui *AXIS VAPIX* e o *AXIS Media Control SDK*. Memória *Flash* para o carregamento de aplicativos embutidos. Esta câmara tem uma desvantagem relativamente à anterior, uma vez que a sua resolução é bastante menor. [AXI13b]

2.3.3 AXIS M1054

A câmara com a designação de AXIS M1054 é uma câmara com qualidade de imagem *HDTV*, o que lhe permite vários fluxos em *H.264* e *Motion JPEG* podem ser fornecidos simultaneamente individualmente otimizados de acordo com as necessidades de qualidade e com as restrições de largura de banda. Esta câmara contém um sensor *PIR* e *LED* de iluminação, que permite também a deteção mesmo no escuro. Possui microfone e alto-falante, que permite tanto entrada como saída de som. Esta câmara pode ser alimentada por meio de *Ethernet* ou fonte de alimentação externa. A *AXIS M1054* pode ser usada facilmente em lojas, restaurantes, hotéis e residências. Contém também Porta de E/S, para conexão com dispositivos externos. A desvantagem desta câmara, é que não é integrada de sistema sem fios. [AXI13d]

2.3.4 AXIS M1011-W

A câmara com a designação de *AXIS M1011-W* é uma câmara com qualidade de imagem *HDTV*, o que lhe permite vários fluxos em *H.264* e *Motion JPEG* e *MPEG-4*. Possui uma varredura progressiva, o que permite montar o conteúdo de uma tela. O seu design é funcional e inteligente e permite conectividade sem fios. A sua instalação, tal como nas câmaras anteriores, também é fácil e flexível. Os pontos fracos desta câmara são a fraca resolução de vídeo e o não suporte de áudio. [AXI13a]

2.3.5 Largura de Banda nos Sistemas de Videovigilância

De uma forma muito sucinta, Largura de Banda é a medida da capacidade de transmissão de um determinado meio, conexão ou rede, determinando a velocidade que os dados passam através

desta rede específica. A Largura de Banda é medida em bits, e não em *bytes*, os quais determinam a medida de capacidade de um determinado meio de transmissão por certa unidade de tempo ,ou seja, *bits* por segundo, por exemplo, Kbits/seg ou Mbits/seg. [Web13]

Uma vez que o projeto usa redes de Sensores Sem Fios (RSSF), é necessário ter em atenção a largura de banda da rede. Neste caso os sensores que irão ter mais impacto, nesta componente, são os sensores de videovigilância. Os principais fatores que afetam a largura de banda nos sistemas de videovigilância, são:

- Número de câmeras;
- Método de gravação (contínuo ou por eventos);
- Número de horas por dia em que a câmera realizará gravações;
- Frames por segundo;
- Resolução de imagem;
- Tipo de compactação de vídeo: *Motion JPEG, MPEG-4, H.264*;
- Complexidade da imagem; [Axi13e]

Uma vez que vários fatores que afetam a largura de banda não podem ser manuseados, torna-se importante focar nos que podem ser alterados, como por exemplo o método de gravação e número de horas de gravação.

2.4 Fusão de Dados

Uma rede de sensores sem fio (RSSF) pode ser projetada com objetivos diferentes. Esta pode ser concebida para recolher e processar os dados provenientes de um ambiente de modo a ter uma melhor compreensão do comportamento da área monitorizada. Esta pode também ser concebida para desencadear um conjunto de ações apropriadas, quando surge no ambiente determinados eventos. [KM11b] A questão fundamental em redes de sensores sem fios (RSSF) é a maneira de processar os dados recebidos. Nesta situação, a fusão de dados surge como uma disciplina que se preocupa com a forma como os dados recolhidos pelos sensores podem ser processadas para aumentar a importância da quantidade de dados. Assim, a fusão de dados pode ser definida como a combinação de múltiplas fontes de dados para obter melhor informação , mais barato, e de maior qualidade.

Com a ajuda do processo de agregação de dados, podemos aumentar a robustez e precisão das informações que são obtidas por toda a rede, além de reduzir tráfego de rede e obter uma melhor eficiência energia dos sensores.

Os dados obtidos pela fusão podem ser caracterizados em: [Ahm12]

- Dados Complementares;

- Dados Redundantes;
- Dados Cooperativos;

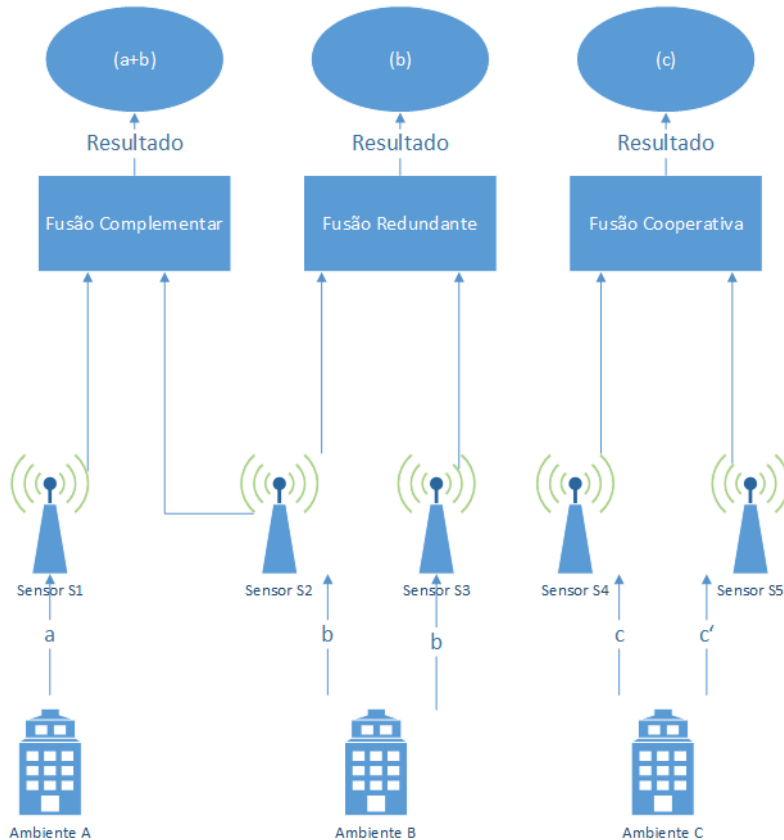


Figura 2.6: Fusão Complementar, Redundante e cooperativa (adaptado de [Ahm12])

2.4.1 Dados Complementares

Este tipo de dados são obtidos a partir de da fusão de 2 ou mais sensores distintos que não depende um do outro, em que a fusão da informação dos sensores resulta numa nova informação mais complexa e completa. A figura 2.6, representa um exemplo de dados complementares, em que do sensor S1 envia uma determinada informação a e o sensor S2 envia uma informação b e a fusão resulta em ab, ou seja, podemos usar informações distintas de diferentes fontes para obter uma nova informação. [Ahm12]

2.4.2 Dados Redundantes

São considerados dados redundantes se duas ou mais fontes independentes fornecem os mesmos dados, estes são fundidos o que é bastante importante para confirmar determinada informação,

dando uma maior confiança aos dados. A figura 2.6 , representa um exemplo de dados redundantes, em que o Sensor S2 e o Sensor S3 observam o mesmo ambiente e retornam os mesmos dados. [Ahm12]

2.4.3 Dados Cooperativos

Este tipo de dados utilizam a informação fornecida pelos dois sensores para obter a informação que não estaria disponível apenas na leitura individual dos sensores, ou seja, ambos os sensores observam o mesmo ambiente, e com a sua fusão consegue-se obter uma vista sobre esse ambiente que não seria possível apenas pela observação dos sensores individualmente. A figura 2.6 mostra esse tipo de fusão quando o sensor S4 e o sensor S5 recolhem c e c' do mesmo ambiente. [Ahm12]

2.5 Arquiteturas Básicas de Fusão de Dados

Existem várias arquiteturas básicas de fusão de dados, sendo as 3 principais: [PDN04]:

- Arquitetura centralizada;
- Arquitectura descentralizada;
- Arquitetura hierárquica;

2.5.1 Arquitetura Centralizada

De acordo com a figura 2.7 todos os sensores processam os seus dados independentemente e enviam diretamente para o nó destino, o qual se encarrega de fazer a fusão destes dados. Neste caso, uma grande quantidade de mensagens pode ocasionar um grande número de colisões no nó destino, porém esta técnica de fusão possibilita que um único nó tenha o poder de processamento necessário para executar a fusão de dados, podendo os demais sensores executarem apenas as atividades de leitura e envio do pacote ao destino. [KM11a]

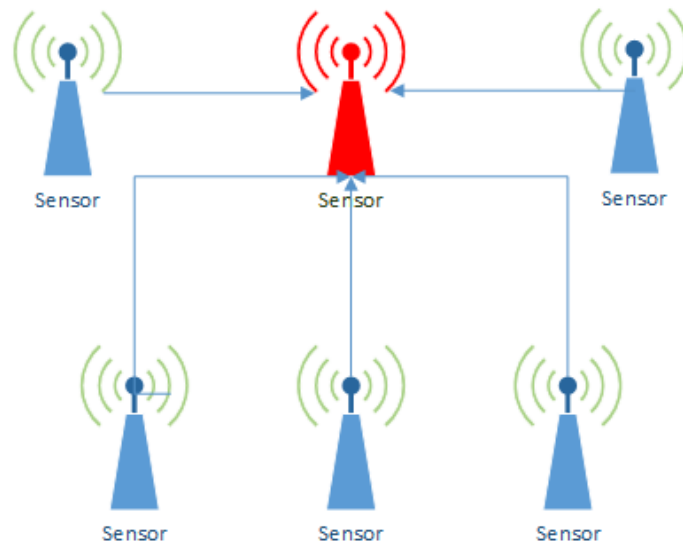


Figura 2.7: Arquitetura Centralizada
(adaptado de [KM11b])

2.5.2 Arquitetura Descentralizada

Neste tipo de arquitetura não existe um nó central que faz a fusão de toda a informação, como mostra a figura 2.8, cada sensor fica com a responsabilidade de receber a leitura proveniente de outro sensor, ler, agregar a sua própria leitura e executar um algoritmo de fusão de dados para encaminhar para o nó seguinte apenas um único pacote contendo apenas uma única leitura resultante da sua fusão de dados. [KM11a]

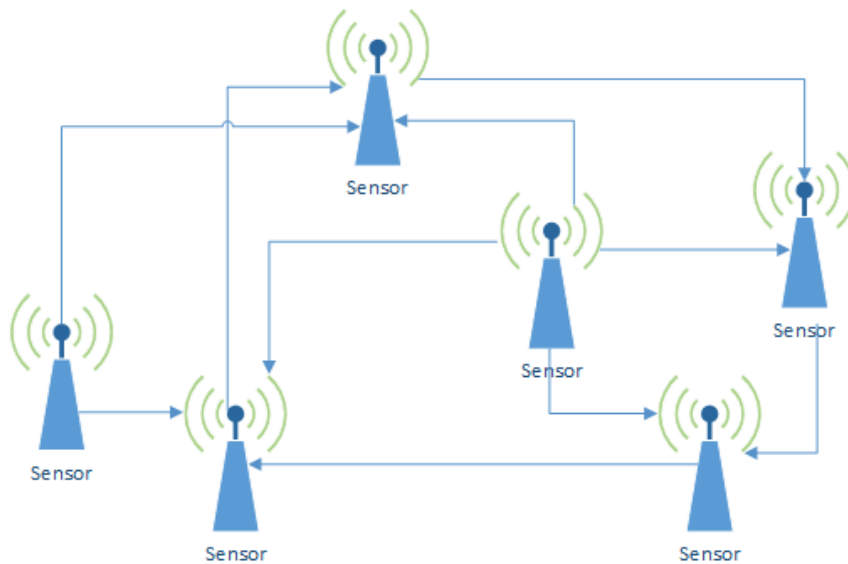


Figura 2.8: Arquitetura Descentralizada
(adaptado de [KM11b])

2.5.3 Arquitetura Hierárquica

Como mostra a figura 2.9, esta arquitetura utiliza as potencialidades das arquiteturas centralizadas e descentralizadas, organizando os nó hierarquicamente. A sua formação é composta de um grupo de nó (um cluster) o qual possui um nó central denominado *cluster head*. Após o *cluster head* receber as mensagens, encaminha os dados recebidos dentro do seu grupo para o *cluster head* raiz, o qual será responsável por tomar a decisão final. O processo de fusão de dados ocorre em cada *cluster head*. [KM11a]

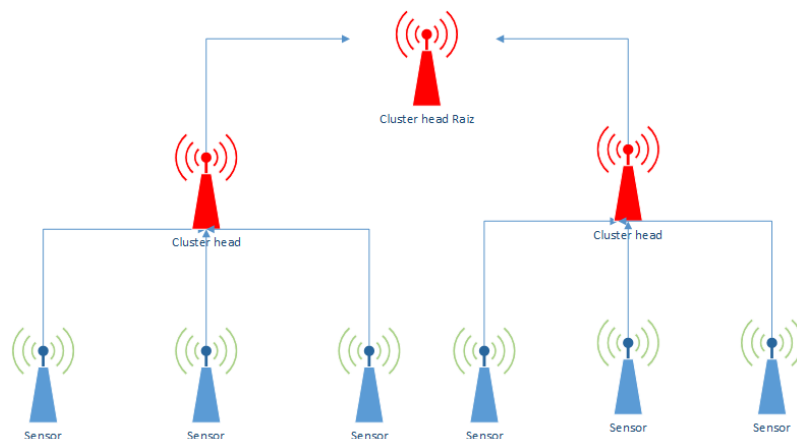


Figura 2.9: Arquitetura Hierárquica
(adaptado de [KM11b])

2.6 Resumo ou Conclusões

Neste capítulo foi apresentada a plataforma e a aplicação que está na origem desta dissertação e que servirá de ponto de partida, bem como as aplicações concorrentes que existem no mercado. Foram também apresentados sensores de videovigilância e quais os principais fatores que afetam a largura de banda. Por fim foi feita uma introdução a temática fusão de dados, pois esta vai ser útil para a resolução do problema proposto.

Revisão Bibliográfica

Capítulo 3

Problema

Neste capítulo é apresentado o problema que motivou esta dissertação e explicada a solução implementada para o resolver.

3.1 Descrição

Portugal é um dos destinos na Europa com crescimento mais alinhado com os princípios do desenvolvimento sustentável, alavancado numa proposta de valor suportada em características distintas e inovadoras do país.

O turismo deve desenvolver-se com base na qualidade do serviço e competitividade da oferta, tendo como motor a criação de conteúdos autênticos e experiências genuínas, na excelência ambiental e urbanística, na formação dos Recursos Humanos e na dinâmica e modernização empresarial e das entidades públicas.

A importância do turismo na economia deve ser crescente, constituindo-se como um dos motores do desenvolvimento social, económico e ambiental a nível regional e nacional. [dP14a]

O setor do Turismo apresenta-se como um dos setores onde a especialização, a inovação e a qualificação assumem particular relevo e importância para a competitividade das empresas. [dP14b]

Segundo a Organização Mundial do Turismo (OMT), as receitas do turismo internacional à escala global aumentaram 4% em 2012, atingindo um novo recorde de 837 mil milhões de euros, segundo a mesma organização (OMT) Portugal em 2012 registou 8,6 mil milhões de euros em receitas turísticas, o que representou um crescimento de 5,6% face ao ano anterior. [dN13].

As últimas três décadas são caracterizadas por um crescimento significativo do número de turistas que procuram alojamentos de baixo custo, um pouco por todo o Mundo. O que inicialmente se considerava ser um fenómeno associado aos turistas Backpackers ou jovens mochileiros, é atualmente compreendido por viajantes de diferentes classes sócio-económicas. Portugal não é exceção e tem vindo a acompanhar, desde há dez anos, esta evolução. Prova disso, é o nascimento e desenvolvimento acentuado de um novo segmento de indústria hoteleira «low-cost»: os

Hostels. [Vol11] Exemplo disso é o crescimento exponencial do número de hostels na cidade do Porto, que nos últimos 5 anos duplicou o número de hostels. [Pub10]

A Freedomgrow decidiu investir no turismo low-cost, visto que é um nicho de mercado em expansão, sendo o desafio desenvolver uma aplicação low-cost agregadora de sistemas de monitorização, críticas para um serviço de hotelaria, de elevado valor acrescentado e obedecendo aos mais rigorosos padrões de qualidade, de forma a obter toda a informação recolhida pelos diversos sistemas de monitorização, numa única plataforma, acessível por qualquer dispositivo fixo ou móvel, para implementar em hotéis de pequenas dimensões, hostels, pousadas entre outras.

Tendo em conta que todos os sensores usados para a monitorização do hotel, são sensores sem fios, e que os sensores de videovigilância necessitam de uma grande largura de banda, torna-se extremamente importante ter uma ligação de rede robusta e fiável. O objetivo do projeto é desenvolver uma aplicação que seja capaz de reduzir custos de instalação de rede e espaço de armazenamento do sistema de videovigilância. A forma mais eficaz de reduzir a largura de banda consumida por estes equipamentos é diminuir a quantidade de informação transmitida, neste caso, pretende-se diminuir o número de imagens transmitidas. Para tal, irão ser usadas informações de outros sensores, que consomem pouca largura de banda, para decidir quando transmitir imagem.

Além da redução do consumo de largura de banda pretende-se melhorar a qualidade da informação transmitida pelos sensores de videovigilância, para tal pretende-se reduzir/eliminar áreas desnecessárias e focar a imagem no que realmente é importante, como por exemplo, eliminar paredes brancas/sem informação.

3.2 Proposta de Valor

Este projeto é muito importante para a FreedomGrow, uma vez que se trata de uma das áreas de negócio onde a empresa se quer focar a curto prazo. O Turismo é uma área em constante crescimento, tendo nos últimos anos atingido valores recorde de receitas. Portugal não é exceção e o Turismo consegue ser uma das poucas áreas onde ao longo dos últimos anos tem havido crescimento [dN13].

Com estes importantes dados de mercado, a FreedomGrow tem interesse em incluir na sua oferta uma solução vocacionada para o setor da Hotelaria, onde se possa ter numa aplicação de monitorização os dados lidos de diversos sensores, bem como informação proveniente de sistemas de videovigilância.

Contudo, tratando-se apenas de um dos mercados verticais da empresa, a solução a desenvolver neste projeto será vocacionada para hotéis de pequena dimensão, tendo a solução que funcionar em locais onde normalmente existem algumas limitações ao nível da largura de banda disponível. Grande parte do trabalho deste projeto é a componente de gestão da largura de banda consumida pela videovigilância e acaba por ser um dos aspetos mais importantes desta solução de monitorização, uma vez que os constrangimentos causados na ligação à Internet dos hotéis serão minimizados ou mesmo anulados. Com esta componente de gestão de largura de banda, as câmaras irão enviar

streaming de vídeo somente quando for necessário, ou seja, quando os sensores que estão no sistema enviarem informação que indique situações anormais.

Desta forma consegue-se uma poupança no consumo de largura, o que é muito valioso para o sistema funcionar na ligação à Internet do próprio hotel, sem obrigar a instalar uma rede dedicada para esta solução. Com este aspeto importante, consegue-se fazer da HotelSense, um produto ainda mais atrativo para os potenciais clientes, pois este fator de rejeição é atenuado ou mesmo eliminado em alguns casos.

3.3 Solução

Como referido anteriormente o principal objetivo do projeto é gerir e otimizar a largura de banda e espaço de armazenamento do sistema de videovigilância, bem como a qualidade da imagem guardada. Para tal foram usadas informações disponíveis de outros sensores para alcançar os objetivos. Tendo em conta os objetivos, o projeto foi dividido em duas componentes:

- Otimização da Largura de Banda;
- Otimização da Qualidade da Informação

Embora tenham sido diferenciadas, estas componentes estão relacionadas, como será mostrado posteriormente.

3.3.1 Otimização da Largura de Banda

Sendo a maioria dos sistemas de videovigilância, sistemas que estão a transmitir/guardar informação 24 horas por dia, e estes requererem grandes requisitos, como por exemplo, largura de banda e espaço para armazenamento, e tendo em conta que os proprietários dos pequenos e médios Hotéis, desejam um sistema fiável, mas com baixo custo de instalação e gestão. Decidiu-se então criar um sistema que apenas transmite informação quando realmente é importante, ou seja, apenas quando há algum tipo de atividade, reduzindo assim uma enorme quantidade de informação transmitida/guardada.

Para obter este sistema foi necessário realizar uma interligação dos vários sensores, com os sensores de videovigilância, ou seja, sempre que algum dos sensores instalados deteta valores fora dos limites (Superior e Inferior), que neste caso são chamados alertas, estes enviam um sinal para a camera e esta inicia a transmissão/gravação.

3.3.2 Otimização da Qualidade da Informação

Quando as cameras do sistema de videovigilância são instaladas, estas tem uma "área de visão" sobre um determinado compartimento, e grande parte do tempo estão a transmitir imagens sem informações relevantes, como por exemplo, áreas sem atividade, paredes.

Após discussão com o orientador da Freedomgrow, decidiu-se criar as seguintes funcionalidades:

Problema

- Criação de um posição *home* para cada camera;
- Criação de um sistema de posicionamento para cada sensor, relativo à camera ao qual este está associado.

3.3.3 Posição Home

Usando as funcionalidades adjacentes às cameras implementas na aplicação, foi criada a funcionalidade para definir uma posição *home*, para cada camera. Com esta funcionalidade o administrador, através da interface da aplicação desenvolvida tem a opção de definir a posição *home* de uma forma simples e intuitiva usando um sistema *joystick* criado para manusear a camera, permitindo assim eliminar áreas desnecessárias.

3.3.4 Sistema de Posicionamento Sensorial

Como visto anteriormente a aplicação apenas irá gravar imagens, quando detetar alguma atividade sensorial fora dos limites, é aqui que entra o sistema de posicionamento Sensorial. No sistema, cada sensor possui uma posição relativa à camera à qual está associado, com a informação da posição do sensor. A camera quando receber a informação de alerta vai rodar para a posição do sensor e iniciar a gravação.

3.3.5 Resumo ou Conclusões

Neste capítulo foi descrito o problema de forma mais detalhada e foi apresentada uma proposta para a sua solução. Foram também enumeradas as funcionalidades expectáveis da solução de modo a cumprir os objetivos.

Capítulo 4

Conceção

Nesta seção importa explorar o processo de conceção da solução. Vão ser apresentados os requisitos que foram definidos para cada módulo bem como a forma e os recursos para os atingir.

4.1 Arquitectura da Solução

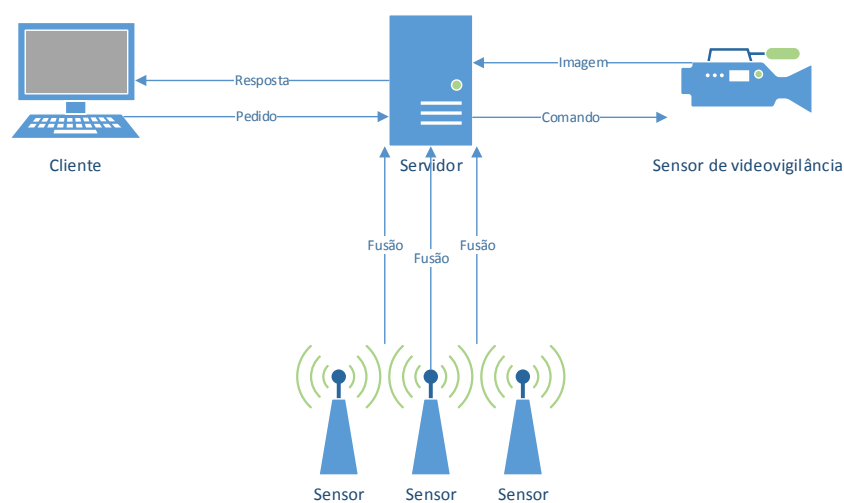


Figura 4.1: Arquitectura da solução

Na figura 4.1 está representado o esquema geral da arquitetura. Os sensores recolhem dados dos ambientes que estão a observar, enviam os mesmos para o servidor que realiza a fusão sensorial. Sempre que recebe um valor fora dos limites estipulados (superior, inferior), ou seja um alerta, este envia um comando para o sensor de videovigilância para efetuar a gravação. O sensor de videovigilância, quando recebe o comando inicia o envio de imagens (Gravação) para o servidor, que por sua vez envia para o utilizador quando este solicita. Desta forma pretende-se

Conceção

conseguir uma redução do tempo de gravação e do espaço de armazenamento do sistema de videovigilância, conseguindo-se assim o objetivo principal do projeto, ou seja, uma otimização da largura de banda, sem afetar o bom funcionamento de todo os sistema. A figura 4.2, mostra de uma forma sucinta o fluxo de dados entre os componentes.

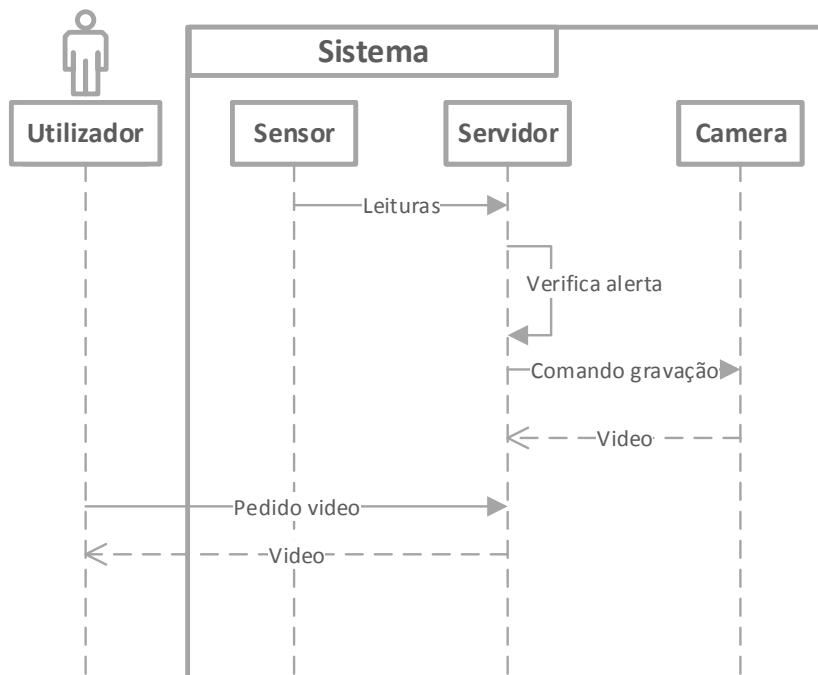


Figura 4.2: Fluxo de Dados entre Componentes

4.2 Casos de Uso

Na aplicação existente, desenvolvida anteriormente, o utilizador pode realizar os caso de uso assinalados a negro na figura 4.3. Os casos de uso assinalado a vermelho são os casos de uso que o utilizador possui após a implementação da solução. No caso de uso "Monitorização", o utilizador pode aceder sempre que achar necessário ao *streaming* de vídeo, para verificar por exemplo o que se passa no momento num determinado local, bem como interagir com as funcionalidades dos sensores de videovigilância(Rodar Camera, Fazer Zoom, Definições). No caso de uso "Ver Vídeo Alerta"o utilizador pode aceder/ver os vídeos guardados de cada alerta arquivado no histórico de alertas. No caso de uso "Definição de Coordenadas", o utilizador atribuir a cada sensor uma posição relativa ao sensor de videovigilância.

Conceção

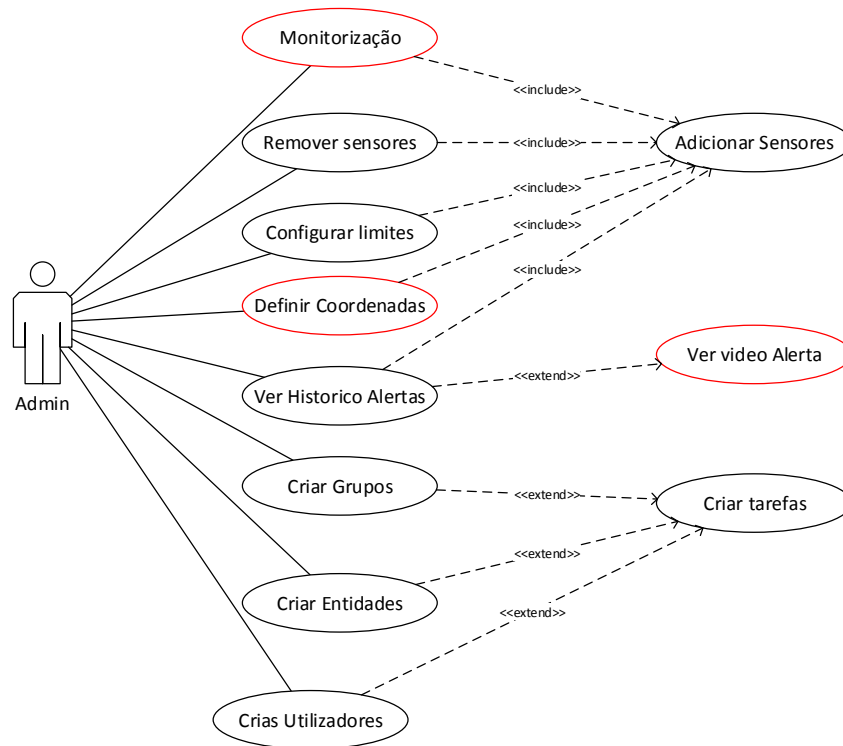


Figura 4.3: Caso de uso - Administrador

4.3 Requisitos

Neste secção serão mostrados os requisitos da aplicação, bem como a abordagem e os requisitos para os alcançar

4.3.1 Integração dos Sensores de Videovigilância

Uma vez que a Plugsense ainda não possui sensores de videovigilância integrados, é necessário realizar essa integração. Os sensores que neste momento a Freedomgrow possui e deseja que sejam integrados, são: um sensor *Axis M1034-W* e um sensor *Foscam FI8918W*.

Para realizar a integração dos sensores é necessário recorrer à documentação das API's disponíveis para cada sensor.

Para a integração do sensor Axis será usada a VAPIX API [Axi13f]. Quanto ao sensor Foscam será usada a API específica. [Fos13]

4.3.2 Otimização da Qualidade da Informação

Para alcançar este requisito, é necessário a criação de um sistema posicional para os sensores. Cada sensor irá possuir coordenadas relativas ao sensor de videovigilância. Com este tipo de

Conceção

abordagem consegue-se melhorar a qualidade da informação transmitida/guardada, pois sempre que ocorre um alerta num determinado sensor, a camera efetua a rotação para a posição do sensor transmitindo assim apenas informação relevante.

As coordenadas de cada sensor são definidas pelo administrador do sistema, usando a interface do sensor de videovigilância implementado no sistema, para posicionar a camera. Quando este posiciona a camera no local pretendido para um determinado sensor, atribui essas coordenadas ao sensor. O diagrama representado na figura 4.4, mostra a sequência de operações necessárias para a atribuição de coordenadas aos sensores pretendidos.

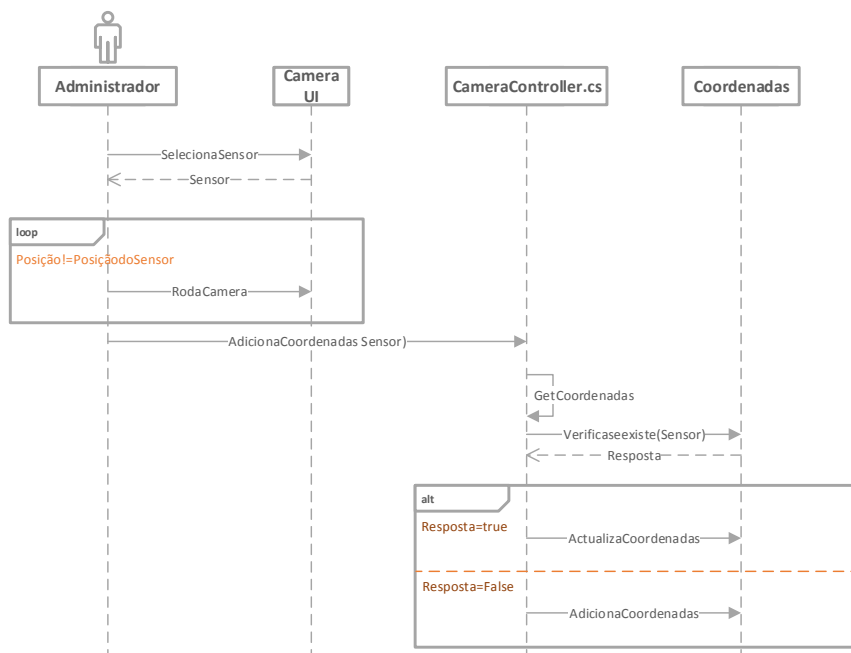


Figura 4.4: Diagrama de Sequência -Adicionar Coordenadas

4.3.3 Otimização do Consumo da Largura de Banda e do Espaço de Armazenamento

Para a realização do objetivo principal do projeto é necessário definir o comportamento do sistema em caso de alerta, ou seja, quais as condições para iniciar a gravação de vídeo. O diagrama representado na figura 4.5, representa a sequência das operações realizadas pelo sistema para realizar a gravação.

Conceção

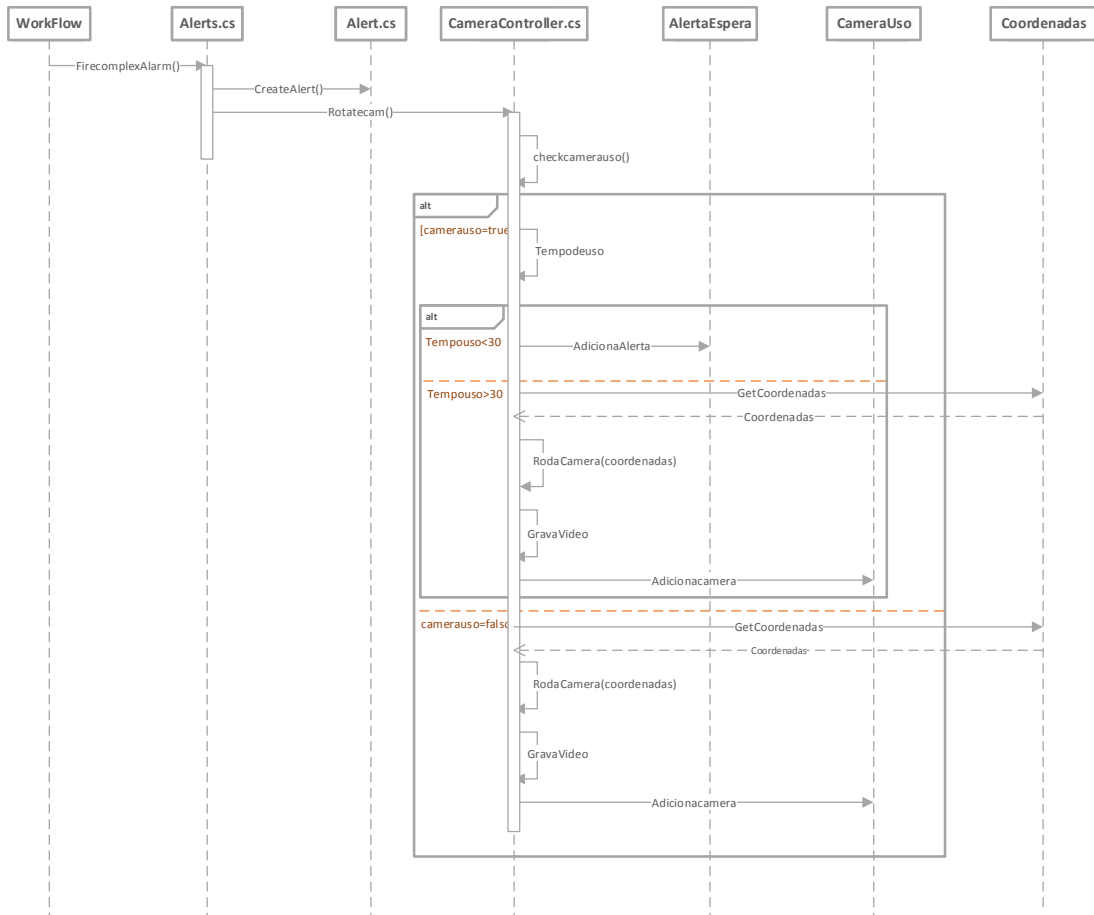


Figura 4.5: Diagrama de Sequência- Alerta

É nos *Work Flows* que são recebidos e tratados os dados, quando ao dados recebidos estão fora dos limites estipulados, é lançado o método *firecomplexalert()* da classe Alerts.cs, este cria o objeto Alerta e invoca o método *rotatecam()* que irá rodar a camera para a posição do sensor e dar inicio à gravação.

O método *rotatecam*, inicialmente verifica se a camera já esta a ser usada, caso esteja a ser usada, verifica se esta a ser usada há menos de 30 segundos(tempo que pode ser definido pelo administrador).Caso esteja, adiciona o alerta aos AlertasEspera para este ser tratado posteriormente. Caso esteja a ser usada a mais de 30 segundos, efetua a rotação da camera para a posição do sensor que lançou o alerta e inicia a gravação.

Caso a camera não esteja a ser usada,efetua a rotação para a posição do sensor que lançou o alerta e inicia a gravação.

Com esta abordagem como se verifica no diagrama representado acima, o sistema consegue tratar múltiplos alertas, adicionando os alertas que não podem ser tratados de imediato a uma fila de espera, intitulada AlertasEspera.

Conceção

Em relação à condição de paragem da gravação, definiu-se que será quando o administrador arquivar o alerta, assim sendo o diagrama representado na figura 4.6, mostra a sequência das operações para a realização da paragem de gravação.

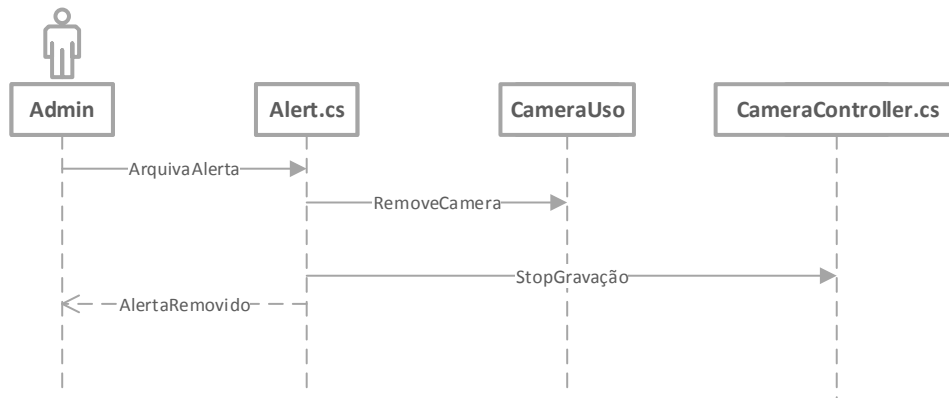


Figura 4.6: Diagrama de sequência- Arquivar Alerta

4.3.4 Visão Geral

De uma forma geral e sucinta, a tabela 4.1, mostra a relação dos requisitos com os casos de uso do utilizador. Quais os requisitos, bem como as funcionalidades necessárias para atingir um determinado caso de uso.

Casos de Uso	Requisitos	Funcionalidades
Monitorização	Integração dos Sensores de Videovigilância	Definir Posição Home Definições do Sensor Definição de Deteções Sistema de Rotação
Definir Coordenadas	Otimização da Qualidade da Informação	Sistema Posicional
Ver Video Alerta	Otimização da Largura de Banda e Espaço de Armazenamento	Rotação do Sensor Gravação de Vídeo

Tabela 4.1: Relação de requisitos com usecases

4.4 Testes Simulados

Uma vez que não existe a possibilidade de testar a aplicação num cenário real, os testes serão realizados nas instalações da FreedomGrow. Será montado um cenário com 2 salas de observação, cada sala irá conter além dos sensores de videovigilância, vários outros sensores, entre os quais:

- Sensores de Temperatura;
- Sensores de Som;
- Sensores de Humidade;
- Sensores de Luminosidade;
- Sensores de Intrusão;

Os sensores mencionados, são sensores que podem ser usados em caso reais, logo é de extrema importância utilizá-los nos testes a realizar.

Serão realizados 2 testes, um de curta duração e um de maior duração. O teste de pequena duração será realizado apenas numa das salas com a duração de aproximadamente 30 minutos, quanto ao teste de maior duração será realizado nas duas salas em simultâneo e com a duração de 4h. Pretende-se com a realização dos testes conhecer o impacto das funcionalidades implementadas na largura de banda e espaço de armazenamento, ou seja, avaliar se a abordagem proposta permite reduzir significativamente consumo de largura de banda e espaço em disco.

4.4.1 Simulação de Monte Carlo

O objetivo da simulação é descrever a distribuição e características dos possíveis valores de uma variável dependente, depois de determinados os possíveis valores e comportamentos das variáveis independentes a ela relacionadas. O Método de Monte Carlo é, portanto, um modelo de simulação que utiliza a geração de números aleatórios para atribuir valores às variáveis que se deseja investigar. Os números podem ser obtidos através de algum processo aleatório. [Eri14]

Após a realização dos testes acima mencionados, será realizada, uma simulação baseada numa simulação de monte Carlo, de forma a conhecer o impacto das implementações a longo prazo. Com este tipo de simulação é possível, utilizando informações retiradas dos testes acima realizados, projetar o impacto das implementações para maiores espaços de tempo. O objetivo da simulação é descrever a distribuição e características dos possíveis valores de uma variável dependente.

Para a realização desta simulação serão usados os seguintes dados.

- **NTA**-Número Total de Alertas;
- **NAA**-Número de Alertas Tipo Aviso;
- **NAP**-Número de Alertas Tipo Perigo;

Conceção

- **LBA**-Consumo de Largura de Banda em Caso de Alerta Tipo Aviso;
- **EAA**-Espaço de Armazenamento em caso de AlertaTipo Aviso;
- **TMA**-Tempo Médio de Arquivo de um Alerta(Perigo) por Parte do Administrador;
- **LBP**-Consumo de Largura de Banda em Caso de Alerta Tipo Perigo;
- **EAP** -Espaço de Armazenamento em Caso de Alerta Tipo Perigo;

A simulação vai ser realizada com o auxílio de uma folha de calculo Excel, onde o valor da variável NTA é obtidos com a ajuda do método *Aleatório()* disponível na folha de calculo. Os valores das variáveis NAA e NAP variam conforme o valor do NTA. Os valores de LBA,LBP,EAA,EAP,TMA serão retirados da análise realizada aos testes de simulação.

Com este tipo de simulação será possível verificar o impacto do número de Alertas no consumo de largura de banda e Espaço de Armazenamento.

4.5 Resumo ou Conclusões

Neste capítulo foram apresentados os planos de conceção das várias funcionalidades a implementar e a forma como vão ser avaliados os resultados.

Capítulo 5

Implementação

Este capítulo serve para descrever os pormenores da implementação. Será descrita a implementação dos diferentes módulos constantes dos requisitos.

5.1 Integração dos Sensores de Videovigilância

No início do projeto foi necessário realizar a integração dos sensores de videovigilância na *framework* da empresa. Foram integrados dois sensores:

- Axis M1034-W;
- Foscam FI8918W;

O projeto inicial, como se verifica na figura 5.1, apenas mostra o *streaming* do sensor Axis. De seguida será detalhada a integração dos dois sensores de videovigilância.

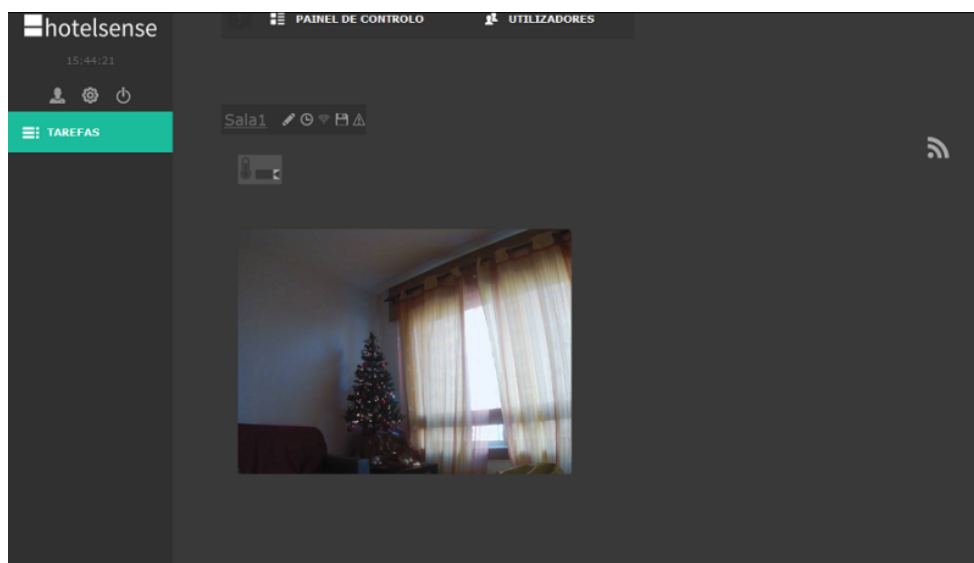


Figura 5.1: Interface do Projeto Inicial

Implementação

5.1.1 Axis M1034-W

Para a integração do sensor Axis M1034-W, foi utilizada a *Vapix api* [Axi13f], esta é uma *api*, que a Axis disponibiliza aos seus parceiros, como é o caso da Freedomgrow. Durante este processo, foram implementadas algumas das principais funcionalidades do sensor, como por exemplo:

- **Video Stream Settings**- O administrador pode configurar vários parâmetros da camera, como por exemplo, resolução, brilho, contraste, entre outras.
- **Detectors**- O administrador pode configurar os valores para os vários tipos de detetores (Movimento, Áudio, etc)
- **Sistema de PTZ(PAN, TILT, ZOOM)**- Uma vez que a camera em causa não possui um sistema mecânico para fazer rotações, é necessário configurar este tipo de sistema.
- **Criação da posição home**-Com esta funcionalidade o administrador pode definir uma posição *home*, aumentando assim qualidade da informação.

A imagem 5.2, mostra a interface após a implementação das funcionalidades acima mencionas.

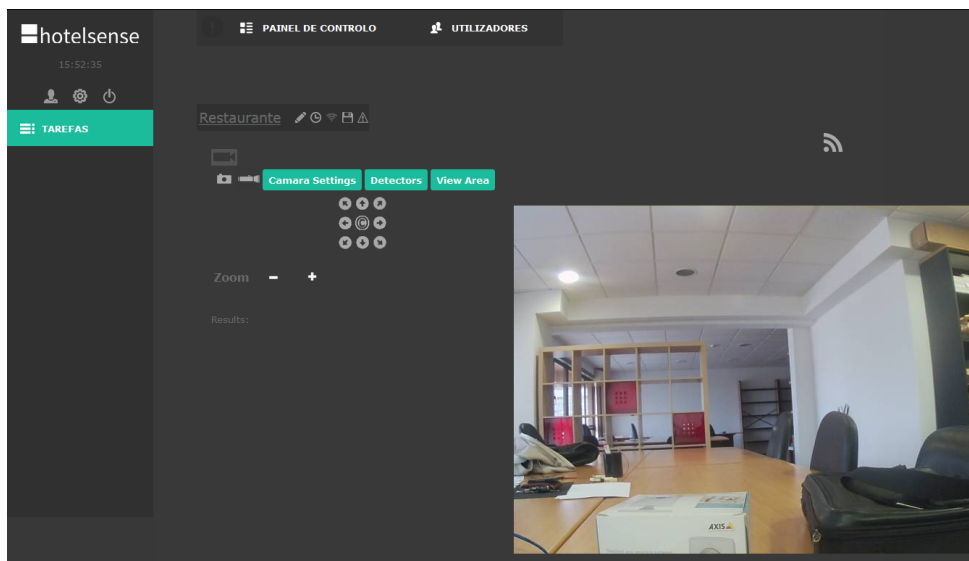


Figura 5.2: Interface - Sensor Axis

5.1.2 Foscam

Para realizar a integração do sensor foscam FI8918W, foi utilizada a documentação disponível da foscam [Fos13]. Durante este processo, foram implementadas algumas das principais funcionalidades do sensor, com por exemplo:

Implementação

- **Video Stream Settings**-O administrador pode configurar vários parâmetros da camera: resolução, brilho e contraste;
- **Detectors**-O administrador pode configurar os valores para os vários tipos de detetores: movimento e áudio
- **Sistema de rotação**- Com esta funcionalidade o administrador pode interagir com o sensor, rodando o sensor nos dois eixos x e y.
- **Configuração do E-mail**-O administrador, pode configurar o e-mail para onde, caso seja definido, o sensor envia imagens ou mensagens de alerta.

A figura 5.3, mostra a interface após a implementação das funcionalidades acima mencionadas.

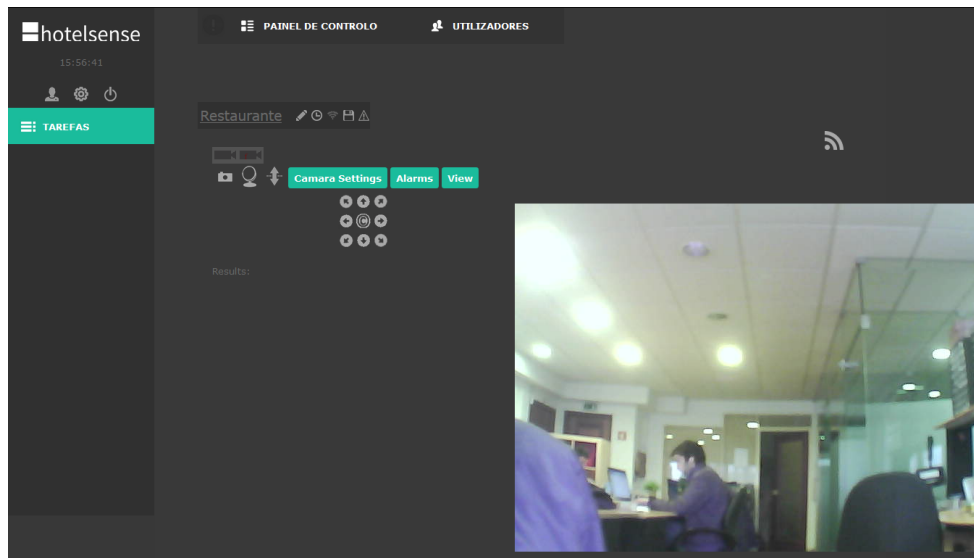


Figura 5.3: Interface - Sensor Foscam

5.1.3 Base de Dados

Para guardar as informações relativas aos sensores de videovigilância, foi necessário adicionar tabelas à base de dados existente. Como se pode verificar na figura 5.4, foram adicionadas 2 novas tabelas, sendo elas: Camera e CameraType. A tabela Camera, é usada para guardar informações relativas ao sensor (ip, user, pass) e ainda informações que são necessárias para a implementação das restantes funcionalidades como é o caso do campo, SensorId, que é usado para implementar as funcionalidades de otimização de largura de banda e qualidade da informação. O campo AgentId é usado para guardar o histórico dos agentes do sensor. O campo CameraTypeId é chave estrangeira usada para ligar à tabela CameraType, que guarda o tipo de sensor, neste caso axis ou foscam.

Implementação

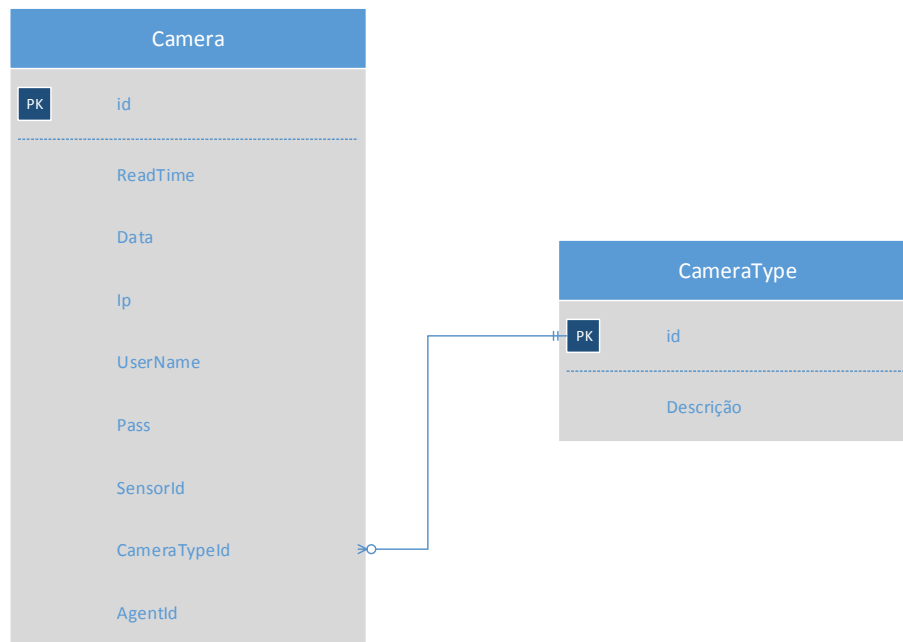


Figura 5.4: Tabelas Camera e CameraType

5.2 Otimização da Largura de Banda/Qualidade da Informação

Para a realização desta componente, como mencionado anteriormente, foi necessário criar as seguintes funcionalidade:

- Fusão de Dados Sensoriais
- Definição da Posição Sensorial.
- Rotação dos sensores para a posição do alerta
- Tratamento de múltiplos Alertas.
- Gravação

5.2.1 Fusão de Dados Sensoriais

A aplicação possui um *Workflow* para cada sensor, este recebe e trata os dados enviados pelos sensores. Sempre que algum dos valores recebidos está fora dos limites (superior ou inferior) este "cria" um Alerta, que posteriormente é tratado por uma classe (*Alerts.cs*). É nesta classe, que é feita a fusão sensorial, por parte da aplicação.

Visto que estamos perante uma arquitetura centralizada, todos os sensores enviam os alertas, para a aplicação, e esta decide qual posteriormente irá tratar. É na classe *Alerts.cs*, que está definida a

Implementação

função *fireComplexAlarm()*, que é invocada pelos *Workflows*. É neste método que é criado e adicionado à base de dados o objeto *Alerta*, é também neste método, como se pode verificar no excerto de código abaixo, que é invocada a função *rotatecam(Alerta)*. Como irá ser mostrado posteriormente, esta função é responsável por realizar as rotações, iniciar a gravação e o tratamento de múltiplos alertas.

```
1     temp.Add(parameter, level.ToString());
2     // New Alert
3     Alert alerta = new Alert(sensor, serie, sensor.Agent, name, parameter,
4         message, level, value.ToString());
5     alerta.CreateAndFlush();
6     CameraController.rotatecam(alerta, 0);
7     mail = true;
```

5.2.2 Definição de Posições

Após discutir com o *product owner* qual seria a melhor forma de integrar o sistema de posições na aplicação, chegou-se à conclusão, que a melhor opção seria atribuir para cada sensor, coordenadas relativas ao sensor de videovigilância. As coordenadas neste caso são os valores que os sensores de videovigilância usam para definirem áreas, ou seja, PTZ para os sensores Axis e a posição para os sensores Foscam.

Para o administrador definir as posições para cada sensor, foi criada uma interface na aplicação, ver figura 5.5. Na interface o administrador utiliza o *joystick* juntamente com os botões de *Zoomin* e *Zoomout*, para rodar e dar profundidade à imagem, quando este achar que a camera está apontar para o sensor escolhido, clica no botão *CreateView*, definindo assim as coordenadas do sensor em questão.

Implementação

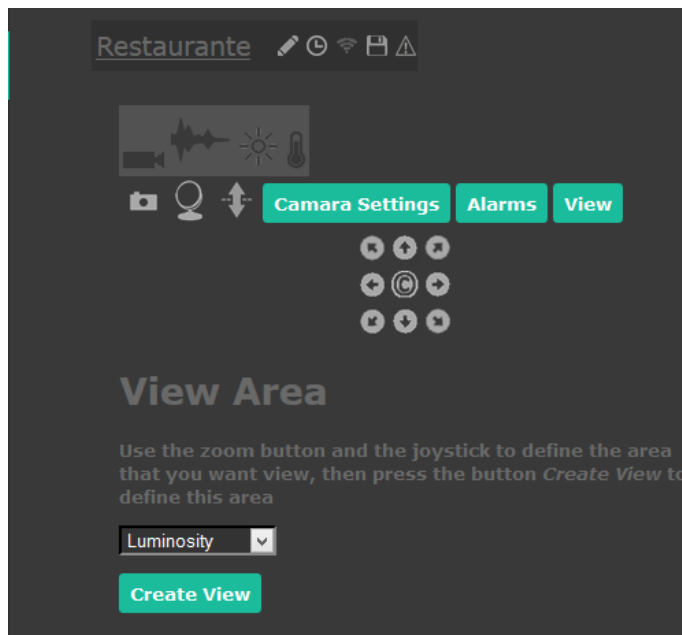


Figura 5.5: Interface -Adicionar Coordenadas

Base de Dados

Para a criação desta funcionalidade foi necessário adicionar a tabela Coordenadas à base de dados para guardar as coordenadas de cada sensor, a figura 5.6 mostra a tabela.

Coordenadas	
PK	Id
	Zoom
	Tilt
	Pan
	FoscamPos
	SensorId
	CameraId

Figura 5.6: Tabela Coordenadas

Implementação

Quando a sensor está associado a um sensor de videovigilância do tipo Axis, utiliza os campos Zoom, Tilt e Pan, deixando o campo FoscamPos a NULL, quando se trata de um sensor do tipo foscam, ocorre o contrário.

Código

Para implementar esta funcionalidade foi necessário criar duas funções, uma para cada tipo de sensor de videovigilância. O código abaixo mostra cada uma das funções, sendo o primeiro a função para o sensor Axis e a segunda para o sensor foscam.

Axis

```
1 public string addcoordenadas(int camera_id, int sensor_id){
2     Camera cam = Camera.Find(camera_id);
3     Sensor sen = Sensor.Find(sensor_id);
4     string ip = cam.IP;
5     string user = cam.UserName;
6     string pass = cam.Pass;
7     string url = "http://" + ip + "/axis-cgi/com/ptz.cgi?query=position";
8     WebClient web = new WebClient();
9     web.Credentials = new NetworkCredential(user,pass);
10    string result = web.DownloadString(url.ToString());
11
12    string[] lines = Regex.Split(result, "\r\n");
13    string[] lines0 = Regex.Split(lines[0], "pan=");
14    string[] lines1 = Regex.Split(lines[1], "tilt=");
15    string[] lines2 = Regex.Split(lines[2], "zoom=");
16    string pan = lines0[1];
17    string tilt = lines1[1];
18    string zoom = lines2[1];
19    int position = 0;
20    string str = null;
21    PlugSenseApp.Models.Coordenadas coordenadas = new PlugSenseApp.Models.
22        Coordenadas();
23    coordenadas = new Coordenadas(sen, cam, zoom, tilt, pan, position);
24
25    Coordenadas[] cord = Coordenadas.FindAll();
26    foreach (Coordenadas cd in cord){
27        if (cd.Sensor.Id.Equals(sensor_id)){
28            if (cd.Camera.Id.Equals(cam.Id)){
29                cd.Pan = pan;
30                cd.Tilt = tilt;
31                cd.Zoom = zoom;
32                cd.UpdateAndFlush();
33                str = "upload";
34                return str;
35            }
36        }
37    }
```

Implementação

```
35     }
36   }
37
38   coordenadas.CreateAndFlush();
39   str = "new";
40   return str;
41 }
```

Foscam

```
1 public string addcoordenadasfoscam(int camera_id, int sensor_id, int pos){
2   Camera cam = Camera.Find(camera_id);
3   Sensor sen = Sensor.Find(sensor_id);
4   string ip = cam.IP;
5   string user = cam.UserName;
6   string pass = cam.Pass;
7   string pan = null;
8   string tilt = null;
9   string zoom = null;
10  string str = null;
11  Coordenadas[] cord = Coordenadas.FindAll();
12
13  foreach (Coordenadas cd in cord){
14    if (cd.Sensor.Id.Equals(sensor_id)){
15      if(cd.Camera.Id.Equals(cam.Id)){
16        int posi = cd.Foscampos;
17        int set = posi - 1;
18        string url = "http://" + ip + "/decoder_control.cgi?user=" + user + "&
19          amp;pwd=" + pass + "&command=" + set;
20        WebClient http = new WebClient();
21        http.OpenRead(url);
22        str = "update";
23        return str;
24      }
25    }
26    pos = pos + 2;
27    PlugSenseApp.Models.Coordenadas coordenada = new PlugSenseApp.Models.
28      Coordenadas();
29    coordenada = new Coordenadas(sen, cam, zoom, tilt, pan, pos);
30    coordenada.CreateAndFlush();
31    str="new";
32    return str;
33  }
```

5.2.3 Rotação dos Sensores para a Posição do Alerta e Tratamento de Múltiplos Alertas.

Tendo o sistema de coordenadas e a integração/fusão sensorial implementados, pode-se implementar as funcionalidades de rotação e tratamentos de múltiplos alertas.

Base de Dados

Para a criação desta funcionalidade foi necessário adicionar tabelas a base de dados, a figura 5.7, mostra as tabelas adicionadas.

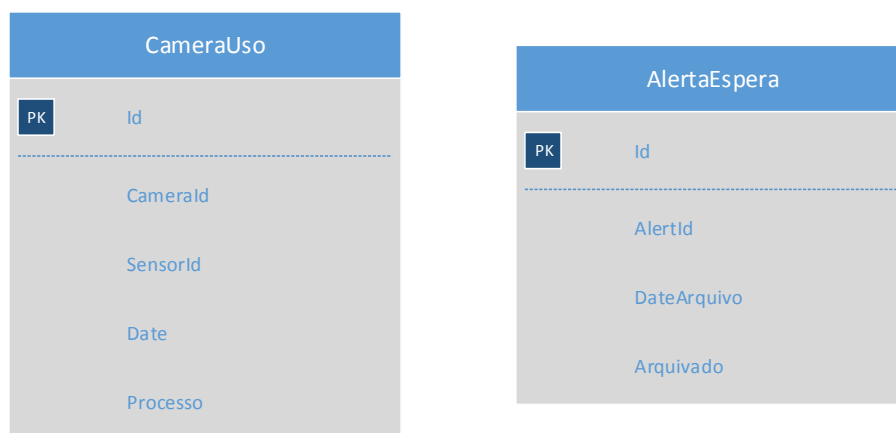


Figura 5.7: Tabelas CameraUso e AlertaEspera

A tabela CameraUso é necessária para guardar qual ou quais as cameras que estão a ser usadas e qual o sensor que as está a usar. A tabela AlertaEspera é usada para guardar os Alertas que não são tratados em caso de múltiplos alertas, o alerta é guardado nesta tabela quando a camera já esta a ser usada.

Código

É a função *rotatecam(Alert alerta, int i)* que trata destas funcionalidades, é também aqui que é invocada a função que inicia a gravação, como se pode verificar no código abaixo.

```

1 public static void rotatecam(Alert alerta, int i){
2
3     /*buscar o sensor que lan ou o alerta*/
4     int id_sensor = alerta.Sensor.Id;
5
6     /*buscar as coordenadas do sensor que lan ou alerta*/

```

Implementação

```
7   Coordenadas []coord = Coordenadas.FindAll();
8   Coordenadas coordenadas=null;
9   foreach (Coordenadas coordenad in coord) {
10      if(coordenad.Sensor.Id.Equals(alerta.Sensor.Id)){
11          coordenadas =coordenad;
12      }
13  }
14  //verifica se a camera j est a ser usada, retorna -1 caso n o esteja a ser
    usada
15  int camuso_id = cameralive(coordenadas.Camera.Id);
16  if (camuso_id !=-1){
17  // n o est livre, verifica se o sensor do Alerta o mesmo que est a ser
    usado
18      if (verificacamerauso(coordenadas.Camera.Id, coordenadas.Sensor.Id) == true
        ) {
19          // O Sensor do alerta j ta a ser tratado, atualiza o Alerta
20          alerta.Videofile = filename;
21          alerta.UpdateAndFlush();
22          }
23      else{
24  // O Sensor do Alerta n o o mesmo, verifica se o Alerta que est a correr
    j corre h mais de 30 seg.
25      int temp = tempodeuso(camuso_id);
26      if (temp < 30){
27          //Corre h menos de 30 seg, adicionar alerta tabela alertas em espera.
28          //verifica se esteja existe na tabelaAlertas em espera
29          bool flag = false;
30          AlertaEspera[] aux = AlertaEspera.FindAll();
31          foreach (AlertaEspera al in aux){
32              if(al.Alerta.Id==alerta.Id){
33                  flag = true;
34              }
35          }
36          if (flag == false){
37          // se n o existir, adiciona aos alertas em espera
38          AlertaEspera al_espera = new AlertaEspera();
39          al_espera.Alerta = alerta;
40          al_espera.Arquivado = false;
41          al_espera.Datearquivo = DateTime.Now;
42          al_espera.CreateAndFlush();
43          }
44          }
45      else{
46  //Corre h mais de 30 seg, roda a camera para o sensor do alerta
47          Camera camera = Camera.Find(coordenadas.Camera.Id);
48          string ipdns = camera.IP;
49          string user = camera.UserName;
50          string pass = camera.Pass;
51          string type = camera.Type.Name;
```

Implementação

```
52         // verifica qual o tipo de camara
53         if (type.Equals("axis")){
54             // camara axis busca as coordenadas do sensor e envia comando para
                rodar camara
55             string pan = coordenadas.Pan;
56             string zoom = coordenadas.Zoom;
57             string tilt = coordenadas.Tilt;
58             string url = "http://" + ipdns + "/axis-cgi/com/ptz.cgi?camera=1&
                pan=" + pan + "&tilt=" + tilt + "&zoom=" + zoom;
59             // chamar um webclient
60             WebClient webClient = new WebClient();
61             webClient.Credentials = new NetworkCredential(user, pass);
62             webClient.OpenRead(url);
63         // update camera em uso, com novo sensor
64             CameraUso cam_uso = CameraUso.Find(camuso_id);
65             cam_uso.Sensor = alerta.Sensor;
66             cam_uso.Date = DateTime.Now;
67             cam_uso.UpdateAndFlush();
68         //Atualiza Alerta
69             alerta.Videofile = filename;
70             alerta.UpdateAndFlush();
71             if (i == 1) {
72                 CameraUso[] cam = CameraUso.FindAll();
73                 cam[0].DeleteAndFlush();
74             }
75         }
76         else{
77             // camara foscama busca as coordenadas do sensor e envia comando para
                rodar camara
78             int slot = coordenadas.Fosc campos;
79             string url = "http://" + ipdns + "/decoder_control.cgi?user=" +
                user + "&pwd=" + pass + "&command=" + slot;
80             // chamar um webclient
81             WebClient webClient = new WebClient();
82             webClient.OpenRead(url);
83         // update camera em uso, com novo sensor
84             CameraUso cam_uso = CameraUso.Find(camuso_id);
85             cam_uso.Sensor = alerta.Sensor;
86             cam_uso.UpdateAndFlush();
87         // Atualiza Alerta
88             alerta.Videofile = filename;
89             alerta.UpdateAndFlush();
90             if (i == 1){
91                 CameraUso[] cam = CameraUso.FindAll();
92                 cam[0].DeleteAndFlush();
93             }
94         }
95
96         throw new NotImplementedException();
```

Implementação

```
97     }
98     }
99     } else{
100 // n o existem cameras em uso, vai tratar do alerta recebido
101 Camera camera = Camera.Find(coordenadas.Camera.Id);
102     string ipdns = camera.IP;
103     string user = camera.UserName;
104     string pass = camera.Pass;
105     string type = camera.Type.Name;
106     // verifica qual o tipo de camara
107     if (type.Equals("axis")){
108         //camara axis busca as coordenadas do sensor e envia comando para rodar
109             camara
110             string pan = coordenadas.Pan;
111             string zoom = coordenadas.Zoom;
112             string tilt = coordenadas.Tilt;
113             string url = "http://" + ipdns + "/axis-cgi/com/ptz.cgi?camera=1&pan=" +
114                 pan + "&tilt=" + tilt + "&zoom=" + zoom;
115             // chamar um webclient
116             WebClient webClient = new WebClient();
117             webClient.Credentials = new NetworkCredential(user, pass);
118             webClient.OpenRead(url);
119             if (i == 1){
120                 // Chama a fun o que inicia a grava o
121                 recordvideo(camera, alerta);
122                 AlertaEspera[] espera = AlertaEspera.FindAll();
123                 espera[0].DeleteAndFlush();
124             }
125             else{
126                 // Chama a fun o que inicia a grava o
127                 recordvideo(camera, alerta);
128             }
129         }
130     } else{
131         // camara foscam busca as coordenadas do sensor e envia comando para rodar
132             camara
133             int slot = coordenadas.Foscamos;
134             string url = "http://" + ipdns + "/decoder_control.cgi?user=" + user + "&
135                 &pwd=" + pass + "&command=" + slot;
136             // chamar um webclient
137             WebClient webClient = new WebClient();
138             webClient.OpenRead(url);
139             if (i == 1){
140                 // Chama a fun o que inicia a grava o
141                 recordvideo(camera, alerta);
142                 CameraUso[] cam = CameraUso.FindAll();
143                 cam[0].DeleteAndFlush();
144             }
145             else{
```

Implementação

```
142     // Chama a função que inicia a gravação
143     recordvideo(camera, alerta);
144     }
145 }
146 throw new NotImplementedException();
147 }
148 }
```

Foi ainda criada a função *checkAlertaespera()*, esta função é uma função que é executada a cada 15 segundos, verificando se existe alertas na tabela *AlertaEspera* que aguardam tratamento, caso haja, esta invoca a função *rotatecam*, com o alerta que se encontra na primeira posição. A função é descrita no código abaixo.

```
1 new Thread(new ThreadStart(this.checkAlertaespera)).Start();
2
3 public void checkAlertaespera() {
4     AlertaEspera[] alert = AlertaEspera.FindAll();
5     if (alert.Length > 0) {
6         CameraController.rotatecam(alert[0].Alerta, 1);
7     }
8     Thread.Sleep(15000);
9 }
```

5.2.4 Gravação

Para realizar esta funcionalidade, foi necessário recorrer a uma aplicação externa, neste caso VLC, por dois motivos:

1. Funcional para todos os tipos de sensores de videovigilância;
2. Sensor Foscam não possui a funcionalidade de gravação;

Pensando já no futuro, e na integração de novos sensores de videovigilância, optou-se por esta abordagem, visto que é uma abordagem funcional para todos os tipos de sensores. O outro motivo é o facto do sensor foscam, não possuir a funcionalidade de gravação de vídeo, logo houve a necessidade de recorrer a outro tipo de aplicações para resolver este problema.

Código

Para a realização deste funcionalidade foi necessário a criação da função *recordvideo(Camera camera, Alert alerta)*, que está descrita no código abaixo.

```
1 public static void recordvideo(Camera camera, Alert alerta) {
```

Implementação

```
2   Sensor sensor = alerta.Sensor;
3   string ip = camera.IP;
4   string user = camera.UserName;
5   string pass = camera.Pass;
6   string type = camera.Type.Name;
7   string targetDir;
8   targetDir = string.Format(@"C:\Users\Pedro\Desktop\HotelSense_lgp_funcional_v1.0\
   C digo\Main Server\PlugSenseServer\videos");
9   //verifica qual o tipo de camera;
10  if(type.Equals("axis")){
11      //Camera axis;
12      int batchid = 0;
13      string url = "axis-cgi/mjpg/video.cgi";
14      filename = "axis_" + camera.Id + "_" + DateTime.Now.ToString() ;
15      //trata filename
16      filename = filename.Replace("/", "_");
17      filename = filename.Replace(":", "_");
18      filename = filename.Replace(" ", "_");
19      //cria um novo processo e adiciona parametros necessarios (parametros
   diferentes para cada tipo de camera)
20  Process batch= new Process();
21      batch.StartInfo.WorkingDirectory = targetDir;
22      batch.StartInfo.FileName = @"C:\Program Files (x86)\VideoLAN\VLC\vlc.exe";
23  batch.StartInfo.Arguments = "http://" + user + ":" + pass + "@" + ip + "/" +
   url + " --sout=#transcode{vcodec=h264,acodec=mpga,channels=1}:standard{dst=
   " + filename + ".mp4}";
24      batch.StartInfo.WindowStyle = ProcessWindowStyle.Hidden;
25      batch.Start();
26  batchid = batch.Id;
27
28      //Adiciona camera   tabela camerauso;
29  CameraUso cam_use = new CameraUso();
30      cam_use.Sensor = sensor;
31      cam_use.Camera = camera;
32      cam_use.Processo = batchid;
33      cam_use.Date = DateTime.Now;
34      cam_use.CreateAndFlush();
35  //Atualiza alerta;
36      alerta.Videofile = filename;
37      alerta.UpdateAndFlush();
38
39  }else if(type.Equals("foscam")){
40      //camera foscam;
41      string url = "videostream.cgi";
42      string filename = "foscam_"+camera.Id + "_" + DateTime.Now.TimeOfDay();
43      //trata filename;
44      filename = filename.Replace("/", "_");
45      filename = filename.Replace(":", "_");
46      filename = filename.Replace(" ", "_");
```

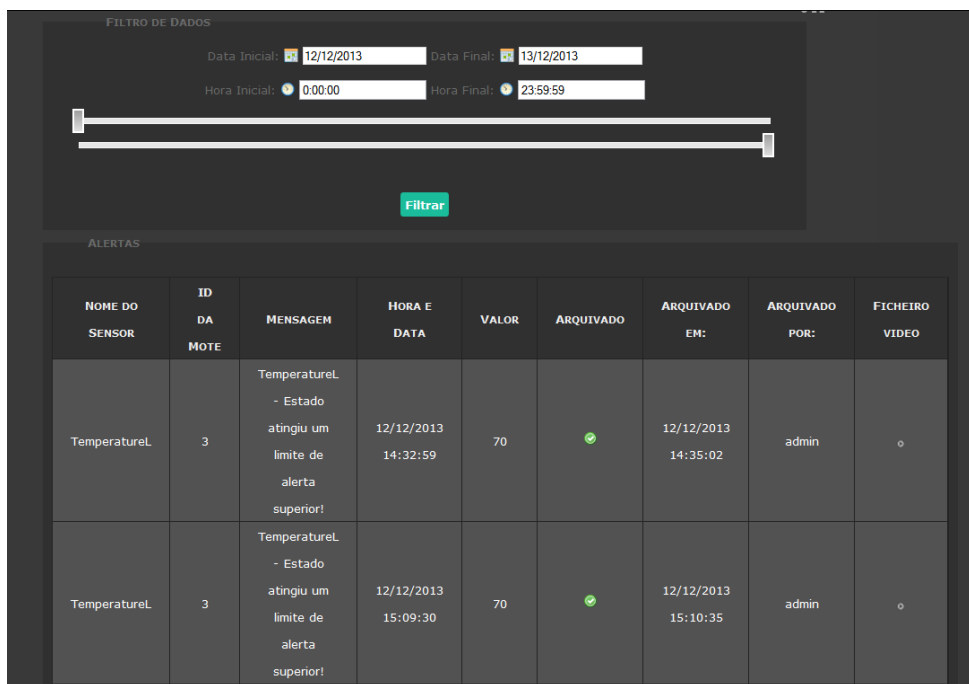
Implementação

```
47 //cria um novo processo e adiciona parametros necess rios (parametros
    diferentes para cada tipo de camera);
48 Process batch = new Process();
49 batch.StartInfo.WorkingDirectory = targetDir;
50 batch.StartInfo.FileName = @"C:\Program Files (x86)\VideoLAN\VLC\vlc.exe";
51 batch.StartInfo.Arguments = "http://" + user + ":" + pass + "@" + ip + "/"
    + url + " --sout=#transcode{vcodec=h264,acodec=mpga,channels=1}:
    standard{dst=" + filename + ".mp4}";
52 batch.StartInfo.WindowStyle = ProcessWindowStyle.Hidden;
53 batch.Start();
54 int batchidf = batch.Id;
55 //Adiciona camera tabela camerauso;
56 CameraUso cam_use = new CameraUso();
57 cam_use.Sensor = sensor;
58 cam_use.Camera = camera;
59 cam_use.Processo = batchidf;
60 cam_use.Date = DateTime.Now;
61 cam_use.CreateAndFlush();
62 //Actualiza alerta;
63 alerta.Videofile = filename;
64 alerta.UpdateAndFlush();
65 }
66 }
```

5.3 Outras funcionalidades

Foi ainda criada a funcionalidade, que permite ao administrado visualizar as gravações na aplicação de uma forma muito simples, basta aceder à *view* que contem o histórico de alertas e seleccionar o alerta que pretende visualizar. A figura 5.8 mostra a interface do histórico de alertas.

Implementação



FILTRO DE DADOS

Data Inicial: 12/12/2013 Data Final: 13/12/2013

Hora Inicial: 0:00:00 Hora Final: 23:59:59

Filtrar

ALERTAS

NOME DO SENSOR	ID DA MOTE	MENSAGEM	HORA E DATA	VALOR	ARQUIVADO	ARQUIVADO EM:	ARQUIVADO POR:	FICHEIRO VIDEO
TemperatureL	3	TemperatureL - Estado atingiu um limite de alerta superior!	12/12/2013 14:32:59	70	✓	12/12/2013 14:35:02	admin	o
TemperatureL	3	TemperatureL - Estado atingiu um limite de alerta superior!	12/12/2013 15:09:30	70	✓	12/12/2013 15:10:35	admin	o

Figura 5.8: View de Histórico de Alertas

Após clicar na coluna "Ficheiro Vídeo", a aplicação irá abrir uma nova *View*, onde será reproduzido o gravação, como se verifica na figura 5.9.

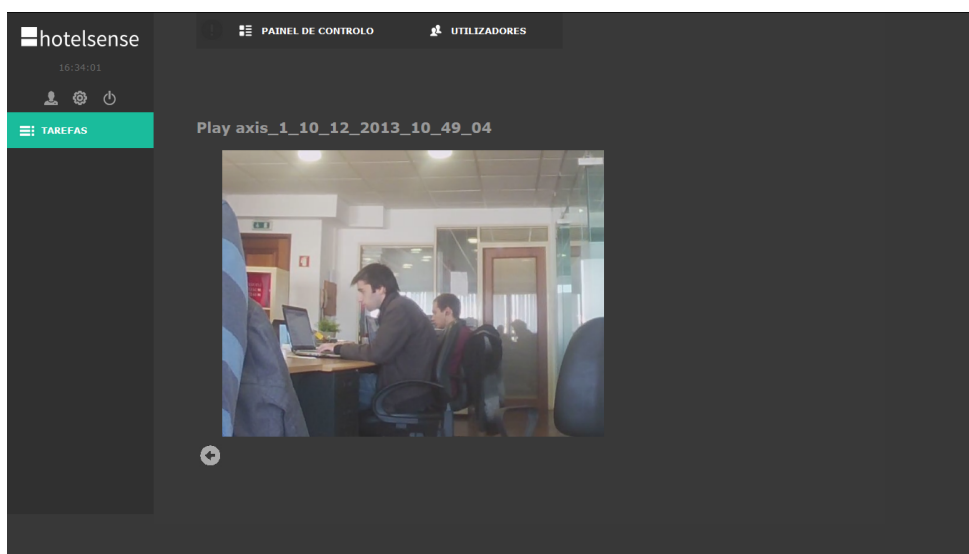


Figura 5.9: View de Visualização das Gravações

5.3.1 Resumo e Conclusão

Neste capítulo foi detalhada a implementação da solução, explicando como foram realizadas as funcionalidades que tinham sido definidas nos requisitos. Foi explicitado o trabalho realizado em cada módulo e descreveu-se o mais importante para cada parte, tal como métodos implementados e principais recursos usados.

Implementação

Capítulo 6

Testes Comparativo

O ideal seria testar o sistema num Hotel, visto que isso não é possível, optou-se por criar um cenário nas instalações da Freedomgrow, de forma a verificar o impacto da implementação da solução no sistema.

6.1 Cenário

Optou-se por criar um cenário com duas salas, cada uma com um sensor de videovigilância e vários sensores dispersos. A Sala1, é uma área de trabalho comum, com vários postos de trabalho, com sensores dispersos, como se pode verificar na figura 6.1. Esta área pode ser comparável a uma sala de jantar de um hotel.

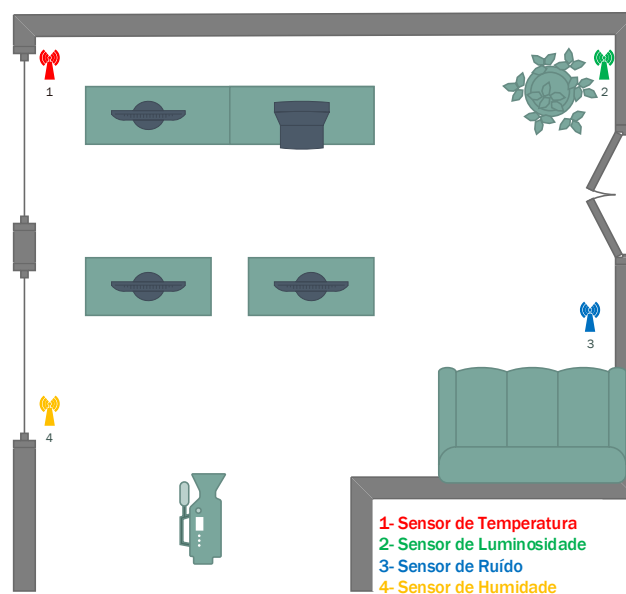


Figura 6.1: Cenário - sala1

Testes Comparativo

Em relação à Sala2, figura 6.2, esta é uma sala de reuniões, apenas com uma mesa central, com sensores dispersos. Esta sala pode ser comparada a uma área restrita num hotel.

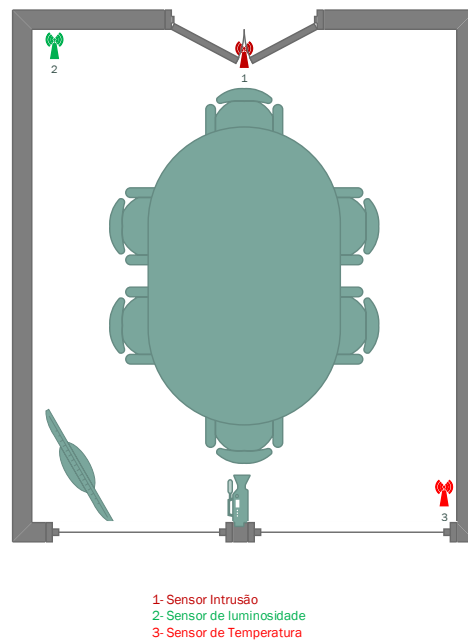


Figura 6.2: Cenário - sala2

Em relação aos sensores optou-se por usar os seguintes, pelo facto de serem sensores de possível utilização nas instalações de um hotel.

- Sensores de Temperatura.
- Sensores de Luminosidade.
- Sensores de Som.
- Sensores de Humidade.
- Sensores de Intrusão.

O teste tem como principal objetivo ver qual o impacto das funcionalidades implementadas na largura de banda e no espaço de armazenamento. O teste foi dividido em duas partes, um teste de pequena duração e um de maior duração. Os testes foram realizados para ambos os sensores de videovigilância (Axis e Foscam). Primeiramente testou-se a aplicação inicial, ou seja, sem nenhuma funcionalidade implementada, anotando-se os valores respetivos do consumo de

largura de banda e espaço de armazenamento. Posteriormente testou-se a aplicação desenvolvida no decorrer do projeto.

Após serem testadas as duas aplicações, nas mesmas condições (temporais e espaciais), realizou-se um pequeno teste comparativo.

6.2 Teste de Pequena Duração

Inicialmente optou-se por realizar um teste de pequenas dimensões/duração (30 mim), para cada um dos casos (com/sem implementação) e para cada um dos sensores de videovigilância. Este teste apenas foi realizado num dos cenário, a sala1, ver 6.1.

6.2.1 Solução Inicial

Nesta secção serão apresentados os resultados da solução inicial para cada um dos sensores de videovigilância.

Foscam

Como foi dito anteriormente, o teste teve a duração de aproximadamente 30 min. Neste caso, uma vez que não há otimizações o sistema esteve a transmitir/gravar imagem durante todo o tempo, independente do número de alertas, obtendo os seguintes resultados.

Em termos de largura de banda, como se pode ver na figura 6.3, houve um consumo de 321MB. Em relação ao espaço de armazenamento, o ficheiro criado pela aplicação, tem um tamanho de 74MB.

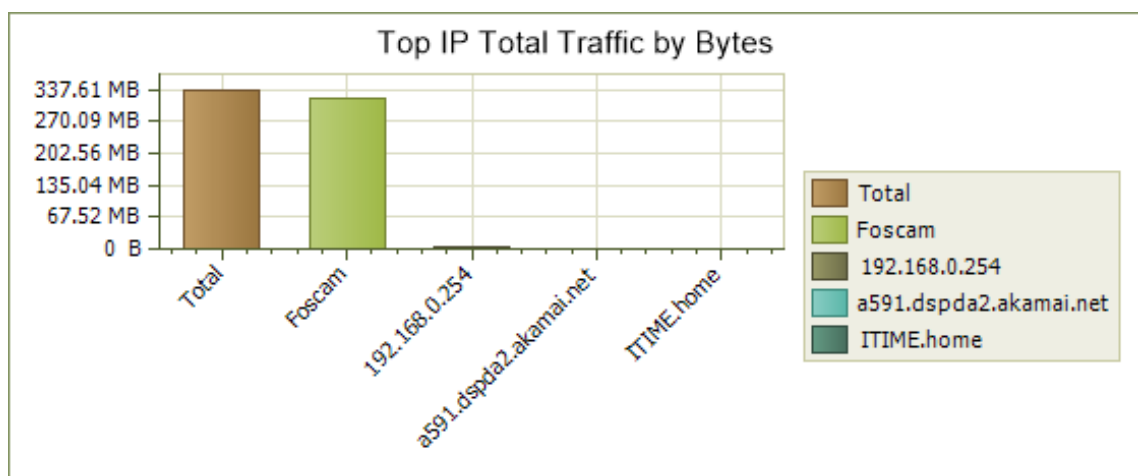


Figura 6.3: Grafico de Consumos- Camera Foscam

Axis

Em relação à camera Axis, quando executado o teste descrito anteriormente, obteve-se os seguintes resultados. Em termos de largura de banda, como se verifica na figura 6.4, houve um consumo de 1.06GB. Quanto ao espaço de armazenamento, o ficheiro criado pela aplicação tem um tamanho de 251MB.

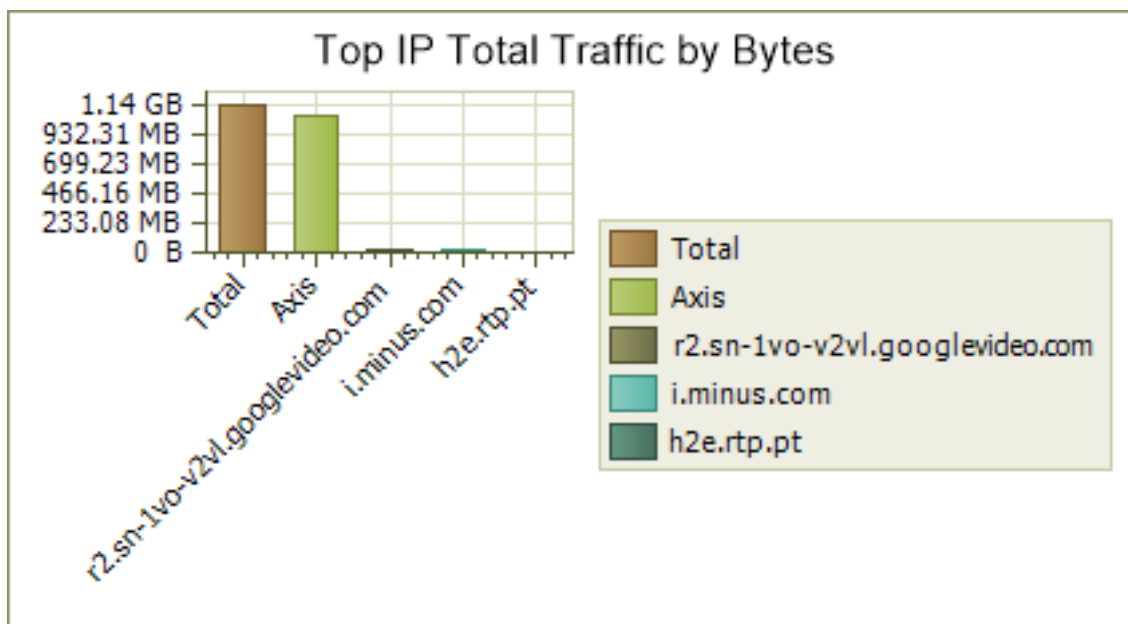


Figura 6.4: Grafico de Consumo- Camera Axis

6.2.2 Solução Implementada

Nesta secção será repetido o teste realizado acima, mas com a solução implementada, para cada uma dos sensores de videovigilância.

Foscam

Antes de iniciar o teste, foram colocados os sensores nas suas posições e definidas posições em relação ao sensor de videovigilância. Quando terminado, deu-se início ao teste. No decorrer do teste, foram detetados 5 alertas nos 4 sensores instalados. A tabela mostra de uma forma mais detalhadas as informações referentes aos alertas.

Testes Comparativo

Início	Sensor	Valor	Arquivado	Ficheiro Criado
10:06	Temperatura	25	10:09	3.60MB
10:10	Som	80	10:12	3.10MB
10:16	Luminosidade	18	10:18	4.32MB
10:24	Som	78	10:28	7.74MB
10:32	Temperatura	26	10:33	1.60MB

Tabela 6.1: Alertas detetados no teste- Foscam

Com esta abordagem, o sistema apenas transmitiu imagem por 5 vezes, ou seja, apenas quando ocorreram alertas, obtendo um consumo de largura de banda, como demonstra a figura 6.5, de 237.80MB. Quanto ao espaço de armazenamento, a soma dos ficheiros criados pela aplicação, é de 20.36MB.

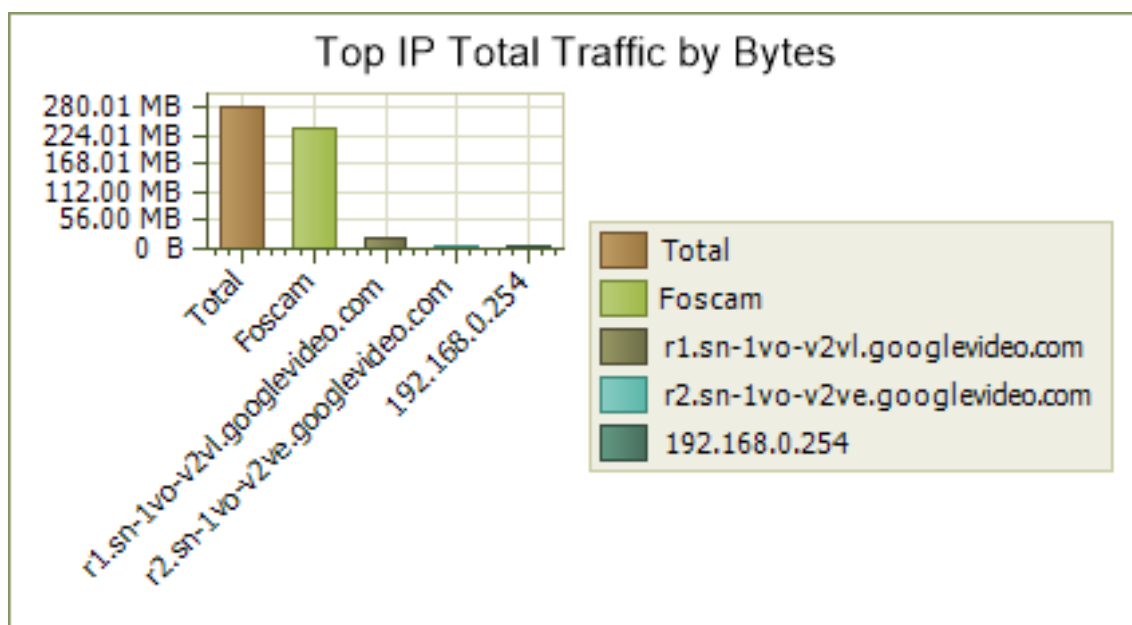


Figura 6.5: Consumo de Largura de Banda da Aplicação Desenvolvida - Camera Foscam

Axis

No decorrer deste teste com a duração de 30 min como referido anteriormente, foram detetados 6 Alertas, nos 4 sensores. A tabela mostra de forma mais detalhada os alertas.

Testes Comparativo

Início	Sensor	Valor	Arquivado	Ficheiro Criado
16:48	Som	97	16:50	6,13MB
16:58	Temperatura	28	17:00	5.65MB
17:02	Humidade	92	17:04	3.36MB
17:04	Humidade	91	17:05	1.76MB
17:08	Luminosidade	18	17:12	7.55MB
17:14	Som	80	17:16	9.47MB

Tabela 6.2: Alertas detetados no teste- Axis.

Com esta abordagem, o sistema apenas realizou gravações quando ocorreu um alerta, obtendo assim um consumo de largura de banda, como demonstra a figura 6.6, de 315.75MB. Quanto ao espaço de armazenamento, a soma dos ficheiros criados pela aplicação, é de 33.92MB.

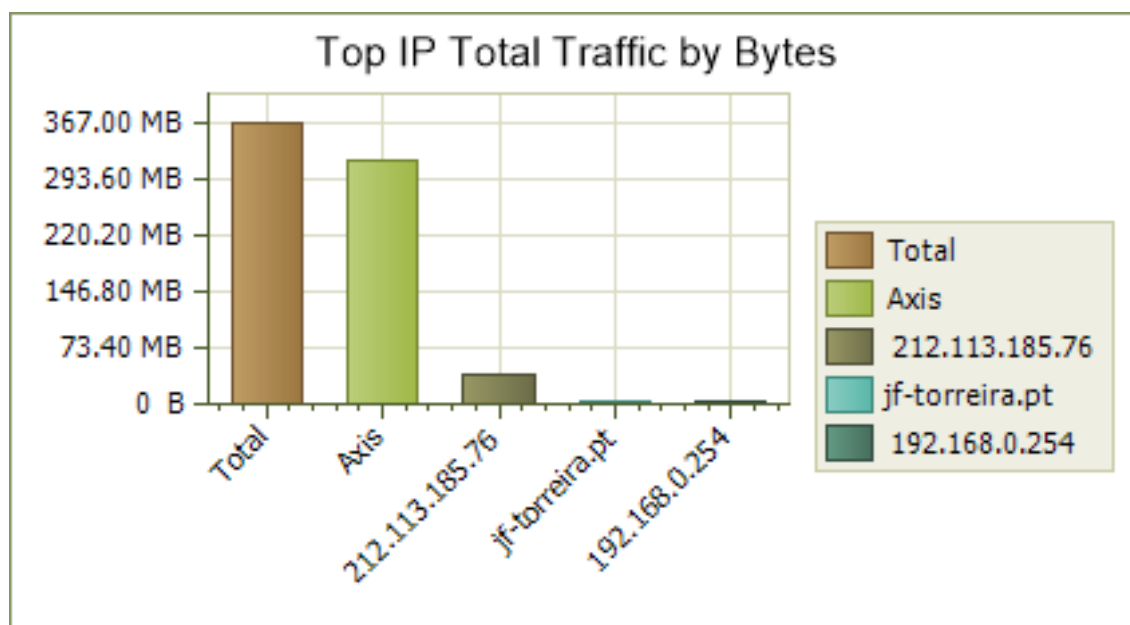


Figura 6.6: Alertas detetados no teste- Axis

6.2.3 Análise dos Resultados e Conclusões

Nesta secção é feita a análise dos resultados do teste para cada caso específico. Foi realizado uma análise simples, demonstrado na tabela abaixo, para cada um dos sensores, para ambas as soluções. Posteriormente foi realizada uma análise da tabela e retiradas as conclusões.

Testes Comparativo

Foscam	Largura de Banda	Espaço de Armazenamento
Aplicação Inicial	321MB	74MB
Aplicação Final	237.36MB	20.36MB
Diferença	83.64MB	53.64MB
Axis		
Aplicação Inicial	1.06GB	251MB
Aplicação Final	315.75MB	33.92MB
Diferença	744.25MB	217.08MB

Tabela 6.3: Análise dos Resultados- Teste de Pequenas Dimensões.

Com a realização deste teste de pequenas dimensões os resultados obtidos com as otimizações realizadas são bastante interessantes pois, como se pode observar na tabela, houve uma redução significativa dos valores a analisar. Sendo o espaço de armazenamento a variável com resultados mais significativos para ambos os sensores.

6.3 Teste de Maior Duração

Este teste teve a duração de 4h,e foi realizado nas duas salas acima mencionadas. Uma vez que só existem 2 sensores de videovigilância disponíveis, colocou-se um em cada sala, ficando assim a sala1 com o sensor Foscam e a sala2 com o sensor Axis.

6.3.1 Solução Inicial

Usando a solução inicial, o sistema efetuou gravação durante todo o tempo, ou seja, efetuou gravação durante 4h em cada um dos sensores de videovigilância.

No final do teste, em termos de largura de banda, como demonstra a figura 6.7 o sensor Axis consumiu 6.8GB e o sensor Foscam 4.2GB, ou seja, houve um consumo total de 11GB. Quanto ao espaço de armazenamento, a soma dos dois ficheiros criados, foi de 2.033GB. O sensor Axis criou um ficheiro com 1.7GB, quanto ao sensor Foscam, este criou um ficheiro de 333MB.

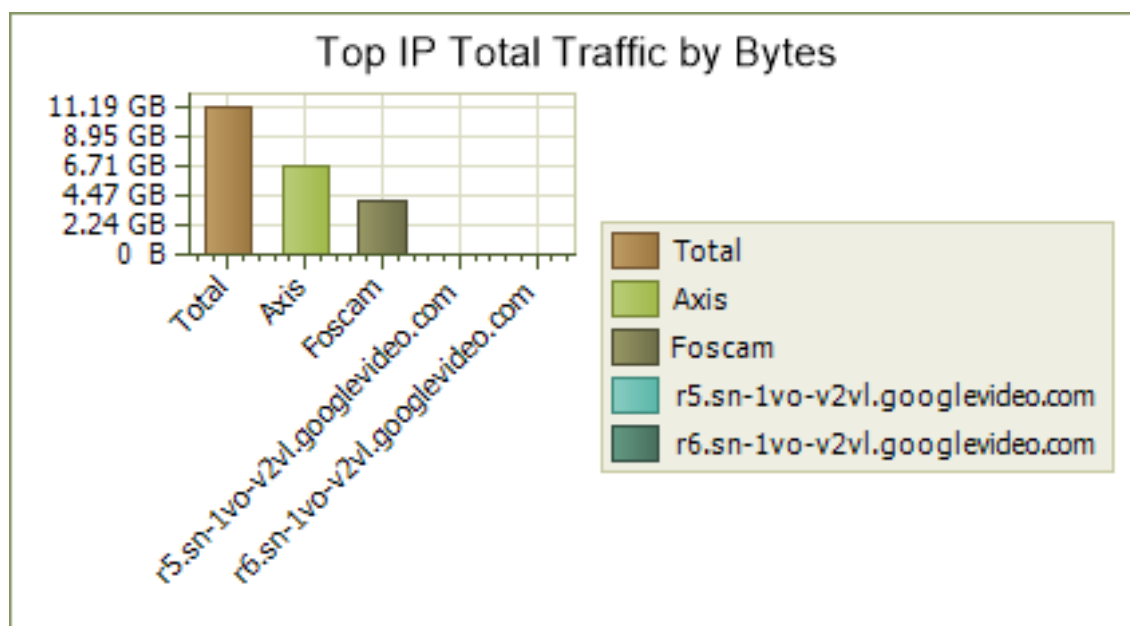


Figura 6.7: Consumo de Largura de Banda- Projecto inicial

6.3.2 Solução Implementada

Como se trata de um teste de maior duração, definiu-se que os sensores iriam enviar dados de 5 em 5 minutos de forma a ser o mais realista possível, pois num cenário real é essa a frequência de envio. Como referido anteriormente o sistema, apenas efetua gravação quando ocorrem alertas.

Os alertas que ocorreram no decorrer do teste, estão detalhados nas tabelas abaixo:

Sala1

Início	Sensor	Valor	Arquivado	Tipo	Ficheiro Criado
9:25	Som	80	9:30	Aviso	8.04MB
9:40	Luminosidade	18	9:45	Aviso	7.51MB
9:55	Som	87	10:05	Aviso	14.1MB
10:15	Temperatura	26	10:20	Perigo	8.77MB
10:40	Som	95	10:50	Aviso	13.01MB
11:05	Humidade	93	11:05	Aviso	9.48MB
11:30	Temperatura	25	11:40	Aviso	12.05MB
12:05	Luminosidade	18	12:10	Aviso	7.17MB
12:15	Temperatura	26	12:30	Perigo	20.3MB

Tabela 6.4: Alertas detetados na Sala1.

Sala2

Início	Sensor	Valor	Arquivado	Tipo	Ficheiro Criado
9:35	Luminosidade	80	9:40	Aviso	23.9MB
9:50	Temperatura	18	10:00	Perigo	19.6MB
10:15	Temperatura	26	10:20	Perigo	11.2MB
10:30	Intrusão	1	10:45	Perigo	24.00MB
11:00	Luminosidade	20	11:05	Aviso	19.01MB
11:30	Temperatura	25	11:40	Alerta	11.6MB
12:00	Luminosidade	18	12:05	Aviso	17.8MB
12:15	Intrusão	1	12:30	Perigo	26.5MB

Tabela 6.5: Alertas detetados na Sala2.

Tendo em consideração o número de alertas detetados no decorrer dos testes, a largura de banda consumida na sala1, que contém o sensor Foscam, foi de 1.009GB. Na sala2, que contém o sensor Axis, foi de 1.12GB. Obtendo um total de largura de banda consumida de 2.169GB.

Quanto ao espaço de armazenamento, a soma dos ficheiros criados pelo sensor Foscam é de 100.71MB. A soma dos ficheiros do sensor Axis é de 153.61MB. Sendo e espaço total usado em disco de 254.36MB.

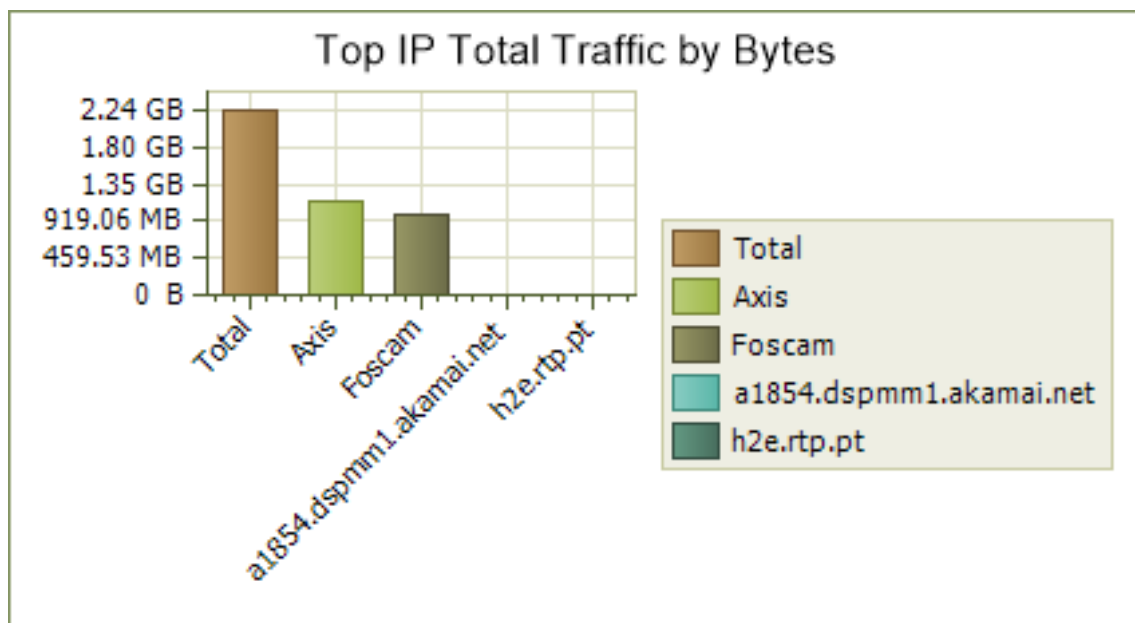


Figura 6.8: Consumo de Largura de Banda- Projecto final

6.3.3 Análise dos Resultados e Conclusões

Foscam	Largura de Banda	Espaço de Armazenamento
Aplicação Inicial	11GB	2.033GB
Aplicação Final	2.169GB	254.36MB
Diferença	8.831GB	1.779GB

Tabela 6.6: Análise dos Resultados- Teste de Maior Dimensão.

Como se pode verificar na tabela acima representada, conseguiram-se melhorias bastantes significativas em ambos os parâmetros de análise. Conseguiu-se em termos de largura de banda uma melhoria na casa dos 80%, em relação ao espaço de armazenamento essa melhoria ainda foi superior atingindo cerca de 87%. Estes valores, são valores bastante benéficos, e mostram que as otimizações realizadas foram um enorme sucesso.

6.4 Simulação de Monte Carlo

Para se ter uma ideia do impacto das melhorias a longo prazo decidiu-se fazer uma simulação de Monte Carlo usando alguns valores do teste realizado anteriormente.

Usando como referencia o teste anterior, e tendo em consideração que os valores seguem uma distribuição uniforme, formulou-se o seguinte modelo:

Variáveis Dependentes

NTA	NAP	NAA
MA+ALEATÓRIO(-10;10)	0,3*NTA	NTA-NAP

Tabela 6.7: Modelo de simulação Monte Carlo-Variáveis Dependentes

Variáveis Independentes

LBA	EAA	LBP	EAP	TMA
valor MB	valor MB	valor*TMA	valor*TMA	valor Min

Tabela 6.8: Modelo de simulação Monte Carlo-Variáveis Independentes

Valores Finais

LBT	EAT
$LBA*NAA+LBP*NAP$	$EAA*NAA+EAP*NAP$

Tabela 6.9: Modelo de simulação Monte Carlo-Valores finais

O valor da média de Alertas (MA) é obtido através de uma distribuição uniforme dos valores obtidos para cada caso (sala1 e sala2).

O Numero Total de alertas (NTA) é obtido a partir da função $MA+ALEATÓRIOENTRE(-10;10)$, ou seja definiu-se que pode ocorrer uma variação de mais ou menos 10 Alertas no final de um dia.

Verificou-se que 30% dos alertas são do tipo Perigo, ou seja, o valor de numero alertas perigo (NAP) é obtido através da seguinte função, $ARRED.EXCESSO.PRECISO(0,3*NTA)$. Quanto ao valor numero alertas Aviso (NAA), este é a diferença entre o número total de alertas e o número de alertas tipo perigo (NTA-NAP).

Os Valores de largura de banda do Alerta de ambos os tipos (LBA,LBP), bem como os valores do espaço de armazenamento de ambos os tipos (EAA,EAP), são obtidos pela análise dos testes. Estes valores variam consoante o sensor de videovigilância, pois os sensores tem diferentes características entre as quais, a qualidade da imagem.

Após definidas as variáveis, o valor total da largura de banda consumida (LBT) é obtida através da seguinte função $(LBA*NAA+LBP*NAP)$.O valor total do espaço de armazenamento (EAT) é obtido através da função $(EAA*NAA+EAP*NAP)$

Sala1

No teste de maior dimensão realizado anteriormente, verificou-se que em 4 horas ocorreram 9 alertas, seguindo uma distribuição uniforme, pode-se afirmar que em 24 horas ocorreram 54 alertas. Tendo em conta que o valor do número médio de alertas (MA) =54, construiu-se o cenário representado na 6.9

Testes Comparativo

SALA1					
Número Total de Alerta (NTA)	Alertas Perigo (NAP)	Alertas Aviso (NAA)	LBTotal	EATotal	
					LBA EAA TMA LBP EAP
					20 8 10 50 14
1	54	17	37	1590	534
2	62	19	43	1810	610
3	54	17	37	1590	534
4	55	17	38	1610	542
5	44	14	30	1300	436
6	52	16	36	1520	512
7	59	18	41	1720	580
8	50	15	35	1450	490
9	58	18	40	1700	572
10	45	14	31	1320	444
11	58	18	40	1700	572
12	52	16	36	1520	512
13	55	17	38	1610	542
14	60	18	42	1740	588
15	62	19	43	1810	610
16	56	17	39	1630	550
17	46	14	32	1340	452
18	61	19	42	1790	602
19	48	15	33	1410	474
20	59	18	41	1720	580
21	46	14	32	1340	452
22	64	20	44	1880	632
23	48	15	33	1410	474
24	62	19	43	1810	610
25	62	19	43	1810	610
26	53	16	37	1540	520
27	51	16	35	1500	504
28	61	19	42	1790	602
29	45	14	31	1320	444
30	55	17	38	1610	542
31	62	19	43	1810	610
TOTAL			49700	16736	

Figura 6.9: Monte Carlo- Sala1

Sala2

No teste de maior dimensão realizado anteriormente, verificou-se que em 4 horas ocorreram 8 alertas, seguindo uma distribuição uniforme, pode-se afirmar que em 24 horas ocorreram 48 alertas. Tendo em conta que o valor do número médio de alertas (MA) = 48, construiu-se o cenário representado na [6.10](#)

Testes Comparativo

SALA2										
	Número Total de Alerta (NTA)	Alertas Perigo (NAP)	Alertas Aviso (NAA)	LBTotal	EATotal	LBA	EAA	TMA	LBP	EAP
						71	19	10	146	22
1	57	18	39	5397	1135,2					
2	47	15	32	4462	936,5					
3	51	16	35	4821	1015,4					
4	42	13	29	3957	835,7					
5	50	15	35	4675	993,5					
6	42	13	29	3957	835,7					
7	58	18	40	5468	1154,2					
8	48	15	33	4533	955,5					
9	46	14	32	4316	914,6					
10	49	15	34	4604	974,5					
11	56	17	39	5251	1113,3					
12	52	16	36	4892	1034,4					
13	49	15	34	4604	974,5					
14	54	17	37	5109	1075,3					
15	52	16	36	4892	1034,4					
16	56	17	39	5251	1113,3					
17	50	15	35	4675	993,5					
18	56	17	39	5251	1113,3					
19	41	13	28	3886	816,7					
20	48	15	33	4533	955,5					
21	53	16	37	4963	1053,4					
22	42	13	29	3957	835,7					
23	52	16	36	4892	1034,4					
24	44	14	30	4174	876,6					
25	50	15	35	4675	993,5					
26	43	13	30	4028	854,7					
27	46	14	32	4316	914,6					
28	52	16	36	4892	1034,4					
29	50	15	35	4675	993,5					
30	43	13	30	4028	854,7					
31	40	12	28	3740	794,8					
			TOTAL	142874	30215,3					

Figura 6.10: Monte Carlo- Sala2

Total

Uma vez que se trata de uma simulação para um mês, o valor total da largura de banda e espaço de armazenamento, é igual ao somatório dos totais das duas salas. A figura 6.11, mostra os totais para ambas as variáveis.

Total	
LB Mensal	192574
EA Mensal	46951,3

Figura 6.11: Monte Carlo- Total

Testes Comparativo

Analisando os resultados demonstrados na [6.11](#) verifica-se que na simulação realizada, a solução implementada tem consumos de largura de banda e espaço de armazenamento respetivamente de 192GB e 47GB. Quanto aos resultados da solução inicial:

$$\mathbf{LBT=11*6*31=2046\ GB}$$

$$\mathbf{EAT=2.033*6*31=378\ GB}$$

Quando comparados os resultados da solução inicial, com os da solução final, os resultados são bastante benéficos, verificando-se aproximadamente as mesmas percentagens de otimização do teste de maior duração realizado anteriormente. As otimizações de largura de banda e espaço de armazenamento, são respetivamente 90% e 88%.

Capítulo 7

Conclusões e Trabalho Futuro

As redes de sensores sem fio são dinâmicas, devido as suas características e aplicações, apresentam diversas limitações que são tratadas diferentemente tendo em conta a sua aplicação. Neste caso concreto a principal limitação é a largura de banda necessária e o espaço de armazenamento necessários para o bom funcionamento do sistema de videovigilância.

Neste relatório foi apresentado o trabalho efetuado ao longo da Dissertação. À revisão bibliográfica foi acrescentada a descrição detalhada do problema bem como uma proposta de solução e foram ainda discriminados o processo de conceção e de implementação da respetiva solução. Após esta última fase foram também efetuados testes à aplicação de forma a verificar o impacto das implementações realizadas.

Para finalizar, tendo em conta o problema e as necessidades explicitadas, julga-se que a solução proposta irá ajudar a reduzir custo, pois com a solução os constrangimentos causados na ligação à Internet dos hotéis serão minimizados ou mesmo anulados. Desta forma consegue-se uma poupança no consumo de largura de banda, o que é muito valioso para o sistema funcionar na ligação à Internet do próprio hotel, sem obrigar a instalar uma rede dedicada para esta solução.

7.1 Cumprimento do Objetivo

Os objetivos inicialmente definidos, a realização e integração de funcionalidades avançadas na aplicação Hotelsense, foram cumpridos. Os módulos foram testados e constatou-se que estão a funcionar corretamente e prontos a serem utilizados. Os testes foram realizados com sucesso e ajudaram a perceber o verdadeiro impacto das integrações no sistema.

Com a implementação de novos sensores à PlungSense Framework, neste caso, os sensores de videovigilância, a Freedomgrow além de acrescenta valor às suas aplicações, aumenta o leque de potenciais clientes, como é o caso do mercado hoteleiro.

Em relação ao projeto final, com a realização dos testes verificou-se que os objetivos propostos foram alcançados com grande sucesso. Exemplo disso são as reduções de cerca de 80% da largura de banda consumida e redução de cerca 87% do espaço de armazenamento. Além da redução

das variáveis referidas anteriormente, as imagens guardada, ou seja, as gravações, contêm informação mais relevante, graças ao sistema de posicionamento sensorial implementado na solução. Acabando assim com gravações sem informação relevante, como é o caso de áreas sem atividade, paredes brancas entre outras.

7.2 Trabalho Futuro

Uma vez que está a ser desenvolvida uma nova versão da PlugSense Framework, com novas funcionalidades interessantes para o projeto HotelSense, o trabalho futuro passa por integrar a atual aplicação com a nova versão da PlugSense Framework.

Uma vez que com a nova versão da PlugSense é possível adicionar plantas de andares de edifícios e associar sensores a essas plantas, um dos problemas do projeto pode ser resolvido. O problema é o facto de sempre que adiciona um sensor de videovigilância, diferente dos existentes, é necessário adicionar novos parâmetros para realizar as interações necessárias ao nível do posicionamento dos restantes sensores.

Para trabalho futuro seria também interessante construir uma classe que fizesse a interligação entre os sensores de videovigilância e o novo sistema de posicionamento *indoor* da PlugSense Framework, ou seja, a classe seria capaz de traduzir os valores das coordenadas do sistema de posicionamento para parâmetros específicos de cada equipamento, tornando o HotelSense capaz de suportar um maior leque de equipamentos de videovigilância.

Bibliografia

- [Ahm12] Abdelgawad Ahmed. *Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks*. Springer, springer edition, 2012.
- [AXI13a] AXIS. Axis m1011-w. http://www.axis.com/products/cam_m1011w/index.htm, Maio 2013.
- [AXI13b] AXIS. Axis m1033-w. http://www.axis.com/pt/products/cam_m1033w/index.htm, Maio 2013.
- [AXI13c] AXIS. Axis m1034-w. http://www.axis.com/pt/products/cam_m1034w/index.htm, Maio 2013.
- [AXI13d] AXIS. Axis m1054. http://www.axis.com/pt/products/cam_m1054/index.htm, Maio 2013.
- [Axi13e] Axis. Considerações sobre largura de banda e espaço de armazenamento. http://www.axis.com/pt/products/video/about_networkvideo/bandwidth.htm, Maio 2013.
- [Axi13f] Axis. Vapix api. http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php, Setembro 2013.
- [Axx13] AxxessIndustries. Hotel system. <http://axxind.com/hospitality/>, Maio 2013.
- [dN13] Jornal de Negocios. Portugal ranking do turismo. http://www.jornaldenegocios.pt/empresas/detalhe/turismo_internacional_em_portugal_gerou_85_mil_milhoes_de_euros_em_2012.html, Maio 2013.
- [dP14a] Turismo de Portugal. Plano estratégico nacional do turismo-revisão e objectivos 2013/2015. http://boasnoticias.sapo.pt/noticias_Turismo-Portugal-%C3%A9-o-melhor-destino-do-mundo-para-2013_15246.html?page=1, Janeiro 2014.

BIBLIOGRAFIA

- [dP14b] Turismo de Portugal. Relatório de sustentabilidade- atuar para o desenvolvimento sustentável 2011. <http://www.turismodeportugal.pt/Portugu%C3%AAs/ProTurismo/destinos/destinostur%C3%ADsticos/Documents/TPrelSust%272011tedV5%206.pdf>, Janeiro 2014.
- [Eri14] Ericb. Simulação de monte carlo. <http://risk.nuvvo.com/lesson/5864-simulacao-de-monte-carlo>, Janeiro 2014.
- [Fer11] Vanessa Ferreira. Economia portuguesa e europeia. <http://ecportuguesaeeuropeia.blogspot.pt/2011/06/turismo-em-portugal-uma-atividade-com.html>, Julho 2011.
- [Fos13] Foscam. Foscam api. http://www.foscam.es/descarga/ipcam_cgi_sdk.pdf, Setembro 2013.
- [Gv14] Guia-viagens. Portugal eleito como um dos melhores destinos em 2014. <http://guia-viagens.aeiou.pt/portugal-eleito-como-um-dos-melhores-destinos-em-2014/>, Janeiro 2014.
- [HDL13] HDL. Hotel solution. http://www.hdlchina.com/Solutions_details.asp?sid=20&cpid=29, Maio 2013.
- [KM11a] Nitin Gupta Kiran Maraiya, Kamal Kant. Sarchitectural based data aggregation techniques in wireless sensor network: A comparative study. Technical report, Department of Computer Science and Engineering ASET, Amity University, Noida (U.P) ,India, 2011.
- [KM11b] Nitin Gupta Kiran Maraiya, Kamal Kant. Study of data fusion in wireless sensor network. Technical report, Department of Computer Science and Engineering ASET, Amity University, Noida (U.P) ,India, 2011.
- [Mar13] Marco Alves. Proposta de dissertação. https://sigarra.up.pt/feup/pt/estagios_empresas.ver_dados_proposta?p_id=153043&pv_perfil=ALU&p_aluno_id=112389, 2013.
- [P313] P3. Turismo: Portugal é o melhor destino do mundo para 2013. <http://p3.publico.pt/vicios/em-transito/5658/portugal-classificado-como-o-melhor-destino-do-mundo-em-2013>, Abril 2013.
- [PACSN] Joel J. P. C. Rodrigues Paulo A. C. S. Neves. Motivação para utilização de ip em redes de sensores sem fios. Technical report.
- [PDN04] Swapnil Patil, Samir R. Das e Asis Nasipuri. Serial data fusion using space-filling curves in wireless sensor networks. *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 82–190, Junho 2004.

BIBLIOGRAFIA

- [Pub10] Publico. Porto-a cidade já está a ganhar com o turismo low-cost. e não tem medo disso. <http://www.turismodeportugal.pt/Portugu%C3%AAs/Clipping/Documents/Mar%C3%A7o%202010/A%20cidade%20j%C3%A1%20est%C3%A1%20a%20ganhar%20como%20turismo%20low%20cost%20e%20n%C3%A3o%20tem%20medo%20disso.pdf>, Fevereiro 2010.
- [TC] Jorge Sá Silva Fernando Boavida Tiago Camilo, André Rodrigues. Redes de sensores sem fios, considerações sobre a sua instalação em ambiente real. Technical report.
- [UMI13] UMIC. Peritos portugueses em conferência sobre redes de sensores organizada pela ocde. http://www.unic.pt/index.php?option=com_content&task=view&id=3245&Itemid=236, Junho 2013.
- [Vol11] Pedro Jorge Tavares Volante. O segmento low-cost da indústria hoteleira em portugal: o caso dos hostels. Master's thesis, Instituto Universitário de Lisboa, 2011.
- [Web13] Web. O que é largura de banda. <http://www.fundacaobradesco.org.br/vv-apostilas/IE8/O%20que%20%C3%A9%20Largura%20de%20Banda.htm>, Maio 2013.