UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

PAC:MAN - SISTEMA DE GESTÃO DO RISCO DE ACIDENTES DE POLUIÇÃO EM ZONAS COSTEIRAS

Celso Filipe Lage de Sousa

PROJETO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

PAC:MAN - SISTEMA DE GESTÃO DO RISCO DE ACIDENTES DE POLUIÇÃO EM ZONAS COSTEIRAS

Celso Filipe Lage de Sousa

PROJETO

Projeto orientado pelo Professor Doutor João Carlos Balsa da Silva
e co-orientado pela Doutora Anabela Pacheco de Oliveira

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

Projeto financiado por:
FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

No âmbito do projeto PAC:MAN: PTDC/AAC-AMB/113469/2009

Trabalho desenvolvido no:

LNEC LABORATÓRIO NACIONAL DE ENGENHARIA CIVIL

Agradecimentos

Gostaria de deixar um obrigado do tamanho do universo à Dra. Anabela Oliveira, chefe do Núcleo de Tecnologias da Informação em Hidráulica e Ambiente onde decorreu este trabalho e minha orientadora pelo lado do LNEC. Sem a sua alegria diária, a sua dinâmica, a sua energia, todo o seu conhecimento e predisposição para ajudar, nada disto teria sido possível.

Os meus maiores agradecimentos aos colegas e investigadores Gonçalo Jesus, João Palha Fernandes, Dr. Alexandre Ribeiro, Dr. Alberto Azevedo, Ricardo Tavares e todos os investigadores e colegas envolvidos no projeto PAC:MAN. As suas preciosas indicações e experiência foram uma mais valia importante para o meu trabalho.

Ao Prof. Dr. João Balsa, meu orientador pelo lado da FCUL um bem-haja pelo apoio. Queria lembrar também os restantes profissionais que aí desempenham as suas funções por todos estes anos de sacrifício e de trabalho em prol de uma instituição que busca a maior qualidade de ensino e a formação de alunos de excelência e, acima de tudo, cidadãos com valores éticos e morais, capazes de ajudar a sociedade e o mundo empresarial com a sua disciplina e rigor. A todos sem exceção os meus melhores cumprimentos.

Ao restante pessoal do Departamento de Hidráulica e Ambiente e do LNEC o meu obrigado, em especial, à Ana Rilo, Joana Contente, Rita Cavalinhos e João Gomes por toda a ajuda disponibilizada e pelos bons momentos de conversa partilhados.

Um abraço enorme a todos os colegas de curso pela paixão comum pela Informática e pelo companheirismo ao longo destes anos, em especial, ao Sérgio Neves que é um exemplo como ser humano. Ele é a prova viva de que apesar de todas as contrariedades, nada na vida é impossível.

Por fim, gostaria de agradecer a toda a minha família. Eles foram desde sempre o meu suporte vital para chegar até aqui. Em especial, aos meus pais António e Amélia e aos meus irmãos Diana e Miguel. A minha força de vencer deve-se em boa parte ao vosso apoio constante. Estarão sempre no meu coração!

Resumo

O efeito da poluição por derrames acidentais nos ecossistemas costeiros motivou a procura e o desenvolvimento de abordagens para planeamento e resposta atempados à emergência com o intuito de proteger os recursos aquáticos. Os sistemas de monitorização da poluição e de modelação existentes são utilizados de forma independente durante acidentes deste âmbito sem a eficácia pretendida. A prevenção do risco de derrame é, habitualmente, feita via planos de contingência com base em estudos simplistas não refletindo o dinamismo da informação nem permitindo o alerta atempado dos gestores costeiros devido ao uso de tecnologia desatualizada.

Os sistemas de gestão de risco, testados com sucesso em desastres ambientais e humanitários, demonstram ser soluções promissoras. A sua adequação permite criar sistemas de gestão de risco mais específicos, como riscos de poluição e gestão da resposta à emergência em zonas costeiras. Esta inovação permite conjugar a modelação costeira de vanguarda para análise de risco, a riqueza de informação ambiental existente para a definição de indicadores de condições propícias à ocorrência de derrames e as tecnologias de comunicação. Obtém-se como resultado um conjunto de meios de alerta precoce e resposta mais eficiente e benéfica do ponto de vista da segurança das populações, da capacidade de atuação dos gestores costeiros e da manutenção dos recursos naturais costeiros.

A adaptação dos módulos do sistema de gestão de risco de acidentes por rotura de barragens SAGE-B permitiu conceber um novo sistema de gestão de risco de poluição em zonas costeiras que incluiu um sistema de alerta precoce resultante da aplicação dos modelos, um sistema de aviso associado e uma base de dados com os recursos em risco e os meios de resposta à emergência para a análise da vulnerabilidade na Ria de Aveiro, obtendo-se uma nova metodologia genérica de planeamento e resposta para riscos de poluição costeira.

Palavras-chave: Sistema de gestão de risco, Sistema de alerta e aviso, Sistemas de informação, *Data mining*, Sistemas tempo-real.

Abstract

The effect of pollution by accidental spills on coastal ecosystems motivated the search and the development of solutions to emergency planning and timely response in order to protect aquatic resources. Nowadays, current pollution monitoring systems and modeling systems are used independently for events of this scope without the desired effectiveness. The risk prevention of oil spill is usually done through contingency plans based on simplistic studies, which do not account for the information dynamics or allow an early warning of coastal managers due to the use of outdated technology.

Risk management systems, successfully tested on environmental and humanitarian disasters, prove to be promising solutions. Their adequation allows the creation of more specific risk management systems like pollution risks and management of emergency response in coastal areas. This innovation allows combining the cutting-edge coastal modeling for risk analysis, the richness of existent environmental information to define indicators of conditions prone to the occurrence of oil spills and communication technologies. This results in a set of tools for a more efficient early-warning and response which is also beneficial from the standpoint of security of the populations, from the ability to act from stakeholders and from the maintenance of coastal natural resources.

The adaptation of the modules of the dam break accidents risk management system SAGE-B incorporates in a new pollution risk management system for coastal areas, including the early-warning system resulting from the application of the models, the associated alert system and a database of resources at risk and means of emergency response for the analysis of vulnerability in the Aveiro lagoon, proposing a new general methodology for planning and response to risks of coastal pollution.

Keywords: Risk management system, Alert and warning system, Information systems, Data mining, Real-time systems.

Conteúdo

Acrónimos

API	Application Programming Interface
ARFF	Attribute-Relation File Format
ASCII	American Standard Code for Information Interchange
CIIMAR	Centro Interdisciplinar de Investigação Marinha e Ambiental
CMOP	Center for Coastal Margin Observation & Prediction
DVD	Digital Video Disk
EMI	External Message Interface
EPSG	European Petroleum Survey Group
FCT	Fundação para a Ciência e a Tecnologia
FCUL	Faculdade de Ciências da Universidade de Lisboa
FOSS	Free and Open Source Disaster Management System
GNOME	General NOAA Operational Modeling Environment
GSM	Global System for Mobile Communication
GUI	Graphical User Interface
HUODINI	Humboldt Disaster Management Interface
IGIS	Internet Geographic Information System
INSEA	Data Integration System for Eutrophication Assessment in Coastal Waters
IP	Internet Protocol
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
KDP	Knowledge Discovery Process

LNEC	Laboratório Nacional de Engenharia Civil
MOHID	Hydrodynamic Model
NTI	Núcleo de Tecnologias da Informação
OGC	Open Geospatial Consortium
OILMAP	Oil Spill Model and Response System
OSCAR	Oil Spill Contingency and Response
PAC:MAN	Sistema de Gestão do Risco de Acidentes de Poluição em Zonas Costeiras
PDF	Portable Document Format
PHP	Hypertext Preprocessor
RDFS	Rapid Deployment Forecast System
ROSETTA	A Rough Set Toolkit for Analysis of Data
RSL	Runtime Shared Libraries
SAGE-B	Sistema de Apoio à Gestão de Emergências em Barragens
SGBD	Sistema de Gestão de Bases de Dados
SIG	Sistema de Informação Geográfica
SIMAP	Integrated Oil Spill Impact Model System
SMPP	Short Message Peer-to-Peer
SMS	Short Message Service
UA	Universidade de Aveiro
UCP	Universal Computer Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
VOILS	Vela OIL Selfe
WAP	Wireless Application Protocol
WFS	Web Feature Service
WMO	World Meteorological Organization
WMS	Web Map Service
XML	Extensible Markup Language

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Os impactos dos derrames acidentais das últimas décadas [1] têm impulsionado o desenvolvimento e a implementação de diversas abordagens para planeamento e resposta à emergência de poluição dos meios aquáticos, incluindo sistemas de monitorização da poluição [43] e sistemas de modelação da evolução da mancha de hidrocarbonetos [2, 24]. Estes últimos englobam modelos numéricos que permitem prever a trajetória dos produtos poluentes. No entanto, quando ocorre um derrame que poderá afetar recursos costeiros, cada uma destas ferramentas é utilizada de forma independente. A prevenção dos riscos em zonas costeiras é, atualmente, baseada em planos de contingência [3], os quais são frequentemente criados a partir de estudos estáticos e simplistas. Também a capacidade de dar o alerta a todos os gestores costeiros é, habitualmente, demasiado lenta para evitar perdas materiais e danos graves nos ecossistemas, por utilizar tecnologias ultrapassadas e limitadas [4].

Os sistemas de gestão de risco [5, 17], que foram aplicados com sucesso para tsunamis [5] e inundações devidas à rotura de barragens [10, 13], podem ser usados para providenciar um enquadramento para riscos de poluição que permita a proteção eficaz dos recursos costeiros. O núcleo de informação destes sistemas, utilizado para identificar as pessoas em risco e os recursos de emergência para o seu salvamento, pode ser adaptado para a identificação dos elementos ecológicos em risco e das perdas ambientais, permitindo ainda integrar de forma eficiente a riqueza de informação sobre os acidentes e as condições ambientais em que estes ocorreram, as quais são fundamentais para suportar sistemas de alerta precoce. Os desenvolvimentos recentes nos sistemas de modelação costeiros, utilizando recursos de elevada *performance* [2, 6], criaram as condições para o seu uso potencial na análise de risco de poluição. Simultaneamente, a popularidade das novas tecnologias de comunicação [7, 8] abriram caminho para a criação de alertas precoces para eventos poluidores.

O projeto PAC:MAN [53] visa investigar os dados de acidentes passados para desenvolver e validar um conjunto de indicadores ambientais de condições atmosféricas e oceanográficas propícias à ocorrência de derrames; a capacidade e eficiência de sistemas

de modelação de elevada precisão para previsão e prevenção do risco de derrame; a fiabilidade, vantagens e escalabilidade de um sistema de alerta baseado em novas tecnologias móveis; a capacidade dos sistemas de tecnologias de informação para integrar e disponibilizar informação ambiental relevante sobre os elementos ecológicos em risco; e o modo de integração destes vários aspetos inovadores num sistema de gestão de risco para alerta precoce e aviso da ocorrência de um derrame nas zonas costeiras.

O estágio realizado no contexto deste projeto centrou-se no desenvolvimento de um sistema informático inovador que integrasse todos os componentes de índole tecnológica para responder às necessidades acima referidas.

Na construção do sistema de gestão de risco para alerta precoce e aviso pretendeu-se seguir uma política de reutilização de sistemas informáticos provados, que foram ou estão a ser aplicados noutras situações, adaptando-os às necessidades específicas do projeto.

1.1 Motivação

Este trabalho foi motivado pelos seguintes fatores:

a) Crescente ocorrência de casos de poluição dos meios aquáticos e das zonas costeiras por derrames acidentais ao longo do tempo;

b) Necessidade de ferramentas integradoras capazes de lidar com o dinamismo da informação associada ao planeamento e resposta à ocorrência de acidentes desta natureza;

c) Os meios de alerta precoce e aviso utilizados atualmente são inadequados e estão desatualizados;

d) Existência de sistemas de gestão de risco postos à prova com sucesso em cenários de desastres naturais; a sua adaptação aos acidentes de poluição de meios aquáticos constitui uma solução inovadora e eficaz na defesa dos recursos costeiros;

e) Desenvolvimentos feitos nos sistemas de modelação costeira para análise do risco de poluição e nas tecnologias de comunicação para alerta atempado de eventos poluidores; estes dois tipos de tecnologias podem ser integrados no novo sistema proposto.

1.2 Contribuições

Este projeto tem como contribuições mais relevantes:

a) Desenvolvimento e validação de um conjunto de indicadores ambientais relativo a condições atmosféricas e oceanográficas propícias à ocorrência de derrames a partir da elevada quantidade existente de dados de acidentes;

b) Análise, implementação e testes de um sistema integrado de gestão de risco para alerta precoce e aviso da ocorrência de um derrame nas zonas costeiras;

c) Base de dados com os recursos em risco e os meios de resposta à emergência para a análise da vulnerabilidade da Ria de Aveiro.

1.3 Contexto Institucional

O presente trabalho decorreu no LNEC que funcionou como unidade coordenadora, enquadrado no projeto PAC:MAN [53] e financiado pela FCT. O projeto está a ser desenvolvido por uma equipa de pesquisa multidisciplinar de três instituições constituída por investigadores das divisões de Tecnologias de Informação e Zonas Estuarinas e Costeiras, do LNEC, dos departamentos de Física e Ambiente, da UA e do CIIMAR, assegurando todas as áreas científicas relevantes. Esta colaboração estende-se à partilha dos resultados científicos, e correspondentes publicações e disseminação em diversas apresentações e conferências.

1.4 Estrutura do Documento

Este documento está organizado da seguinte forma:

- No primeiro capítulo, denominado *Introdução*, é feita uma introdução ao projeto e um resumo do trabalho realizado. São apresentadas as motivações, as contribuições, o contexto institucional e a estrutura deste documento.
- No segundo capítulo, denominado *O Projeto*, são apresentados os objetivos, o contexto subjacente, o plano de trabalho e o modelo de desenvolvimento de *software* adotado.
- No terceiro capítulo, denominado *Estado da Arte*, é feito um enquadramento com trabalhos relacionados e o estado da arte dos sistemas e tecnologias mais importantes para a área científica deste projeto. São apresentados conceitos teóricos relevantes no contexto das atividades executadas durante este projeto.
- No quarto capítulo, denominado *Análise do Problema*, é explorado o problema inicial, que meios foram utilizados e possíveis alternativas.
- No quinto capítulo, denominado *Desenho da Solução*, é apresentado o modelo de dados do sistema e a arquitetura escolhida para a conceção do *software*.
- No sexto capítulo, denominado *Implementação da Solução*, é feita uma abordagem pormenorizada sobre a forma de implementação do sistema, que tarefas foram levadas a cabo e que ferramentas foram efetivamente utilizadas.

- No sétimo capítulo, denominado *Avaliação*, são apresentados os testes realizados ao sistema, os respetivos resultados alcançados e uma análise crítica.
- No oitavo capítulo, denominado *Conclusões*, é apresentado um resumo do trabalho desenvolvido, as conclusões, comentários críticos acerca do trabalho realizado e dos resultados obtidos e linhas orientadoras de trabalho futuro referindo as tarefas futuras e os pontos que poderão ser melhorados.

Capítulo 2

O Projeto

O trabalho associado ao projeto englobou, em primeiro lugar, a análise de dados *in-situ* e de deteção remota para acidentes de poluição ocorridos na plataforma Ibérica Atlântica e sua zona costeira, para apoiar o desenvolvimento dos indicadores de condições de risco. Estes indicadores foram depois validados para a Ria de Aveiro, a qual foi escolhida pela sua importância ambiental e económica. Com base nestes indicadores, foram definidos um conjunto de cenários de ocorrência de acidentes, os quais conjuntamente com os resultados de uma análise de vulnerabilidade local e um conjunto de modelos sofisticados [2], foram utilizados para avaliar análises de risco de poluição, e para desenvolver uma nova metodologia de prevenção deste tipo de riscos. Este sistema de modelação de derrames simula os processos relevantes às escalas adequadas e está acoplado com um sistema de modelação da circulação forçada conjuntamente por ondas, correntes e vento. A metodologia proposta é a base de um sistema inovador de alerta precoce, que combina de modo eficiente as condições ambientais propícias à ocorrência de acidentes com previsões detalhadas do percurso e da transformação das plumas de poluente. Este sistema de alerta alimenta um sistema de aviso, baseado em SMS e outras tecnologias, o qual foi analisado em termos de eficiência e escalabilidade para números crescentes de utilizadores e para definir os conteúdos ótimos do aviso.

Os dois sistemas estão ligados através de um sistema de gestão do risco, adaptado para derrames a partir do sistema proposto em [10]. Esta infraestrutura inclui uma base de dados dos elementos em risco e dos recursos de resposta à emergência, e foi adaptado para a análise de vulnerabilidade da Ria de Aveiro. O resultado final do projeto constitui uma nova metodologia genérica de planeamento e resposta para riscos de poluição costeira, baseada nas várias ferramentas e análises propostas.

2.1 Objetivos

Os objetivos deste estágio, correspondentes às tarefas principais a que o estagiário esteve associado na proposta de projeto, englobaram as seguintes atividades:

a) Análise de dados presentes em bases de dados de modelação relativos à Ria de Aveiro e à região costeira, orientada ao risco identificado nos padrões de circulação, através da utilização de técnicas de *data mining* e tendo em conta os dados disponíveis de indicadores de condições ambientais propícias a acidentes e de acidentes de poluição;

b) Testes ao protótipo do sistema de aviso de emergência baseado em serviços móveis de mensagens;

c) Desenvolvimento e validação de um sistema integrado de gestão de risco para alerta precoce e aviso de acidentes de poluição em zonas costeiras através da adaptação do sistema SAGE-B e utilizando a plataforma *Sahana* Vesuvius [52].

2.2 Planeamento

Numa primeira fase foi realizada a análise das actividades associadas a este projeto e um levantamento do estado da arte. Efetuou-se a pesquisa de artigos, livros e documentação diversa e a respectiva redação de um relatório preliminar.

Numa segunda fase o estagiário realizou a tarefa de análise de dados disponíveis em bases de dados de modelação da Ria de Aveiro e da região costeira desenvolvidas no âmbito de projetos de pesquisa anteriores, do LNEC e da UA, tendo em conta indicadores de condições ambientais propícias a acidentes e acidentes passados, dados esses provenientes do trabalho de colegas neste projeto, e utilizando técnicas de *data mining* [33], como *clustering* ou redes neuronais.

Numa terceira fase, participou nos testes ao protótipo aplicacional do sistema de aviso de emergência baseado em serviços móveis de mensagens.

Numa quarta fase, realizou o desenvolvimento e a validação do sistema de gestão de risco de emergência SAGE-*spill* adaptando o núcleo de informação acerca de acidentes por quebra de barragem do SAGE-B para acidentes de poluição na área costeira.

Após a redação do relatório preliminar era expectável alcançar a implementação de uma primeira versão dos principais módulos do sistema a desenvolver, por forma a obter o máximo de resultados satisfatórios possível até ao final da duração do estágio de mestrado, correspondente a cerca de nove meses. A duração total da atividade do aluno no projeto compreendia um período total de vinte e seis meses. Durante o tempo de duração do estágio a taxa de esforço foi de 100%.

A Figura A.8 apresenta o mapa de *Gantt* relativo ao planeamento das tarefas associadas a este estágio, as datas de início e fim e a respectiva duração de realização.

2.3 Metodologia

O modelo de desenvolvimento de *software* utilizado foi o modelo de desenvolvimento ágil (*Agile Modeling* [84] - Figura 2.1). Este método define iterações curtas que visam reduzir o ciclo de vida de desenvolvimento para acelerar a concepção de *software*. Uma versão mínima é desenvolvida, as funcionalidades vão sendo integradas iterativamente com base nos requisitos e nos testes ao longo de todo o ciclo de desenvolvimento. É dada maior ênfase à interação entre os parceiros, à parte prática de desenvolvimento de código e a uma maior abertura à mudança em detrimento dos processos e instrumentos adotados, da definição e cumprimento de um plano rígido, da criação pormenorizada de documentação e da negociação contratual.

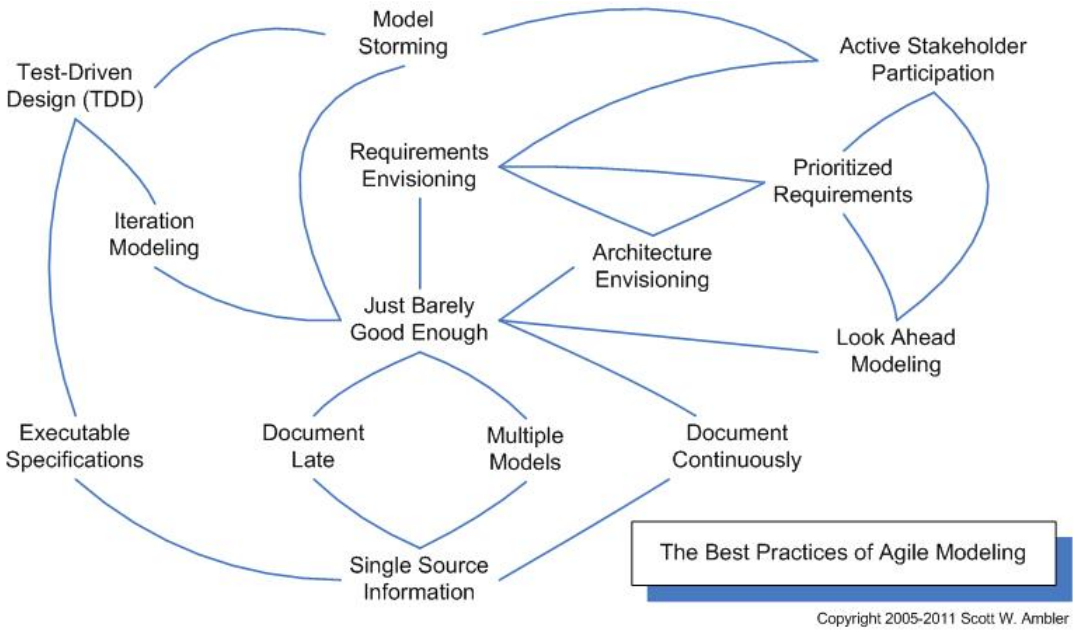

Figura 2.1: Práticas e Princípios de Agile Modeling

Este método surgiu devido à dificuldade em obter requisitos completos do cliente no início do projeto o que levou a ser criada uma capacidade de adaptação a esta instabilidade decorrente das mudanças de contexto e de especificações durante o processo de desenvolvimento. Esta flexibilidade permite ao cliente participar desde o início do projeto obtendo uma primeira versão do *software* mais rapidamente.

Capítulo 3

Estado da Arte

Segue-se a descrição do estado da arte em sistemas de gestão de risco e emergência aplicados a diversos tipos de calamidades, sistemas de previsão em tempo-real, sistemas de alerta e aviso e modelos de simulação de derrames, complementada com exemplos específicos em diversas áreas sociais e científicas para cada tipo de sistema e alguns conceitos teóricos relevantes. Apresenta-se ainda o estudo realizado sobre as principais ferramentas para *data mining* existentes.

3.1 Conceitos Relevantes

Nesta secção são abordados conceitos e definições relevantes para este estágio: *data mining*, o processo de descoberta de conhecimento e alguns dos modelos e algoritmos de *data mining* mais importantes.

3.1.1 Data Mining

A elevada quantidade de dados na posse das empresas foi reconhecida como um recurso valioso pelos decisores, que a utilizam para retirar conhecimento útil, estudar tendências e analisar os fatores que influenciam, pelo que se torna vantajoso realizar *data mining*.

Data mining representa a procura (orientada por objetivos) e extração de padrões [18] ou conhecimentos relevantes, não conhecidos inicialmente, de um conjunto de dados, através de métodos de análise e previsão. Esses padrões podem ser utilizados sobre informação relativa a cenários futuros, desconhecida à partida, para identificação precoce de situações de potencial risco, permitindo otimizar processos. Na prática, *data mining* tem dois objetivos principais: previsão e análise. No primeiro caso, procura prever valores futuros desconhecidos de outras variáveis de interesse envolvendo valores ou campos numa base de dados. No segundo caso, foca-se em procurar e encontrar padrões interpretáveis pelos humanos que descrevam os dados.

3.1.2 Processo de Descoberta de Conhecimento

O processo de descoberta de conhecimento (KDP [20]) começa com um conjunto de dados de treino, e uma metodologia para desenvolver a melhor representação possível da estrutura dos dados e sobre a qual se obtém o conhecimento [19]. Esse conhecimento adquirido pode ser utilizado durante a execução do processo a conjuntos maiores de dados assumindo que estes têm estrutura semelhante com a obtida nos dados amostra. A Figura 3.1 esquematiza este processo.

Figura 3.1: Processo de Descoberta de Conhecimento

Este processo engloba quatro etapas: definição de objetivos, preparação de dados, extração de conhecimento e interpretação de resultados. A partir dos dados em bruto, *definem-se objetivos* científicos consoante a descrição e a qualidade dos dados a analisar. A *preparação dos dados* está relacionada com a optimização: combinação de fontes de dados, transformação de atributos, limpeza de dados, seleção de dados, seleção de atributos e visualização de dados [21]. A *extração de conhecimento* é feita de acordo com modelos de *data mining* (*clustering*, classificação, regras de associação, regressão, etc) e os padrões são gerados segundo uma forma representativa (regras de classificação, tendências, modelos de regressão, árvores de decisão, etc). Os *modelos de previsão* gerados que tenham alta qualidade de uma perspetiva analítica dos dados são avaliados segundo o objectivo de aplicação.

3.2 Métodos de Data Mining

Descreve-se abaixo os métodos de *data mining* relevantes: classificação e *clustering*.

3.2.1 Classificação

A *classificação* identifica a aprendizagem de uma função que mapeia os novos dados com classes pré-definidas através de um algoritmo de classificação. O modelo de classificação é baseado em dados históricos e factos conhecidos, divididos nos conjuntos de treino e de

teste. Essa função é aprendida a partir dos dados de treino. O algoritmo de classificação é, portanto, treinado numa parte dos dados e a sua precisão testada em dados independentes. Existem diversos algoritmos de aprendizagem supervisionada (a quantidade e a especificidade das classes com que os dados já existentes são categorizados é conhecida) tais como árvores de decisão ou redes neuronais.

3.2.2 Clustering

O *clustering* pretende obter um conjunto finito de *clusters* ou categorias para descrever os dados. Corresponde à aprendizagem não supervisionada porque o número e o tipo de classes não são conhecidos à partida. Estas categorias são agrupamentos naturais em que os elementos de cada *cluster* partilham em maior grau uma característica em comum em relação aos restantes elementos nos outros *clusters*. Apesar de ser um método menos preciso que os métodos de aprendizagem supervisionada, é muito útil quando não há dados de treino disponíveis.

O K-Means é o algoritmo mais usado na prática para encontrar *clusters*. Dado um valor k inicial correspondente ao número de sementes que identificam, cada qual, o seu *cluster*, os dados são divididos por esses k conjuntos não vazios consoante a semente que lhes estiver mais próxima. Após isso, é calculada uma média ponderada para cada *cluster*, das n dimensões de entre todos os elementos nesse *cluster*. Isto permitirá obter o centróide de cada *cluster*. Cada elemento é novamente atribuído ao *cluster* que tenha o centróide mais próximo de si até que todos os elementos estejam atribuídos [23]. A distância euclidiana é normalmente a forma mais usada para calcular a distância entre um ponto no diagrama e o centróide.

3.3 Sistemas de Gestão de Risco e Emergência

A plataforma *Sahana* FOSS [51] foi implementada com o intuito de ajudar a gerir o impacto do *tsunami* ocorrido na Ásia em 2004 [5], tendo depois sido desenvolvida e aplicada em outras catástrofes mais recentes com vista a generalizar a aplicação para qualquer tipo de emergência.

O *Sahana* tem como objectivo primário ser um projeto integrador de um conjunto de aplicações *Web*, interrelacionadas, de gestão de catástrofes e fornecedoras de soluções para responder aos problemas humanitários neste âmbito, com a envolvência direta das partes interessadas.

Esta plataforma foi construída em PHP com base de dados MySQL, o que é vantajoso do ponto de vista de reutilização de código, podendo também utilizar uma base de dados PostgreSQL. A natureza modular da interface *Web* do *Sahana* permite a criação de novos módulos para um sistema de gestão de risco independente do SAGE-B (ver descrição abaixo). O seu *design* permite uma rápida configuração dos processos de suporte à gestão

de desastres e a criação de soluções flexíveis na administração das funcionalidades necessárias consoante os diferentes contextos de atuação. O núcleo da plataforma é composto por uma camada de API's e bibliotecas que, por sua vez, é composta pelas camadas de módulos núcleo e de módulos opcionais. As camadas exteriores têm acesso às funcionalidades das camadas interiores [5]. A informação espacial associada à mitigação, monitorização e gestão de desastres, é tratada e visualizada segundo um sistema SIG. É possível consultar informação espacial através do Google Maps ou através do OpenLayers.

Existem três versões do *Sahana*: Eden, Vesuvius e Mayon. A versão de produto Eden é uma plataforma flexível, rica, modular e altamente configurável para gestão das necessidades humanitárias mais críticas durante um desastre. Tem versões para trabalho conectado e desconectado. O *Sahana* Vesuvius foca-se principalmente na procura do paradeiro de pessoas perdidas e assistência à triagem hospitalar. Inclui captura fotográfica e intercâmbio de dados. Em oposição às outras versões, está orientado para múltiplos desastres por base de dados em vez de uma instância nova por cada desastre o que revela a sua apetência para a prevenção de desastres. É mais fácil de desenvolver e concentra-se em menor grau que o Eden na otimização da instalação de campo feita com *hardware* portátil em condições de conectividade intermitente. A versão Mayon é uma solução de gestão de emergência para municípios e organizações que engloba um número alargado de eventos e cenários de desastre e respetivos planos de atuação, gestão de meios, de recursos e de pessoal.

A versão da plataforma escolhida para utilização foi o *Sahana* Vesuvius [52]. Os módulos desta plataforma, inclusive aqueles que foram adaptados e melhorados, constituem a camada aplicacional do sistema de alerta precoce e aviso construído. A sua infraestrutura fornece suporte à informação espacial.

O sistema de informação SAGE-B [10, 26, 31] foi desenvolvido no LNEC utilizando a plataforma *Sahana*, sendo baseado na existência de planos de emergência para barragens e tirando partido do grande conhecimento acumulado sobre acidentes em barragens. É um sistema dinâmico de apoio à emergência que tem por objetivo facilitar o acesso, a partilha e a gestão da informação associada a acidentes por roturas em barragens. A informação relevante para a gestão destas emergências está em constante alteração pelo que se torna mais adequado ter um sistema que suporte e atualize dinamicamente todo esse conhecimento permitindo uma resposta pronta e mais eficaz à situação de emergência por parte das entidades competentes em detrimento dos planos estáticos. Estes são mais suscetíveis a falhas na obtenção e gestão da informação com impactos negativos na tomada de decisão e com consequências no sucesso da resposta ao desastre.

O SAGE-B gere a informação relevante em tempo-real através de módulos integrados que ajudam a definir o nível de alerta de uma barragem e as ações a tomar consoante o nível de cada caso. Integra também um sistema de simulação de emergência de inundação.

Está estruturado em onze módulos (Figura 3.2), oito deles internos. Os três módulos externos representam possíveis expansões futuras do sistema. Alguns dos módulos foram implementados como serviços *Web* sobre uma base de dados MySQL e com uma interface em PHP. O SAGE-B adaptou alguns dos módulos da plataforma *Sahana* visto que partilham requisitos semelhantes para além de incluir suporte *Web* GIS. Este suporte GIS ajuda as entidades de emergência que utilizam o SAGE-B a gerirem espacialmente os incidentes em tempo-real. A sua natureza modular permite a integração de novos módulos e a adição de desenvolvimentos e melhorias constantes.

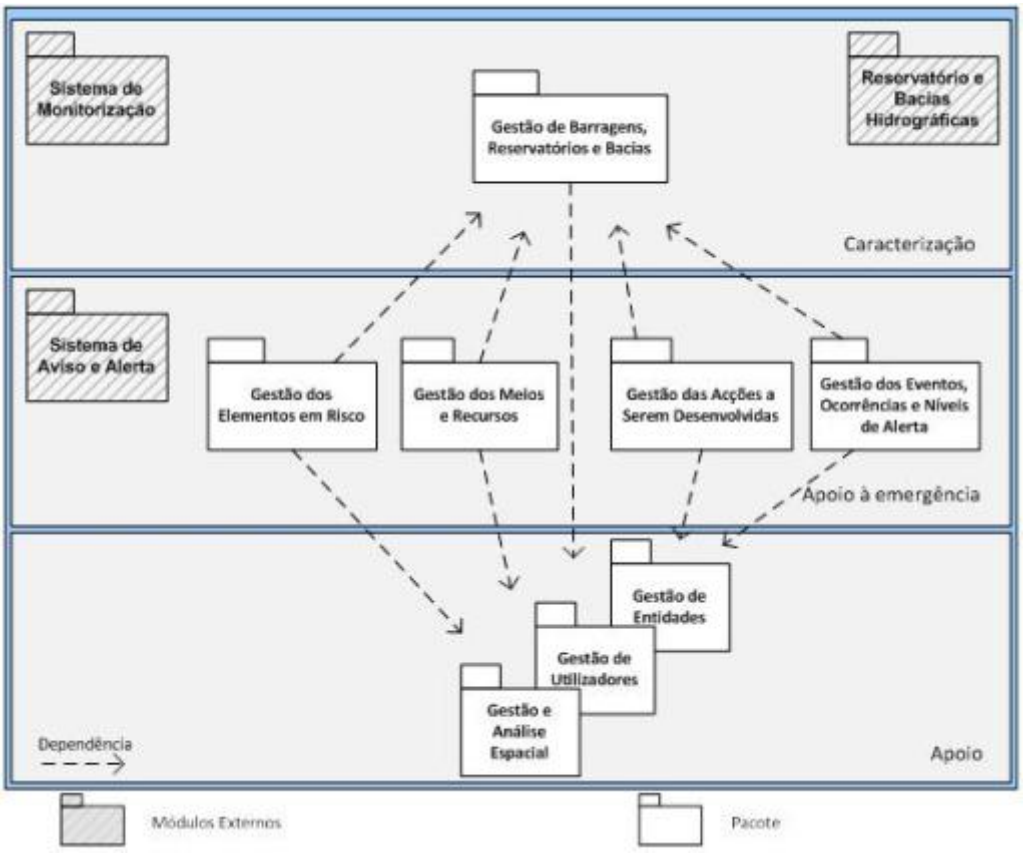

Figura 3.2: Estrutura do Sistema de Informação SAGE-B

Os módulos principais incluem:

a) um módulo de *Gestão de Barragens* que contém informação sobre a barragem e os seus órgãos hidráulicos e segurança;

b) um módulo de *Gestão de Elementos em Risco* que permite a caracterização dos elementos em risco, edificações e população residente na área de risco;

c) um módulo de *Gestão de Meios e Recursos* que inventaria todos os meios e recursos disponíveis para acorrer a uma emergência: viaturas, maquinaria, unidades de saúde, espaços para alojamento temporário;

d) um módulo de *Gestão de Ações a Desenvolver* que lista todos os procedimentos e ações de resposta à emergência;

e) um módulo de *Gestão e Análise Espacial* que disponibiliza funcionalidades para a representação geográfica e a análise espacial, bem como funcionalidades de acesso e navegação em mapas, constituindo a parte do repositório onde são coligidos temas geográficos como redes de vias de comunicação, redes hidrográficas, localização das barragens, mapas gerados com as zonas de risco;

f) um módulo de *Gestão de Utilizadores* que permite configurar os diferentes perfis de utilizadores de sistema e criar novos utilizadores;

g) um módulo de *Gestão de Entidades* que inventaria todas as entidades com responsabilidade na gestão de risco.

Neste projeto foi estudada a viabilidade da adaptação de alguns dos módulos do SAGE-B e respetivo reaproveitamento de código para o novo sistema de alerta precoce e aviso construído. Isto permite o uso de estruturas e funcionalidades semelhantes que na sua essência não necessitem de ser totalmente reinventadas.

Para além do SAGE-B e do *Sahana*, existem diversos exemplos de sistemas semelhantes.

O IGIS é um sistema de apoio à decisão espacial *Web-based open-source* para agilizar a resposta a desastres naturais. É definido como um conjunto de aplicações e serviços de informação geográfica que pode aceder via Internet ou sem fios a esse tipo de informação, a ferramentas de análise espacial e a serviços *Web* GIS [9]. Integra um sistema GIS, bases de dados espaciais e imagens de satélite.

O HUODINI é um sistema para integração e visualização flexíveis de fontes heterogéneas de informação (inclusive redes sociais) para a gestão de desastres [11].

Outros sistemas de gestão de informação relativa a desastres foram desenvolvidos tendo por base as tecnologias geo-informáticas [14]. Por exemplo, um servidor que possibilita aceder a informação com base em imagens de satélite de observação da Terra [12], um sistema de satélites de recolha de informação com utilidade para a gestão de desastres e para projetos humanitários [15] e um caso de estudo para a gestão do desastre de avalanche e alerta precoce numa autoestrada chinesa [25].

Sistemas na área da telemedicina [16] também foram considerados na gestão da resposta a emergências.

3.4 Sistemas de Previsão em Tempo-Real

Os sistemas de previsão em tempo real permitem fazer previsões de fenómenos ambientais a curto prazo. São plataformas informáticas que combinam os resultados de modelos de simulação e dados em tempo *quasi*-real.

O RDFS-PT [46] é um sistema de previsão em tempo-real para a costa Portuguesa. Integra modelos de simulação numérica (SELFE e WaveWatch3) e dados em tempo *quasi*-real das águas estuarinas e costeiras Portuguesas. Estes dados permitem validar os resultados dos modelos em diferentes variáveis como os níveis de água, salinidade, temperatura e velocidades. Baseia-se na tecnologia Virtual Columbia River [47], desenvolvida pelo CMOP.

Para além do RDFS-PT, existem diversos exemplos de sistemas de previsão em tempo-real aplicados nas mais diversas áreas científicas e sociais. Por exemplo, o INSEA [45] é um projeto em investigação para a gestão integrada de informação de modelos matemáticos, medidas locais e dados de observações por satélite para previsão em tempo-real da qualidade das águas em zonas costeiras, sistemas para gestão de tráfego nas estradas [27] e cálculo e apresentação aos passageiros [28] do tempo estimado de chegada de diversos meios de transporte a uma determinada estação/paragem, sistemas para avaliação e previsão de eventos extremos como inundações [41] ou relacionados com a área da meteorologia [35, 36, 39, 40].

3.5 Sistemas de Alerta e Aviso

O sistema de gestão de risco construído integra um sistema de aviso de emergência baseado em serviços de mensagens móveis (SMS e outras tecnologias) [7, 8] que é despoletado pelo sistema de alerta precoce. O objetivo é aumentar a velocidade de aviso, a cobertura espacial e a riqueza da informação enviada seguindo a aproximação teórica descrita em [7].

São exemplos os sistemas de alerta precoce para a poluição dos lençóis aquáticos [34], para desastres induzidos por condições meteorológicas desfavoráveis [37], para terramotos [38], para medição de deslizamentos de terra [32] ou para diversos desastres naturais e ecológicos [42].

3.6 Modelos de Simulação de Derrames

Os sistemas de modelação de derrames permitem simular os processos físicos, químicos e biológicos que determinam a propagação dos derrames e a sua transformação, sendo por vezes acoplados aos modelos hidrodinâmicos. Os resultados destes modelos permitem a análise potencial de risco de poluição e a prevenção e mitigação destes acidentes via utilização de computação de elevado desempenho. Os modelos mais utilizados atualmente são os de partículas com abordagem numérica Lagrangeana, distinguindo-se também o uso de abordagens numéricas Eulerianas e Eulerianas-Lagrangeanas. As principais diferenças, vantagens e desvantagens entre estas abordagens estão descritas em [24].

Existem diversos exemplos interessantes de sistemas de modelação de derrames de

hidrocarbonetos entre os quais o OSCAR [29, 30], o OILMAP [49], o SIMAP [50], o GNOME [48], o VOILS [24], o Virtual Columbia River [47] e o MOHID [44], que permitem realizar a modelação e simulação de cenários e processos.

3.7 Ferramentas para Data Mining

Foi efetuada a pesquisa e estudo de diversas ferramentas existentes vocacionadas para a realização de *data mining*. Esta pesquisa tinha como objetivo encontrar a ferramenta mais indicada para essa tarefa, tendo em conta o tipo de dados utilizado por cada uma, os algoritmos que usa, o que cada uma permitia fazer, as vantagens e as desvantagens de cada uma delas e o seu grau de utilidade no contexto deste trabalho.

Os requisitos iniciais para escolha da aplicação que seria utilizada passavam pelas seguintes características: ser *open-source*, permitir uma utilização simples e intuitiva, ter documentação acessível e completa e apresentar elevada *performance*.

Destacam-se em seguida algumas das aplicações mais relevantes.

3.7.1 Weka

O Weka [55, 66, 71] é um *package* de *data mining* que contém um conjunto de algoritmos de aprendizagem automática. Dentro desse vasto conjunto inclui vários dos algoritmos considerados mais importantes como o Naive Bayes, EM, AdaBoost, K-Means ou o *Nearest Neighbour* [65].

Foi desenvolvida em Java e corre em quase todas as plataformas. Apresenta uma GUI sendo também possível a sua utilização via linha de comandos. Inclui vários esquemas de classificação, associação, descoberta de regras de associação, agrupamento, etc.

Utiliza ficheiros ASCII com o formato ARFF. Estes incluem os atributos da instância do *dataset*, o tipo de cada atributo e valores possíveis no caso de tipos nominais, bem como os valores dos atributos das várias instâncias de dados a serem classificadas. Os algoritmos podem ser chamados diretamente sobre o *dataset* ou podem chamar-se a partir de código em Java.

Tem versões para processamento distribuído (Weka4WS [67]) e processamento paralelo (Weka-Parallel [68]). Fornece suporte ao pré-processamento automático de dados geográficos para a realização de *data mining* sobre informação espacial através de uma versão extendida do Weka 3.4 denominada Weka-GDPM [56, 69].

Existe igualmente a possibilidade de se utilizar um *package* do R chamado RWeka [70] para aceder ao Weka. O RWeka contém o código da interface, o *package* RWekajars contém o *jar* do Weka.

Esta ferramenta é passível de ser utilizada em imensos cenários. As seguintes referências são exemplos: extracção de informação genética [72], análise de informação

biológica relacionada com vírus [73], comparação de classificadores utilizando dados ligados a fraude com cartões de crédito [74] e obtenção de informação musical [75].

A eficiência da ferramenta, a documentação completa, a possibilidade de pré-processar a informação espacial automaticamente, a implementação da maioria dos algoritmos das diferentes categorias de *data mining* (árvores, *bayesianos*, regras, *lazy*, baseados em funções, etc) e a facilidade de utilização fazem do Weka a aplicação ideal para este projecto.

3.7.2 KNIME

O KNIME [78] é uma plataforma modular baseada no Eclipse com variadas *features*. Permite o acesso, transformação, análise estatística, *data mining*, previsão e visualização de dados.

Integra todos os módulos de análise do Weka. Existem *plugins* adicionais que permitem executar *scripts* do R.

Permite a importação e exportação de dados nos formatos mais comuns (ARFF, CSV) e suporta muitos dos algoritmos das categorias mais usuais (regras de indução, *bayes*, redes neuronais, árvores de decisão, *clustering*, etc).

3.7.3 R

O R [59] é uma linguagem altamente extensível para manipulação de dados, computação estatística e visualização gráfica. Encontra-se entre os ambientes mais utilizados para análise de dados [81]. Permite chamar e correr código em C, C++ e Fortran em tempo de execução podendo também os seus objetos serem manipulados a partir de código em C e Java.

Existem diversos *packages* adicionais para o R como o Rattle [58] que fornece uma GUI para *data mining* no R.

3.7.4 RapidMiner

O RapidMiner [57] é uma solução com versões *open-source* e código fechado, escrita em Java, para tarefas de aprendizagem automática e *data mining*.

Inclui diversos operadores para importação e exportação de dados em diferentes formatos de ficheiro bem como vários esquemas de aprendizagem para tarefas de regressão, classificação e *clustering*.

Para grandes quantidades de dados apresenta-se como uma solução poderosa ao ser capaz de lidar com uma crescente complexidade de processos. Mostra-se flexível ao ter uma natureza modular e transparente para os processos de análise de dados e fornecendo mais etapas de análise (operadores) que o Weka [55] e mais possibilidades de combinação destas. Permite trabalhar directamente sobre bases de dados com biliões de transacções

ao contrário do Weka que necessita que sejam utilizados exemplos mais simples. A *performance* é inferior à do Weka se bem que a Enterprise Edition permite ganhos escaláveis em relação aproximada do número de núcleos da máquina e usando uma versão paralela do algoritmo de aprendizagem.

Inicialmente, a habituação a todas as características do RapidMiner e à sua utilização pode ser árdua.

O estudo da previsão dos preços e tendências no mercado financeiro [76] ou a análise de dados biomédicos relacionados com doenças [77] são alguns exemplos de áreas onde esta ferramenta foi utilizada.

3.7.5 Outras Ferramentas

De entre várias outras ferramentas existentes capazes de cumprir tarefas de *data mining* haveria muito mais a referir. Descrevem-se de seguida e de um modo geral, mais algumas ferramentas válidas no contexto deste projecto.

O Orange [60] é um *software* orientado para aprendizagem e *data mining*. Inclui componentes para as tarefas de pré-processamento, filtragem, modelação, avaliação e exploração. Foi implementado em C++ e Python. Fornece suporte para análise de dados, visualização e *scripting*.

O ROSETTA [54] é uma ferramenta para modelação de conhecimento escrita em C++ e direcionada para a análise de dados tabulares. O seu *design* permite fornecer suporte a todas as fases dos processos de *data mining* e de descoberta de conhecimento como o pré-processamento dos dados, a computação de conjuntos de atributos iniciais, a geração, validação e análise de padrões descritivos ou regras *if-then*. Tem uma GUI intuitiva, existindo também uma versão do programa em linha de comandos que pode ser invocada por *scripts* em Python e Perl. Diversos algoritmos de computação podem ser usados com o núcleo do ROSETTA, tais como os presentes na biblioteca RSES que está embebida nesta ferramenta. O ROSETTA teve origem num trabalho ligado à análise de dados médicos [61] e foi utilizada em áreas como a análise de dados biológicos [79] e seleção de atributos em bases de dados de companhias aéreas [80].

Ferramentas/Caraterísticas	Desempenho	Facilidade de Utilização	Capacidade de Processamento	Dimensão dos Dados	*Open-Source*
Weka	3	4	4	4	Sim
KNIME	3	3	3	3	Sim
R	3	3	4	4	Sim
RapidMiner	3	3	4	4	Sim
Orange	2	3	3	3	Sim
ROSETTA	2	3	3	3	Sim
IBM SPSS Modeler	5	4	5	5	Não
STATISTICA	4	3	5	5	Não
SAS	5	5	5	5	Não
					Escala: 1-Mau 2-Fraco 3-Razoável 4-Bom 5-Muito Bom

Tabela 3.1: Comparação Entre Ferramentas de Data Mining

Ferramentas como o IBM SPSS Modeler [62], STATISTICA [63] ou SAS [64] por serem proprietárias não se enquadravam nos requisitos pretendidos.

A Tabela 3.1 apresenta uma tabela comparativa das ferramentas de *data mining* enunciadas.

Capítulo 4

Análise do Problema

Neste capítulo é dada ênfase à análise dos principais pontos que foram tratados durante o trabalho. Estes podem ser divididos em três níveis: estudo e concretização de bases de dados geográficas para os dados geográficos disponíveis da aplicação dos modelos WaveWatch3 (modelo de ondas) [82] e SELFE (simulação de níveis, correntes, salinidade e temperatura) [24, 83], de acidentes marítimos e do sistema *Sahana*; tarefas relacionadas com *data mining* dos dados armazenados em base de dados, dos dois modelos referidos; análise, desenho, programação e testes de novas funcionalidades ao nível do módulo de mensagens do *Sahana* (componente de aviso do sistema).

4.1 Tipologia da Informação

Segue-se a referência às características da diversa informação existente e aos obstáculos encontrados para o armazenamento da mesma.

4.1.1 WaveWatch3

Os dados resultantes das corridas diárias dos modelos referidos estavam acessíveis num servidor específico denominado *luna*. Estes *outputs* correspondem às previsões diárias para cada variável de saída do modelo. Estão armazenados numa pasta de previsões específica de cada modelo e local que inclui as pastas com a identificação do dia, mês e ano, respectivos.

Para o caso do WaveWatch3 há dados dos anos de 2010, 2011 e 2012 sendo que por motivos de armazenamento, os dados mais antigos se encontram comprimidos. O ficheiro de recolha de informação mais importante no contexto deste trabalho denomina-se *tab33.ww3*. É um ficheiro de formato ASCII onde a organização dos dados se encontra feita por colunas. Inclui atributos relacionados com a direção das ondas (pontos cardeais), o período da onda (valor da Física) e a altura significativa (1/3 das maiores ondas registadas). A Figura 4.1 mostra o cabeçalho do ficheiro de um determinado dia, identificando os diferentes atributos e alguns registos de exemplo.

tab33.ww3

Date	Time h m s	Hs (m)	L (m)	Tr (s)	Dir. (d.N)	Spr. (deg)	fp (Hz)	p_dir (d.N)	p_spr (deg)
20111201	0 00 00	3.159	182.1	10.15	294.9	40.48	0.0731	317.7	13.54
20111201	1 00 00	3.237	185.6	10.24	295.2	41.39	0.0726	317.4	13.32
20111201	2 00 00	3.292	187.5	10.28	295.3	42.20	0.0722	317.0	13.19
20111201	3 00 00	3.336	188.7	10.31	295.3	42.87	0.0718	316.4	12.98
20111201	4 00 00	3.350	188.4	10.30	295.1	43.26	0.0724	315.7	12.72
20111201	5 00 00	3.354	188.3	10.31	294.9	43.21	0.0733	315.1	12.33
20111201	6 00 00	3.335	187.6	10.30	294.8	42.86	0.0740	314.7	11.93
20111201	7 00 00	3.322	186.3	10.27	294.6	42.68	0.0744	314.5	11.50
20111201	8 00 00	3.300	184.1	10.20	294.2	42.75	0.0746	314.6	11.17
20111201	9 00 00	3.289	181.4	10.12	293.5	43.10	0.0748	314.8	10.89
20111201	10 00 00	3.281	177.4	9.99	292.2	43.81	0.0750	315.1	10.67
20111201	11 00 00	3.285	172.7	9.84	290.4	44.67	0.0752	315.3	10.63
20111201	12 00 00	3.292	167.1	9.66	288.0	45.60	0.0754	315.7	10.64
20111201	13 00 00	3.310	161.5	9.47	285.4	46.42	0.0756	316.0	10.69
20111201	14 00 00	3.328	155.4	9.27	282.4	47.12	0.0759	316.3	10.75
20111201	15 00 00	3.357	149.6	9.08	279.3	47.63	0.0764	316.5	10.84
20111201	16 00 00	3.389	143.7	8.89	276.1	47.72	0.0772	313.1	10.78
20111201	17 00 00	3.429	138.4	8.73	273.3	47.27	0.0781	313.1	10.83
20111201	18 00 00	3.481	133.4	8.58	271.1	46.44	0.0788	313.0	10.85
20111201	19 00 00	3.519	130.6	8.51	272.8	43.95	0.0793	313.0	10.92
20111201	20 00 00	3.495	130.8	8.54	278.3	40.95	0.0798	313.1	11.00
20111201	21 00 00	3.412	133.1	8.63	284.8	38.39	0.0802	313.1	11.16
20111201	22 00 00	3.415	131.5	8.58	289.8	35.69	0.0805	313.2	11.35
20111201	23 00 00	3.494	128.5	8.49	293.7	33.17	0.0808	313.1	11.69

Figura 4.1: Exemplo de Dados de Output do Modelo WaveWatch3

Pode-se visualizar que é gerado um registo de hora em hora. A coluna *Hs* representa a altura significativa, a coluna *Tr* é referente ao período da onda, a coluna *fp* apresenta a frequência de pico, a coluna *L* apresenta o comprimento de onda, a coluna *Spr* representa a distribuição, a coluna *p_dir* apresenta a direção de pico, a coluna *p_spr* é o *spread* de pico e a coluna *Dir* é referente à direção das ondas.

Estes dados seguem o formato padrão para dados de previsão e de histórico meteorológico *.grb* da WMO [85].

O objetivo principal consistiu no processamento adequado e na inserção dos vários registos dos diferentes ficheiros de texto numa única base de dados PostgreSQL para posterior análise por *data mining*.

4.1.2 SELFE

Para o caso do SELFE também estavam disponíveis dados dos anos de 2010, 2011 e 2012 sendo que, novamente, por razões relacionadas com o armazenamento, os dados mais antigos se encontram em formato *tar.gz*.

Os ficheiros mais importantes relativos aos atributos elevação, temperatura, salinidade e velocidade denominam-se, respetivamente, *1_elev.61*, *1_temp.63*, *1_salt.63*, *1_hvel.64*.

Estes ficheiros encontram-se em formato binário, sendo necessário encontrar e aplicar uma forma de conseguir obter os dados relativos a estes atributos.

O objetivo principal passava pelo processamento adequado e pela inserção dos vários valores de cada atributo na base de dados criada no PostgreSQL para posterior análise por *data mining*.

4.1.3 Acidentes

Para o caso dos dados relativos a acidentes marítimos houve a necessidade de os guardar numa base de dados geográfica única no PostgreSQL para posterior análise e visualização espacial dos dados num servidor de mapas.

Esta tarefa envolvia a transformação para o formato *geometry* do PostGIS das coordenadas de latitude e longitude específicas que identificam, através de pontos, os locais onde ocorreram esses incidentes no mapa.

Perante o fornecimento dos dados existentes em formato Excel, houve necessidade de os organizar e importar para ficheiros de texto a fim dos dados contidos serem processados adequadamente e inseridos nas tabelas respectivas da base de dados criada.

4.1.4 Sahana

O *Sahana* encontrava-se inicialmente configurado para a utilização do SGBD MySQL, motivando uma revisão e adaptação para PostgreSQL de vários ficheiros com código SQL da BD dispersos pelos vários módulos. Nesta migração da BD de MySQL para PostgreSQL vários detalhes foram tidos em consideração:

a) os tipos das chaves primárias de algumas tabelas que tiveram de ser alterados de BIGINT AUTO_INCREMENT para BIGSERIAL;

b) os tipos de outros atributos tiveram de ser também alterados (como BLOB para BYTEA e DATETIME para TIMESTAMP);

c) a criação de domínios para alguns atributos que tinham um conjunto específico de valores pré-determinado;

d) atualização do CURRENT_TIMESTAMP através da criação de TRIGGER e FUNCTION;

e) a correção no uso de plicas e aspas entre outras questões.

Uma configuração necessária para o funcionamento correto do sistema de aviso, ao nível do PHP, no ficheiro *php.ini*, foi a alteração do parâmetro *short_open_tag* de *Off* para *On* para que a interface do *Sahana* reconhecesse também as *tags* alternativas do PHP da forma <? ou <?=.

Durante a execução dos processos de instalação do *Sahana* foi gerado um ficheiro *sahana.conf* na pasta *conf.d* do Apache. Este permite mapear a diretoria do *Sahana* (*/usr/share/sahana/www*) com o espaço URL (*link/alias* denominado */sahana*) e com as permissões de acesso.

Também, a utilização do *Sahana* via *browser* requereu a alteração e adaptação do ficheiro *setup.inc* para que o sistema ficasse operacional. Este ficheiro, escrito em PHP, é executado quando se acede pela primeira vez a *http://localhost/sahana* e permite a criação do ficheiro de configuração *sysconf.inc*. A instalação inicial do sistema divide-se nos seguintes passos:

a) Verificação de que todas as dependências são satisfeitas (bibliotecas presentes);

b) Configuração da base de dados (informação de utilizadores e de ligação), verificação da ligação ao PostgreSQL e execução de cada um dos ficheiros SQL da BD;

c) Configuração do Sahana (informação de utilizadores e detalhes de instalação);

d) Geração do ficheiro de configuração *sysconf.inc* com toda a informação preenchida na pasta *conf* (*link* simbólico para */etc/sahana*).

Para efeitos de versão de produção e caso se pretenda que não apenas o *localhost* mas qualquer IP possa aceder a esta instalação do *Sahana*, é necessário editar o ficheiro de configuração do Apache *httpd.conf* através da indicação de *Allow all* em vez de *Allow from 127.0.0.1* na *tag Directory* do *alias* criado.

Após a configuração e instalação da plataforma *Sahana*, analisaram-se as funcionalidades existentes e futuras com o objetivo de sistematizar a melhoria da componente de aviso. Para tal, recorreu-se à criação de protótipos em papel (Apêndice A.1.4) que permitiram esquematizar a organização da interface do *Sahana* para as novas funcionalidades em mente, quais os controlos a ser utilizados de uma perspetiva informal sem aprofundar questões relativas à implementação, que informação deveria o sistema apresentar e receber, bem como a análise da navegação no sistema.

4.2 Data Mining

Para executar os processos de *data mining* sobre os dados dos modelos WaveWatch3 e SELFE houve a necessidade de desenvolver testes à ligação da ferramenta Weka (instalada na máquina a partir do *download* feito da página oficial) com o PostgreSQL. O estabelecimento desta ligação necessitou de algumas configurações, tais como, a presença dos ficheiros *.jar* do Weka e do JDBC para PostgreSQL no *classpath*, a extração e edição do ficheiro *DatabaseUtils.props* do *weka.jar* com o mapeamento dos tipos de dados do PostgreSQL, a instalação do *driver* JDBC e a configuração da ligação à base de dados.

Por fim, a execução do Weka através da linha de comandos na máquina de desenvolvimento procedia-se da seguinte forma:

java -Xmx768m -cp weka.jar;postgresql-9.1-901.jdbc4.jar weka.gui.explorer.Explorer

A opção *-Xmx768m* define o tamanho máximo de memória que a pilha da JVM pode atingir. A *performance* da ferramenta será mais baixa quanto mais memória a pilha do programa estiver a usar e quanto mais próximo esse valor estiver do tamanho máximo da pilha. Foi tomada esta medida, de maneira a aumentar a memória disponível para a pilha deste programa, criando-se as condições para um aumento da *performance* da ferramenta (na aplicação de algoritmos de *data mining* por exemplo) e diminuindo a ocorrência de *OutOfMemoryExceptions*.

A Figura 4.2 demonstra uma *query* efectuada com sucesso via Weka à tabela *selfe-data_values* e os registos resultantes.

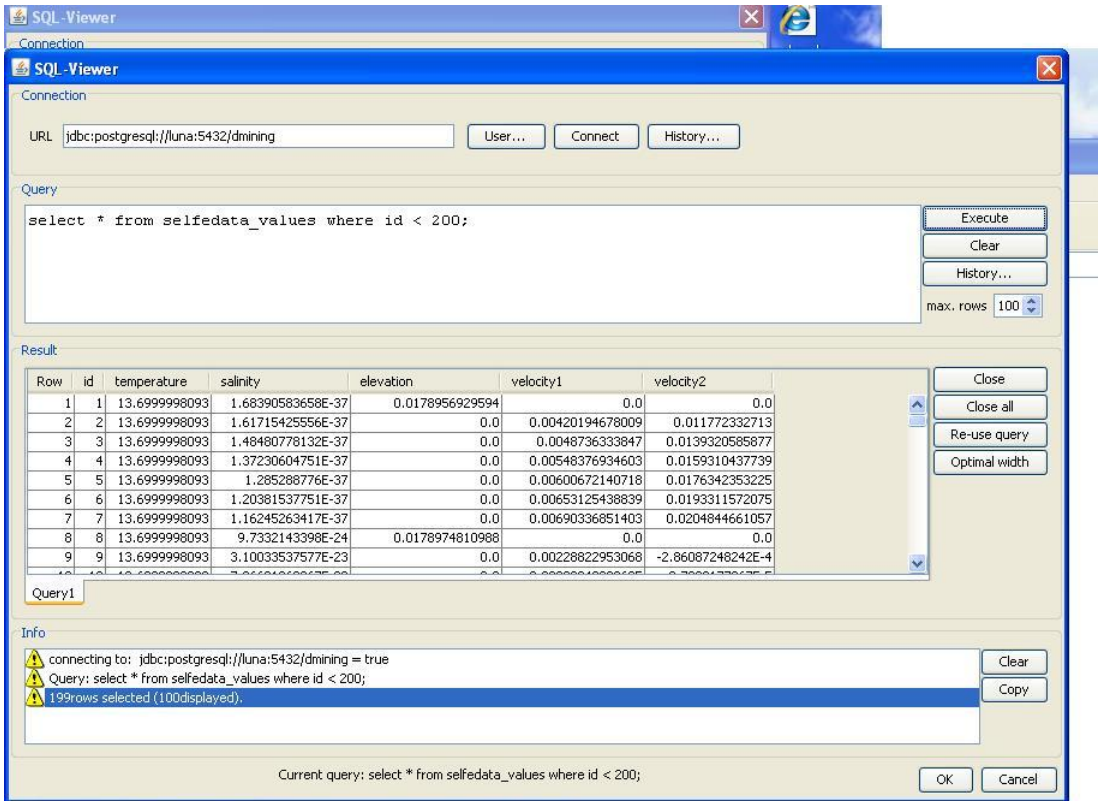

Figura 4.2: Query aos Dados SELFE via Weka

A Figura 4.3 apresenta os *clusters* resultados da execução do algoritmo COBWEB sobre os dados do modelo WaveWatch3 obtidos da BD e o gráfico com a distribuição de valores para uma configuração do eixo do *xx*, eixo do *yy* e cor a servir como exemplo de demonstração. Os dados e os respectivos *clusters* podem ser exportados para *.csv* ou *.arff*.

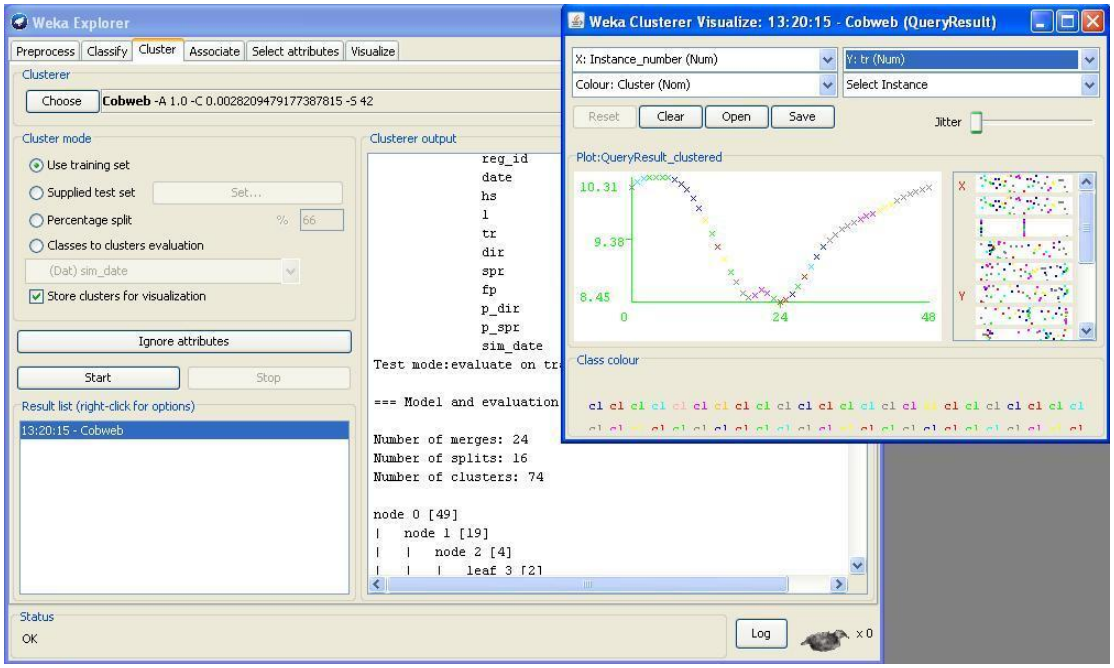

Figura 4.3: Resultados da Execução do Algoritmo COBWEB nos Dados WaveWatch3

Desta forma foram criadas as condições para que estes dados pudessem ser alvo de *data mining*, tarefa que está a ser realizada e que trará resultados à luz do dia sobre os dados processados brevemente.

Capítulo 5

Desenho da Solução

Neste capítulo são apresentadas as opções estruturais que foram tomadas na resolução dos problemas que se colocaram durante este trabalho. Nomeadamente, ao nível do desenho das bases de dados e dos programas que fazem uso dessas bases de dados.

5.1 Modelo de Dados

Nesta secção é dado ênfase aos modelos de informação criados, relacionados com cada tipo de dados envolvido.

5.1.1 WaveWatch3

A análise da organização do ficheiro permitiu concluir que era apenas necessária uma tabela simples dentro da base de dados para guardar estes dados. No Apêndice B encontra-se o código SQL correspondente.

Optou-se pela criação de um *script* em Java que permitisse o processamento adequado e a inserção mais rápida e simples dos dados de alguns dos ficheiros *tab33.ww3*, nomeadamente os mais recentes (dos últimos três meses de 2011 à altura) que não se encontravam comprimidos, de modo a testar o programa. Esta verificação permitiu ter maior segurança quanto à correta execução do que era pretendido para um nível escalável de informação disponível obtida por este modelo.

5.1.2 SELFE

Tendo em conta a existência de código em Python para leitura desta informação que poderia ser usado (para além da existência de código em Fortran para o mesmo fim caso se tivesse optado por esta abordagem), decidiu-se utilizar essas funções presentes no ficheiro *pyselfe_AA.py* para a leitura dos dados dos ficheiros binários de atributos. Em particular, o método *read_time_series* partindo de um pequeno exemplo de teste em Python. Este

método recebe o nome do ficheiro da característica, o número de ficheiros a ler dessa característica e o número do ficheiro por onde se vai começar a ler que aparece no início do nome do ficheiro. Como retorno obtemos uma estrutura *mdata* com os dados extraídos dessa característica (salinidade, temperatura, velocidade, elevação). Esta estrutura é uma matriz de quatro dimensões. Inclui o tempo *t* desde o início da simulação, o número de nós, de níveis e de variáveis dessa característica.

Para o caso da temperatura e da salinidade esta estrutura retorna 96 tempos *t**22464 nós*7 níveis*1 variável = 15095808 registos diferentes como demonstra a Figura 5.1.

Para o caso da elevação esta estrutura retorna 96 tempos *t**22464 nós*1 nível*1 variável = 2156544 registos diferentes como demonstra a Figura 5.2.

Para o caso da velocidade esta estrutura retorna 96 tempos *t**22464 nós*7 níveis*2 variáveis = 30191616 registos diferentes como demonstra a Figura 5.3.

Estes registos correspondem aos diferentes valores que foram inseridos nas tabelas *selfedata_indices* e *selfedata_values*. Os valores mais relevantes correspondem às variáveis de cada característica pelo que se decidiu separá-los numa tabela diferente da restante informação (*t's*, *nodes* e *levels*).

A Figura A.5 contém o diagrama UML simples com as tabelas criadas e o tipo de cada atributo. De salientar a criação de uma tabela *selfedata_simulation* com as datas de cada simulação, uma tabela *selfedata_values* com os valores de temperatura, salinidade, elevação e velocidades (como existem duas variáveis diferentes de velocidade existem dois valores em BD) e uma tabela *selfedata_indices* com os índices de ligação entre as duas tabelas acima descritas (de datas e de valores) e que contém também os tempos, os nós e os valores obtidos de *mdata*. O Apêndice B apresenta o código SQL relativo às tabelas criadas para conter estes dados.

Optou-se pela criação de um programa em Python para o correto tratamento e inserção desta informação em BD. Esta escolha deveu-se a estarmos a utilizar funções já existentes criadas na mesma linguagem. Foi necessário ter em consideração a elevada quantidade de informação a inserir pelo que a *performance* foi um ponto importante a ter em conta no modo como poderia ser realizada esta operação em tempo útil e para quantidades cada vez maiores de dados.

5.1.3 Acidentes

Os ficheiros de texto a processar continham os dados divididos por colunas em número igual ao número de colunas da tabela respetiva.

Havia cinco tipos principais de dados: acidentes, navios envolvidos, condições meteorológicas na altura do acidente, produtos transportados e imagens de satélite existentes.

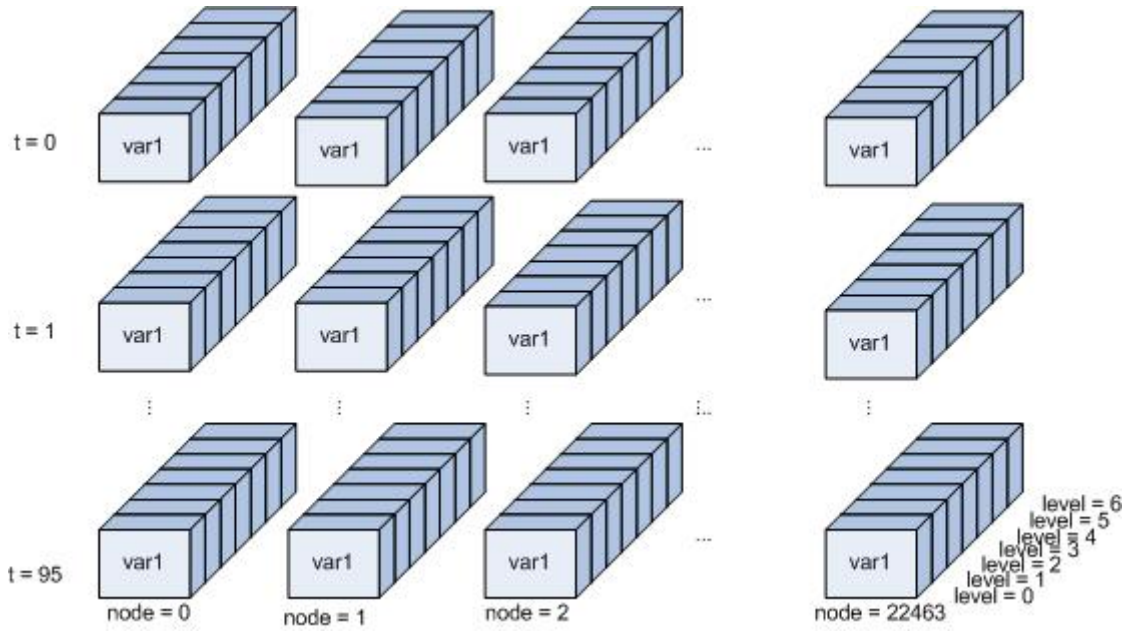

Figura 5.1: Matriz de Registos dos Atributos Temperatura e Salinidade

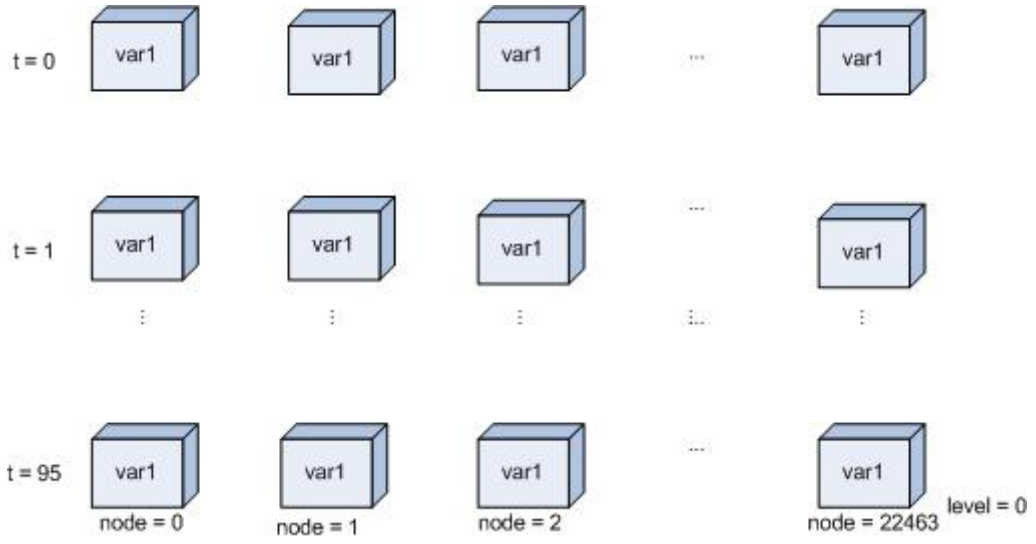

Figura 5.2: Matriz de Registos do Atributo Elevação

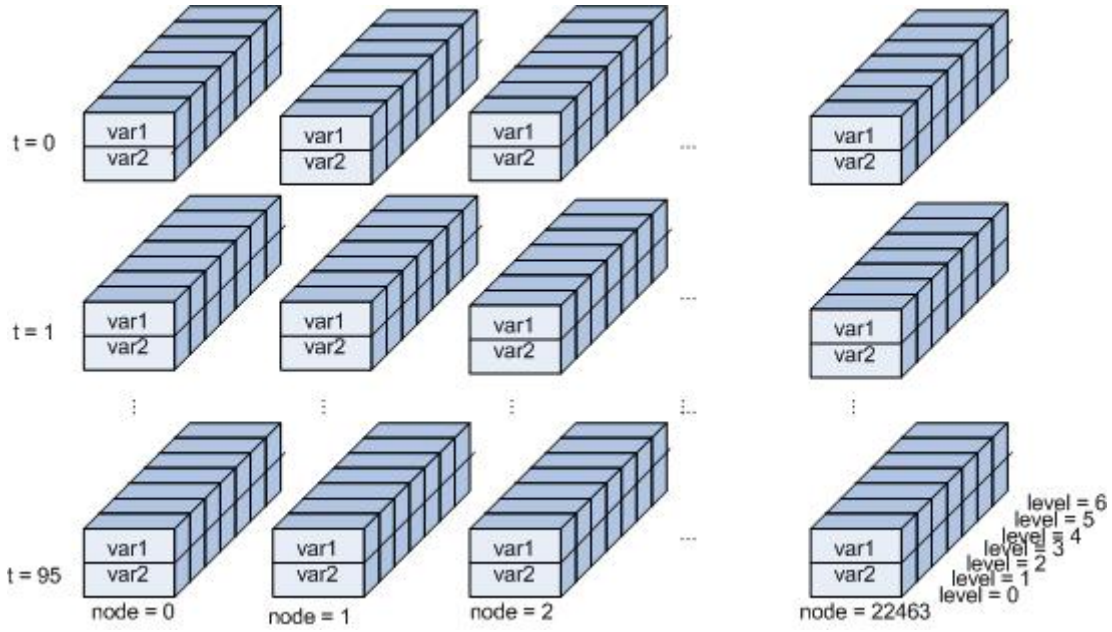

Figura 5.3: Matriz de Registos do Atributo Velocidade

A Figura A.6 apresenta o diagrama UML relativo às tabelas criadas e relações entre si. O Apêndice B apresenta o código SQL das tabelas criadas.

5.1.4 Sahana

Os ficheiros SQL adaptados para PostgreSQL são apresentados no Apêndice B (exceto o código relativo a INSERT's que não foi colocado em anexo por razões de espaço ocupado nesse documento).

A Figura A.7 demonstra o modelo UML com as várias tabelas do *Sahana* e respetivas associações principais. Este foi obtido recorrendo à ferramenta DbVisualizer e à sua capacidade de ligação a BD's PostgreSQL (entre outras), de criação de esquemas dinâmicos a partir dos vários componentes da BD (tabelas, sequências, índices, etc) e de exportação desses esquemas para ficheiro de imagem ou outros (*jpeg*, *png*, *gif*, *pdf*, *svg*).

As funcionalidades adicionadas e sua apresentação vêm descritas no Apêndice A.1.

5.2 Model/View/Controller

MVC (*Model/View/Controller*) é um padrão de arquitetura fundamental a nível do desenvolvimento *web* e do desenvolvimento de GUI *software*.

O *Model* é o objeto da aplicação contendo as partes de lógica e de dados da aplicação. Uma alteração no estado do *Model* implica uma atualização no que é mostrado pela *View* e nas operações de controlo do *Controller*.

A *View* traduz o *Model* no ecrã. Existe um protocolo entre estas duas componentes

que permite a atualização da *View* sempre que o estado do *Model* sofre modificações. Pode haver vários *Views* diferentes para um mesmo *Model*.

O *Controller* controla a interação com o utilizador. Tem operações de controlo do estado do *Model* e dos *outputs* da *View* fazendo a ligação entre estes dois componentes.

O trabalho associado à implementação das funcionalidades do sistema *Sahana* segue este padrão de *design*. A componente *View* é representada pelo código do formulário com os controlos mostrados no *browser* quando se acede a uma determinada funcionalidade do *Sahana*. A componente relativa ao *Controller* é representada pelo código que realiza a validação da informação que é inserida nesses controlos após ser feito o *submit* do formulário. Estes *inputs* são direcionados em seguida para o *Model*. Com base nos *inputs* dados, o *Model* realiza operações de atualização da base de dados ou operações de consulta de informação (o *Model* no *Sahana* é representado pelo código que realiza estas operações). A *View* é notificada pelo *Controller* desta alteração de estado do *Model* e atualiza os seus *outputs* com o novo conteúdo. Por exemplo, a inserção de uma nova área de risco faz com que a lista de áreas de risco a editar apresentada na interface contenha todas as áreas de risco em BD mais a nova área de risco inserida.

A Figura 5.4 apresenta o funcionamento geral do padrão MVC em PHP.

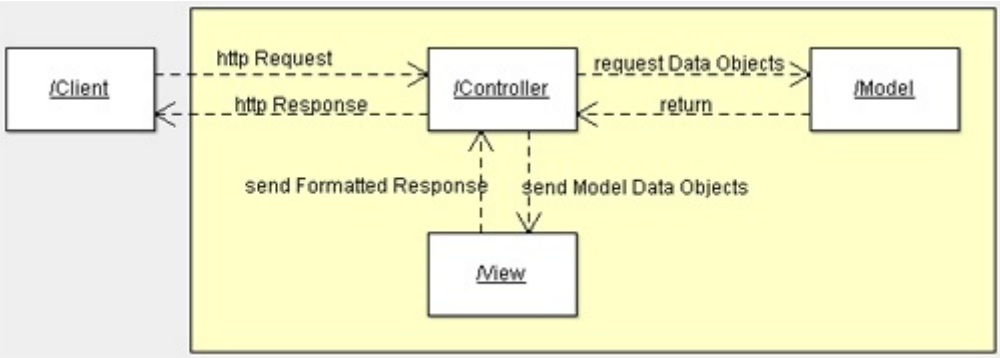

Figura 5.4: Diagrama Colaboração MVC

5.3 Arquitetura

A arquitetura da aplicação SAGE-*spill* é apresentada na Figura 5.5 onde são mostradas as interações entre os utilizadores da aplicação e os diferentes componentes de *software* e *hardware*. Contém um sistema de informação geográfica (SIG) típico e o esquema relativo ao sistema de alerta e aviso.

5.3.1 Sistemas de Informação Geográfica

Sendo a informação associada a catástrofes e desastres muito diversificada, é de grande utilidade para o planeamento e tomada de decisão em cenários desta índole, ter um SIG

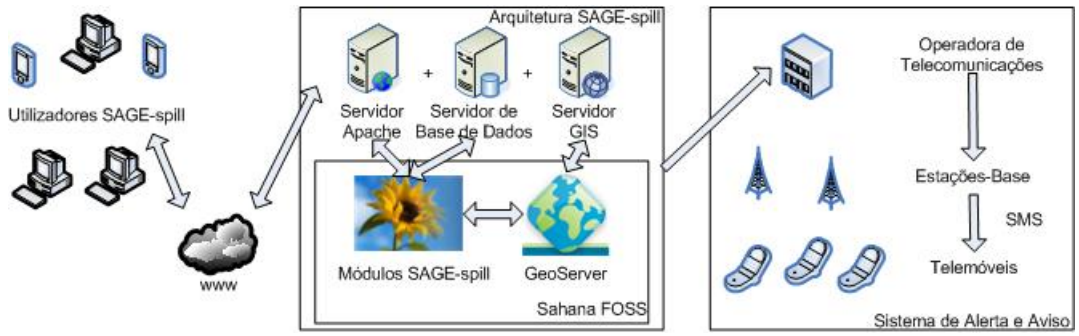

Figura 5.5: Arquitetura do Sistema SAGE-spill

que permita fornecer suporte para gestão e visualização de informação espacial.

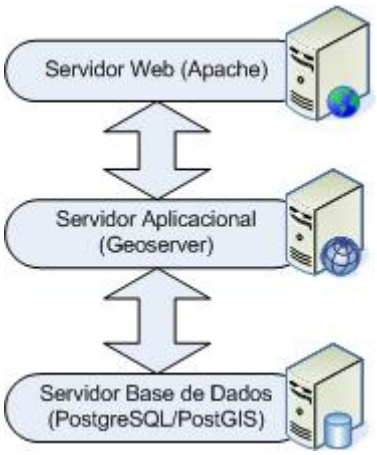

Figura 5.6: Arquitetura do Sistema de Informação Geográfica

Os sistemas de informação geográfica típicos são compostos por um servidor *web*, um servidor de bases de dados e um servidor SIG. A arquitetura do SIG que foi utilizado no âmbito deste trabalho é demonstrada na Figura 5.6. Recorreu-se à utilização da combinação Apache/PostgreSQL(PostGIS)/GeoServer(OpenScales).

5.4 JDBC

A JDBC API é usada no acesso aos dados do WaveWatch3 e dos dados geográficos de acidentes através dos programas em Java criados para tal, bem como no acesso do Weka aos dados em BD.

Esta API escrita em Java facilita a vida dos programadores que pretendem criar este tipo de aplicações permitindo gerir a ligação e o acesso à informação em bases de dados relacionais, o envio de *query's* e *update's* e o processamento dos resultados recebidos dessas *query's*.

A aplicação pode comunicar diretamente com a fonte de dados (como mostrado na Figura 5.7) ou ter uma camada intermédia de serviços (lógica de negócio) que faz o en-

caminhamento da comunicação entre o cliente e o SGBD, gere o controlo de acesso e as modificações que podem ser feitas nos dados e facilita a criação de aplicações no cliente.

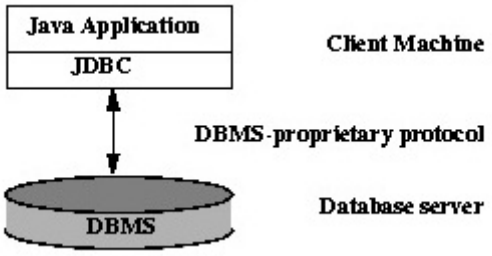

Figura 5.7: Comunicação Direta Entre Aplicação e Fonte de Dados

A JDBC API pode comunicar com fontes de dados de diferentes naturezas em ambiente distribuído.

Neste trabalho foi usado o *.jar* para PostgreSQL *postgresql-9.1-901.jdbc4*.

Capítulo 6

Implementação da Solução

Neste capítulo são apresentados os passos de implementação realizados e as ferramentas que foram utilizadas tanto a nível do tratamento e inserção das diferentes fontes de informação a tratar como a implementação de novas funcionalidades a nível do sistema *Sahana* incluindo o estudo sobre a utilização de mapas para apresentação de informação geográfica.

6.1 Ambiente de Desenvolvimento

A máquina utilizada durante o desenvolvimento tinha instalado o sistema operativo Fedora 16. Porém, foram necessárias algumas ferramentas de forma a suportar todo o desenvolvimento feito. Instalou-se o PostgreSQL, o Apache e o PHP bem como algumas bibliotecas relacionadas com estas ferramentas. Esta instalação foi maioritariamente concretizada através do *software* de gestão de pacotes *yum* disponível no Fedora.

6.1.1 PostgreSQL

Para a instalação do PostgreSQL foi necessário realizar os seguintes passos:

a) Correr os seguintes comandos pela ordem que são apresentados como utilizador *root*:

yum -y install postgresql postgresql-server postgresql-contrib php-pgsql

service postgresql initdb

b) Podem ser aplicadas as modificações que forem do interesse do utilizador aos ficheiros *postgresql.conf* e *pg_hba.conf* reiniciando em seguida o PostgreSQL com o comando:

pg_ct -o '-i' restart -m fast

c) Mudando para o utilizador *postgres* pode-se em seguida criar uma base de dados e aceder-lhe através dos comandos:

su - postgres

createdb <database_name>

psql <database_name>

6.1.2 Apache

Para a instalação do Apache foi necessário realizar os seguintes passos:

a) Correr o seguinte comando como utilizador *root*:

yum -y install httpd

b) As configurações do Apache são feitas no ficheiro */etc/httpd/conf*

6.1.3 PHP

Para a instalação do PHP foi necessário realizar os seguintes passos:

a) Correr o seguinte comando como utilizador *root*:

yum -y install php php-soap php-devel php-pear

b) Criar um ficheiro *index.php* com o seguinte conteúdo:

<? php phpinfo(); ?>

c) Criar o ficheiro */etc/httpd/conf.d/phpinfo.conf* com o seguinte conteúdo relativo ao *alias* e permissões:

<Directory <directory_where_is_index.php_file>

order deny,allow

deny from all

allow from localhost

</Directory>

d) Reiniciar Apache

service httpd restart

e) Aceder ao *localhost* no *browser* para verificar a correta instalação.

6.1.4 Psycopg2

Este *driver* é usado na comunicação entre o Python e o PostgreSQL à semelhança do JDBC para aplicações Java. Este foi o *driver* usado no *script* em Python para tratamento dos dados do SELFE.

Esta API foi desenvolvida em C. Tem uma boa *performance* e uma implementação leve. Consegue trabalhar bem com *arrays* do PostgreSQL convertendo-os para listas do Python sem ser necessário criar código manualmente para fazer o *parse* dos *arrays*. Um aspecto a melhorar passa pelo suporte do lado do servidor a *prepared statements*.

6.1.5 ADOdb

O ADOdb é uma biblioteca que aplica uma camada de abstração entre o PHP e o SGBD. Este *driver* suporta entre outros o PostgreSQL. Este foi o *driver* utilizado ao longo do código de implementação das funcionalidades do *Sahana* para acesso aos dados em BD.

Apresenta boa *performance*, evolução contínua, documentação completa, uma grande comunidade de utilizadores, é *open-source* e suporta as principais funcionalidades a nível de SQL requeridas por clientes empresariais.

6.2 Inserção de Informação em BD

Esta secção explica pormenorizadamente como foram implementados os programas de processamento de informação do WaveWatch3, SELFE e acidentes. Aborda também a implementação de novas funcionalidades no sistema *Sahana*.

6.2.1 WaveWatch3

O código do *script* criado (Apêndice B) recorre à utilização de um ficheiro XML com os dados de ligação à base de dados (*host*, porto, nome da BD, *login* e *password*). Posteriormente, é feito o *parse* das *tags* deste ficheiro para obter os valores respectivos.

O código faz a chamada do comando *find* da *bash* para encontrar todos os nomes absolutos de ficheiros *tab33.ww3* na diretoria *..forecastsww3-nadata2011* e subdiretorias. Faz depois o *parsing* dos dados dos ficheiros encontrados e insere a informação na BD, tendo particular atenção aos tipos dos atributos (*dates* e *floats*).

Para a ligação à BD foi necessário utilizar o *.jar* do JDBC para PostgreSQL. Este foi colocado na mesma diretoria dos *.class*. Para correr o *script* foi necessário editar a *classpath*, o que poderia ser feito de várias formas: em *runtime* no comando para correr o programa indicando o nome da diretoria onde estavam os *.class* e a localização do ficheiro do JDBC, ou antes de correr o programa exportando a *classpath* com estes dois caminhos ou editando esta variável no ficheiro *.bashrc*.

6.2.2 SELFE

O código do programa em Python criado (Apêndice B) para solucionar este problema usa uma solução semelhante em alguns aspetos à implementada para o WaveWatch3.

O código faz a chamada do comando *find* da *bash* utilizando a classe Popen com os vários componentes do comando a executar. Neste caso, pretendia-se executar uma pesquisa nas pastas de diversos dias das previsões para Aveiro, procurando em quais delas existiam ficheiros com os nomes de alguma das características em jogo. Sendo quatro as características diferentes, foram feitas quatro procuras. O resultado de cada procura é uma lista de nomes absolutos correspondentes ao caminho para cada ficheiro dessa característica específica. As quatro listas resultantes são colocadas dentro de uma lista com a seguinte estrutura:

[[temp_file_name1, temp_file_name2, ...], [salt_file_name1, salt_file_name2, ...], [elev_file_name1, elev_file_name2, ...], [hvel_file_name1, hvel_file_name2, ...]]

A criação da conexão ao PostgreSQL também é feita obtendo os dados da ligação a partir do *parse* de um ficheiro XML. É necessário utilizar o adaptador de bases de dados PostgreSQL denominado Psycopg2 para Python. A partir da ligação pode ser criado um cursor para poder executar os comandos SQL.

Em seguida, são obtidas as quatro estruturas *mdata* a partir do conteúdo dos quatro ficheiros de características correspondentes a uma determinada data (esta é obtida a partir do nome absoluto de um dos quatro ficheiros de características), recorrendo ao método *read_time_series* já previamente programado e apresentado acima. A lista dos quatro *mdatas* e da data a que correspondem é passada ao método que trata da inserção da informação.

A data é inserida em BD se ainda não existir. Utilizando o número maior de tempos *t*, de nós horizontais e de níveis verticais entre as quatro características e a lista de *mdatas*, são construídos dois *memory files* (*string buffers*). Para cada registo a inserir de um ficheiro foram criados dois dicionários (um com os valores do id do registo, do *t*, do *node* e do *level* e outro com os valores das várias características arredondados). Os arredondamentos seguidos foram: temperatura (em graus centígrados), 2 casas decimais (ex: 13.69); salinidade (sem unidades no Sistema Internacional), 2 casas decimais (ex: $1.68e^{-37}$); elevação (em metros), 3 casas decimais (ex: 0.017); velocidade (em m/seg), 3 casas decimais (ex: $0.000704997917637229 \sim 0.001$).

Recorreu-se ao uso das sequências do Python para a criação destes dois objectos StringIO. Cada registo foi sendo anexado aos já existentes (os dados dos dois dicionários foram sendo anexados ao *memory file* respetivo). O cursor recebeu como argumentos os dois StringIO para ser feita a inserção de todos os registos em cada um dos dois *memory files*. Esta inserção foi realizada através da chamada do método *copy_from* que anexa a uma tabela o conteúdo do *string buffer* exatamente como este foi construído. Como tal, o nome das tabelas indicadas onde se insere a informação deve corresponder à estrutura do

memory file.

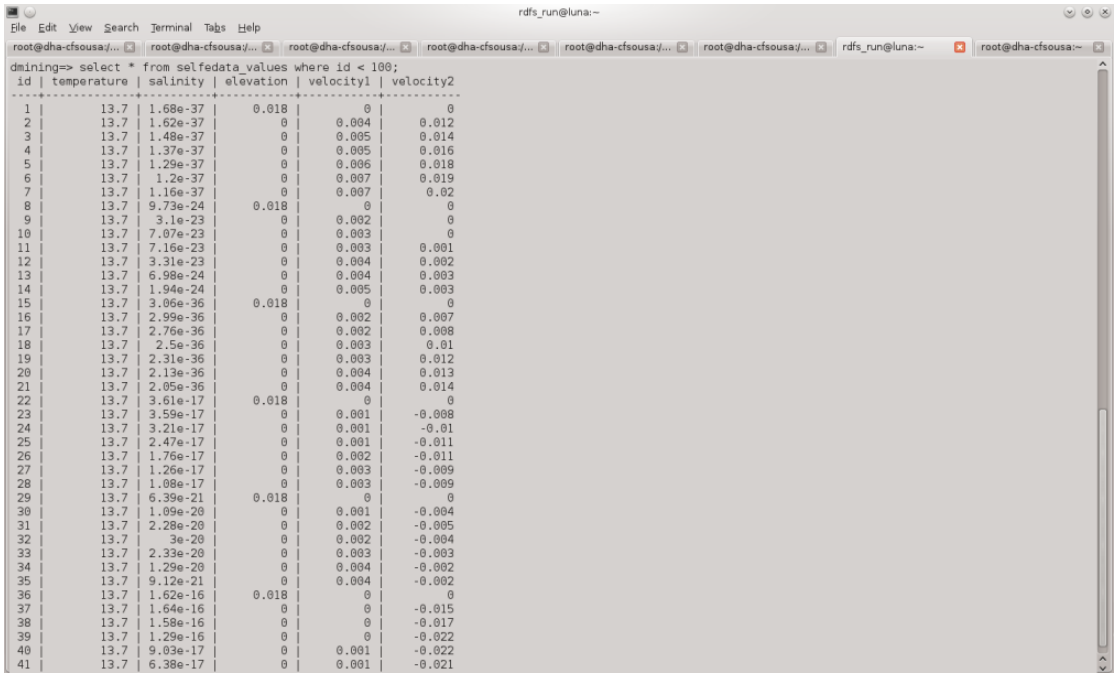

Figura 6.1: Valores Características SELFE em BD PostgreSQL

A Figura 6.1 apresenta alguns valores das características (temperatura, salinidade, elevação e velocidades) que foram inseridos.

Um novo TABLESPACE numa localização com capacidade apropriada, para conter o conteúdo das tabelas descritas, teve de ser criado no PostgreSQL de forma a permitir um crescimento escalável da base de dados.

A Figura 6.2 apresenta os atributos das sequências de cada uma das tabelas o que nos permite visualizar o número de ficheiros de dados inseridos (cerca de 170) e o número de registos diferentes nas tabelas de índices e de valores (cerca de 2500 milhões). A única coluna relevante é a coluna *last_value*. Este valor é calculado através da multiplicação do número de ficheiros inseridos (171) pelo tamanho *standard* da estrutura *mdata* (96 * 22464 * 7 * 1 = 15095808 registos). A inserção dos dados relativos a estes valores demorou cerca de dois dias a ser realizada o que permitiu uma melhoria de *performance* significativa (cerca de 18 vezes mais rápida a concluir esta tarefa em relação à primeira versão do *script* programada e executada).

Relativamente ao acima descrito, há dois pontos importantes a reter: a utilização de sequências e de *string buffers* permitiu resolver o problema de excesso de consumo de memória em cópias de e para dicionários muito grandes. A utilização de *copy_from* em vez de *executemany* permitiu diminuir substancialmente (na ordem das 6 horas para cerca de 18 minutos por ficheiro) o tempo de inserção da informação (testado com as funções de tempo do Python). O tempo de inserção da informação de um dia de exemplo é mostrado na Figura 6.3.

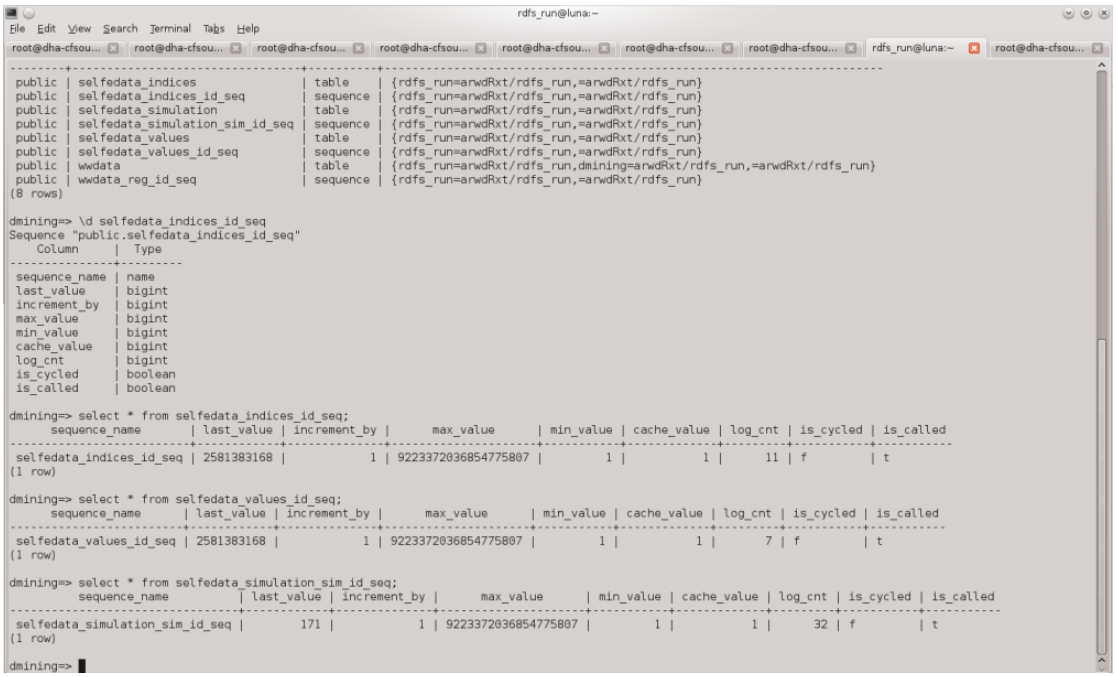

Figura 6.2: Sequências das Tabelas SELFE

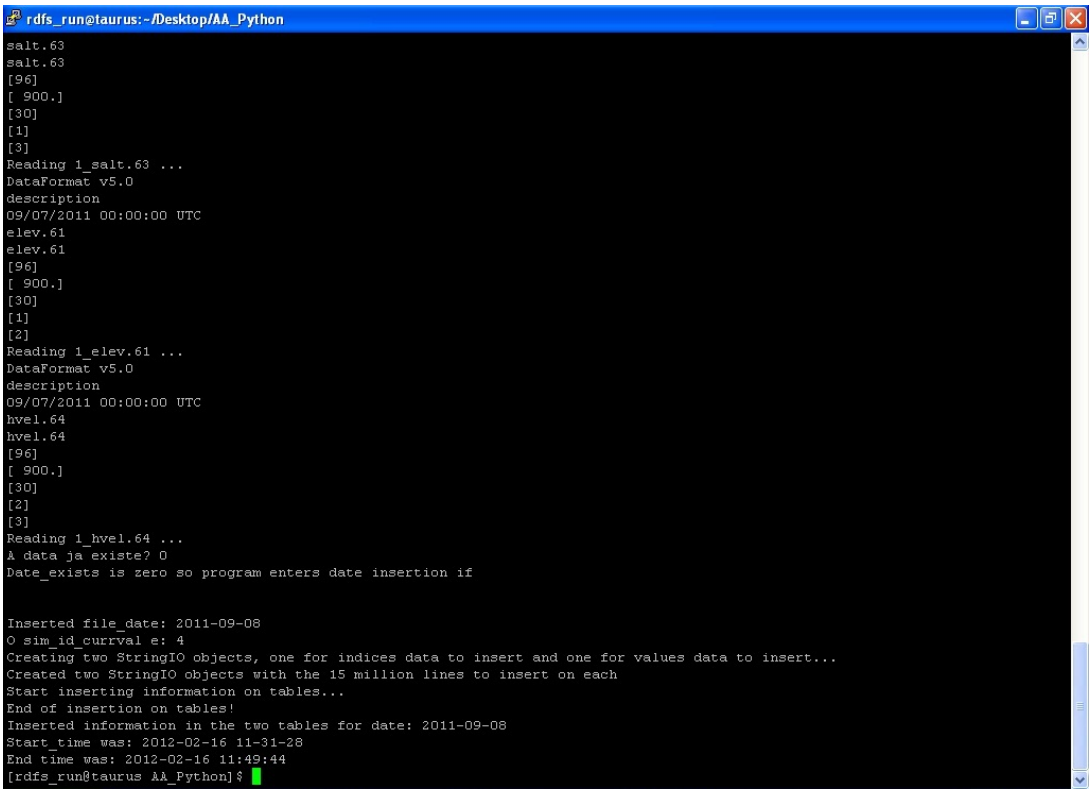

Figura 6.3: Tempo Inserção da Informação de um Dia no Modelo SELFE

6.2.3 Acidentes

Foi criado um programa em Java (Apêndice B) que permite processar os ficheiros de texto e fazer a inserção dos dados no PostgreSQL.

Os dados para a ligação são obtidos a partir do *parsing* de um ficheiro XML. Foi necessário dar indicação à *classpath* da localização do *.jar* do JDBC para PostgreSQL à semelhança do programa feito para processar os dados do modelo WaveWatch3. Os dados nos ficheiros são lidos e a sua informação é inserida na BD recorrendo ao uso de *prepared statements* próprias para cada tabela (tendo em conta os atributos e os tipos de dados de cada uma).

6.2.4 Sahana

Todo o código relativo aos *forms* com as novas funcionalidades criadas está disponível no Apêndice B.

Após o levantamento de requisitos (Apêndice A.1) determinou-se a implementação das funcionalidades de envio de mensagens para todos os contatos num determinado local onde tenha ocorrido um acidente, de inserção de novas áreas de riscos para armazenar a informação de locais onde ocorram acidentes e de edição da informação associada às áreas de risco.

O ficheiro *main.inc* do módulo de mensagens contém o código de lançamento das diferentes funções relativas a este módulo. O *include* do ficheiro *message_forms.inc* permite a execução do código associado a essas funções.

Para o caso do envio de mensagens para um determinado local era necessário ter um campo textual para a mensagem a enviar, uma lista que permitisse escolher a área a ser abrangida pelo aviso (esta foi preenchida através da consulta à BD utilizando ADOdb para retornar os resultados relativos aos nomes, valores e níveis de risco das áreas de risco existentes), a inserção do *flash* OpenScales para apresentação da área de risco escolhida no mapa, dois campos *readonly* apresentando o valor (ou intervalo) de risco em minutos associado à área de risco escolhida, um campo com a cor associada ao nível de risco dessa área e um botão com código Javascript por trás para efectuar o envio da mensagem.

Ao escolher outra área de risco na lista de áreas de risco, também os valores apresentados nos restantes campos são modificados em conformidade, recorrendo-se ao uso de jQuery, com a informação que estiver associada a essa nova escolha em BD.

Para a integração dos mapas no *browser* foi usado o exemplo na pasta *bin* do OpenScales que engloba os *.swf* e *.js* necessários. Na fase de configuração no *browser* do *flash* foi necessário resolver um erro de carregamento dos *.swf* com a identificação *Error #2046* através da indicação da opção *-static-link-runtime-shared-libraries=true*. Esta foi acrescentada aos argumentos do comando de *build* do *flex_builder* no projecto criado na ferramenta Eclipse para o código do OpenScales. Feito um novo *build* ao projeto para compilar

os *.swf* permitiu que estes fossem mostrados corretamente no *browser* e que o problema fosse resolvido.

Em tempo de compilação, o compilador *mxmlc* presente no Flex Builder associa dinamicamente a aplicação às RSL's necessárias a esta. O Adobe Flash Player precisa de verificar a autenticidade das assinaturas dessas RSL's para se certificar que o *flash player* não corre código suspeito.

A Figura 6.4 mostra o ecrã com os vários controlos criados e valores de exemplo obtidos da base de dados do *Sahana* (consultar código respetivo no Apêndice B).

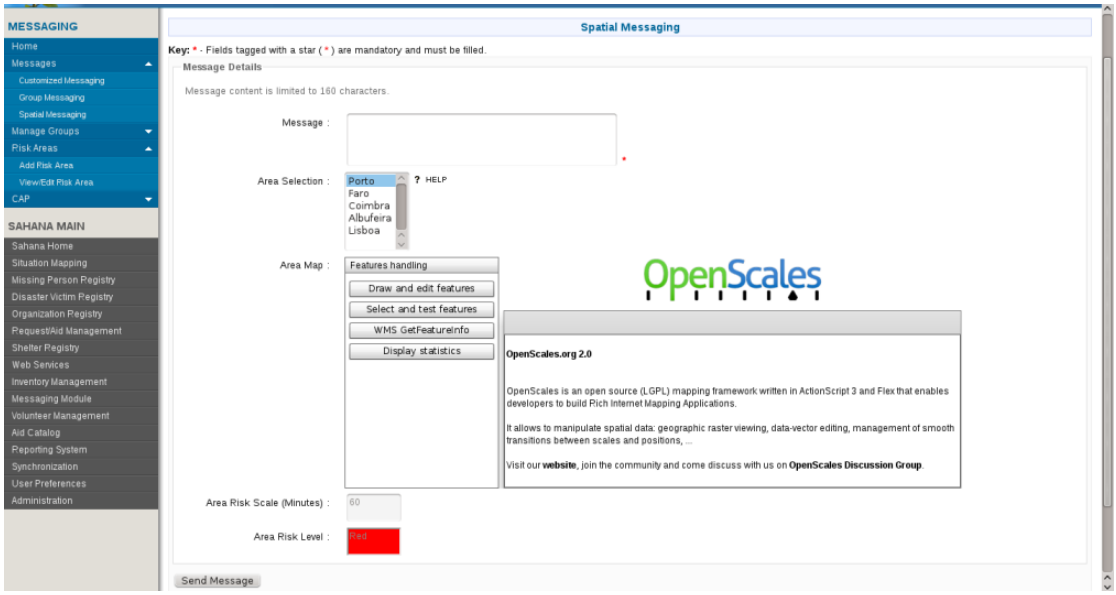

Figura 6.4: Ecrã Spatial Messaging

Para o caso da inserção de novas áreas de risco era necessário ter uma caixa de texto para indicação do nome da área de risco a inserir, um campo de *upload* do ficheiro relativo ao polígono da área de risco (*kml*, *shapefile*), um campo textual com a descrição associada à área de risco para permitir inserir descrições mais pormenorizadas, caixas para indicação do valor (ou intervalo) em minutos para evacuação das pessoas nessa área de risco, caixas *readonly* com a escala de cores relativas ao nível de risco desta área para escolha e botões para limpeza dos campos e para guardar a informação em base de dados. Houve a necessidade de criar uma tabela de áreas de risco na BD do *Sahana* para armazenar estes dados. A informação inserida nos campos é devidamente validada e a sua inserção no PostgreSQL é corretamente efetuada. Todos os campos são devidamente validados, sendo que havia a necessidade de identificar se estávamos a inserir uma área de risco com um valor ou um intervalo em minutos para o *risk scale* e saber se o nome (único) da área de risco a inserir já tinha sido inserido antes. A Figura 6.5 apresenta o ecrã demonstrativo desta funcionalidade.

Para o caso da edição de informação associada às áreas de risco foi criado um primeiro ecrã (Figura 6.6) com a lista de áreas de risco em BD incluindo a informação do nome, do

valor (ou intervalo) em minutos disponível para evacuar a área e a cor associada ao nível de risco. Apresenta também o mapa com a área escolhida na lista e um botão para edição. Apenas é possível escolher uma área para edição de cada vez.

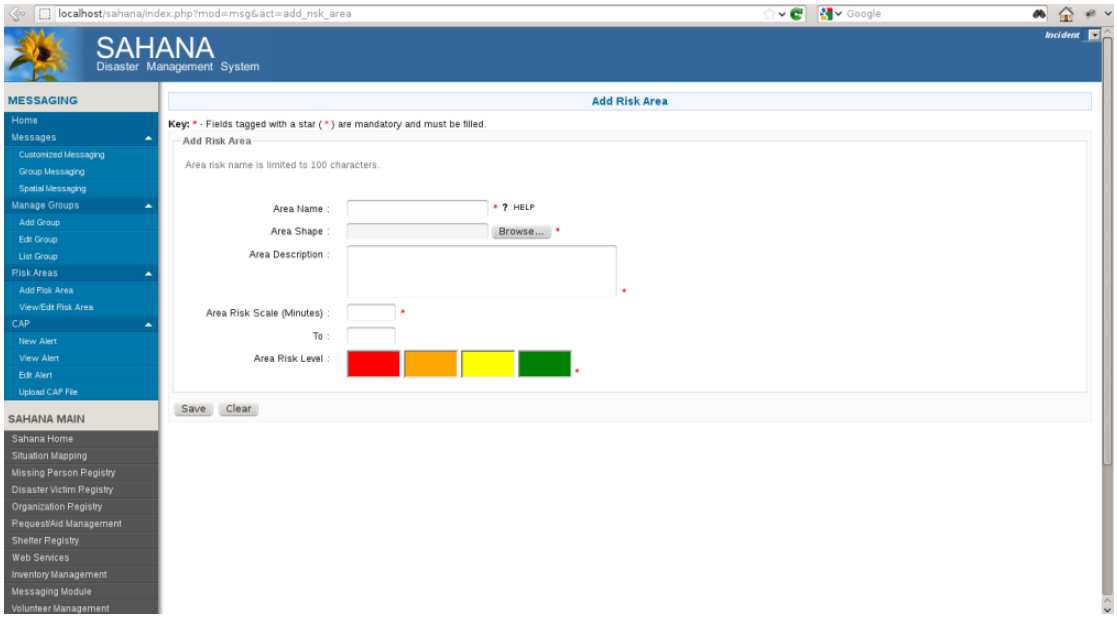

Figura 6.5: Ecrã Add Risk Area

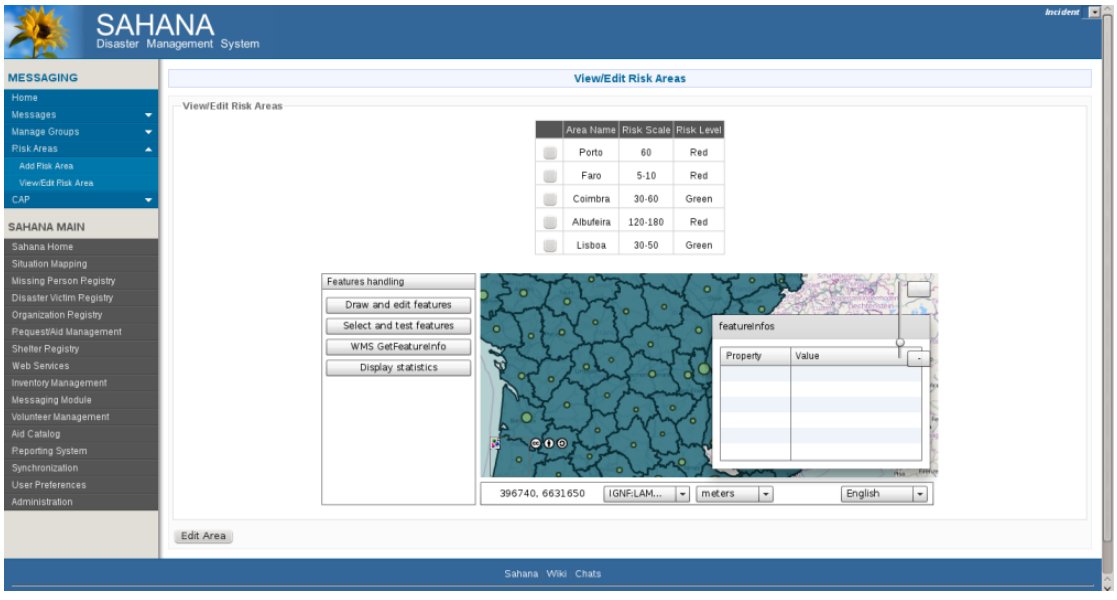

Figura 6.6: Ecrã View/Edit Risk Area Checkboxes

O segundo ecrã (Figura 6.7) é mostrado quando se indica que se quer editar uma determinada área escolhida da lista. Aqui são mostrados os campos textuais com o nome da área atual, valor (ou intervalo) em minutos para evacuação atual, cor do nível de risco atual, campo de upload do ficheiro relativo ao polígono identificador da área de risco e o

campo textual com a descrição atual bem como botões para salvar a informação editada, apagar o registo desta área de risco da BD e limpar os campos. As validações a efetuar foram essencialmente as mesmas realizadas para a inserção de uma nova área de risco.

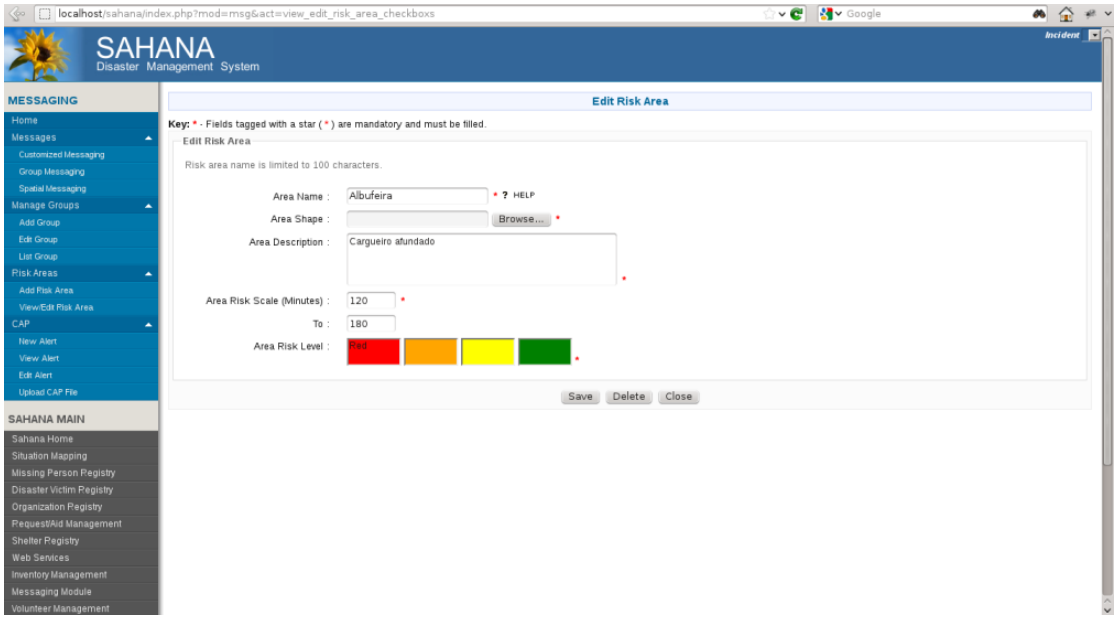

Figura 6.7: Ecrã View/Edit Risk Area Edition

6.3 PostGIS

O PostGIS é uma tecnologia que fornece suporte geográfico para bases de dados PostgreSQL. Permite gerir objetos espaciais tal como se fosse qualquer outro tipo de dados numa BD.

Existem três factores que favorecem a otimização da *performance* em BD's espaciais: suporte à indexação espacial multidimensional para o suporte eficiente às operações espaciais, tipos de dados espaciais utilizados fazem referência a formas (temos como exemplos os pontos, as linhas e os polígonos) e funções espaciais em SQL para pesquisa de informação espacial.

A sua instalação foi realizada através do *yum* na máquina de trabalho. Para as bases de dados dos modelos, dos acidentes e do *Sahana* foi necessário ativar este suporte através da execução dos seguintes passos:

a) Ativar o suporte à linguagem procedimental PL/pgSQL na BD visto muitas das funções do PostGIS estarem escritas nesta linguagem através do comando:

createlang plpgsql <nome_da_BD>

b) Carregar as definições de objetos e funções do PostGIS através do comando:

psql -d <nome_da_BD>-f postgis.sql

c) Carregar as definições em *spatial_ref_sys.sql* para um conjunto completo de coordenadas EPSG em sistema:

psql -d <nome_da_BD>-f spatial_ref_sys.sql

A criação de tabelas com dados geográficos segue a forma descrita no seguinte excerto de código:

```
CREATE TABLE "public"."acidentes" (gid serial PRIMARY KEY,
"zid" int4,
"name" varchar(20),
"elev" float8,
"icon" int2);
SELECT AddGeometryColumn('public','acidentes','the_geom','-1','
 POINT',2);
INSERT INTO "public"."acidentes" ("zid","name","elev","icon",
 the_geom) VALUES ('0','FV Blanca Ines','0','81',
'0101000000AF269E158D4422C0A8CBED0F32944440');
```

Neste excerto de código de exemplo, é criada uma coluna geométrica específica utilizando a função de gestão *AddGeometryColumn*. Esta recebe o nome da tabela existente, o nome da coluna de tipo geométrico a criar, o *srid*, o tipo de dados e a dimensão. É também inserido um registo de exemplo na tabela criada. Também é possível efetuar o carregamento de dados geográficos em *shapefiles* utilizando o carregador shp2pgsql em vez de usar código SQL formatado.

A transformação da representação KML das coordenadas para o tipo *geometry* do PostGIS é feita através do construtor geométrico *ST_GeomFromKML*. Basta correr o seguinte excerto de código na base de dados geográfica através do psql (ferramenta de interação com o PostgreSQL através do terminal):

```
SELECT ST_GeomFromKML('
      <LineString>
            <coordinates>-71.1663,42.2614
                  -71.1667,42.2616</coordinates>
      </LineString>');
```

6.4 GeoServer

O GeoServer é um servidor WFS/WMS *open-source* escrito em Java para edição e partilha de informação geoespacial.

Para instalar o GeoServer basta realizar os seguintes passos:

a) Fazer o *download* da última versão em binário estável (independente do sistema operativo);

b) Exportar a variável de ambiente JAVA_HOME com o caminho para o *jre*:

export JAVA_HOME=/usrt/lib/jvm/jre

c) Fazer o *unzip* e mudar para a directoria *geoserver2.1/bin* para poder executar o *script startup.sh*:

unzip geoserver2.1

cd geoserver2.1/bin

./startup.sh

d) Aceder a *http://localhost:8080/geoserver* e fazer *login* com os dados:

user: admin

password: geoserver

Para disponibilizar os dados geográficos no GeoServer basta criar uma *data store* com os dados na BD geográfica através da indicação dos dados de ligação ao PostgreSQL e uma *feature type* (gerando a respetiva *bounding box*). Os dados carregados podem ser visualizados em seguida no mapa carregando no *link* do nome da *layer* criada em *Map Preview*.

6.5 OpenScales

O OpenScales é uma biblioteca para apresentação de informação relativa a mapas em *Web browsers*.

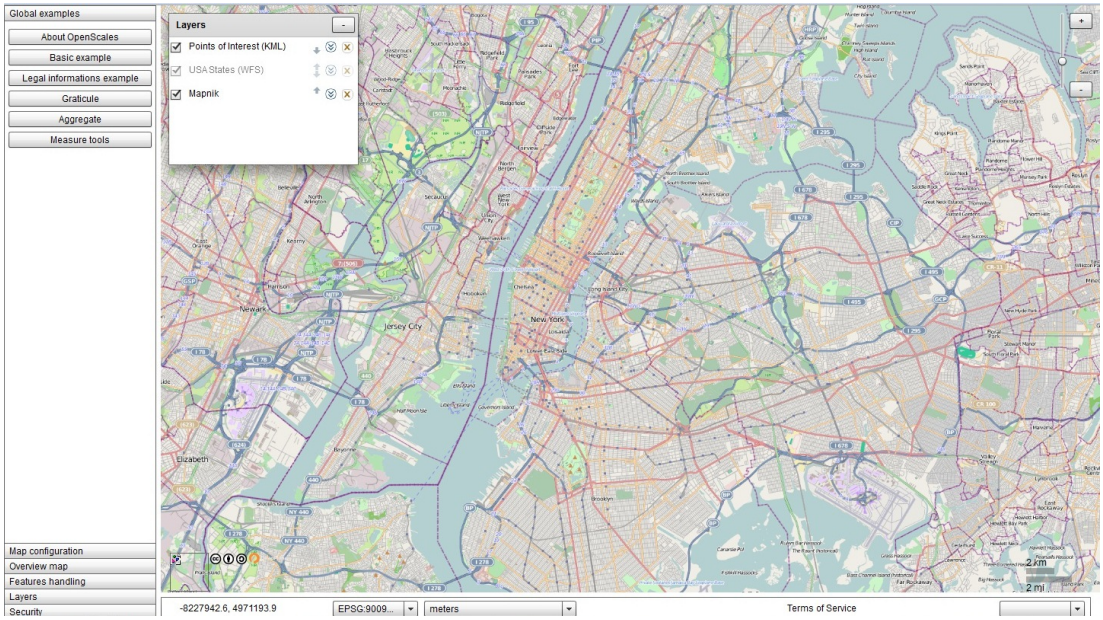

Figura 6.8: Visualização Mapa com Dados Geográficos no OpenScales

Esta *framework* baseia-se no *Flex* pelo que foi necessário instalar o Adobe Flash Builder para Linux através do atualizador de extensões do Eclipse. Também houve a necessidade de instalar o Flex SDK de forma a ter o compilador *mxmlc* para compilar projetos *flex*.

A Figura 6.8 mostra o exemplo de um mapa com dados geográficos no OpenScales.

Capítulo 7

Avaliação

Neste capítulo são apresentados os principais testes da componente de aviso do sistema e cenários associados.

7.1 Testes

Segue-se uma descrição dos testes feitos ao código da componente de aviso, nomeadamente uma breve explicação dos testes unitários, de integração e de sistema realizados.

7.1.1 Unitários

Foram feitos à medida que o código ia sendo desenvolvido, com vista a detetar erros de programação, sendo aplicados a blocos de código. Dada a sua informalidade não foram descritos.

7.1.2 Integração

Como não houve lugar à integração de novos módulos criados de raiz, não houve lugar a testes de integração. A integração das funcionalidades programadas foi realizada de modo direto no módulo de *messaging* já existente no sistema *Sahana*. Como tal, a verificação de que a conformidade destas em relação às restantes funcionalidades do sistema era cumprida foi feita de forma informal nos testes unitários.

7.1.3 Sistema

Não houve necessidade de realizar testes de sistema intensivos como por exemplo testes de *stress* ou testes de desempenho, devido à simplicidade das funcionalidades implementadas e dos dados de *input* e de *output*. Como tal, a *performance* na gestão da informação e na realização das operações associadas não é um ponto crítico. Fizeram-se apenas verificações da não existência de atrasos significativos (5 segundos) nas leituras e escritas em BD, na navegação pela interface e na interação com os controlos.

7.2 Cenário-Teste Simples

A Figura 7.1 apresenta o caso de teste em que se utiliza um telemóvel para simulação de um SMS Center.

No sistema *Sahana*, o Kannel (WAP Gateway *open-source* desenvolvido em C, utilizado neste trabalho) obtém por http a mensagem a enviar e o número de telemóvel (ou endereço de *email* visto suportar conversão de media a partir de *email* e outros formatos). O Kannel funciona, portanto, como SMS Gateway (membro de rede de telecomunicações) para redes GSM (redes celulares que utilizam uma frequência de transmissão especifica) facilitando o envio e receção de SMS's.

A conexão ao SMS Center é feita através dos protocolos de SMS Center existentes tais como o EMI/UCP ou o SMPP que se baseiam na troca de pacotes onde vão codificadas a mensagem e o endereço de destino dessa mensagem.

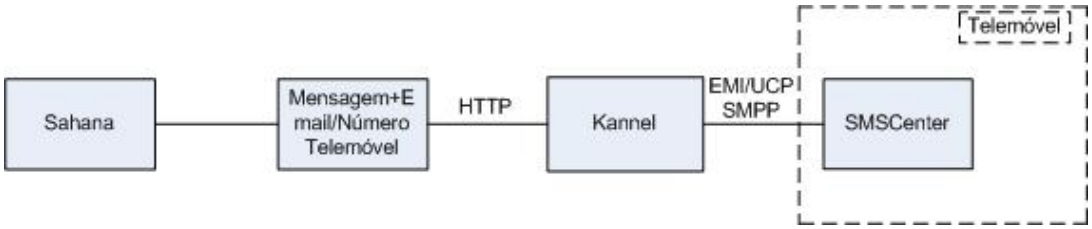

Figura 7.1: Simulação Teste de SMSCenter Utilizando Telemóvel

Os testes realizados neste cenário incluíam dois modos. Um primeiro modo em que se realizava *loopback*, fazendo um *ping* ao *localhost* e um segundo modo que utilizava um telemóvel para simular um SMS Center, cenário explicado acima.

7.3 Cenários-Teste com Operadora

A Figura 7.2 apresenta o caso de teste em que o SMS Center é a própria operadora.

O Kannel recebe por *http* do sistema *Sahana* a mensagem a enviar e o ficheiro *kml* ou a *shapefile* que definem a área alvo do aviso. Utilizando os protocolos de SMS Center EMI/UCP ou SMPP, a operadora é informada e pode assim verificar e obter os números que estão nas estações-base que cobrem aquela área e enviar a mensagem para todos os números que estejam nesse espaço.

Contámos com o apoio de algumas operadoras de telecomunicações no mercado para que fosse possível pôr em prática este cenário.

Foram definidos três cenários de teste, em locais relevantes no contexto deste projeto, com o operador de mensagens: um cenário numa zona de risco de baixa densidade populacional (Barragem Pedrogão), outro cenário numa zona de risco de média densidade populacional (Aveiro) e outro cenário numa zona de risco de grande densidade populacional (Lisboa). Segue-se uma descrição sucinta de cada um deles.

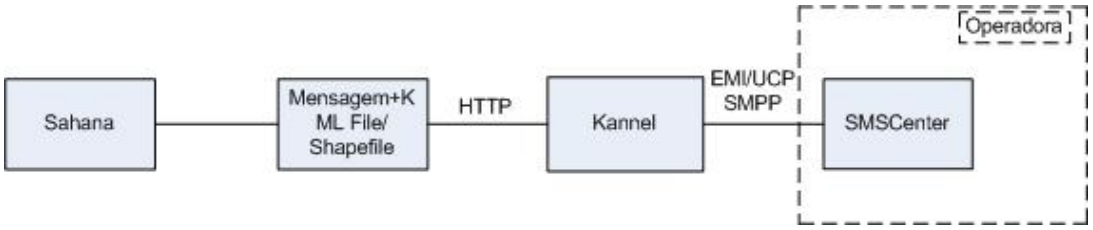

Figura 7.2: Simulação Teste de SMSCenter Com Operadora

7.3.1 Barragem Pedrogão

Este cenário pretende simular uma quebra de barragem e respetiva inundação que pode atingir os dois concelhos onde a barragem se localiza e o estudo dos acontecimentos que podem ocorrer no momento da abertura de comporta para libertação de diversas quantidades de água.

7.3.2 Aveiro

Este cenário pretende simular uma inundação na Ria de Aveiro.

7.3.3 Lisboa

Este cenário pretende simular um *tsunami* nos distritos de Lisboa e Setúbal. Este cenário baseia-se na informação do terramoto de 1755 segundo a qual, o tempo de chegada da onda desde o epicentro até Sagres foi de cinco minutos, até Lisboa foi de vinte e cinco minutos e até ao Porto uma hora. Estes tempos representam o tempo disponível para evacuação nas respectivas zonas de risco.

Capítulo 8

Conclusões

O volume de dados existente constituiu um candidato ideal para a utilização de técnicas inteligentes de informação como *data mining*. Os padrões resultantes da aplicação do método de *clustering* sobre esse volume de dados e os pesos de redes neuronais aplicados aos vários indicadores de condições ambientais propícias a acidentes resultantes de um processo de aprendizagem artificial decorrente de acidentes de poluição anteriores permitiu examinar: a probabilidade de ocorrência de padrões de circulação, outros padrões de circulação relevantes que pudessem fomentar a obtenção de condições de acidente específicas para a costa, o porto e a Ria de Aveiro e fornecer uma base sólida para a análise de risco da Ria de Aveiro. Os desenvolvimentos efetuados permitirão contribuir para a validação de um conjunto de indicadores ambientais de possibilidade de acidente e para a deteção de padrões propícios à ocorrência de acidentes.

O sistema de aviso baseado em tecnologias de comunicação enviará automaticamente os avisos quando as condições propícias a acidente são cumpridas e é alimentado pelos componentes da fase anterior resultantes do trabalho de colegas: o sistema de alerta precoce, os índices de vulnerabilidade e a análise de risco da zona costeira de Aveiro. Os testes realizados ao protótipo deste sistema de aviso foram levados a cabo com base em informação real.

O sistema de gestão de risco criado integra as componentes de alerta precoce e aviso resultantes das tarefas anteriores utilizando tecnologias muito difundidas e *open-source*. Tem uma natureza dinâmica e arquitetura modular à semelhança do SAGE-B, integrando uma componente SIG, para melhor transmissão das zonas em risco.

Os resultados obtidos foram satisfatórios a vários níveis:

- A nível da procura de soluções e meios capazes de lidar com a gestão e armazenamento em bases de dados dos diferentes tipos de informação geográfica disponíveis no departamento;
- Procurou-se criar ferramentas que pudessem ser tidas em conta no futuro como sendo uma mais-valia para trabalhar com quantidades de dados várias vezes superiores às que foram utilizadas durante estes meses de projeto;

- Os programas criados podem continuar a ser utilizados com confiança constituindo uma base sólida de desenvolvimento nesta área e um ponto de partida para futuros desenvolvimentos;
- A nível do desenvolvimento para sistemas de gestão de desastres;
- A nível da aprendizagem adquirida na área dos sistemas de informação geográfica (SIG).

8.1 Trabalho Futuro

No futuro proceder-se-á a uma aposta maior na classificação dos dados das ondas relativos ao modelo WaveWatch3 e SELFE.

Procurar-se-á trabalhar mais sobre a escalabilidade da base de dados em PostgreSQL para quantidades de dados em larga escala dos modelos SELFE e WaveWatch3. No primeiro caso, temos menos de duzentos dias inseridos, o que tendo em conta a informação existente relativa a alguns anos a esta parte bem como a informação que for sendo obtida daqui em diante, permite antever quantidades de dados na ordem do 1 Terabyte ou mais. Para o segundo caso, como foram inseridos dias relativos a três meses do final de 2011 (cerca de 200 MB de informação em base de dados), será interessante inserir os dados relativos aos restantes meses dos vários anos, situação um pouco semelhante com o caso do SELFE em termos da quantidade de informação disponível e da dinâmica na sua obtenção (diária) através do sistema de previsão em tempo real RDFS-PT.

A nível do sistema *Sahana*, serão estudados e desenvolvidos novos módulos e novas funcionalidades em módulos já existentes que se pensem ser uma mais-valia para os objetivos centrais do sistema. Procurar-se-á também otimizar a integração dos mapas por exemplo através da edição do código Javascript é possível alterar o tamanho dos mapas no cliente OpenScales. Também é possível ativar e desativar *layers*, mudar a informação das *layers*, adicionar mais *layers*, etc. Um dos próximos assuntos a estudar será o envio de coordenadas Javascript para o *Flex*.

Bibliografia

[1] Penela-Arenaz, M., Bellas, J. and Vázquez, E. (2009). "Chapter Five: Effects of the Prestige Oil Spill on the Biota of NW Spain 5 Years of Learning," in *Advances in Marine Biology*, 56:365-396;

[2] Azevedo, A., Oliveira, A., Fortunato, A., Bertin, X. (2009). "Application of an Eulerian-Lagrangian oil spill modeling system to the Prestige accident: trajectory analysis," in *Proceedings of the 10^{th} International Coastal Symposium. Journal of Coastal Research, SI 56*, pp.777-781;

[3] Sanchez, S. (2008). "Risk assessment in ports," *Lectures of the Risk Management in Civil Engineering Course*;

[4] Rogers, G., Sorensen, J. (1988). "Diffusion of Emergency Warnings," in *The Environmental Professional*, 10:281-294;

[5] Careem, M., De Silva, C., De Silva, R., Raschid, L., Weerawarana, S. (2006). "Sahana: Overview of a Disaster Management System," in *Proceedings of the International Conference on Information and Automation*, pp.361-366;

[6] Costa, M., Oliveira, A., Rodrigues, M., Azevedo, A. (2009). "Application of Parallel, High-Performance Computing in Coastal Environmental Modeling: Circulation and Ecological Dynamics in the Portuguese Coast," in *Proceedings of the 3^{rd} Iberian Grid Infrastruture Conference*, Hernandez Garcia et al (eds), pp.375-386;

[7] Fernandes, J. P. (2008). "Emergency Warnings with Short Message Service," in *Coskun HG; Cigizoglu HK; Maktav MD (Eds.), Integration of Information for Environmental Security*. Dordrecht, Netherlands: Springer, 6p;

[8] Jagtman, H. M. (2010). "Cell broadcast trials in The Netherlands: Using mobile phone technology for citizens' alarming," in *Reliability Engineering and System Safety*, Vol.95, I.1, pp.18-28;

[9] Herold, S., Sawada, M., Wellar, B. "Integrating Geographic Information Systems, Spatial Databases and the Internet: A Framework for Disaster Management," Laboratory for Applied Geomatics and GIS Science (LAGGISS), Department of Geography, University of Ottawa, 2005;

[10] Jesus, G., Santos, M. A., Fernandes, J. P. (2010). ”SAGE-B: A Dynamic Dam Break Flood Emergency Management System,” in *Disaster Management 2009*, New Forest, United Kingdom, in review;

[11] Fahland, D., Gläßer, T. M., Quilitz, B., Weißleder, S., Ulf, L. ”HUODINI - Flexible Information Integration for Disaster Management,” in *Proceedings ISCRAM2007*, pp.255-262, 2007;

[12] Wood, H. M. ”The use of earth observing satellites for hazard support,” in *Geoscience and Remote Sensing Symposium, 2001. IGARSS 01. IEEE 2001 International*, July 2001, pp.135-137;

[13] Rodrigues, A. S., Santos, M. A., Santos, A. D., Rocha, F. (2002). ”Dam-Break Flood Emergency Management System,” in *Water Resources Management*, vol.16, no.2, 2002, pp.489-503;

[14] Hussain, M., Arsalan, M., Siddiqi, K., Naseem, B., Rabab, U. ”Emerging geoinformation technologies (git) for natural disaster management in Pakistan: an overview,” in *Proceedings of* 2^{nd} *International Conference on Recent Advances in Space Technologies, 2005. RAST 2005*, June 2005, pp.487-493;

[15] Wegmuller, U., Wiesmann, A., Strozzi, T. and Werner, C. ”Envisat asar in disaster management and humanitarian relief,” in *Geoscience and Remote Sensing Symposium, 2002. IGARSS 02. 2002 IEEE International*, June 2002, pp.2282-2284;

[16] Olariu, S., Maly, K., Foudriat, E. and Yamany, S. ”Wireless support for telemedicine in disaster management,” in *Proceedings of Tenth International Conference on Parallel and Distributed Systems, 2004. ICPADS 2004*, July 2004, pp.649-656;

[17] Pradhan, A. R., Laefer, D. F. and Rasdorf, W. J. (2007). ”Infrastruture Management Information System Framework Requirements for Disasters,” in *Journal of Computing in Civil Engineering*, March-April, pp.90-101;

[18] Jain, Anil K., Duin, Robert P. W., Mao, Jianchang. ”Statistical Pattern Recognition: A Review,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22, No.1, January 2000;

[19] Nagabhushana, S. *Data Warehousing OLAP and Data Mining*. New Age International, 2006, Volume II;

[20] Cios, K. J., Pedrycz, W., Swiniarski, R. W., Kurgan, L. A. *Data Mining, A Knowledge Discovery Approach*. XV, 606 p, Hardcover, Chapter 2, Springer, 2007;

[21] Olafsson, Siggi. ”Operations Research and Data Mining,” in *Proceedings* 20^{th} *European Conference on Operational Research Rhodes*, Greece, July 2004;

[22] Kröse, Ben, van der Smagt, Patrick. *An Introduction to Neural Networks*. Eighth edition, November 1996;

[23] Jimenez, Tino. "Using Neural Networks in Data Mining," 19 February 2009;

[24] Azevedo, Alberto. (2010). "Sistema integrado de modelação para apoio à prevenção e mitigação de acidentes de hidrocarbonetos em estuários e orla costeira," tese de doutoramento, Departamento de Engenharia Geográfica, Geofísica e Energia, Faculdade de Ciências da Universidade de Lisboa, 224 páginas;

[25] Liu, X., Liu, Y., Li, L., Ren, Y. (2009). "Disaster Monitoring and Early-Warning System for Snow Avalanche Along Tianshan Highway," in *Proceedings of Geoscience and Remote Sensing Symposium, 2009. IGARSS 2009. 2009 IEEE International*;

[26] Santos, M. A., Jesus, G., Fernandes, J. P. "Instrumentos de apoio à gestão do risco de cheias," in *Proceedings of International Meeting On Emergency Management*, 2008;

[27] Harvey, Susan. "Real Time Traffic Information Systems," March 1998;

[28] Higginson, Martin. "Public Transport Passenger Information," 1999;

[29] Reed, M., French, D., Rines, H., Rye, H. (1995). "A three-dimensional oil and chemical spill model for environmental impact assessment," in *Proceedings of the 1995 International Oil Spill Conference*, pp.61-66;

[30] Aamo, O. M., Reed, M., Lewis, A. (1997). "Regional contingency planning using the OSCAR oil spill contingency and response model," in *Proceedings of the 20th Arctic and Marine Oil Spill Program. AMOP Technical Seminar*, Vancouver, Canada, 1997;

[31] Jesus, G., Santos, M. A., Fernandes, J. P., Oliveira, A. "Development of a Dambreak Flood Emergency Information System," in *Proceedings of the 7th International ISCRAM Conference*, Seattle, USA, May 2010;

[32] Lollino, G., Arattano, M., Cuccureddu, M. (2002). "The use of the automatic inclinometric system for landslide early warning: the case of Cabella Ligure (north-Western Italy)," in *Physics and Chemistry of the Earth*, vol.27:1545-1550;

[33] Han, J., Kamber, M., Pei, J. (2001). *Data Mining: Concepts and Techniques*. Academic Press, 533 pages;

[34] Weihong, Z., Yongsheng, Z., Jun, D., Mei, H. "An Early Warning System for Groundwater Pollution Based on GIS," in *Proceedings 2011 International Symposium on Water Resource and Environmental Protection (ISWREP)*, College of Environment & Resources, Jilin University, China, 20-22 May 2011;

[35] Samet, R., Tural, S. "Web Based Real-Time Meteorological Data Analysis and Mapping Information System," in *International Journal of Education and Information Technologies*, Issue 4, Volume 4, 2010;

[36] Tu, J. C. (1996). "Cockpit Weather Information (CWIN) System," in *Proceedings of the IEEE 1996 National Aerospace and Electronics Conference, 1996. NAECON, 1996*, vol.1:29-32;

[37] Zhu, R., Liu, Y., Jiang, H., Yin, Z. (2011). "Visualization of Weather-Induced Disaster Warning Information System using Google Earth API based on Mashup," in *Proceedings of 2011 International Conference on Multimedia Technology (ICMT)*, pp.3789-3793;

[38] Li, X. (2011). "An Intelligent System for Earthquake Early Warning," in *Proceedings of 2011 3rd International Workshop on Intelligent Systems and Applications (ISA)*, pp.1-4;

[39] Graber, Hans C. et al. "Real-Time Forecasting System of Winds, Waves and Surge in Tropical Cyclones," Rosenstiel School of Marine & Atmospheric Science (RSMAS), Division of Applied Marine Physics (AMP), University of Miami, 2006;

[40] Baron, Sr. Robert O. et al. "System and methods for distributing real-time site specific weather information," Jan 2000;

[41] Weerts, A. H., Schellekens, J., Weiland, F. S. "Real-Time Geospatial Data Handling and Forecasting: Examples from Delft-FEWS Forecasting Platform/System," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2010;

[42] Grasso, V. F. (2001). "Early-Warning Systems: State-of-Art Analysis and Future Directions," Draft Report, United Nations Environment Programme (UNEP), vol.1, i.2, pp.1-66;

[43] *CleanSeaNet*. Setembro, 2011 <http://cleanseanet.emsa.europa.eu/>;

[44] *MOHID*. Setembro, 2011 <http://www.mohid.com/>;

[45] *INSEA*. Setembro, 2011 <http://www.insea.info/>;

[46] *RDFS-PT*. Setembro, 2011 <http://ariel.lnec.pt/>;

[47] *Virtual Columbia River*. Setembro, 2011 <http://www.stccmop.org/datamart/virtualcolumbiariver/>;

[48] *GNOME (General NOAA Operational Modeling Environment)*. Maio, 2012;

[49] *ASA (Applied Science Associates), OILMAP (Oil Spill Model and Response System)*. Outubro, 2011 <http://www.asascience.com/software/oilmap/index.shtml>;

[50] *ASA (Applied Science Associates), SIMAP (Integrated Oil Spill Impact Model System)*. Outubro, 2011 <http://www.asascience.com/software/simap/index.shtml>;

[51] *Sahana FOSS*. Outubro, 2011 <http://sahanafoundation.org/>;

[52] Contributors of wiki.sahanafoundation.org. "agasti:vesuvius:start". *Sahana WIKI*. Outubro, 2011 <http://wiki.sahanafoundation.org/doku.php/agasti:vesuvius:start>;

[53] *PAC:MAN*. Novembro, 2011 <http://www.lnec.pt/organizacao/dha/nti/estudos_id/pacman>;

[54] *ROSETTA*. Novembro, 2011 <http://www.lcb.uu.se/tools/rosetta/index.php>;

[55] *Weka*. Novembro, 2011 <http://www.cs.waikato.ac.nz/ml/weka/index.html>;

[56] Bogorny, V., Palma, A. T., Engel, P. M., Alvares, L. O. "Weka-GDPM - Integrating Classical Data Mining Toolkit to Geographic Information Systems," in *SBBD Workshop on Data Mining Algorithms and Applications (WAAMD2006)*, Florianopolis, Brazil, October 16-20 2006;

[57] *RapidMiner*. Novembro, 2011 <http://rapid-i.com/content/blogcategory/38/69/lang,en/>;

[58] *Rattle*. Dezembro, 2011 <http://rattle.togaware.com/>;

[59] *R*. Novembro, 2011 <http://www.r-project.org/>;

[60] *Orange*. Dezembro, 2011 <http://orange.biolab.si/>;

[61] Øhrn, A. (1999). "Discernibility and Rough Sets in Medicine: Tools and Applications," PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway. NTNU report 1999:133, IDI report 1999:14, ISBN 82-7984-014-1. 239 pages;

[62] *IBM SPSS Modeler*. Dezembro, 2011 <http://www-01.ibm.com/software/analytics/spss/products/modeler/>;

[63] *STATISTICA Data Miner*. Dezembro, 2011 <http://www.statsoft.com/#>;

[64] *SAS*. Dezembro, 2011 <http://www.sas.com/>;

[65] Wu, X. et al. "Top 10 algorithms in data mining," in *Knowledge and Information Systems*, Vol.14, No.1 (2008), pp.1-37, 24 December 2007;

[66] Witten, I. H., Frank, E., Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. Third Edition, Morgan Kaufmann, January 2011;

[67] Talia, D., Trunfio, P., Verta, O. "The Weka4WS framework for distributed data mining in service-oriented Grids," in *Concurrency and Computation: Practice and Experience*, vol.20, n.16, pp.1933-1951, Wiley InterScience, November 2008;

[68] Celis, S., Musicant, D. R. (2002). "Weka-Parallel: Machine Learning in Parallel," Technical Report, Department of Mathematics and Computer Science, Carleton College;

[69] Bogorny, V., Tietböhl, A. (2006). "Extending the Weka Data Mining Toolkit to support Geographic Data Preprocessing," Technical Report - RP 354, 26p, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, July 2006;

[70] Hornik, K., Buchta, C., Zeileis, A. (2009). "Open-Source Machine Learning: R Meets Weka," in *Computational Statistics*, vol.24, numb.2, pp.225-232;

[71] Hall, M. et al. (2009). "The WEKA, Data Mining Software: An Update," in *ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations Newsletter*, Volume 11, Issue 1, pp.10-18, November 2009;

[72] Wang, Y. et al. (2005). "Gene selection from microarray data for cancer classification - a machine learning approach," in *Computational Biology and Chemistry*, Volume 29, Issue 1, pp.37-46, Feb 2005;

[73] Pillai, S., Good, B., Richman, D., Corbeil, J. (2003). "A new perspective on V3 phenotype prediction," in *AIDS Research and Human Retroviruses*, Vol.19, I.2, pp.145-149, Feb 2003;

[74] Gadi, M. F. A., Wang, X., do Lago, A. P. "Credit Card Fraud Detection with Artificial Immune System," in *Proceedings of 7th International Conference on Artificial Immune Systems, ICARIS 2008, Phuket, Thailand August 10-13, 2008. Lecture Notes in Computer Science*, Berlin: Springer, Vol.5132, pp.119-131, 2008;

[75] Wieczorkowska, A., Kolczynska, E. (2008). "Identification of dominating instrument in mixes of sounds of the same pitch," in *Proceedings of 17th International Conference on Foundations of Intelligent Systems (ISMIS'08). Lecture Notes in Artificial Intelligence*, Berlin: Springer, Vol.4994, pp.455-464;

[76] Sivakumar, P. B., Mohandas, V. P. (2010). "Performance Comparison of Attribute Set Reduction Algorithms in Stock Price Prediction - A Case Study on Indian Stock Data," in *Proceedings of First International Conference on Swarm, Evolutionary, and Memetic Computing, SEMCCO 2010, Chennai, India, December 16-18, 2010. Lecture Notes in Computer Science*, Springer, Vol.6466, pp.567-574;

[77] Han, J., Rodriguez, J. C., Beheshti, M. (2009). "Discovering Decision Tree Based Diabetes Prediction Model," in *Proceedings of 2008 International Conference on Advanced Software Engineering and its Applications, ASEA 2008. Communications in Computer and Information Science*, Springer, Vol.30, pp.99-109;

[78] *KNIME*. Dezembro, 2011 <http://www.knime.org/>;

[79] Khan, A., Revett, K. "Data mining the PIMA dataset using rough set theory with a special emphasis on rule reduction," in *Proceedings of 8th International Multitopic Conference, IEEE INMIC 2004*. pp.334-339, December 2004;

[80] Gürbüz, F., Özbakir, L., Hüseyin, Y. (2011). "Data mining and preprocessing application on component reports of an airline company in Turkey," in *Expert Systems with Applications*, Vol.38, I.6, pp.6618-6626, June 2011;

[81] Muenchen, E. R. "The Popularity of Data Analysis Software." Dezembro, 2011 <http://sites.google.com/site/r4statistics/popularity>;

[82] *WAVEWATCH III Model*. Maio, 2012;

[83] *SELFE*. Maio, 2012;

[84] Ambler, Scott W., Lines, M. *Disciplined Agile Delivery (DAD): A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press, ISBN: 0132810131;

[85] *World Meteorological Organization*. Junho, 2012.

Apêndice A

Outputs

A.1 Levantamento de Requisitos

A.1.1 Requisitos Funcionais

Spatial Messaging:

a) O sistema deve ser capaz de enviar uma mensagem para uma área de risco à escolha (escolhida através de uma *dropdownlist*);

b) A operadora recebe a área e a mensagem a enviar, identifica todos os contatos na área em questão e efetua o aviso correspondente;

c) O sistema deve mostrar o tempo disponível para aviso das populações e o nível de risco com uma cor identificativa de acordo com as cores dos níveis de alerta meteorológicos.

Add Risk Area:

a) O sistema deve permitir a inserção de uma nova área de risco;

b) Para tal, deve permitir a indicação do nome da nova área de risco, do caminho para o ficheiro de dados geográficos referente ao polígono que identifica a área de risco, da descrição da área de risco, do valor ou intervalo de tempo em minutos disponível para avisar as populações e do nível de risco.

View/Edit Risk Area:

a) O sistema deve permitir a visualização de todas as áreas de risco;

b) O sistema deve permitir a edição de cada uma delas;

c) Para tal, deve permitir a indicação de um novo nome para a área de risco, de um novo caminho para o ficheiro (*kml*, *shapefile*) de dados geográfico, de uma nova descrição da área de risco, de um novo valor ou intervalo de tempo em minutos disponível para avisar as populações e de um novo nível de risco.

A.1.2 Requisitos Não Funcionais

a) Usabilidade - A interface com o utilizador deve ser simples, intuitiva, autoexplicativa e de fácil operação;

b) Disponibilidade - O sistema deve estar disponível para os utilizadores 24 horas por dia, 7 dias por semana;

c) Confiabilidade - A informação apresentada e armazenada deve ser consistente;

d) Segurança - O sistema deve ter *login* e validação dos dados do utilizador;

e) Tempo de Resposta - A página deve demorar no máximo 5 segundos a carregar o HTML na página desde que é feito o pedido ao sistema. O tempo que o tráfego demora pela rede não conta;

f) Escalabilidade - O sistema deve ser escalável;

g) Portabilidade - O sistema deve ser independente da plataforma.

A.1.3 Caso de Uso

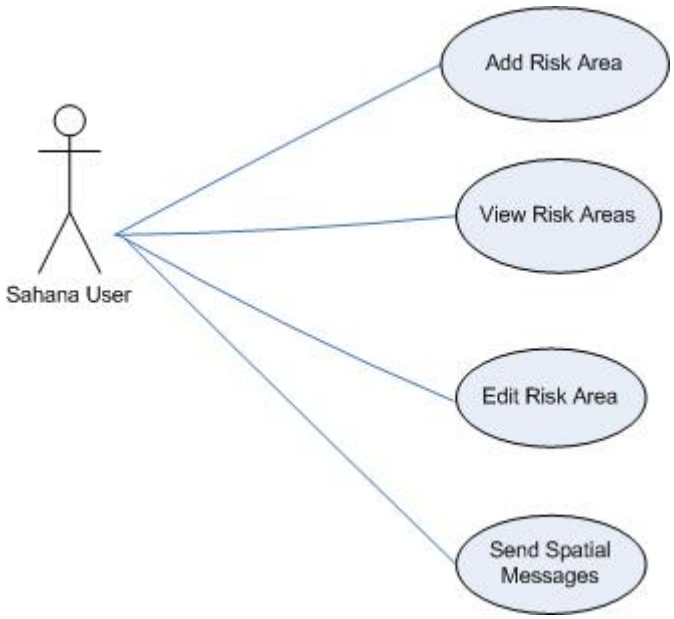

Figura A.1: Caso de Uso Novas Funcionalidades Componente de Aviso

A.1.4 Protótipos

Spatial Messaging:

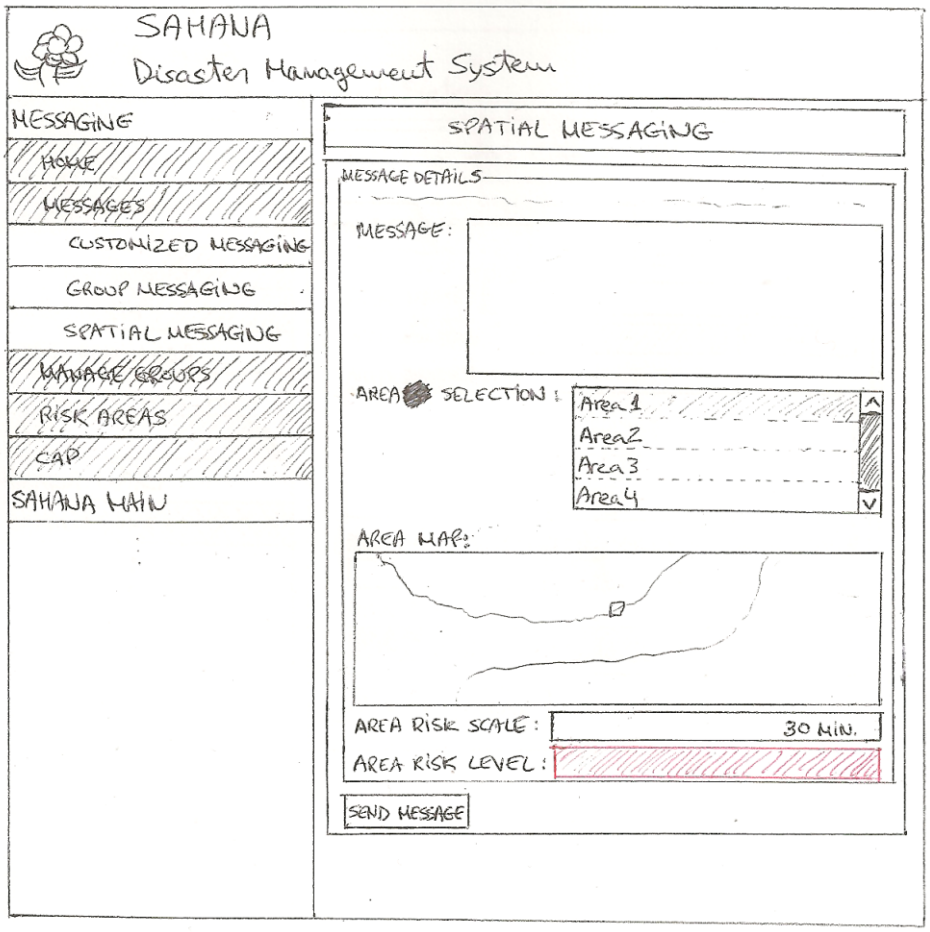

Figura A.2: Protótipo Spatial Messaging

Add Risk Area:

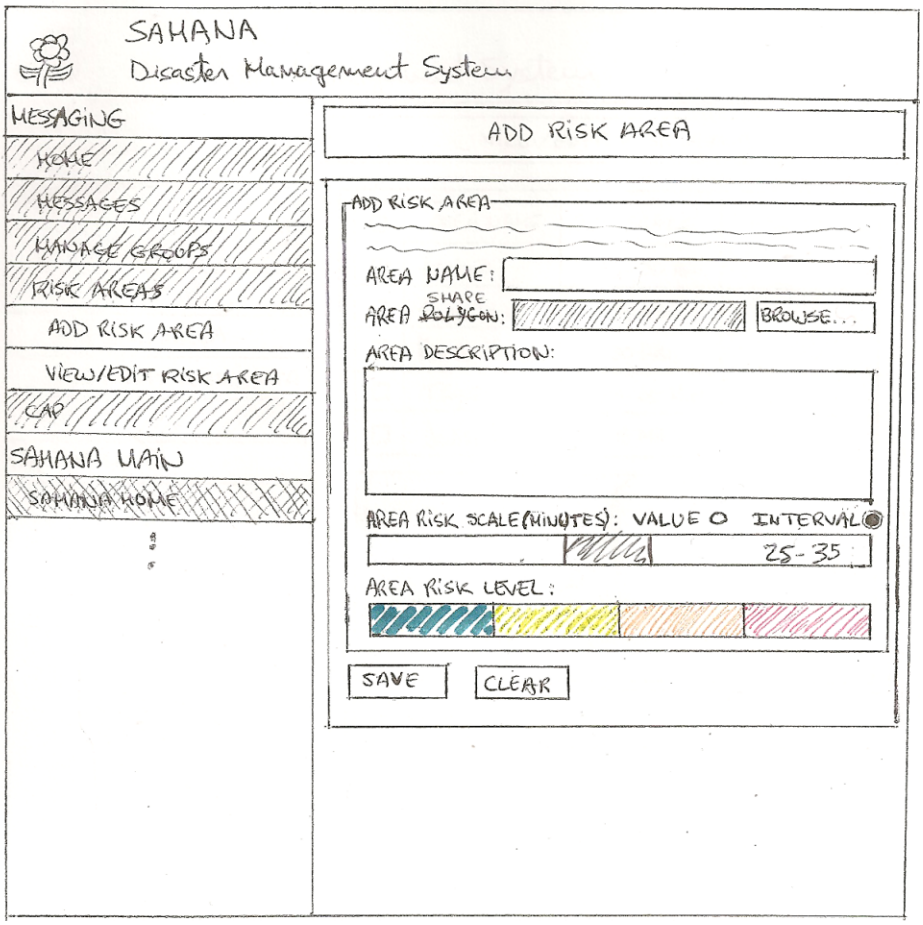

Figura A.3: Protótipo Add Risk Area

View/Edit Risk Area:

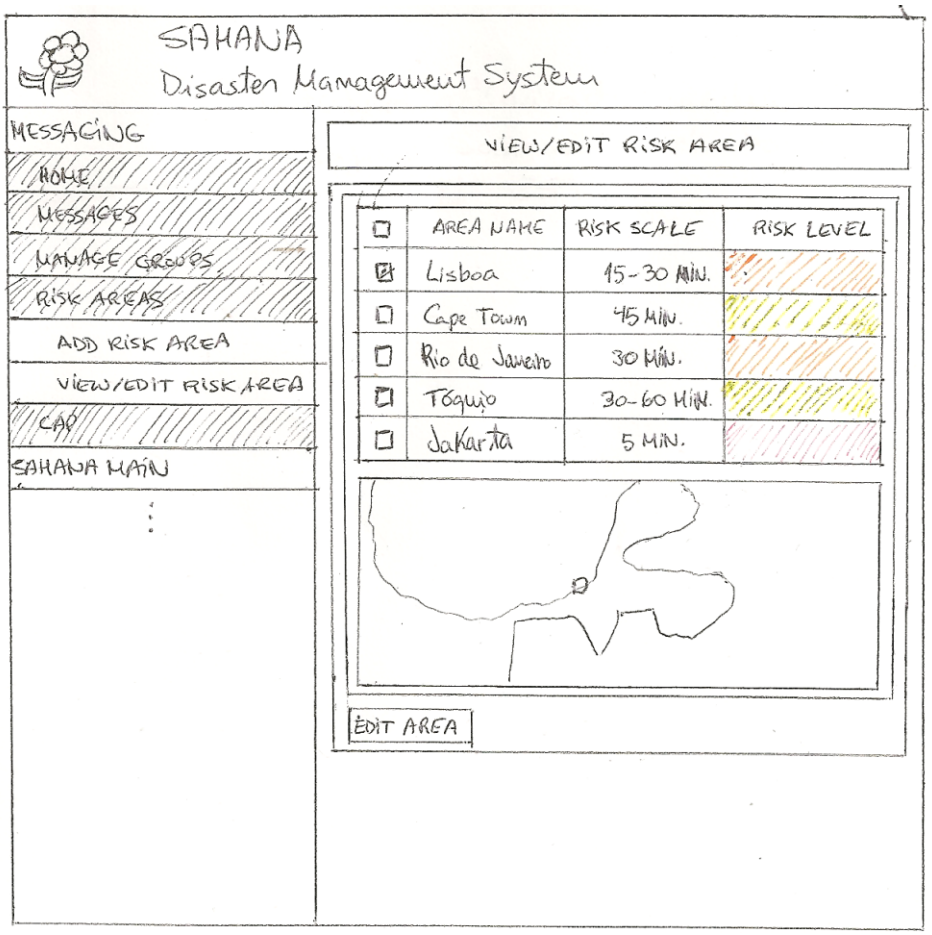

Figura A.4: Protótipo View/Edit Risk Area

A.2 Desenho Detalhado

A.2.1 Modelos de Dados

SELFE:

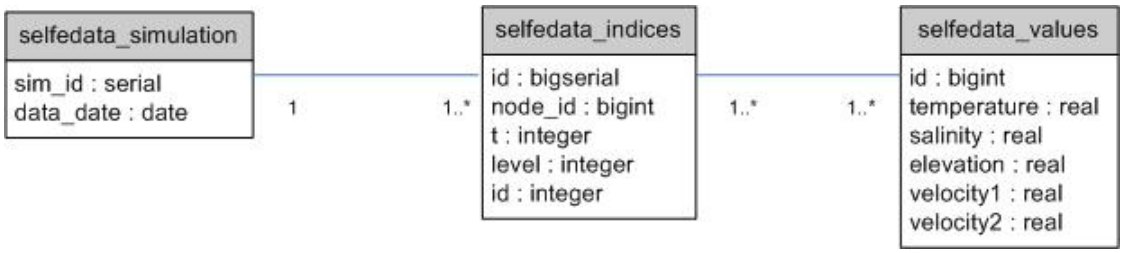

Figura A.5: UML BD SELFE

BD Acidentes:

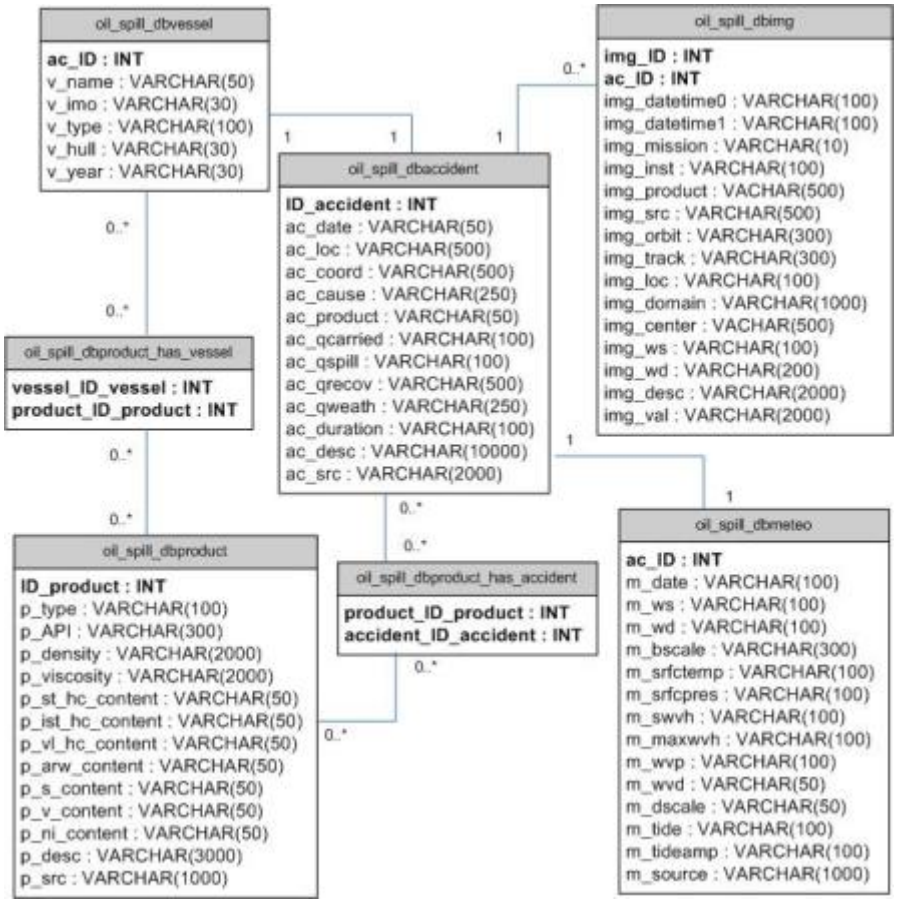

Figura A.6: UML BD Acidentes

BD Sahana:

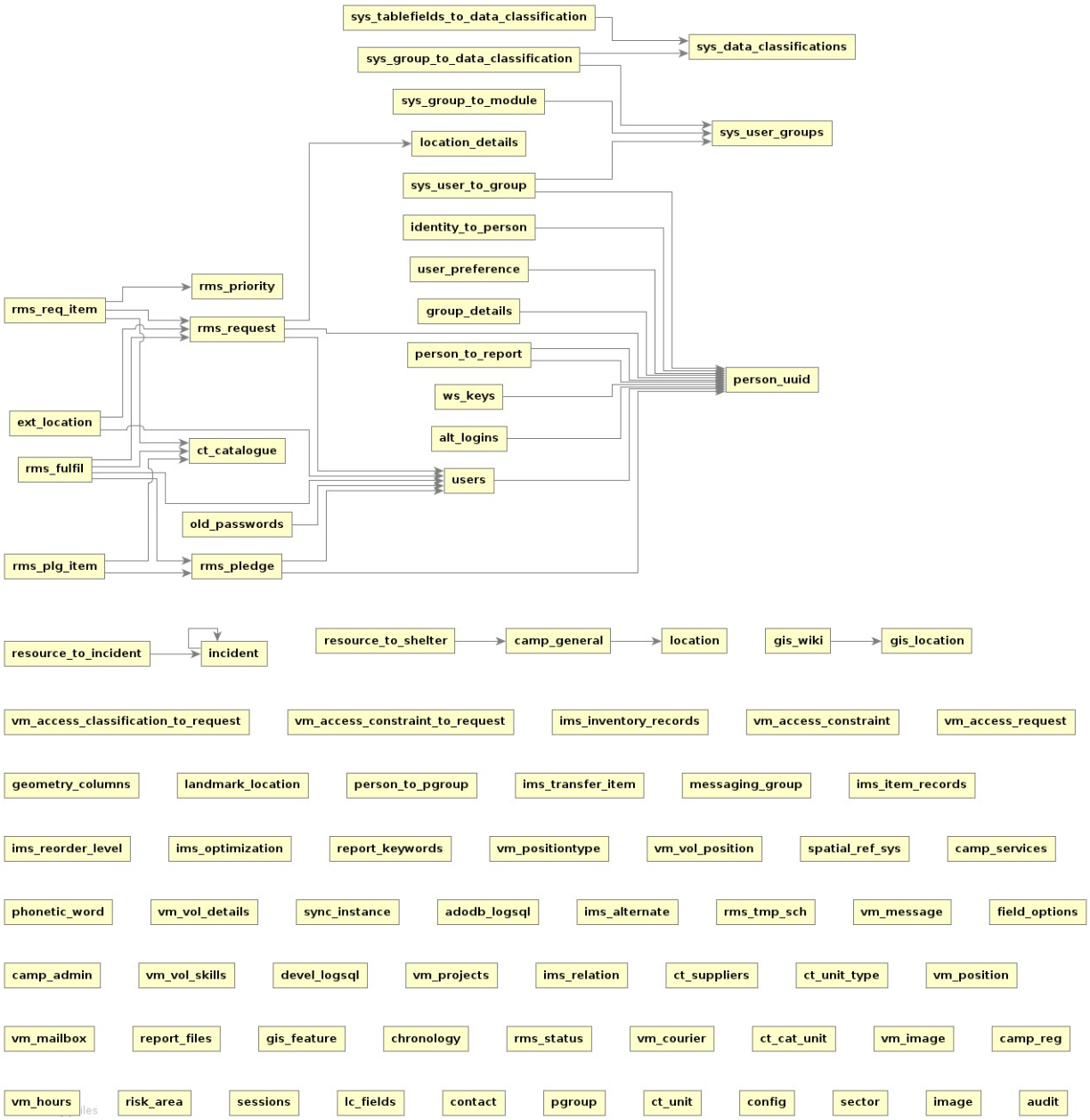

Figura A.7: UML BD Sahana

A.3 Mapa de Gantt

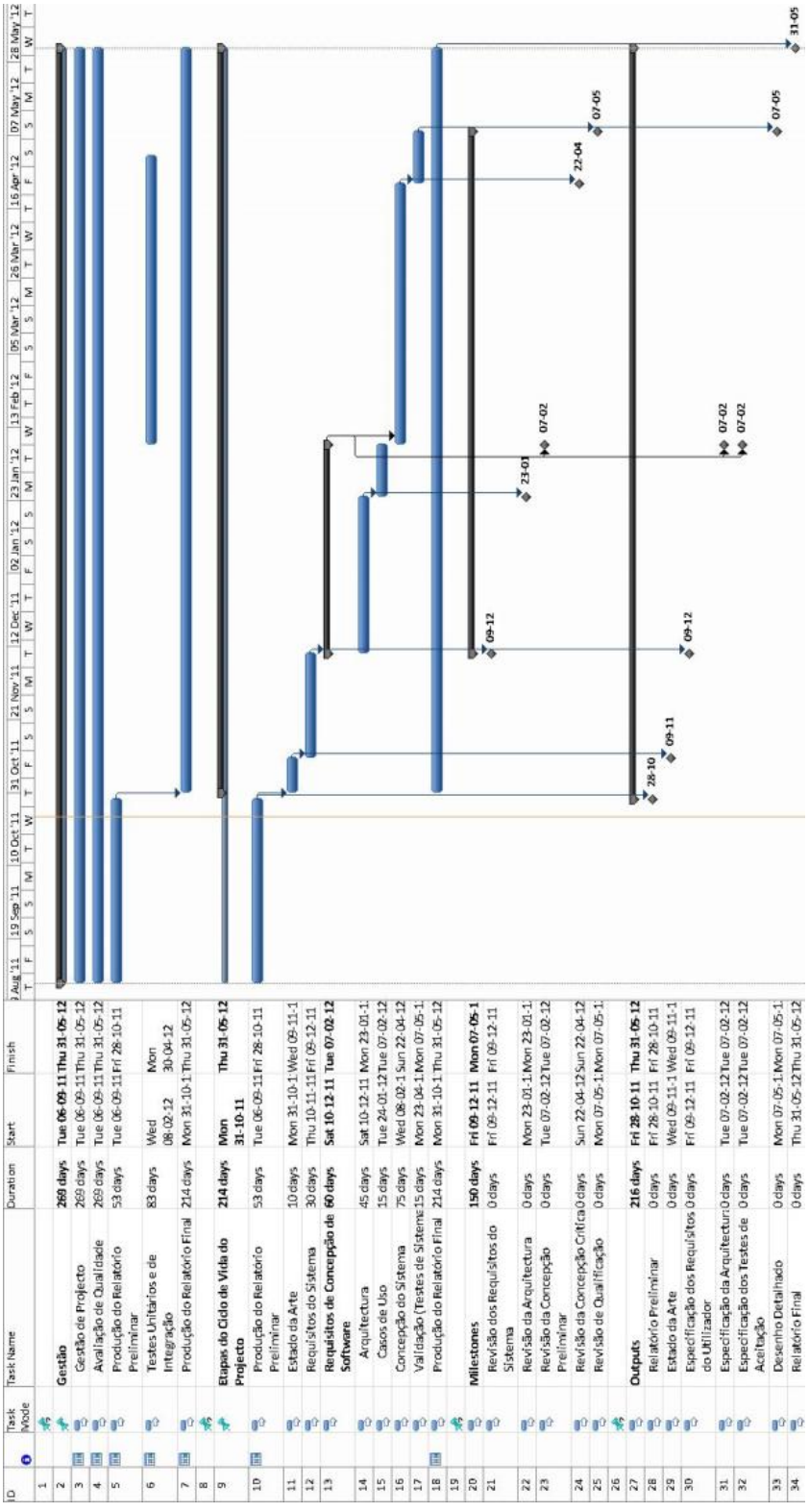

Figura A.8: Mapa de Gantt com o Planeamento do Trabalho

Apêndice B

Excertos de Código

Junto com este relatório foi entregue um DVD com pastas contendo o código fonte do sistema *Sahana*, dos programas criados para tratamento e inserção da informação dos modelos WaveWatch3 e SELFE e da informação relativa a acidentes. Inclui também um ficheiro de texto com o Apêndice B contendo o código dos ficheiros mais relevantes no âmbito deste trabalho.