

**Instituto Federal de Educação, Ciência e
Tecnologia Fluminense**

**Programa de Pós-graduação em Sistemas Aplicados à
Engenharia e Gestão**

**Integração do sistema APRS com a Plataforma de Gestão
Humanitária Sahana Eden: Rastreamento de unidades
móveis e estações de sensores**

Whanderley Souza Freitas
2018

**Instituto Federação de Educação, Ciência e Tecnologia Fluminense
Programa de Pós-graduação em Sistemas Aplicados à Engenharia e Gestão**

**Integração do sistema APRS com a Plataforma de Gestão Humanitária
Sahana Eden: Rastreamento de unidades móveis e estações de sensores**

WHANDERLEY SOUZA FREITAS

Renato Gomes Sobral Barcellos

(Orientador)

Dissertação de mestrado submetido para obtenção do grau de **Mestre** no Programa de Pós-graduação em Sistemas Aplicados à Engenharia e Gestão, Área de Concentração em Sistemas Computacionais.

Whanderley Souza Freitas

2018

Instituto Federação de Educação, Ciência e Tecnologia Fluminense
Programa de Pós-graduação em Sistemas Aplicados à Engenharia e Gestão

WHANDERLEY SOUZA FREITAS

Projeto submetido como requisito para elaboração da dissertação para obtenção do grau de **Mestre** no Programa de Pós-graduação em Sistemas Aplicados à Engenharia e Gestão, Área de Concentração em Sistemas Computacionais.

PROJETO APRESENTADO EM 26/05/2017

Renato Gomes Sobral Barcellos

Dr. em geociências IFFluminense

(Orientador)

Rogério Atem de Carvalho

Dr. em engenharia de produção IFFluminense

(Co-orientador)

Luiz Gustavo Lourenço Moura

Dr. em engenharia de sistemas e computação IFFluminense

Milthon Erthal Junior

Dr. em Produção vegetal IFFluminense

AGRADECIMENTOS

A presente dissertação de mestrado não poderia ser concluída sem o inestimável apoio de várias pessoas, que ao longo do processo ajudaram direta ou indiretamente.

Inicio agradecendo a DEUS, por ter me guiado e colocado pessoas tão especiais a meu lado, sem as quais certamente não teria este trabalho não seria possível!

Não posso deixar de agradecer ao meu orientador, Professor Doutor Renato Gomes Sobral Barcellos, por toda a paciência e empenho com que me orientou neste trabalho.

Gostaria agradecer à minha família e amigos pelo apoio incondicional que me deram, especialmente aos meus pais e avós que me guiam pela vida desde meu nascimento até os dias de hoje.

Desejo igualmente agradecer a todos os meus colegas do Mestrado em Sistemas Aplicados a Engenharia e Gestão, cujo apoio e amizade estiveram presentes em todos os momentos.

Aos amigos de trabalho da DGTI pelo apoio, paciência, atenção e força que prestaram. Para não correr o risco de não lembrar-me de alguém não vou identificar ninguém, aqueles a quem este agradecimento se dirige sabê-lo-ão, desde já os meus agradecimentos.

Não poderia de deixar de agradecer a minha noiva e minha futura sogra, pelo apoio e paciência em meus momentos de ausência, pois sem o o apoio destas este trabalho não seria possível ou no mínimo seria uma jornada muito mais difícil.

A nova fonte de poder não é o
dinheiro nas mãos de poucos, mas
informação nas mãos de muitos.

John Naisbitt

RESUMO

O auxílio às vítimas e gerenciamento de serviços essenciais como buscas, envio de suplementos e atendimento médico, são ações que demandam de coordenação logística, técnica e de pessoal, objetivando a otimização dos custos e a garantia de maior alcance nas escalas espacial e temporal. Faz-se necessário a utilização de ferramentas que auxiliem na coordenação dos recursos tanto humanos quanto materiais, visando a mitigação dos prejuízos e principalmente a preservação de vidas. Esta coordenação dos recursos demanda de um sistema de gestão que integre as diversas informações da catástrofe, como a localização dos recursos em áreas estratégicas e quais tarefas estão sendo realizadas, por quem e em qual local. Visando conferir uma maior efetividade na gestão de desastres este estudo tem como objetivo explicar o processo de integração do sistema APRS (Automatic Packet Reporting System) à uma plataforma de ajuda humanitária, bem como demonstrar sua aplicabilidade e eficiência na realidade brasileira a partir dos dados de uma cidade. Para este fim, foi escolhida a plataforma de ajuda humanitária Sahana Eden, desenvolvida pelo Sri Lanka durante o evento da Tsunami em 2004. A integração da plataforma ao sistema APRS, tem o intuito de automatizar a entrada de dados referentes a geolocalização de unidades móveis e dados oriundo de sensores, agilizando a tomada de decisão a partir da geração de dados mais confiáveis e do controle das ações de resgate durante o evento catastrófico. A integração do APRS ao Sahana Eden mostrou-se um excelente recurso no que diz respeito a mitigação de prejuízos em meio a catástrofes, uma vez que permite uma disponibilidade de dados maior, mais precisa e de forma contínua.

ABSTRACT

Assistance to victims and management of essential services such as searches, submission of supplements and medical care are actions that require logistic, technical and personnel coordination, aiming at optimizing costs and guaranteeing greater reach at the spatial and temporal scales. It is necessary to use tools that aid in the coordination of both human and material resources, aiming to mitigate damages and especially the preservation of lives. This coordination of resources requires a management system that integrates the various information of the catastrophe, such as the location of resources in strategic areas and what tasks are being carried out, by whom and in which location. The objective of this study is to explain the process of integrating the APRS (Automatic Packet Reporting System) system into a humanitarian aid platform, as well as to demonstrate its applicability and efficiency in the Brazilian reality, based on data from a city. To this end, the Sahana Eden humanitarian aid platform, developed by Sri Lanka during the Tsunami event in 2004, was chosen. The integration of the platform into the APRS system is intended to automate the entry of data concerning the geolocation of mobile units and data from sensors, streamlining decision making from the generation of more reliable data and the control of rescue actions during the catastrophic event. The integration of the APRS into Sahana Eden has proven to be an excellent resource for disaster mitigation as it allows for greater, more accurate and continuous data availability.

LISTA DE FIGURAS

Figura 1 – Diagrama expressando a iteração dos termos pesquisados na base <i>scopus</i> . . .	10
Figura 2 – Exemplos de <i>templates</i> feitos sobre a plataforma <i>Sahana Eden</i>	14
Figura 3 – Exemplo de fluxo de informações uma vez a plataforma em produção. mo- dificado de (NURI S.T., 2016)	15
Figura 4 – Exemplo de uma lista de organizações cadastradas no <i>Sahana Eden</i>	16
Figura 5 – Exemplo de uma lista de voluntários cadastrados no <i>Sahana Eden</i>	16
Figura 6 – Exemplo de uma lista de itens.	17
Figura 7 – Exemplo de uma lista de ativos.	18
Figura 8 – Exemplo de mapa exibindo alguns registros.	18
Figura 9 – Infraestrutura do APRS(Alterado de (HANSSEN, 2015)).	19
Figura 10 – Estrutura dos servidores APRS (APRS..., 2016).	20
Figura 11 – Esquema da tabela de veículos do <i>Sahana Eden</i>	22
Figura 12 – Esquema da tabela 'asset_asset' do <i>Sahana Eden</i>	23
Figura 13 – Esquema da tabela 'sit_trackable'	24
Figura 14 – Esquema da tabela 'sit_presence'	25
Figura 15 – Esquema do fluxo de dados para inserção de registro de localizações de veí- culos	28
Figura 16 – Tela de cadastro de localização	29
Figura 17 – Exemplo do cadastro do <i>callsign</i> de um veículo	30
Figura 18 – Esquema do fluxo de dados para inserção de registro de localizações de veí- culos via APRS	31
Figura 19 – Log do serviço recebendo pacotes do servidor APRS.	32
Figura 20 – Exemplo de registros de localização na plataforma <i>Sahana Eden</i>	34
Figura 21 – Exemplo de veículo sendo mostrado no mapa.	35
Figura 22 – Representação da tabela 'sensor_sensor_station'	37
Figura 23 – Representação da tabela 'sensor_sensor_station_registry'	38
Figura 24 – Representação da tabela 'sensor_sensor_property'	40
Figura 25 – Exemplo de estação de sensores sendo representada no mapa	44
Figura 26 – Exemplo de <i>popup</i> com informações dos sensores.	45
Figura 27 – Pontos de interesse mostrados no mapa do <i>Sahana Eden</i>	48
Figura 28 – Detalhe de um ponto de interesse mostrado no mapa do <i>Sahana Eden</i>	49
Figura 29 – Exemplo de registro de incidente no mapa.	50
Figura 30 – Exemplo de unidades móveis próximo ao local de um registro de incidente no mapa.	51

Figura 31 – Mapa com relevos.	52
Figura 32 – Mapa com relevos, estações de sensores e dados históricos.	53
Figura 33 – Proposta de estrutura de comunicação e gestão com a inserção do APRS. . .	54
Figura 34 – Commit da correção de erro na mudança de idiomas.	55
Figura 35 – Commit para tradução de termos presentes na página inicial.	55
Figura 36 – Mapa do site aprs.fi	56

LISTA DE TABELAS

Tabela 1 – Tipologia de eventos geradores de desastres e suas consequências à sociedade em 2013(Quantitativo Humano) (CENAD, 2014).	6
Tabela 2 – Descrição de trabalhos relevantes	10
Tabela 3 – Configuração de veículos para exibição no mapa.	35
Tabela 4 – Configuração de estações de sensores para exibição no mapa.	44

SUMÁRIO

LISTA DE FIGURAS	1
LISTA DE TABELAS	3
SUMÁRIO	4
1 INTRODUÇÃO	6
2 REVISÃO DA LITERATURA	9
2.1 Análise Bibliométrica	9
2.2 Gestão de desastres	11
2.3 <i>Hyogo framework</i>	12
2.4 <i>Sendai Framework</i>	13
2.5 <i>Sahana Eden</i>	13
2.5.1 Registros das organizações envolvidas	15
2.5.2 Gestão dos recursos humanos	16
2.5.3 Inventário	17
2.5.4 Ativos	17
2.5.5 Mapa	18
2.6 APRS	19
3 MATERIAL E MÉTODOS	21
3.1 Rastreamento de unidades móveis	21
3.1.1 Estrutura do banco de dados	22
3.1.2 Aquisição de dados do sistema APRS	26
3.2 Rastreamento de sensores	26
4 RESULTADO E DISCUSSÃO	28
4.1 Integração com o sistema APRS	28
4.1.1 Rastreamento de unidades móveis	28
4.2 Rastreamento de sensores	36
4.2.1 Estrutura do banco de dados	36
4.2.2 Aquisição de dados do sistema APRS	40
4.3 Instalação em produção e inserção de dados	46
4.4 Estudo de caso	49
4.4.1 Rastreamento de unidades móveis	49
4.4.2 Rastreamento de estações de sensores	51

4.5 Contribuições Geradas	54
5 CONSIDERAÇÕES FINAIS	57
Referências	58

1 INTRODUÇÃO

Os fenômenos naturais potencialmente destrutivos como chuvas torrenciais, terremotos e furacões acontecem invariavelmente, podendo tornar-se um desastre natural de acordo com os impactos causados à sociedade. [Tobin \(1997\)](#) diferencia riscos naturais e desastres. Segundo este autor um risco natural representa o risco potencial de interação entre os seres humanos que estão constantemente expostos às forças da natureza, enquanto desastres caracterizam-se por eventos atuais, que podem ou não causar mortes, entretanto usualmente acarretam impactos econômicos em alguma proporção.

[Wahlstrom e Guha-Sapir \(2015\)](#) Wahlstrom (2015) aponta que no período de 1995 à 2015 aproximadamente 51 milhões de pessoas foram afetadas de alguma forma por catástrofes climáticas no Brasil, o que coloca o país como o único país das Américas a constar na lista dos dez países mais afetados por tais fenômenos. Como pode ser visto na Tabela 1 no ano de 2013 o Brasil teve mais de 18 milhões de pessoas afetadas por desastres naturais no ano de 2013 ([CENAD, 2014](#)).

	Óbitos	Feridos	Enfermos	Desabrigados	Desalojados	Desaparecidos	Outros	Afetados
Alagamentos	4	279	4.306	44.330	48.260	0	180.641	277.820
Chuvas intensas	30	468	3.607	25.585	103.278	5	1.604.303	1.757.668
Deslizamentos	41	133	88	6.721	6.843	0	219.530	233.356
Enxurradas	38	787	2.324	17.266	118.074	92	778.694	931.608
Erosão	1	86	466	2.826	3.964	0	343.906	351.249
Estiagem	9	5.020	68.047	2.040	10.009	2	11.194.527	11.953.305
Geadas	0	0	0	0	0	0	0	0
Granizo	1	95	8	5.551	9.967	0	160.315	176.936
Incêndios florestais	0	0	139	113	145	0	2.970	3.367
Inundações	36	1.461	13.283	59.023	208.274	6	1.083.402	1.389.454
Vendaval	4	166	33	4.553	6.552	0	107.432	123.735
Outros	19	78	67.395	7.601	1.039	0	1.282.603	1.358.735
TOTAL	183	8.573	159.696	175.609	516.405	105	16.958.323	18.557.233

Tabela 1 – Tipologia de eventos geradores de desastres e suas consequências à sociedade em 2013(Quantitativo Humano) ([CENAD, 2014](#)).

Os desastres naturais são pouco explorados no conjunto de pesquisas sobre meio ambiente no Brasil. Resultando na dificuldade de acesso à informações atualizadas e consistentes sobre aspectos como, por exemplo, as formas de organização social, as percepções do risco das populações atingidas, a capacidade de auto-organização e o aprendizado dos atores sociais, entre outras que formam conjunto de informações indispensáveis para formulação e implemen-

tação de medidas de confrontação de desastres(MATTEDI; BUTZKE; others, 2001).

As informações necessárias para resposta às crises humanitárias devem ser oportunas e detalhadas, ao passo que as circunstâncias destas crises tornam difícil a coleta dos dados(THIEREN, 2005). Diante deste quadro as plataformas de ajuda humanitária contribuem disponibilizando os dados para que os atores envolvidos possam atuar na mobilização de recursos para os locais onde são realmente necessários.

Em situações de catástrofes a tomada de decisão é crítica, pois existe um fluxo de informação dinâmico numa situação em que a maior parte dos dados relevantes são heterogêneos, não podendo ser previstos e carregados anteriormente (CORDEIRO; CAMPOS; BORGES, 2015).

A precisão e a velocidade com que as informações chegam aos atores responsáveis pela tomada decisão é crucial para obtenção de êxito no gerenciamento de catástrofes. Diante disso recursos que tornem a entrada de dados automática, podem ser uma boa alternativa quando integrados à uma plataforma de ajuda humanitária, pois os dados seriam coletados de forma mais rápida e evitando possíveis erros humanos.

Dentre as informações necessárias para tomar decisões corretas em cenários catastróficos, estão a localização de dos atores envolvidos, localização de recursos, transportes, equipes de resgate e etc. Dado esta necessidade surge o desafio da implementação de um sistema de transmissão de dados que seja confiável em cenário caótico.

Diante do exposto o sistema APRS apresenta-se como uma boa alternativa. Tal sistema foi projetado para suportar a troca rápida e confiável de informações locais em tempo real. O conceito, que remonta a meados dos anos 80, é que todas as informações relevantes são transmitidas imediatamente a todos na rede e todas as estações capturam essa informação(BRUNINGA,).

Fundamentalmente, o APRS é um protocolo de comunicação de pacotes para disseminar dados para todos em uma rede em tempo real. Sua característica mais visível é a combinação de rádio por pacotes com a rede de satélites GPS (*Global Positioning System*), permitindo aos radioamadores exibirem automaticamente as posições das estações de rádio e outros objetos nos mapas em um computador. Outros recursos não diretamente relacionados ao relatório de posição são suportados, como relatórios de estação meteorológica, localização de direção e mensagens(Wade, 2000).

O sistema APRS pode ser utilizado para fazer o aporte automático de dados, como posição de veículos, agentes e condições climáticas, tornando mais ágil e precisa a tomada de decisão, o que na prática pode evitar perda de vidas humanas. A aquisição de dados pode ser feita diretamente dos servidores que compõe o sistema APRS o de sites de terceiros, como o caso do site <http://aprs.fi>, que disponibiliza dados do APRS.

Após a aquisição dos dados surge um outro problema que é a organização e disponibilização destes dados de forma concisa e clara para os atores necessários. Neste contexto sistemas de informações pode dar uma grande contribuição.

Dentre os sistemas de apoio a gestão de desastre destaca-se o *Sahana Eden*. O *Sahana Eden* é um sistema de código aberto criado especificamente para gestão de desastres. Este tem como características altamente flexível e adaptável a diferentes necessidades e cenários(KOHLI et al., 2011).

Este trabalho visa o estudo acerca da criação de funcionalidades que ofereçam a opção da entrada de dados na plataforma *Sahana Eden* serem feitas a partir da integração com a tecnologia APRS. Essa integração será feita principalmente para o aporte dos dados referentes a geolocalização de unidades móveis e o recolhimento de dados de estações de sensores.

2 REVISÃO DA LITERATURA

2.1 Análise Bibliométrica

A fim de realizar um levantamento bibliométrico acerca das publicações correlatas a este trabalho, no que tange aos seus principais objetos de estudo, foi realizada uma pesquisa bibliográfica na base de conhecimento Scopus (www.scopus.com). Para tanto foram utilizadas as seguintes palavras-chave:

- "Automatic Position Reporting System" e sua abreviação "APRS"
- "Sahana Eden"
- "disasters management"

O comando utilizado para pesquisar o termo "Automatic Position Reporting System" e sua abreviação "APRS" é demonstrado a seguir:

```
1 (TITLE-ABS-KEY ( "Automatic Position Reporting System" ) OR TITLE-ABS-KEY ( aprs ) )
```

O comando supracitado retornou 630 resultados. Vale ressaltar que muitos dos resultados retornados se tratam de falsos positivos uma vez que o termo 'APRS' também é utilizado em outras áreas, principalmente medicina na qual abarca 433 dos resultados retornados.

Em seguida foi realizada uma pesquisa a fim de levantar publicações referentes a plataforma *Sahana Eden*, conforme abaixo:

```
1 TITLE-ABS-KEY ( "Sahana Eden" )
```

Esta pesquisa permitiu a constatação de que a literatura presente na base *scopus* apresenta poucos trabalhos relacionado a plataforma *Sahana Eden* publicados. A pesquisa retornou apenas dois resultados que são brevemente apresentados a seguir na tabela [2](#).

Tabela 2 – Descrição de trabalhos relevantes

Título	Descrição
Five years of extra credit in a studio-based course: An effort to incentivize socially useful behavior	Este trabalho não é voltado de fato para o Sahana Eden, apenas citando-o como exemplo de software livre
Ushahidi and Sahana Eden open-source platforms to assist disaster relief: Geospatial components and capabilities	O trabalho busca investigar a utilização do sahana eden e um outro software open-source, chamado Ushahidi para gerentes de desastres rastrear pessoas afetadas por desastres.

Logo após uma pesquisa com o intuito de consolidar uma base teórica acerca da gestão de desastres, tal pesquisa foi realizada na base *scopus*, conforme a seguir:

1 TITLE-ABS-KEY ("disasters management")

Esta pesquisa apresentou um número maior de publicações retornando 12008 resultados distribuídos em diversas áreas.

Após esta primeira fase novas pesquisas foram feitas mesclando os termos, como pode ser visto na Figura 2.1.

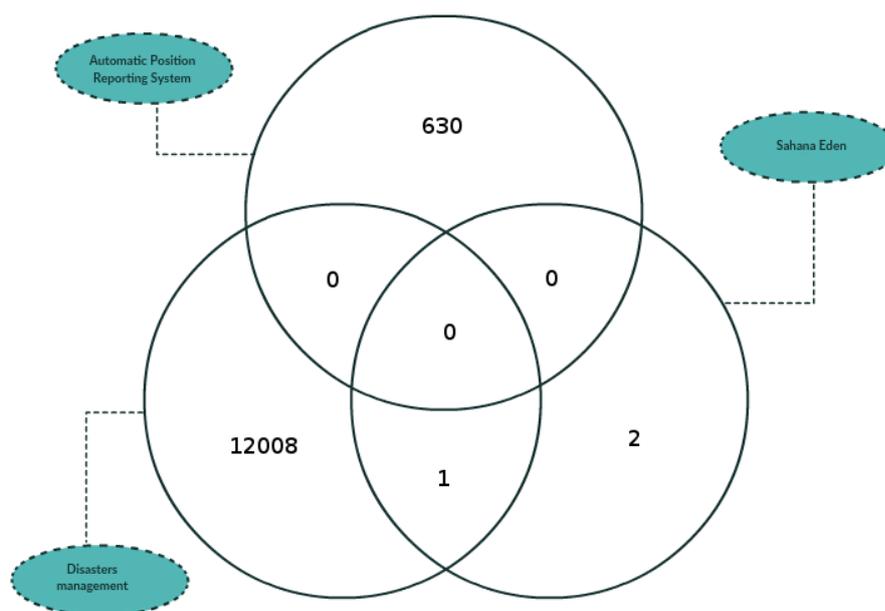


Figura 1 – Diagrama expressando a iteração dos termos pesquisados na base *scopus*

Todos os resultados obtidos foram pesquisados no dia 04/02/2018. Como pode ser visto as intersecções dos assuntos alvos deste trabalho não possuem publicações. Isto deve-se ao fato de que o georreferenciamento via APRS não ser uma *feature* presente na plataforma de ajuda humanitária *Sahana Eden* o que evidencia a relevância dos resultados que espera-se deste trabalho.

Diante disso o arcabouço teórico para desenvolvimento deste trabalho, sobre tudo no que diz respeito ao *Sahana Eden* será derivado de outras fontes tais como: Documentação oficial, sites e estudo do próprio código fonte da aplicação.

2.2 Gestão de desastres

Para o entendimento deste conceito é necessário antes definirmos o que é desastre. [Perdikaris \(2014\)](#) no capítulo introdutório define desastre como um estado no qual, uma população, grupo populacional, ou um indivíduo é incapaz de lidar com os efeitos adversos de um evento extremo sem ajuda externa. O impacto de um evento extremo pode incluir danos físicos significativos ou destruição, perda de vidas, ou mudança drástica para o meio ambiente. É um fenômeno que pode causar danos à vida, à propriedade e destruir a vida econômica, social e cultural dos povos.

Outra definição para desastre seria a seguinte: "Uma séria perturbação do funcionamento de uma comunidade ou de uma sociedade que envolva perdas e impactos generalizados de natureza humana, material, econômica ou ambiental, que ultrapassa a capacidade da comunidade ou da sociedade afetada de lidar com os seus próprios recursos"([UNISDR, 2009](#)).

[Pinkowski \(2008\)](#) relata que desastres pode apresentar varias definições como as supracitadas. Porém muitas definições apresentam as seguintes características.

- Perturbação dos padrões normais de vida. Essa interrupção é geralmente grave e também pode ser súbita, inesperada e generalizada.
- Efeitos humanos como perda de vidas, lesões, dificuldades e efeitos adversos na saúde.
- Efeitos sobre a estrutura social, tais como a destruição ou danos aos sistemas governamentais, edifícios, comunicações e serviços essenciais.
- Necessidades comunitárias como abrigo, alimentação, vestuário, assistência médica e assistência social.

Cada desastre traz consigo um risco em potencial. O risco de desastres compreende diferentes tipos de perdas potenciais que são muitas vezes difíceis de quantificar. Contudo, com o conhecimento dos riscos prevalentes e dos padrões de desenvolvimento populacional e socioeconômico, os riscos de catástrofes podem ser avaliados e mapeados, em termos gerais pelo menos (UNISDR, 2009).

Com o passar do tempo a gestão de desastres tem evoluído para cada vez mais tratar do risco de desastres deixando de ser apenas medidas reativas, para tratar de meios preventivos como pode ser visto na definição de UNISDR (2015): "O gerenciamento de riscos de desastres é a aplicação de políticas, processos e ações de redução de risco de desastres para prevenir novos riscos, reduzir o risco de desastres existente e gerenciar riscos residuais, contribuindo para o fortalecimento da resiliência".

Para ser eficaz, portanto, a gestão de desastres precisa ser implementada como uma atividade abrangente e contínua, e não como uma reação periódica a circunstâncias de desastres individuais. Conseqüentemente, os representantes nacionais encarregados das responsabilidades em matéria de gestão de catástrofes têm de lidar com uma vasta gama de questões políticas, de organização, operacionais e outras (PINKOWSKI, 2008).

2.3 *Hyogo framework*

Foi realizado no período 18 a 22 de janeiro de 2005 na província de Hyogo no Japão a "*The World Conference on Disaster Reduction*". A Conferência foi realizada com o intuito de fazer um balanço dos progressos alcançados na redução do risco de desastres desde a conferência de Yokohama de 1994 e fazer planos para a próxima década. Quatro documentos foram produzidos, que são o principal resultado da Conferência Mundial sobre a Redução de Desastres. Os documentos representam o empenho da comunidade internacional para lidar com a redução de desastres e para engajar-se em um plano de ação determinado e orientado a resultados para a próxima década (UNISDR, 2005).

Dentre os resultados da conferência destaca-se a criação do *Hyogo framework*. (ISDR, 2005) faz uma revisão dos aprendizados e desafios enfrentados desde a conferência de Yokohama de 1994. O documento define metas, prioridades, como implementar e acompanhar medidas que visam a redução dos riscos decorrentes de desastres, bem como medidas para mitigar prejuízos durante os desastres, no período de 2005 a 2015.

O *Hyogo Framework* trabalha com a ideia de se criar sociedades resilientes. Tal termo é mencionado em (REDUCTION, 2004), na qual o define resiliência como sendo a capacidade de um sistema, comunidade ou sociedade potencialmente exposta a riscos de se adaptar, resistindo ou mudando para alcançar e manter um nível aceitável de funcionamento e estrutura. Isso é determinado pelo grau em que o sistema social é capaz de se organizar para aumentar sua ca-

pacidade de aprender com desastres passados para uma melhor proteção futura e para melhorar as medidas de redução de riscos.

2.4 *Sendai Framework*

Com o avanço das pesquisas na área de redução de riscos de desastres, ficou notório que muitas vezes não é o perigo que determina um desastre, mas a vulnerabilidade, exposição e capacidade da população de antecipar, responder e recuperar de seus efeitos ou seja o poder de resiliência (AITSI-SELM I et al., 2015).

Em março de 2015, em *Sendai* no Japão, foi renovado o compromisso com a redução do risco de desastres. Foi firmado um novo *framework*, chamado *Sendai Framework* para redução do risco de desastres. Em teoria este *framework* foi baseado em lições aprendidas com a implementação do *framework de Hyogo*, durante a última década. O *framework* abrange um conjunto de metas e prioridades no intuito de conferir uma maior resiliência á áreas afetadas por pequenos e grandes desastres (POTERIE; BAUDOIN, 2015).

2.5 *Sahana Eden*

A palavra "Sahana" significa "alívio" ou "ajuda compassiva" em Sinhala, uma das línguas do Sri Lanka. A Sahana Software foi inicialmente criada pela comunidade de tecnologia da informação no Sri Lanka para ajudar o país a se recuperar após o terremoto e tsunami de 2004. A comunidade Sahana cresceu gradualmente a fim de cooperar com os profissionais de emergência, universitários e voluntários no processo de desenvolvimento de software (NGO, 2013).

A plataforma de ajuda humanitária *Sahana Eden* foi e tem sido utilizada na maioria dos desastres desde o *tsunami* que atingiu o oceano Índico de 2004 e mais recentemente durante o terremoto do Haiti em 2010 e o furacão Sandy. Atualmente é utilizado por diversas organizações como a Cruz Vermelha Internacional, Associação Portuguesa dos Bombeiros Voluntários (APBV), o Escritório de Gerenciamento de Emergências da Cidade de Nova York e a Ásia Centro de Preparação para Desastres, dentre outros (TEMNIKOVA¹; BIYIKLI; BOON, 2013) (KOHLI et al., 2011).

O objetivo principal do *Sahana Eden* é fornecer aos gestores de desastres ferramentas para minimizar o impacto negativo de desastres, através do rastreamento das necessidades das pessoas afetadas e da coordenação de agências de emergência. Além disso, a plataforma pode permitir que gestores de desastres e profissionais de campo deem respostas as emergências. Permite inclusive, que todo o mundo acessem e compartilhem as informações necessárias para reduzir o sofrimento humano desde de a preparação à fase pós catástrofes (NGO, 2013).

Desastres de natureza diferentes apresentam características diferentes, além disso cada

nação responde de um modo diferente a desastres. Assim o *Sahana Eden* foi feito sobre conceitos que permite que ele seja facilmente adaptado para atender as necessidades das mais variadas situações de desastres (PRUSTALIS; SILVA, 2010). A Figura 2 demonstra várias páginas iniciais provenientes de modificações feitas sobre a plataforma *Sahana Eden*.

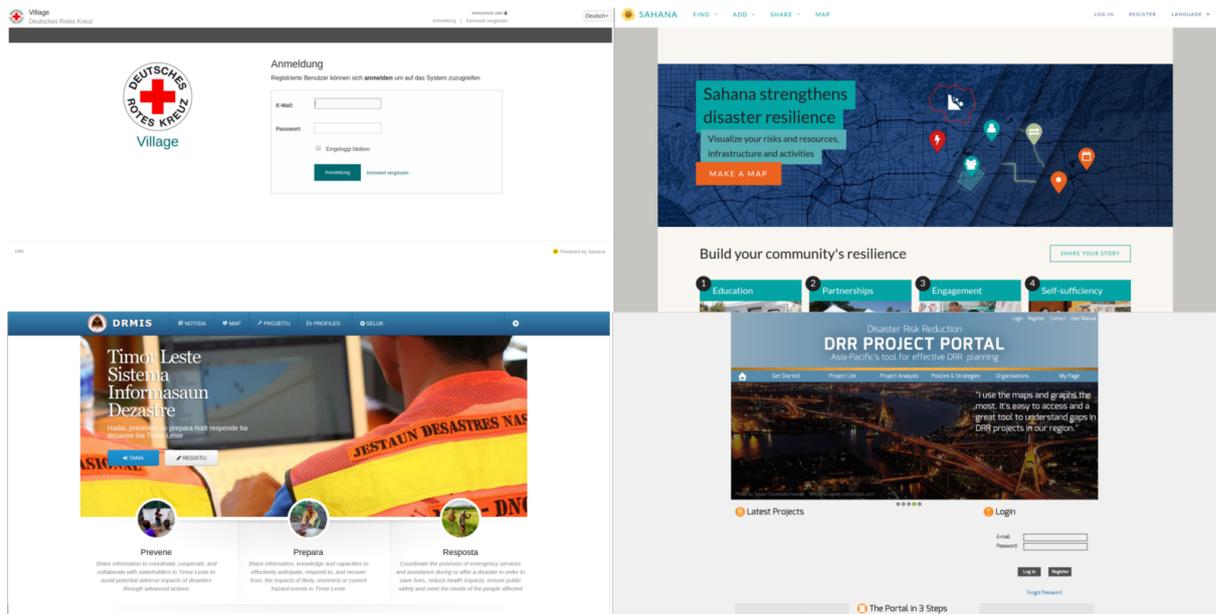


Figura 2 – Exemplos de *templates* feitos sobre a plataforma *Sahana Eden*

A partir do momento que é configurado para uma determinada situação e colocado em produção ele pode seguir os mais variados fluxos de informações até que os mesmos estejam disponibilizados na plataforma. Na Figura 3, é possível ver um exemplo deste fluxo.

Apesar da plataforma *Sahana Eden* ser flexível no que diz a configuração e expansão das funcionalidades, Cordeiro, Campos e Borges (2015) defendem que sistemas baseados em modelos de banco de dados relacional não são adequados para lidar com informações de estrutura imprevista, pois a rigidez de seu esquema aumenta as dificuldades na tarefa de integrar os dados, o que geralmente resulta na perda de semântica.

Como apresentado na seção 2.1, há poucos trabalhos publicados acerca do *Sahana Eden*. Diante disto as informações a seguir foram retiradas do (KOHLI et al., 2011), que é um guia oficial criado pela *Sahana Foundation*.

O *Sahana Eden* é desenvolvido utilizando a linguagem *python* e o *framework web2py*, contando com vários módulos que podem ser utilizados e adaptados para satisfazer as necessidades de cada situação na qual será empregado. As sub-sessões a seguir fazem um resumo de

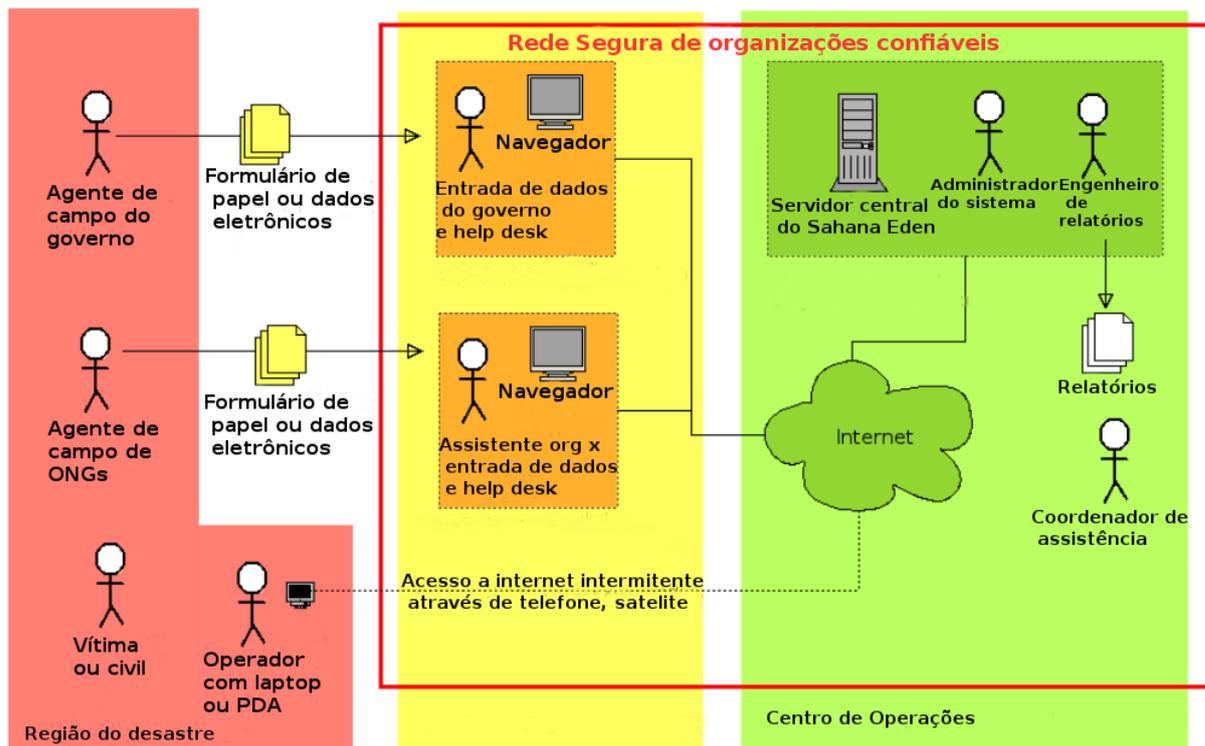


Figura 3 – Exemplo de fluxo de informações uma vez a plataforma em produção. modificado de (NURI S.T., 2016)

alguns dos principais módulos que compõem o *software* e que de alguma forma envolvidos ou são afetados pelas modificações criadas durante este trabalho.

2.5.1 Registros das organizações envolvidas

Durante um desastres diversas organizações se envolvem na gestão de toda situação, seja no registro de ocorrência, entrega de mantimentos ou apoiando as pessoas que necessitam. O *Sahana Eden* é capaz de registrar e acompanhar as organizações que estão ativas em diferentes contextos, proporcionando o acompanhamento e cooperação entre as diferentes organizações envolvidas na gestão do desastre.

O sistema permite que as organizações registrem seus escritórios, armazéns e bases de campo, que podem ser mapeados, bem como *links* para outros módulos, como Recursos Humanos, Ativos e Inventário. A figura 4 mostra um exemplo de uma lista de organizações cadastradas no *Sahan Eden*.

Organizations

Search: ? HELP

Type:

Home Country:

[Clear Filter](#) [Saved Filters](#)

Show 16 entries Link to this result | Export as:

Showing 1 to 16 of 16 entries

	Name	Acronym	Organization Type	Website
Open	Barangay Ayala Alabang test	BAA	Government	http://www.barangayayalaalabang.com/
Open	Civil Defence Department	APM	Government	
Open	CVISNET Foundation, Inc.	CVISNET	NGO	http://www.cvis.net.ph
Open	Disaster Early Warning System	DEWS	Academic	
Open	Fireteam	FT	NGO	
Open	Homeless First Response	HFR	NGO	http://www.hfrbmt.wixsite.com/website
Open	National Disaster Risk Reduction and Management Council	NDRRMC	Government	http://www.ndrrmc.gov.ph/
Open	National Museum of the Philippines		Government	http://www.nationalmuseum.gov.ph/index.html
Open	olathe cert	ocert	NGO	http://olatheks.org
Open	Radio Amateurs Emergency Network (UK)	RAYNET	Private	
Open	Sebassur07		Private	
Open	Sozy		Private	

Figura 4 – Exemplo de uma lista de organizações cadastradas no *Sahana Eden*.

2.5.2 Gestão dos recursos humanos

Durante a gestão de desastres há varias pessoas envolvidas, sejam voluntários, funcionários que trabalham nas diferentes organizações, desempenhando papeis diferentes simultaneamente. O Sistema *Sahana Eden* possui o módulo de recursos humanos que pode auxiliar na gestão das pessoas envolvidas. Ele rastreia as pessoas que estão, quais habilidades elas têm e ajuda a garantir que todos estão efetivamente envolvidos com o trabalho que precisa ser feito. *Sahana Eden* também pode ser usado para fornecer uma lista de contatos para garantindo que as pessoas certas possam ser contatadas no momento certo. A figura 5 mostra um exemplo de uma lista com o cadastro de voluntários.

Volunteers

Search:

Organization:

[More Options](#) [Clear Filter](#) [Saved Filters](#)

Show 5 entries Link to this result | Export as:

Showing 1 to 5 of 5 entries

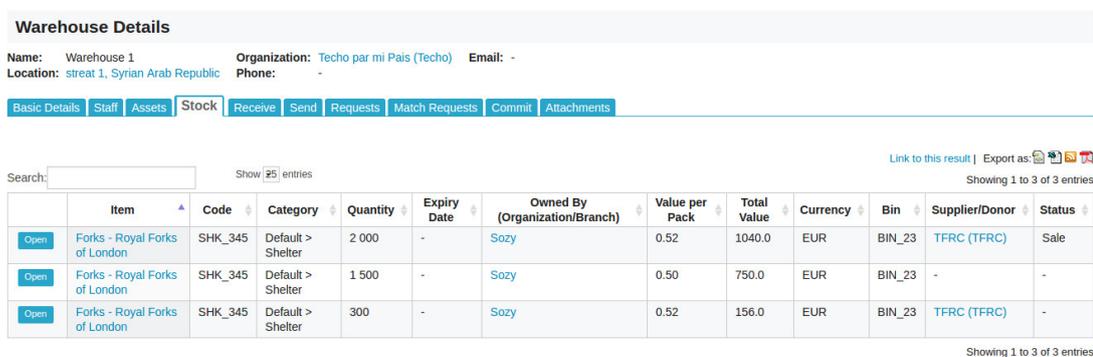
	Person	Sex	Volunteer Role	Organization	Mobile Phone	Email	Program	Home Address	Trainings	Certificates
Open	Allison Cline	female	Law Enforcement	Civil Defence Department (APM)	-	ieowhfo@hotmail.com	-	-	-	RN - Registered Nurse
Open	arjan	female	-	Civil Defence Department (APM)	-	admin@gmail.com	-	-	-	-
Open	Joe smith	male	Opperations	olathe cert (ocert)	-	jsmith@example.com	-	-	-	-
Open	Kabi		-	Disaster Early Warning System (DEWS)	-	epkabi@gmail.com	-	-	-	-
Open	Ram	male	Health_Inspector	Disaster Early Warning System (DEWS)	-	kannan09mca@gmail.com	-	-	-	-

Showing 1 to 5 of 5 entries

Figura 5 – Exemplo de uma lista de voluntários cadastrados no *Sahana Eden*.

2.5.3 Inventário

As organizações atuam fornecendo itens básicos de vida para as pessoas afetadas por desastres naturais, bem como dando às comunidades ferramentas que necessitam para restaurar seus meios de subsistência. O *Sahana Eden* pode ser usado para gerenciar inventários de itens e combinar pedidos com depósitos dentre outras facilidades. Operacionalmente, o *Sahana Eden* pode ser usado para registrar e automatizar transações de envio e recebimento de remessas. *Sahana Eden* suporta vários catálogos de Itens, bem como fornece itens alternativos para garantir um uso mais eficaz dos suprimentos. A figura 6 ilustra uma lista de itens de um determinado armazém.



The screenshot shows the 'Warehouse Details' page for 'Warehouse 1'. It includes fields for Name, Location, Organization, and Email. Below this is a navigation bar with tabs for 'Basic Details', 'Staff', 'Assets', 'Stock', 'Receive', 'Send', 'Requests', 'Match Requests', 'Commit', and 'Attachments'. The 'Stock' tab is active, displaying a table of items. The table has columns for Item, Code, Category, Quantity, Expiry Date, Owned By, Value per Pack, Total Value, Currency, Bin, Supplier/Donor, and Status. Three items are listed, all with a quantity of 2,000, 1,500, and 300 respectively. The supplier for all items is 'Sozy'.

	Item	Code	Category	Quantity	Expiry Date	Owned By (Organization/Branch)	Value per Pack	Total Value	Currency	Bin	Supplier/Donor	Status
Open	Forks - Royal Forks of London	SHK_345	Default > Shelter	2 000	-	Sozy	0.52	1040.0	EUR	BIN_23	TFRC (TFRC)	Sale
Open	Forks - Royal Forks of London	SHK_345	Default > Shelter	1 500	-	Sozy	0.50	750.0	EUR	BIN_23	-	-
Open	Forks - Royal Forks of London	SHK_345	Default > Shelter	300	-	Sozy	0.52	156.0	EUR	BIN_23	TFRC (TFRC)	-

Figura 6 – Exemplo de uma lista de itens.

2.5.4 Ativos

Durante um desastre vários ativos são utilizados como veículos, itens de socorro, equipamentos de rádio e etc. O *Sahana Eden* é capaz de gerenciar ativos, além de rastrear onde este encontram-se, pra quem foram designados, bem como em que condições eles estão. Isso garante um uso eficaz e eficiente dos ativos disponíveis. Um exemplo de uma lista de ativos é mostrado na figura 7.

Assets

Search: ? HELP

Category:

[More Options](#) [Clear Filter](#) [Saved Filters](#)

Show 25 entries [Link to this result](#) | Export as:

Showing 1 to 7 of 7 entries

	Category ▲	Item	Asset Number	Assigned To	Organization	Facility	Country	State / Province	County / District	City / Town / Village	Village / Suburb	Condition	Comments
Open	Standard > EQUIP	Heater	232334	-	TFRC (TFRC)	Corporate office (Office)	India	-	-	-	-	Unknown	-
Open	Standard > EQUIP	Ambulance & Equipment	1	-	Disaster Early Warning System (DEWS)	Warehouse (Warehouse)	India	-	-	-	-	Unknown	-
Open	Standard > EQUIP	Fire Pump	2	-	Disaster Early Warning System (DEWS)	Warehouse (Warehouse)	India	-	-	-	-	Unknown	-
Open	Standard > EQUIP	Wrench		-	Barangay Ayala Alabang test (BAA)	Barangay Ayala Alabang DRRMC (Office)	Philippines	Tabon Cave Complex	-	-	-	Unknown	jodjomatojsamot
Open	Standard > EQUIP	Boat	1234	-	Civil Defence Department (APM)	APM Kota Bharu (Office)	Malaysia	-	-	-	-	Unknown	-
Open	Standard > EQUIP	Ambulance & Equipment	1	-	Disaster Early Warning System (DEWS)	Warehouse (Warehouse)	India	-	-	-	-	Unknown	-
Open	Standard > EQUIP	Boat	3	-	Disaster Early Warning	Warehouse (Warehouse)	India	-	-	-	-	Unknown	-

Figura 7 – Exemplo de uma lista de ativos.

2.5.5 Mapa

O *Sahana Eden* possui uma funcionalidade de mapeamento que permite que todos os dados baseados em geolocalização sejam visualizados em um mapa. Essas informações também podem ser pesquisadas usando uma seleção de limites com base no mapa. Os mapas ajudam a fornecer uma consciência situacional, como pode ser visto na figura 8, que é essencial quando se planeja preparar-se para uma catástrofe.

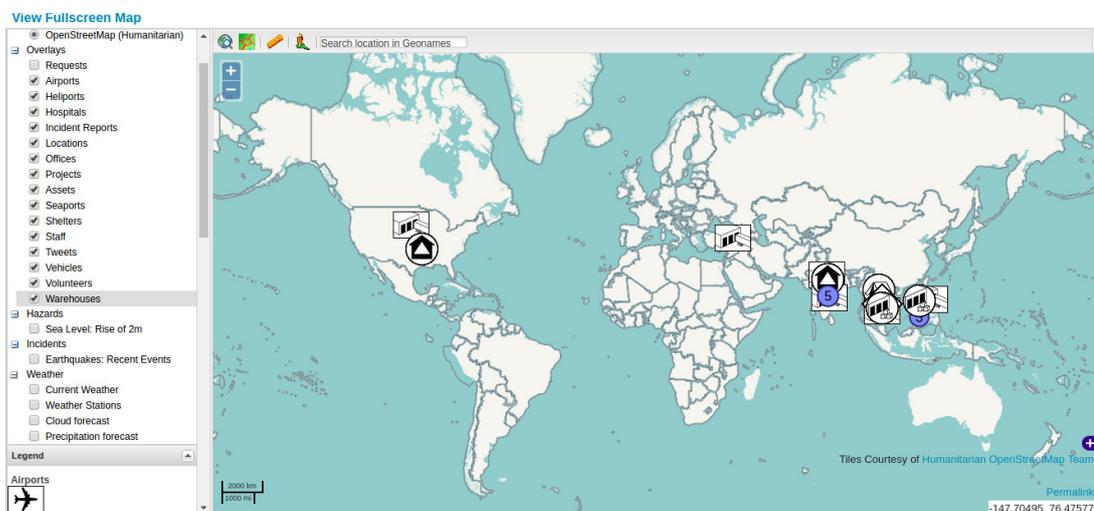


Figura 8 – Exemplo de mapa exibindo alguns registros.

2.6 APRS

APRS é a abreviatura de *Automatic Position Reporting System*, que foi proposto por Bob Bruninga em 1992 na *TAPR / ARRL Digital Communications Conference*. O APRS é um protocolo de comunicação para a disseminação de dados em *broadcast* em uma rede em tempo real. Ele utiliza o sistema de posicionamento global (GPS), permitindo aos radioamadores exibir automaticamente as posições das estações de rádio e outros objetos. Outros recursos não diretamente relacionados ao envio do geoposicionamento são suportados, como relatórios de estações meteorológica, por exemplo (GROUP et al., 2000).

A tecnologia APRS pode ser utilizada para os mais diferentes fins. Chaiyasoonthorn, Hongyim e Mitatha (2015), por exemplo propõe a utilização do sistema APRS para transmitir dados de radiação e posição de robôs autônomos em áreas com contaminadas. O estudo relata sucesso na comunicação a uma distância de 20 quilômetros entre o robô e um I-gate com uma antena de recepção localizada em uma torre de rádio com 60 metros de altura.

Outro relato da literatura feito por Pereira e Júnior (2006), mostra a tecnologia APRS sendo empregada na monitoração de uma viatura utilizada na entrega de medicamentos no tratamento supervisionado da tuberculose. Os autores consideram ainda viável o emprego do sistema APRS para o rastreamento de ambulâncias, de UTI's móveis, de equipes de tratamento supervisionado e de outras formas de transporte relacionadas à assistência na área de saúde.

Já Tonder (2005) propõe uma pilha de protocolo melhorada com o intuito de melhorar a *performance* do APRS, substituindo a camada de controle de acesso de mídia (MAC).

Na figura 9 está demonstrada a arquitetura básica e interação entre os componentes da infraestrutura do APRS. Os pacotes de dados são transmitidos por pontos móveis. Para estender a rede os pacotes podem ser retransmitidos por repetidores digitais. *Internet gateways*, também conhecidos como *igates*, são responsáveis por coletar os pacotes de dados e publicarem eles na internet. A cobertura de repetidores ou *gateways* depende da topografia, altura da antena, etc. Um alcance de 50 km não é incomum (HANSSEN, 2015).

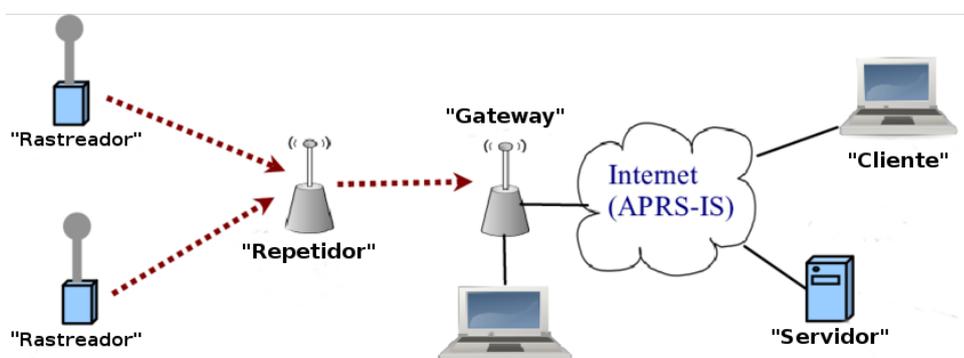


Figura 9 – Infraestrutura do APRS (Alterado de (HANSSEN, 2015)).

Ainda observando a figura 9 nota-se a presença do componente denominado 'serviço de Internet APRS' ou mais comumente chamado de APRS-IS que nada mais é do que uma rede mundial de servidores interligados. *Gateways*(Igates) ou qualquer aplicativo pode se conectar ao serviço APRS-IS para transmitir ou receber dados.

A figura 10 mostra a organização a nível mundial dos servidores da rede APRS. A rede possui alguns servidores principais onde passa todo o fluxo da rede. Logo depois vêm os servidores de segundo nível e os servidores regionais. Atualmente um *balancer* foi implementado a fim de distribuir a carga de maneira mais eficiente entre os servidores que estejam *online* no momento.

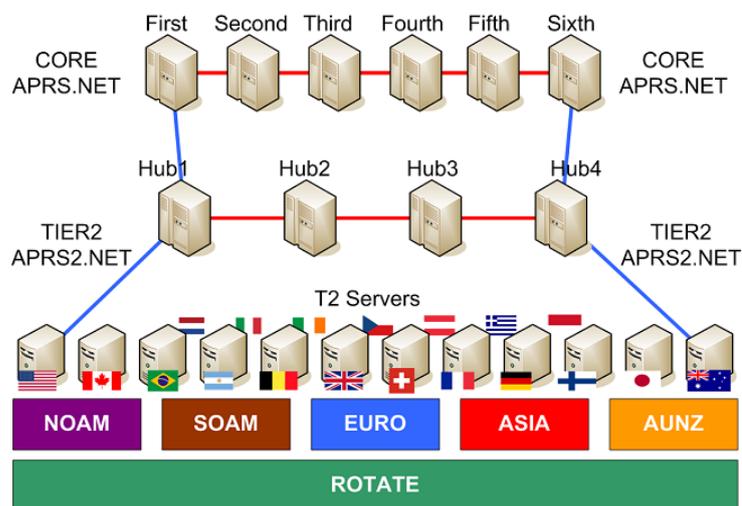


Figura 10 – Estrutura dos servidores APRS (APRS..., 2016).

3 MATERIAL E MÉTODOS

Este capítulo descreve a metodologia adotada para a integração do APRS a plataforma de ajuda humanitária *Sahana Eden*, para rastreamento de unidades móveis e aquisição de dados de estações de sensores.

Com o intuito de confirmar a viabilidade, bem como os benefícios da integração do APRS com a plataforma *Sahana Eden*, serão utilizados dados reais para efeito de teste. No entanto até que o sistema esteja apto a ser utilizado em produção em situações reais de catástrofes há uma série de passos a serem implementados até chegar ao objetivo final, estes passos estão listados abaixo.

- Instalação do sistema *Sahana Eden* em modo de desenvolvimento.
- Estudo do sistema dos conceitos técnicos do sistema *Sahana Eden*, bem como das tecnologias a este correlatos.
- Estudo e implementação de ferramentas que viabilizem a leitura de dados dos servidores do sistema APRS.
- Integração dos dados extraídos dos servidores do APRS à plataforma *Sahana Eden*, para rastreamento de veículos.
- Integração dos dados extraídos dos servidores do APRS à plataforma *Sahana Eden*, para rastreamento de sensores.
- Instalação do sistema em produção.
- Inserção de dados reais, para analisar como estes relacionam-se com os dados oriundos do sistema APRS.

3.1 Rastreamento de unidades móveis

Em se tratando de gerência de desastres a ciência por parte do tomador de decisão de onde estão localizados seus recursos móveis é de total importância para coordenar os atendimentos a população afetada de forma eficiente.

Esta seção irá expor o processo desenvolvido para a integração do sistema APRS à plataforma *Sahana Eden* para possibilitar o rastreamento de unidades móveis. A seção se subdividirá em duas subseções, uma relacionada a exploração e entendimento da estrutura do banco de dados correlato ao georreferenciamento de unidades móveis, e outra relacionada a aquisição de dados do sistema APRS.

3.1.1 Estrutura do banco de dados

Para utilização dos dados do APRS para rastreamento de veículos, torna-se necessário entender como está organizada a estrutura do banco de dados do *Sahana Eden*. Para tal pode-se ver na Figura 11.

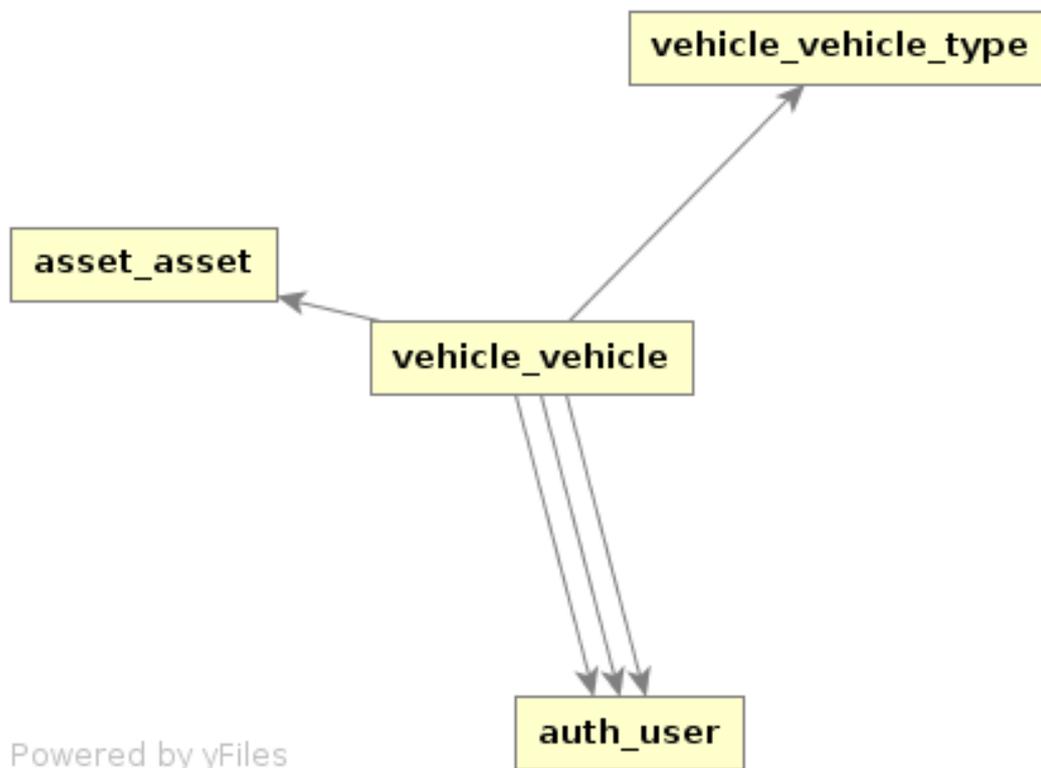


Figura 11 – Esquema da tabela de veículos do *Sahana Eden*

Como pode ser visto a tabela de veículos(*vehicle_vehicle*) possui uma referência para tabela '*asset_asset*', o que pode ser entendido como se veículo fosse uma especialização de *assets*, ou seja os veículos são um tipo de recurso dentre os inúmeros tipos que são gerenciados na plataforma *Sahana Eden*. Além disso a tabela de veículos também tem um relacionamento com a tabela '*vehicle_vehicle_type*', que é o cadastro com a informação do tipo de veículo.

Este é o ponto inicial para entendermos a estrutura das tabelas do banco de dados interessantes no que diz respeito ao cadastro e georreferenciamento de veículos. Na Figura 12 a representação da tabela '*asset_asset*' e seus relacionamentos.

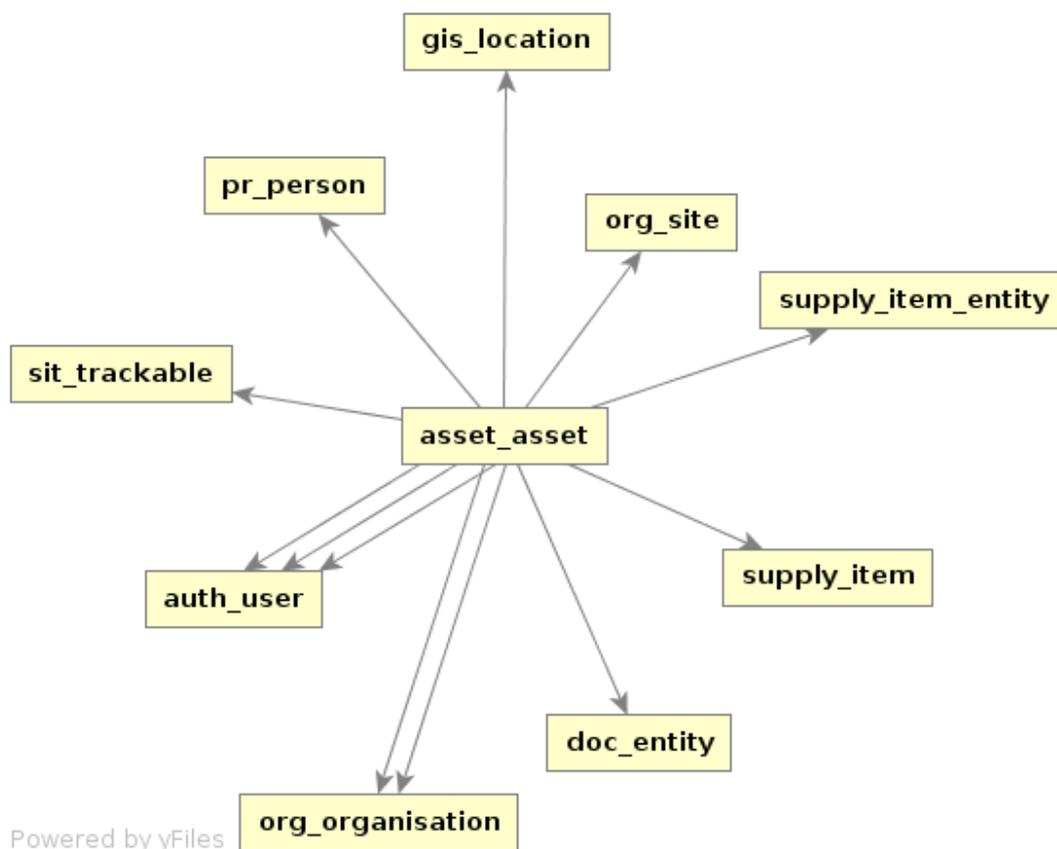


Figura 12 – Esquema da tabela 'asset_asset' do *Sahana Eden*

Nesta figura pode-se ver a tabela 'asset_asset' com seus inúmeros relacionamentos. Dentre os seus relacionamentos dois são relevantes para a questão do georreferenciamento e são eles 'gis_location' e 'sit_trackable'. A tabela 'gis_location' pertence ao módulo 'GIS', módulo que tem dentre as suas tratativas o manejo de informações referentes a localização, bem como a manipulação de mapas. A tabela 'gis_location', é a tabela responsável uma determinada posição, sendo utilizada em inúmeras situações pelo sistema. Dentre outros possui os atributos de latitude e longitude, que são utilizados para o posicionamento de um determinado recurso no mapa, por exemplo.

Uma vez entendido a função da tabela 'gis_location', pode-se notar na Figura 12 que a tabela 'asset_asset' faz uma referência direta a tabela 'gis_location', o que nos permite afirmar que essa relação seria do tipo 1 para 1 (um recurso tem uma localização) ou no máximo muitos para 1 (muitos recursos tem referência para a mesma localização), vale ressaltar este último caso seria possível de acordo com a modelagem do banco, porém não acontece na prática. De qualquer forma é possível notar que essa abordagem de modelagem não contempla a realidade na qual é necessário fazer o rastreamento do recurso através de dados históricos de localização.

Todavia a plataforma *Sahana Eden* dispõe de recursos para o rastreamento dos seus recurso, permitindo o cadastro de varias localizações através do tempo. Isto leva análise da tabela 'sit_trackable', que por sua vez permite que um determinado recurso seja rastreável. Na Figura 13 tem-se uma visão detalhada acerca da tabela 'sit_trackable'.

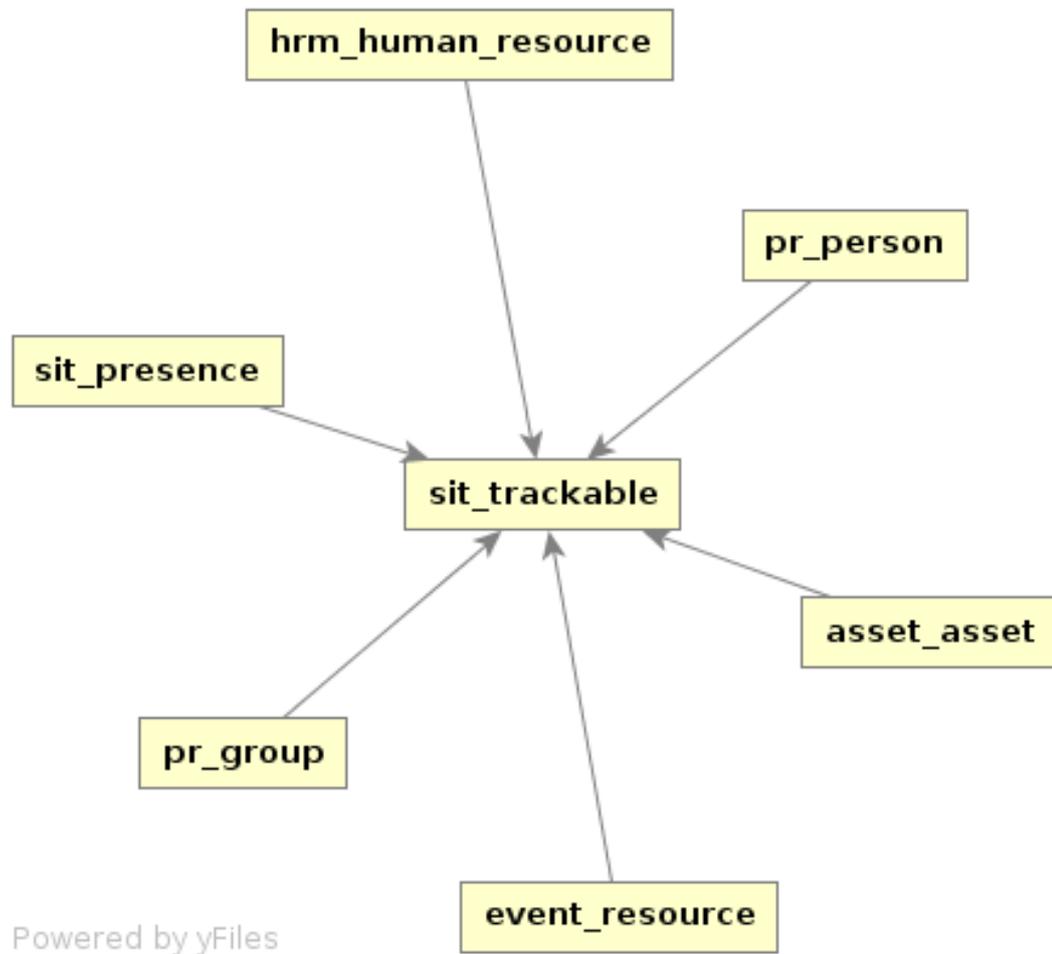


Figura 13 – Esquema da tabela 'sit_trackable'

Na imagem é possível notar a tabela 'sit_trackable' e outras tabelas que mantêm referência a ela. Como dito anteriormente tabela 'sit_trackable' permite que um determinado recurso seja rastreável, para além disto outros tipos de elementos tem referência para ela, ou seja são rastreáveis. Porém dentre as tabelas que mantêm relacionamento com 'sit_trackable', existe a tabela 'sit_presence' que não é um elemento rastreável propriamente, mas sim a tabela que registra uma ocorrência no sentido de registrar a localização do elemento em um determinado

momento. Esta relação entre 'sit_presence' e o elemento na qual ele registra uma localização é feita através da tabela 'sit_trackable'.

Seguindo adiante tem-se a Figura 14 mostra o esquema de organização da tabela 'sit_presence' e suas conexões.

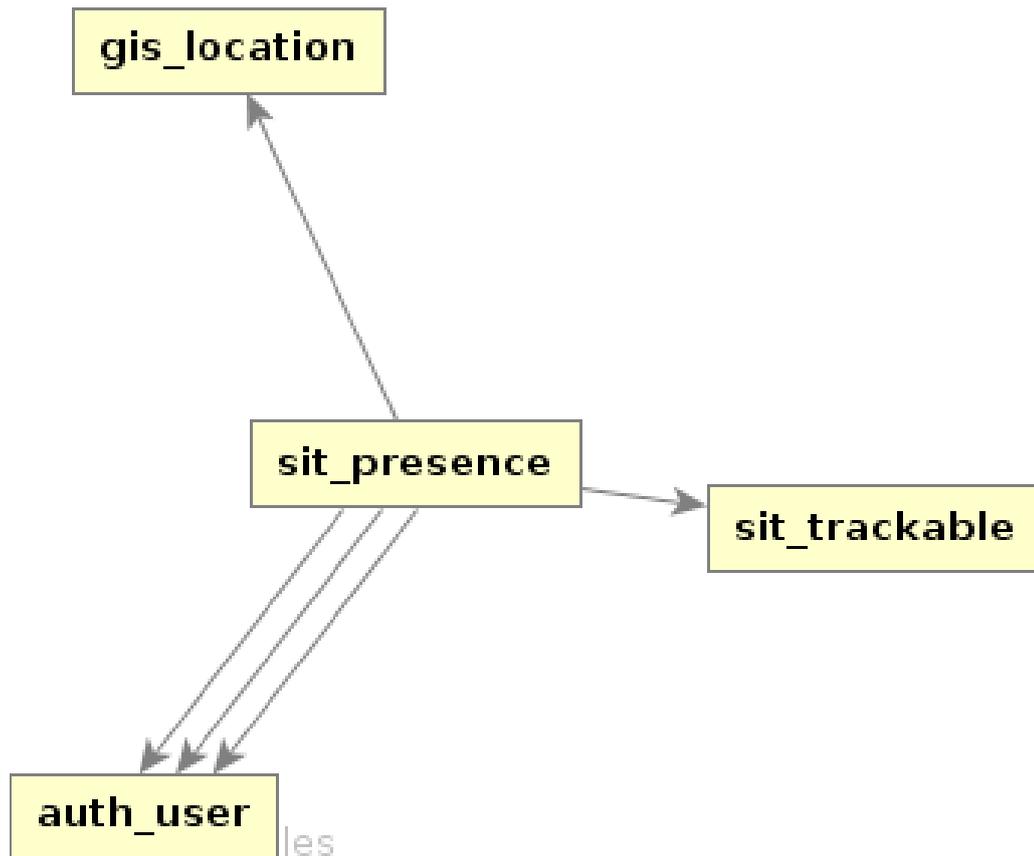


Figura 14 – Esquema da tabela 'sit_presence'

Como pode ser visto na imagem a tabela 'sit_presence' tem uma referência para a tabela 'sit_trackable', que é a via por onde se registra de localização de diversos elementos. Uma outra relação na qual a tabela 'sit_presence' mantém é com a tabela 'gis_location', que como dito anteriormente é a tabela que guarda as informações de latitude e longitude, ou seja a localização de fato de um elemento.

Com o estudo da tabela 'sit_presence' encerra-se o estudo acerca da modelagem do banco de dados para entender qual o caminho necessário para inserção de dados para gerar dados na qual seja possível expressar a posição de uma unidade móvel através do tempo.

Uma vez apresentada a estrutura da organização do banco de dados para poder inserir os dados relacionados a localização de unidades móveis, vale a pena fazer um resumo do 'caminho' dos dados. Relembrando temos a seguinte quência: A tabela 'vehicle_vehicle' tem uma relação com a tabela 'asset_asset' o que implica em dizer que os veículos são um tipo de recurso. A tabela 'asset_asset' tem uma relação com a tabela 'sit_trackable' o que significa que os recursos são rastreáveis. A tabela 'sit_trackable' por sua vez se relaciona com a tabela 'sit_presence', que é a tabela responsável por guardar cada episódio no que diz respeito o registro de localização. A tabela 'sit_presence' se relaciona com a tabela 'gis_location' que é a tabela responsável por conter as informações de latitude e longitude. Com isso fecha-se o ciclo de entendimento da estrutura do banco de dados para registro de posicionamento de veículos.

3.1.2 Aquisição de dados do sistema APRS

Uma vez dominado a estrutura do banco de dados, no que diz respeito as tabelas utilizadas para o registro do histórico de localização dos diversos tipos de veículos, chega-se a segunda tarefa para alcançar o objetivo final, que são os meios para aquisição de dados oriundos do sistema APRS.

Para obtenção de dados do sistema APRS, a princípio criou-se uma biblioteca *python* que acessa a API do site 'aprs.fi', e coleta os dados de interesse. Todavia este site é um sistema privado e tem em seus termos de uso a não garantia de disponibilidade, optou-se por acessar diretamente os servidores que compõe a rede APRS.

Foi então criado um serviço que acessa um determinado servidor APRS e recebe as informações das unidades móveis e as insere no banco de dados da plataforma *Sahana Eden*.

3.2 Rastreamento de sensores

Em meio a catástrofes ou mesmo como forma de prevenção, é de suma importância que o tomador de decisão tenha a sua disposição informações que o permita tomar decisões de maneira mais eficientes e assertivas possíveis.

Velocidade do vento, temperatura, chuva e etc, estão dentre as informações mais importantes para a gestão de desastres. Ter este tipo de informação atrelados as outras informações já existente na plataforma *Sahana Eden*, pode ampliaria de forma significativa o poder de reação as adversidades.

A plataforma *Sahana Eden* funciona totalmente de forma configurável e modularizada, permitindo habilitar e desabilitar módulos, conforme a necessidade. Diante disto criou-se um novo módulo que possibilita a gestão de sensores na plataforma. Além de criar todo arcabouço para o cadastro e leitura de sensores, também produziu-se a integração com o sistema APRS, para que aquisição de dados dos sensores fossem feita de forma automática.

Para o cadastro das estações de sensores foi necessário a criação da estrutura de banco de dados que dá suporte a isto. O módulo para gestão de sensores adicionou três tabelas à estrutura de banco de dados do *Sahana Eden*.

A aquisição de dados dos servidores APRS, foi feita da mesma forma da descrita para o rastreamento de veículos. Ou seja tem-se um serviço que funciona paralelamente à plataforma *Sahana Eden*. Onde este serviço acessa o banco de dados da aplicação para recolher os *callsigns* das estações de sensores cadastrados para que através destes *callsigns* filtrar os dados provenientes do servidor APRS.

Foi adicionado ao serviço de acesso aos servidores APRS uma estrutura de controle para identificar se o dado pacote originou-se de uma estação meteorológica. Uma vez que o pacote possua o atributo *'weather'*, os dados do pacote APRS são persistidos no banco de dados.

Finalizada a criação da funcionalidade de rastreamento dos sensores o sistema foi colocado em produção e criado um *script* para que fosse feito a inserção de dados reais provenientes do planos de contingência do município de Santa Maria Madalena do estado do Rio de Janeiro.

4 RESULTADO E DISCUSSÃO

Serão apresentadas as opções que foram conferidos a plataforma *Sahana Eden* a partir da implementação de novas funcionalidades. Essa exposição dar-se-á com a explicitação da interação entre os dados reais inseridos, conforme mostrado na seção 4.3, bem como os dados oriundos do sistema APRS.

Logo depois serão apresentadas algumas contribuições efetivas para a comunidade, que a princípio não faziam parte do escopo principal deste trabalho, mas que foram geradas no decorrer do processo de desenvolvimento deste trabalho.

4.1 Integração com o sistema APRS

4.1.1 Rastreamento de unidades móveis

Uma vez dominado a estrutura do banco de dados, no que diz respeito as tabelas utilizadas para o registro do histórico de localização dos diversos tipos de veículos, chega-se a segunda tarefa para alcançar o objetivo final, que são os meios para aquisição de dados oriundos do sistema APRS.

Essa subseção tem como objetivo elucidar como deu-se o processo de criação do mecanismo utilizado para recolher os dados dos servidores do APRS e persistir tais dados na base do sistema *Sahana Eden*.

Antes de de analisar como se dá por padrão o fluxo de dados para o registro de localização de unidades móveis. A Figura 15 ilustra o cenário padrão de entrada de dados.

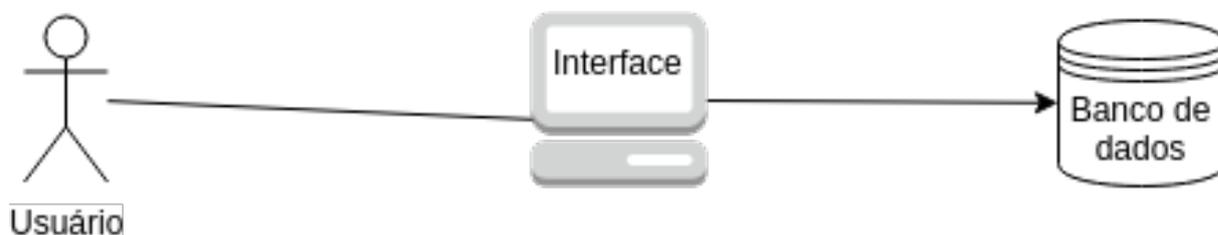


Figura 15 – Esquema do fluxo de dados para inserção de registro de localizações de veículos

Como pode ser visto na imagem tem-se um usuário que pode ser um coordenador, um agente de campo ou até mesmo civis, dependendo de como for configurado a plataforma *Sahana*

Eden. O usuário faz a inserção dos dados de localização através da interface da plataforma, conforme mostrado na Figura 16.



The screenshot shows a web interface with a navigation bar at the top containing tabs: 'Detalhes do Ativo', 'Vehicle Details', 'contatos', 'GPS Data', and 'registro'. The 'registro' tab is active. Below the navigation bar is a form titled 'INCLUIR REGISTRO'. The form contains the following fields and controls:

- 'data/hora:': A text input field with a 'Clear' button to its right.
- 'Latitude:': A text input field.
- 'Longitude:': A text input field.
- 'Use deg, min, sec': A small blue link text.
- 'Place on Map': A button with a location pin icon.
- 'Endereço:': A wide text input field.
- 'Speed:': A text input field.
- 'Accuracy:': A text input field.
- 'armazenar': A blue button.
- 'Cancelar': A red button.

Figura 16 – Tela de cadastro de localização

A imagem ilustra a tela de cadastro de localização para um determinado veículo. Como pode ser visto o usuário cadastra informações como: data/hora, latitude e longitude, o que permite criar um histórico de localizações.

A necessidade da inserção deste tipo de dados de forma manual, além de ser um processo custoso, pode tornar os dados cadastrados pouco precisos, pois o usuário pode inseri-los de forma equivocada. Diante disso foi criada a funcionalidade de rastreamento das unidades móveis através do sistema APRS, visando a entrada automática de dados.

O primeiro passo no sentido da integração foi criar no cadastro de veículos um campo para inserção dos *callsign*. O *callsign* identifica de forma única um elemento APRS. Na Figura 17 é mostrado o cadastro do *callsign* para um determinado veículo.

O sistema APRS em todo o mundo possui várias ferramentas para acesso e visualização de dados. Dentre as ferramentas uma comumente utilizada para este fim é o site 'aprs.fi', que funciona como um concentrador de dados da rede APRS, além de exibir os dados em um mapa.

Figura 17 – Exemplo do cadastro do *callsign* de um veículo

Com o intuito de obter os dados do sistema APRS, criou-se uma biblioteca *python* que acessa a API do site 'aprs.fi', e coleta os dados de interesse. Porém como o site é um sistema privado e que tem em seus termos de uso a não garantia de disponibilidade, optou-se por acessar diretamente os servidores que compõe a rede APRS.

O objetivo da integração é disponibilizar uma forma alternativa de entrada de dados, tornando a entrada de localização de unidades móveis uma atividade livre de interferências humanas, mais rápida e precisa. Na figura 18 é representado o esquema de como se dá o fluxo de dados, uma vez adotado a integração com o sistema APRS.

Como demonstrado na Figura 15 foi criado um serviço que roda em paralelo com a plataforma *Sahana Eden*. Este serviço funciona acessando o banco de dados da aplicação para recolher os *callsign* cadastrados nos veículos e através destes *callsigns* filtrar os dados provenientes do servidor APRS.

O sistema APRS disponibiliza seus dados através de vários servidores dividido em vários níveis. Neste trabalho foi utilizado o servidor brasileiro com endereço 'brazil.aprs2.net' que utiliza a porta 14579.

Uma vez que o serviço é iniciado, ele começa a monitorar o tráfego de dados no servidor APRS. A Figura 19 exemplifica o log do serviço recebendo pacotes do servidor APRS.

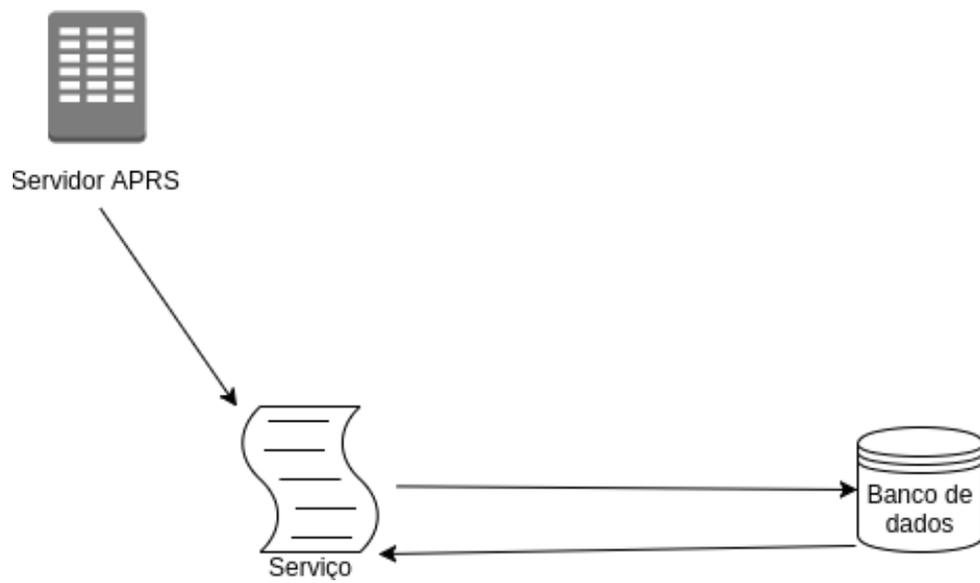


Figura 18 – Esquema do fluxo de dados para inserção de registro de localizações de veículos via APRS

```

PY3ADY-5>APDR14,TCPIP*,qAC,T2BRAZIL:=2706.785/05236.65Mf051/001/147.180MHz/A=002379 Rede Sul APRS
PU3MEM-10>APRX29,qAR,PV3NZ-10:12838.125I05306.02WPHG5160 /A=001496 Rede Sul APRS - Ibruba/RS
PU2YYP-2>APRS,PV2RJ-15,WIDE1*,qAR,PV2XH-15:12320.125/04633.18W#Rede Brasll APRS Matripora SP
PV5TDJ-5>RUS32R-13,WIDE1-1,WIDE2-2,qAR,PV5TD-1:1P31 [U]
PU3LSA-5>APDR14,TCPIP*,qAC,T2BRAZIL:=2910.225/05110.88WS171/003/A=002562 (54)991091532 146.400
PY4MAB-C>APDU61,TCPIP*,qAC,T2EDM:=2148.435/04634.01NS000/000 /A=000000 www.aprsbrasil.com
PV3YJU-10>APU25N,TCPIP*,qAC,BRASIL:=2117052PY3YJU VIA ISS 145.570 Local
PY4BBB-13>APRS,TCPIP*,qAC,T2BRAZIL:=021190622107.135/04254.90W_170/002g004t082r000p018P018h99b10155L000eCumulusFO
PU2AMA-7>OUT83W,PT20N-15,WIDE1*,WIDE2-2,qAR,PT2AP-1:'L^cL }>/'^Ac}13
PU2AMA-7>APT314,PT20N-15,WIDE1*,WIDE2-2,qAR,PT2AP-1->Aguilar - pu2ama@hotmail.com
PU3JVS-9>APDR13,TCPIP*,qAC,BRASIL:=3009.135/05108.88WS107/011/A=000123 adventure team
PY4RMT-APRS,WIDE1-1,WIDE2-1,qAR,PU4FLS-1:=1953.655/04350.86wr /REPETIDORA MORRO DO TATU/147.150 + 600 DPL732/GRI
PU1JFB-9>APDR14,TCPIP*,qAC,T2KA:=2253.795/04307.61MK083/012/A=-00003
PV3ADY-5>APDR14,TCPIP*,qAC,T2BRAZIL:=2706.785/05236.65Mf225/001/147.180MHz/A=002376 Rede Sul APRS
PU2KHZ>APAGM,TCPIP*,qAC,T2BRAZIL:>FLAVIO LIMEIRA SP
PV5KMR-15>APZBRT,WIDE1-1,WIDE2-1,qAR,PYSAGC-15:12431.225/05140.43W#PHG62604REDE SUL APRS(B03)
PU3DSS-10>APU25N,TCPIP*,qAC,BRASIL:=2921.505I05051.25W8IGATE GRAMADO RS
PU3DSS-13>APU25N,TCPIP*,qAS,PU3DSS-10:021185722921.505/05051.25W_181/000g00t072r000p000P000h0b10135L000ESTACAO GRAMADO - RS
PU3LSA-5>APDR14,TCPIP*,qAC,T2BRAZIL:=2910.215/05110.88WS352/002/A=002555 (54)991091532 146.400
PV3SB-5>APDR14,TCPIP*,qAC,T2IRELAND:=2946.875/05547.03WS115/002/A=000301 https://aprsdroid.org/
PU5RJ-5>APDR13,TCPIP*,qAC,T2HUN:=2513.065/05059.25W/083/002/146.520MHz/A=002639 Supervisor Substituto REER 2 CORPDEC - Santin - Rede ARCESUL -42999174180
PV5BPJ-15>APRS,WIDE1-1,WIDE2-1,qAR,PV5TD-1:12553.425/04857.43W# www.labre-pr.org.br//div PR/SC- M.Perdidos 1440mts Pot 45 Watts Antena HAAT 20mts,Gain 6dbt
PV5ME-13>APRS,TCPIP*,qAC,T2BRAZIL:=2516.755/04902.74W_270/006g007t079r000p000P000h05b10195Teste WX WRF Arduino
PV5MF-10>APDU12,TCPIP*,qAC,T2BRAZIL:=2444.245T05344 02WA
PR2MTN-15>QSP33V,PT20N-15*,WIDE3-2,qAR,PT2AP-1:LNW#APRS DIGI Barro Alto-CO :Use W1-1,W2-2
PU3MEM-10>APRX29,qAR,PV3NZ-10:12838.125I05306.02WPHG5160 /A=001496 Rede Sul APRS - Ibruba/RS
PU3JVS-9>APDR13,TCPIP*,qAC,BRASIL:=3009.135/05108.88WS110/000/A=000121 adventure team
PU3M1W-1>APU25N,TCPIP*,qAC,BRASIL:>2100162PU3M1W A. PRADO IGATE
PV3SB-5>APDR14,TCPIP*,qAC,T2IRELAND:=2946.875/05547.03WS328/001/A=000318 https://aprsdroid.org/
PV5TD-5>RUS08X,WIDE1-1,WIDE1-1,WIDE1-2,qAR,PV5FL-1:'M ;l 6u/
PU2LZL-9>RS347R,WIDE1-1,WIDE2-1,qAR,PV2PE-1:'JG^nI/"/<0>439.650MHz T123 -500 "
PV1IBB-5>APD001,TCPIP*,qAC,PV1IBB-GS:PV1IBB B *21190622251.615D04328.16NaRNG0001 440 Voice 431.00000MHz +0.0000MHz
PV5MNS>APRS,TCPIP*,qAC,T2BRAZIL:=021190722326.575/05156.13W_239/002g007t089r000p000P000h02b10136L440eCumulusFos
PV2FIT-6>APDR14,TCPIP*,qAC,T2HUN:=2330.915/04630.09W /A=002613 https://aprsdroid.org/
PV3MO-13>APU25N,TCPIP*,qAS,PV3MO-10:021190722922.835/05053.17W_015/000g005t072r000p001P001h51b10185PY3MO.com - Estacao Gramado RS
PU3LSA-5>APDR14,TCPIP*,qAC,T2BRAZIL:=2910.215/05110.88WS000/000/A=002552 (54)991091532 146.400
PU2AMA-7>OUT83W,PT20N-15,WIDE1,PP2PM-15*,WIDE2-1,qAR,PP2AR-1:'L^cL 3>/'^Ad}13
PV1RSU-5>APD001,TCPIP*,qAC,PV1RSU-GS:PV1RSU C *21191122256.995D04313.78WaRNG0000 Voice
PP5CFL-13>APRS,TCPIP*,qAC,BRASIL:=12733.825/04837.51W_000/000g000t077r000p000P000h43b10103ESP8266 Rede APRS Sul
PY4PCF-1>APU25N,TCPIP*,qAC,T2BRAZIL:=IY0KIC/x88 B--REDE ARUR DE APRS-- [UV32N]
PP5ERE-13>APRAS,TCPIP*,qAC,BRASIL:=2736.015/04841.11W_160/011g...E080...p...P...h59b10131APRS on Raspberry PI with DHT11
PU3JVS-9>APDR13,TCPIP*,qAC,BRASIL:=3009.135/05108.91WS263/004/A=000117 adventure team
PV2HY-15>APNU38,WIDE,qAR,PV2XH-15:12325.205/04642.00W#PHG4150 Rede VHF de APRS

```

Figura 19 – Log do serviço recebendo pacotes do servidor APRS.

Uma vez que a entrada de dado é referente a algum veículo cadastrado, fato que é verificado através do *callsign*, ele insere os dados de latitude e longitude no banco de dados.

O serviço de acesso ao servidor APRS foi escrito em *python*, utilizando a biblioteca 'aprslib', disponível no endereço "<https://github.com/rossengeorgiev/aprs-python/tree/master/aprslib>". No trecho de Código 4.1 exibe os comandos para conexão com o servidor APRS.

```

1 AIS = aprslib.IS("callsign", "senha", host="brazil.aprs2.net", port="14579"
)
2 aprs_names = [field.aprs_id for field in
3 list(db(s3db.vehicle_vehicle.aprs_id != None).select(s3db.vehicle_vehicle.
aprs_id))
4 AIS.connect()
5 AIS.consumer(callback, raw=False)

```

Código 4.1 – Acesso ao servidor APRS

Na primeira linha é instanciada a classe 'IS' que é abstração do servidor APRS. Como primeiro argumento tem-se o *callsign* do usuário, que é um identificador único do usuário. Logo depois, tem-se a senha do usuário, o endereço do servidor e a porta, respectivamente. Como dito anteriormente foi usado um servidor nacional já que os veículos de interesse estão todos em território nacional.

Logo depois tem-se uma *query* que retorna os códigos de identificação dos veículos. Esses códigos serão usados posteriormente para filtrar o fluxo de dados proveniente do servidor APRS.

Em seguida conecta-se de fato no servidor. Por último através do método '*consumer*' indica-se que todos os pacotes recebidos devem ser enviados como parâmetro ao método '*callback*'. O argumento '*raw*', por sua vez, quando verdadeiro indica que o pacote deve ser passado de forma bruta à função '*callback*', caso contrário será passado o resultado do método '*parse*'.

A seguir no Código 4.2, é demonstrado a função *callback*, que tem como incumbência receber os pacotes oriundos do servidor APRS e verificar se este é originário de um dos veículos de interesse.

```
1 def callback(packet):
2     print packet
3     parsed_packet = aprslib.parse(packet)
4     if parsed_packet['from'] in aprs_names:
5         create_gis_location(parsed_packet)
```

Código 4.2 – Função de *callback*

Este método tem uma implementação relativamente simples. A princípio ele apenas para fins de *feedback*, imprime na tela o pacote que recebeu como parâmetro, o que resulta no log visto na Figura 19.

Adiante faz-se o *parse* do pacote bruto, o que transforma os dados em um dicionário. Este dicionário é atribuído à variável '*parsed_packet*'.

Na linha seguinte é verificado se o *callsign* do pacote recebido pertence a um dos cadastrados. Caso o pacote seja oriundo de algum dos veículos cadastrados, ele é enviado à função '*create_gis_location*', que é apresentada no Código 4.3.

```
1 def create_gis_location(packet):
2     track_id = db(s3db.vehicle_vehicle.aprs_id == packet['from']).select(
3     join=db.sit_trackable.on((s3db.vehicle_vehicle.asset_id == s3db.
4     asset_asset.id)
5     & (s3db.asset_asset.track_id == s3db.sit_trackable.id))).first().
6     sit_trackable.id
7     location_id = s3db.gis_location.insert(lat=packet['latitude'], lon=
8     packet['longitude'])
```

```

6  print packet
7  presence_id = s3db.sit_presence.insert(timestamp = datetime.datetime .
    utcnow(), track_id = track_id, location_id = location_id)
8  s3db.commit()

```

Código 4.3 – Função de criação de registro de localização

Esta função tem como incumbência persistir os dados oriundos do sistema APRS no banco de dados. A primeira coisa que esta função faz é selecionar o id da tabela 'sit_trackable' que se relaciona com o veículo do pacote APRS.

Posteriormente cria-se um registro de localização e por último cria-se um registro da tabela 'sit_presence', unindo as tabelas 'sit_trackable' e 'gis_location'. Desta forma é criado um novo registro de localização para um veículo.

A medida que o serviço recebe os pacotes do servidor APRS ele cria os registros de localização tendendo assim a ter uma quantidade maior de registro do que se fosse feito manualmente. A Figura 20 mostra um exemplo de registros criados na plataforma *Sahana Eden*.

The screenshot shows the 'GPS Data' tab in the Sahana Eden interface. It displays a table with 3 entries. The table has columns for 'data/hora', 'Localização', 'Endereço', 'Speed', and 'Accuracy'. Each row has an 'Editar' button. The data shown is as follows:

	data/hora	Localização	Endereço	Speed	Accuracy
Editar	22/03/2018 20:27	23.609000 S, 46.611000 W	-	-	-
Editar	22/03/2018 20:29	23.609000 S, 46.611000 W	-	-	-
Editar	22/03/2018 20:33	23.609000 S, 46.611000 W	-	-	-

Figura 20 – Exemplo de registros de localização na plataforma *Sahana Eden*.

Esta seção expôs os passos para atribuir a funcionalidade de criação automática de registros de localização e por consequência o rastreamento de veículos via sistema APRS. Entretanto é necessário configurar a plataforma *Sahana Eden* para que exiba a posição atual (último registro de localização) do veículo.

É possível através de arquivos 'csv' criar uma pré-carga no banco de dados do *Sahana Eden*, estes arquivos contém dados de configuração, bem como outros dados que serão carregados ao primeiro acesso à plataforma.

Para que seja exibido no mapa a localização de cada veículo, foi adicionado ao arquivo 'gis_layer_feature.csv' uma linha de configurações referente aos veículos. A tabela 3 mostra os atributos mais importantes para e os valores atribuídos, bem como a descrição do que é cada atributo.

Atributo	Valor	Descrição
Name	Vehicles	Identificação do recurso que será exibido.
Controller	vehicle	<i>Controller</i> que será onde encontra-se a função que será chamada.
Function	vehicle	Função quando é carregado o mapa e que retorna a localização do recurso em questão.
Popup Format	{number} (T('Vehicle'))	Conteúdo que será exibido em um <i>popup</i> ao parar o mouse sobre o ícone do recurso

Tabela 3 – Configuração de veículos para exibição no mapa.

Como explicado na Tabela 3, assim que carregado o mapa é feita uma chamada assíncrona para a função e *controller* cadastrados. Essa função responde esta chamada em formato geojson, contendo a localização do recurso. Após todo o exposto a plataforma *Sahana Eden* torna-se apta a exibir no mapa a localização dos veículos, conforme mostrado na Figura 21.

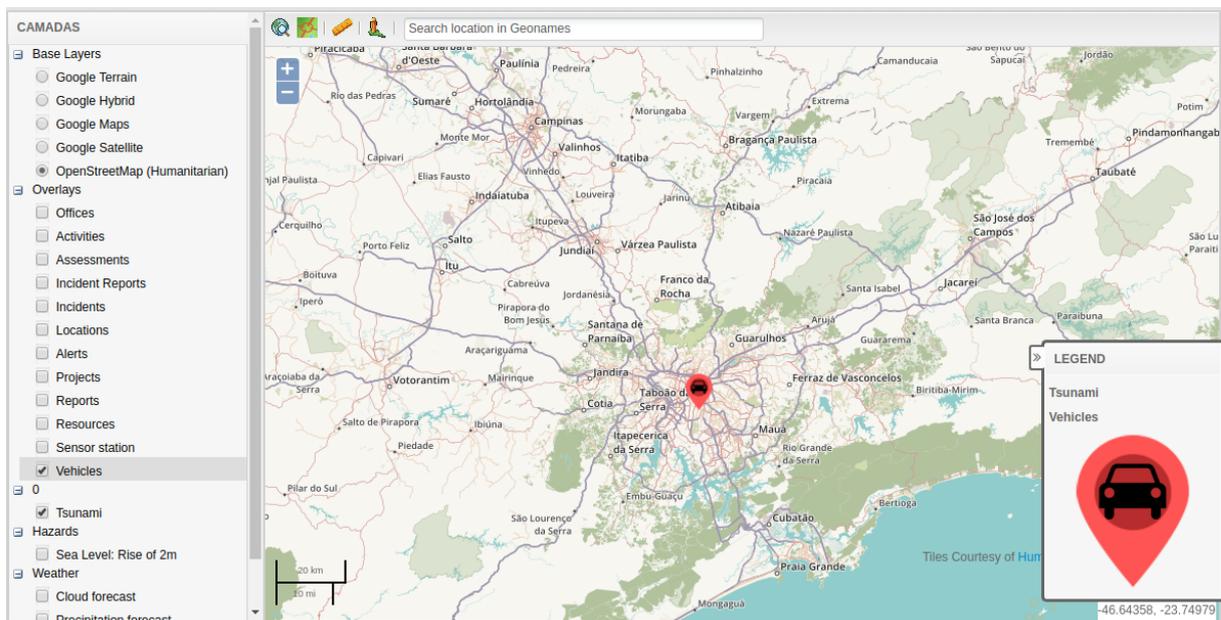


Figura 21 – Exemplo de veículo sendo mostrado no mapa.

4.2 Rastreamento de sensores

Em meio a catástrofes ou mesmo como forma de prevenção, é de suma importância que o tomador de decisão tenha a sua disposição informações que o permita tomar decisões de maneira mais eficientes e assertivas possíveis.

Velocidade do vento, temperatura, chuva e etc, estão dentre as informações mais importantes para a gestão de desastres. Ter este tipo de informação atrelados as outras informações já existente na plataforma *Sahana Eden*, pode ampliar de forma significativa o poder de reação as adversidades.

A plataforma *Sahana Eden* funciona totalmente de forma configurável e modularizada, permitindo habilitar e desabilitar módulos, conforme a necessidade. Diante disto criou-se um novo módulo que possibilita a gestão de sensores na plataforma. Além de criar todo arcabouço para o cadastro e leitura de sensores, também produziu-se a integração com o sistema APRS, para que aquisição de dados dos sensores fossem feita de forma automática.

As subseções a seguir tem como objetivo demonstrar os principais pontos da criação do módulo da gestão de sensores, bem como a integração com o sistema APRS.

4.2.1 Estrutura do banco de dados

Nesta subseção é apresentada a estrutura do banco de dados referentes as tabelas e suas relações, que foram utilizadas para dar suporte ao cadastro dos sensores bem como os valores de leitura destes. Os dois principais arquivos dos módulo são o *model* e o *controller* a seguir são apresentados os principais pontos acerca do *model* e o banco de dados.

Este módulo foi concebido de forma a permitir o cadastro de estações de sensores, que é caracterizado por ser um conjunto de sensores. O Código 4.4 mostra a código do arquivo *model* para criação da tabela de estação de sensores.

```
1      tablename = "sensor_sensor_station"
2          self.define_table(tablename ,
3              Field("description" ,
4                  notnull=True ,
5                  label = T("Description") ,
6              ) ,
7              Field("aprs_id" ,
8                  notnull=False ,
9                  label = T("APRS ID") ,
10             ) ,
11             location_id(
12                 widget = S3LocationSelector(show_address = False ,
13                 show_postcode = False ,
14                 show_latlon = True ,
15             ) ,
```

Código 4.4 – Criação da tabela de estação de sensores

O código mostra a definição da tabela "sensor_sensor_station". Por convenção do *Sahana Eden* o nome das tabelas é iniciado pelo nome do módulo a qual pertence. Como pode ser visto no código a tabela "sensor_sensor_station" possui o campo 'description', que é responsável por armazenar uma breve descrição acerca daquela estação de sensores.

Um outro campo que compõe a tabela "sensor_sensor_station" é o "aprs_id", que armazena o *callsign* que identifica aquela estação no sistema APRS. Este campo é opcional, uma vez que a utilização de sensores poderá funcionar independente do sistema APRS.

E por último tem-se a função 'location_id' que cria uma referência para a tabela 'gis_location'. Um dos parâmetros desta função é o *widget* que recebe uma instância de 'S3LocationSelector', isso define que o que o cadastro da localização da estação de sensores usará um *widget* que mostra um mapa que o usuário pode clicar em um determinado local para marcar a localização.

Para uma melhor abstração da estrutura de banco de dados acerca da tabela "sensor_sensor_station", pode-se ver a representação gráfica da tabela na Figura 22.

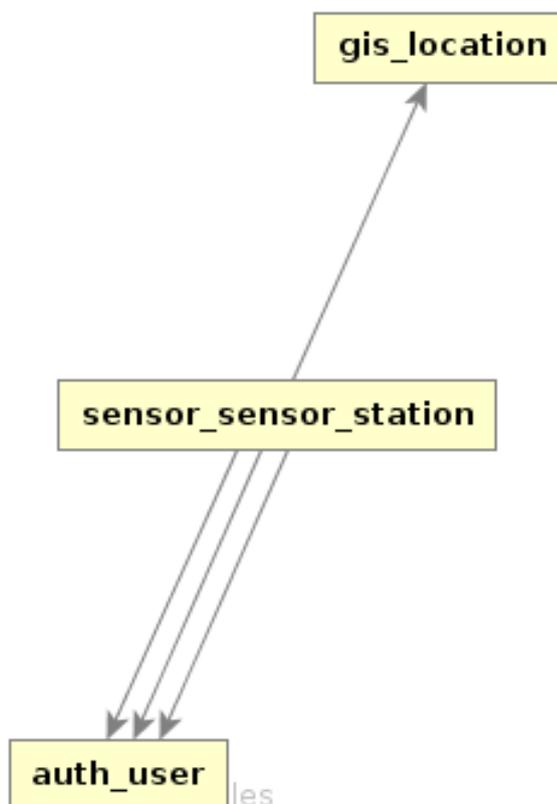


Figura 22 – Representação da tabela 'sensor_sensor_station'

Como foi dito e pode ser visto na figura 'sensor_sensor_station' relaciona-se com a tabela 'gis_location'.

Como dito anteriormente as estações de sensores, são um conjunto de vários sensores. Cada sensor deste é responsável por retornar a leitura de alguma grandeza. Além da exibição das últimas leituras dos sensores da estação, também é interessante que se mantenha uma base histórica da leituras dos sensores.

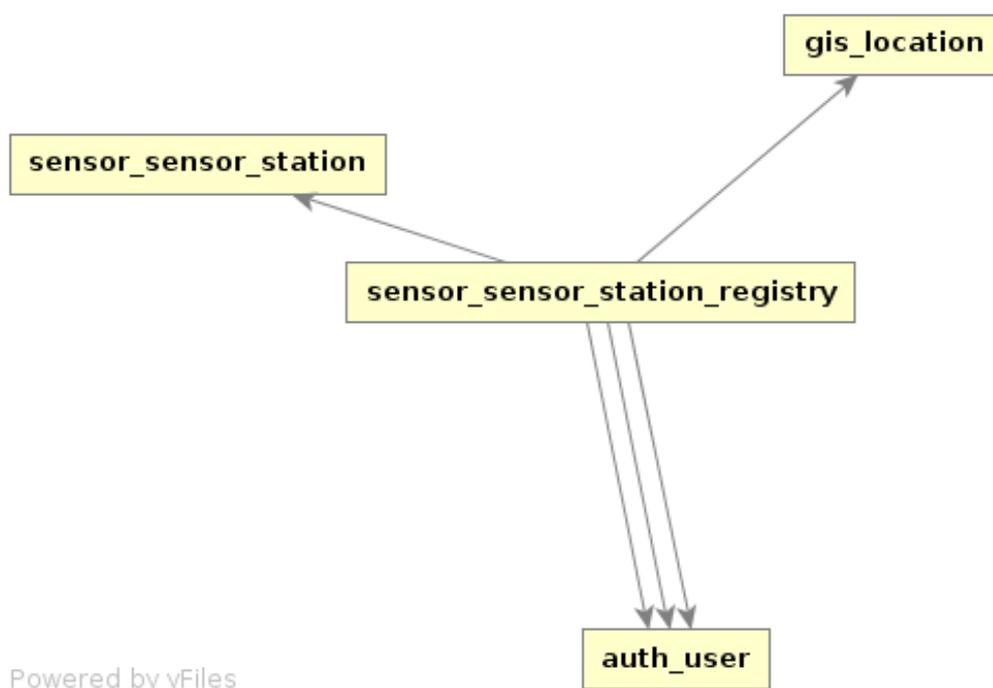
Para o armazenamento série histórica da leitura dos sensores, foi criada a tabela 'sensor_sensor_station_registry'. O Código 4.5, mostra código que dá origem a tabela "sensor_sensor_station_re

```
1  tablename = "sensor_sensor_station_registry"  
2      self.define_table(tablename ,  
3      sensor_sensor_station_id() ,  
4      location_id(  
5      widget = S3LocationSelector(show_address = False ,  
6      show_postcode = False ,  
7      show_latlon = True ,  
8      )))
```

Código 4.5 – Criação da tabela de registro de leitura da estação de sensores

Este código cria uma tabela com o nome 'sensor_sensor_station_registry'. Tal tabela apresenta uma referência para a tabela 'sensor_sensor_station', na qual pertence o registro de leitura. Além da referência para a tabela 'sensor_sensor_station', também existe uma referência para tabela 'gis_location'.

A Figura 23 exibe de maneira gráfica a representação da tabela 'sensor_sensor_station_registry' e seus relacionamentos.



Powered by yFiles

Figura 23 – Representação da tabela 'sensor_sensor_station_registry'

A figura mostra as relações da tabela "sensor_sensor_station_registry" e suas relações. A tabela "sensor_sensor_station_registry" tem uma chave estrangeira para a tabela "sensor_sensor_station", ou seja a estação de sensores da qual o registro de leitura pertence.

Um outro ponto importante a ser ressaltado é a chave estrangeira que a tabela "sensor_sensor_station_registry" possui para a tabela "gis_location". Esta relação confere ao registro de estações de sensores uma característica interessante, uma vez que isto permite o suporte por parte da plataforma *Sahana Eden* estações de sensores móveis.

Por último temos a criação da tabela "sensor_property". Esta tabela tem por função armazenar a grandeza e o valor coletado. O código 4.6, mostra o código de criação da tabela "sensor_property".

```
1      tablename = "sensor_property"
2      self.define_table(tablename,
3                          sensor_sensor_station_registry_id(),
4                          Field("name",
5                              label = T("Name"),
6                          ),
7                          Field("value", "double",
8                              label = T("Value"),
9                          ),
10     )
```

Código 4.6 – Criação da tabela registro da leitura dos sensores

Este código cria uma tabela com o nome 'sensor_property'. Tal tabela apresenta uma referência para a tabela 'sensor_sensor_station_registry', na qual pertence o registro de leitura do sensor em questão. Além da referência para a tabela 'sensor_sensor_station_registry', também foi criado o campo 'name', que registra o nome da grandeza, tal como: Temperatura, velocidade do vento e etc. Além da grandeza há o registro do valor lido para tal grandeza.

A Figura 24 exibe de maneira gráfica a representação da tabela 'sensor_sensor_property' e seus relacionamentos.

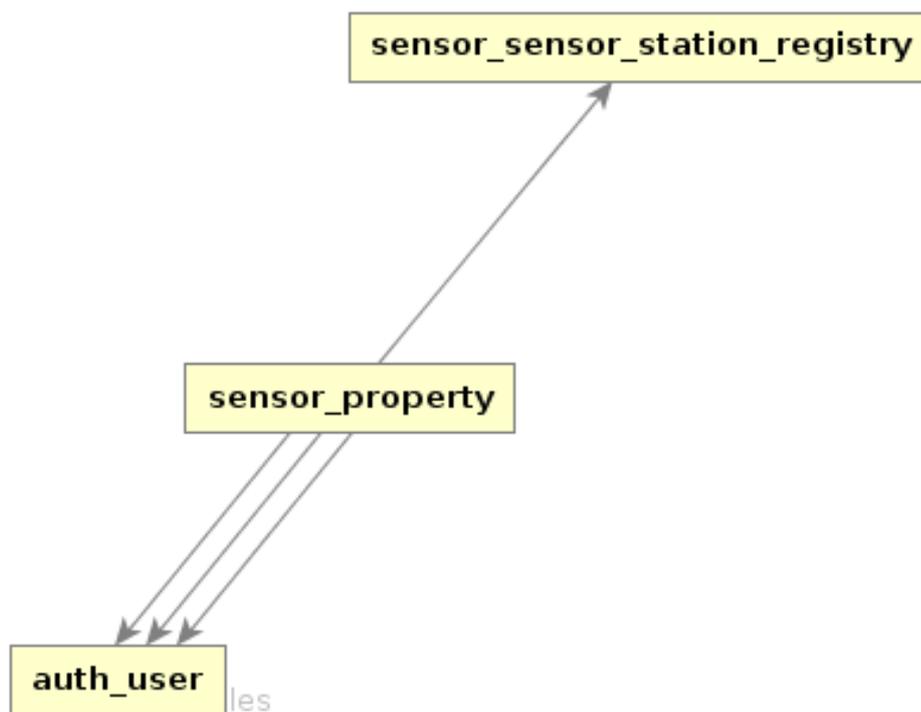


Figura 24 – Representação da tabela 'sensor_sensor_property'

Como pode ser visto a tabela "sensor_property" possui uma chave estrangeira para a tabela "sensor_sensor_station_registry", que indica o registro de leitura da grandeza aferida. A tabela "sensor_property" foi criada para conferir uma maior flexibilidade na inserção de dados de sensores na plataforma, uma vez que permite a o registro de dados de forma irrestrita.

4.2.2 Aquisição de dados do sistema APRS

Uma vez explanado a criação e a estrutura do banco de dados que sustenta o cadastro de estações de sensores, chega-se ao passo seguinte que é a aquisição automática de dados oriundos do sistema APRS.

Essa subseção tem como objetivo elucidar como deu-se o processo de criação do mecanismo que permitiu o recolhimento dos dados dos servidores do APRS, acerca dos sensores e persistir tais dados na base do sistema *Sahana Eden*. Estes dados são basicamente dados provenientes de estações meteorológicas, mas como citado anteriormente a modelagem dos dados permite a inserção de dados das mais variadas formas e origem.

Após a criação das tabelas que armazenam as informações a cerca das estações de sensores, foram criados os artifícios para a aquisição de dados dos servidores APRS e inserção na plataforma *Sahana Eden*.

A aquisição de dados dos servidores APRS, foi feita da mesma forma da descrita para o rastreamento de veículos. Ou seja tem-se um serviço que funciona paralelamente à plataforma *Sahana Eden*. Onde este serviço acessa o banco de dados da aplicação para recolher os *callsigns* das estações de sensores cadastrados para que através destes *callsigns* filtrar os dados provenientes do servidor APRS.

Foi adicionado ao serviço de acesso aos servidores APRS uma estrutura de controle que utiliza o atributo *'weather'* do pacote APRS, para identificar se o dado pacote originou-se de uma estação meteorológica. Uma vez que o pacote possua o atributo *'weather'*, os dados do pacote APRS são persistidos no banco de dados.

Além da dinâmica da integração com o sistema APRS, será exposto alguns pontos do processo de criação do módulo de sensores, além do já apresentado na subseção 4.2.1.

A dinâmica entre a aquisição dos dados e a persistência destes, permanece a mesma ilustrada na Figura 18. Ou seja tem-se um serviço que funciona em paralelo com a plataforma *Sahana Eden*. Onde este serviço acessa o banco de dados da aplicação para recolher os *callsigns*, tanto dos veículos como das estações de sensores cadastrados para que através destes *callsigns* filtrar os dados provenientes do servidor APRS.

Nesta etapa foi adicionado ao serviço de acesso ao sistema APRS, o recolhimento dos *callsigns* cadastrados para as estações de sensores. Portanto foi adicionado a *query* de aquisição dos *callsigns* dos veículos, a obtenção dos *callsigns* das estações de sensores, conforme demonstrado no Código 4.7

```
1 AIS = aprslib.IS("callsign", "senha", host="brazil.aprs2.net", port="14579")
2 aprs_names = [field.aprs_id for field in
3 list(db(s3db.vehicle_vehicle.aprs_id != None).select(s3db.vehicle_vehicle.aprs_id)) + list(db(s3db.sensor_sensor_station.aprs_id != None).select(s3db.sensor_sensor_station.aprs_id)) ]
4 AIS.connect()
5 AIS.consumer(callback, raw=True)
```

Código 4.7 – Inicialização do serviço de acesso ao sistema APRS com a aquisição dos *callsigns* das estações de sensores.

Como pode ser visto o código de inicialização do serviço continua praticamente o mesmo, com exceção das linha dois e três, onde agora é concatenado com a lista de *callsigns* dos veículos, os *callsigns* oriundos das estações de sensores.

Uma vez feita atribuídos a variável 'aprs_names', os *callsigns* dos veículos e estações de sensores, o serviço passa a monitorar o fluxo de pacotes no servidor APRS. A cada pacote que é recebido pelo serviço, este pacote é repassado a função 'callback'. Conforme demonstrado no Código 4.2, uma vez que a função recebe um pacote, esta verifica se o mesmo possui como *callsign* um dos que constam na variável 'aprs_names'. Caso o pacote pertença a um dos veículos ou estações de sensores cadastradas, este é passado a função 'create_gis_location'.

A função 'create_gis_location', vista no Código 4.3 foi incrementada de forma a suportar a criação de registros de leitura das estações de sensores. A função é demonstrada no Código 4.8.

```

1 def create_gis_location(packet):
2     if packet.has_key('weather'):
3         sensor_station = db(s3db.sensor_sensor_station.aprs_id == packet['
4         from']).select().first()
5         sensor_station_location = db(s3db.gis_location.id == sensor_station
6         .location_id).select().first()
7         sensor_station_location.update_record(lat=packet['latitude'], lon=
8         packet['longitude'])
9         location = s3db.gis_location.insert(lat=packet['latitude'], lon=
10        packet['longitude'])
11        registry = s3db.sensor_sensor_station_registry.insert(location_id =
12        location.id,
13        sensor_sensor_station_id = sensor_station.id)
14        for name, value in packet['weather'].items():
15            s3db.sensor_property.insert(sensor_sensor_station_registry_id=
16            registry.id, name=name, value=value)
17        else:
18            track_id = db(s3db.vehicle_vehicle.aprs_id == packet['from']).
19            select(
20            join=db.sit_trackable.on((s3db.vehicle_vehicle.asset_id == s3db.
21            asset_asset.id)
22            & (s3db.asset_asset.track_id == s3db.sit_trackable.id))).first().
23            sit_trackable.id
24            location_id = s3db.gis_location.insert(lat=packet['latitude'], lon=
25            packet['longitude'])
26            print packet
27            prsence_id = s3db.sit_presence.insert(timestamp = datetime.datetime.
28            utcnow(), track_id = track_id, location_id = location_id)
29            s3db.commit()

```

Código 4.8 – Função 'create_gis_location' adequada a criação de registro das estações de sensores.

A função 'create_gis_location' preservou o código que cria os registros de localização dos veículos, ao passo que adicionou uma estrutura de controle que utiliza o atributo 'weather' do pacote APRS, para identificar se o dado pacote originou-se de uma estação meteorológica. Caso o pacote não possua o atributo 'weather' ele é considerado como originário de um veículo e segue o fluxo descrito na seção 3.1.

Uma vez que o pacote possua o atributo 'weather', inicia-se o processo de persistência dos dados provenientes da estação meteorológica. A princípio seleciona-se a estação através do atributo 'from', que é o *callsign* do pacote APRS.

Uma vez selecionada a estação de sensores é atualizada sua localização, onde é substituída as informações referentes a latitude e longitude pelas informações contidas no pacote APRS. Esta informação será utilizada para o georreferenciamento da estação de sensores no mapa.

Logo após são criados registros da tabelas "sensor_sensor_station_registry" e "gis_location", que são relacionadas. Logo após é criado um registro da tabela "sensor_property", para cada item do atributo 'weather' do pacote APRS. Na criação dos registros da tabela "sensor_property" é passado como parâmetro o registro da tabela "sensor_sensor_station_registry", anteriormente criado.

Com isto conclui-se o processo de aquisição de dados dos servidores APRS e persistência destes, no que refere-se a estações de sensores. Porém alguns outros passos foram necessários até que as estações de sensores pudessem ser exibidas no mapa da plataforma *Sahana Eden*, bem como as informações de leitura desta estação de sensores.

Como exposto anteriormente é necessário configurar a plataforma *Sahana Eden* para que exiba a posição atual da estação de sensores, bem como os valores das grandezas lidas presentes no último registro.

É possível através de arquivos 'csv' criar uma pré-carga no banco de dados do *Sahana Eden*, estes arquivos contém dados de configuração, bem como outros dados que serão carregados durante o primeiro acesso à plataforma.

Para que seja exibido no mapa a localização de cada estação de sensores, bem como os valores das leituras dos sensores, foi adicionado ao arquivo 'gis_layer_feature.csv' uma linha de configurações referente as estações de sensores. A tabela 4 mostra os atributos mais importantes para e os valores atribuídos, como também a descrição do que cada atributo representa.

Como explanado na Tabela 3, assim que carregado o mapa é feita uma chamada assíncrona para a função e *controller* cadastrados. Essa função responde esta chamada em formato

Atributo	Valor	Descrição
Name	Sensor station	Identificação do recurso que será exibido.
Controller	sensor	<i>Controller</i> que será onde encontra-se a função que será chamada.
Function	sensor_station_gis	Função quando é carregado o mapa e que retorna a localização do recurso em questão.
Popup Format	<h4>T('Sensor Station') </h2> <p>description</p>	Conteúdo que será exibido em um <i>popup</i> ao parar o mouse sobre o ícone do recurso

Tabela 4 – Configuração de estações de sensores para exibição no mapa.

geojson, contendo a localização do recurso. Após todo o exposto a plataforma *Sahana Eden* torna-se apta a exibir no mapa a localização das estações de sensores, conforme mostrado na Figura 25.

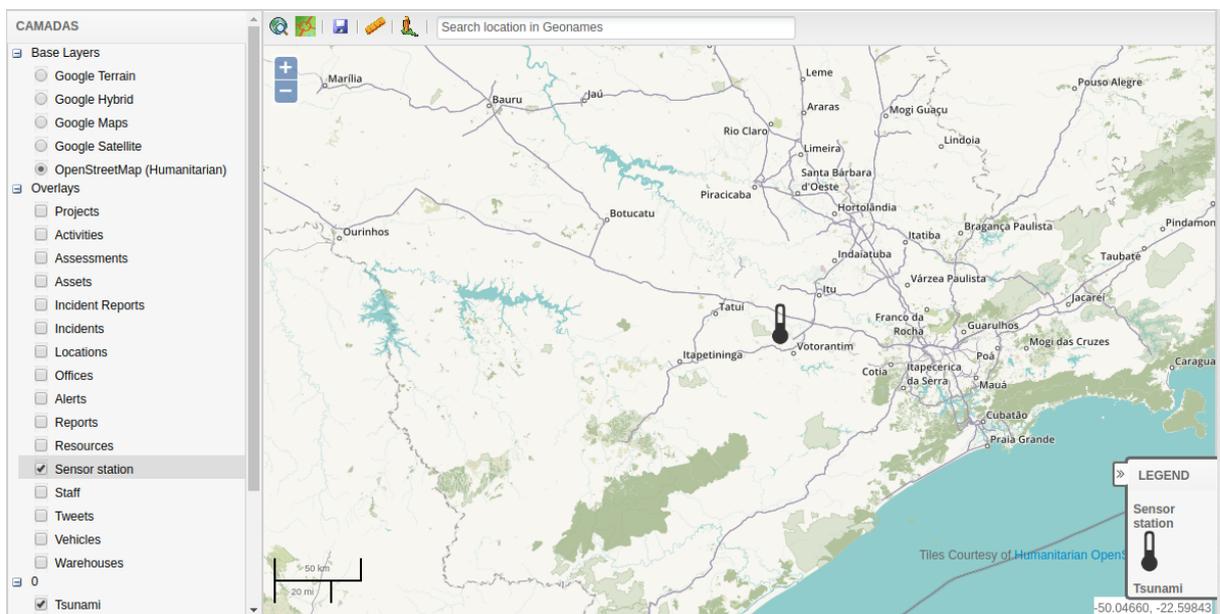


Figura 25 – Exemplo de estação de sensores sendo representada no mapa

Como pode ser visto o mapa exibe a localização da estação de sensores, todavia tão importante quanto a localização da estação de sensores, é a exibição dos dados referentes a leitura dos sensores. Para tal é necessário implementar uma resposta a uma chamada que é feita quando o ícone da estação de sensores é clicado. O Código 4.9 mostra o principal trecho onde é implementado uma função que será chamada ao clicar no ícone no mapa, retornando um trecho html que será renderizado em um *popup*.

```

1 if r.representation == "plain":
2     last_registry = r.record.sensor_sensor_station_registry.select().last()

```

```

3     property_table = TABLE()
4     for sensor_property in last_registry.sensor_property.select():
5         property_table.append(TR(TD(sensor_property.name), TD(
6             sensor_property.value)))
7     output['item'].append(property_table)
8 return output

```

Código 4.9 – Resposta para exibição das informações da leitura dos sensores no mapa.

Uma vez que o ícone da estação de sensores é clicado faz-se uma chamada assíncrona para o *controller* configurado. Esta chamada tem como parâmetro *representation* o valor "plain". Após identificar-se que o *request* têm como como parâmetro *representation* o valor "plain", ou seja trata-se de um chamado oriundo do mapa para formação do *popup* com as informações. A princípio é carregado o último registro da estação em questão. Logo após é criado uma tabela que têm em cada linha duas células, contendo o nome da o nome da propriedade e seu valor respectivamente. Após o retorno do *controller* é exibido um *popup* contendo as informações de leitura dos sensores, conforme ilustra a Figura 26.

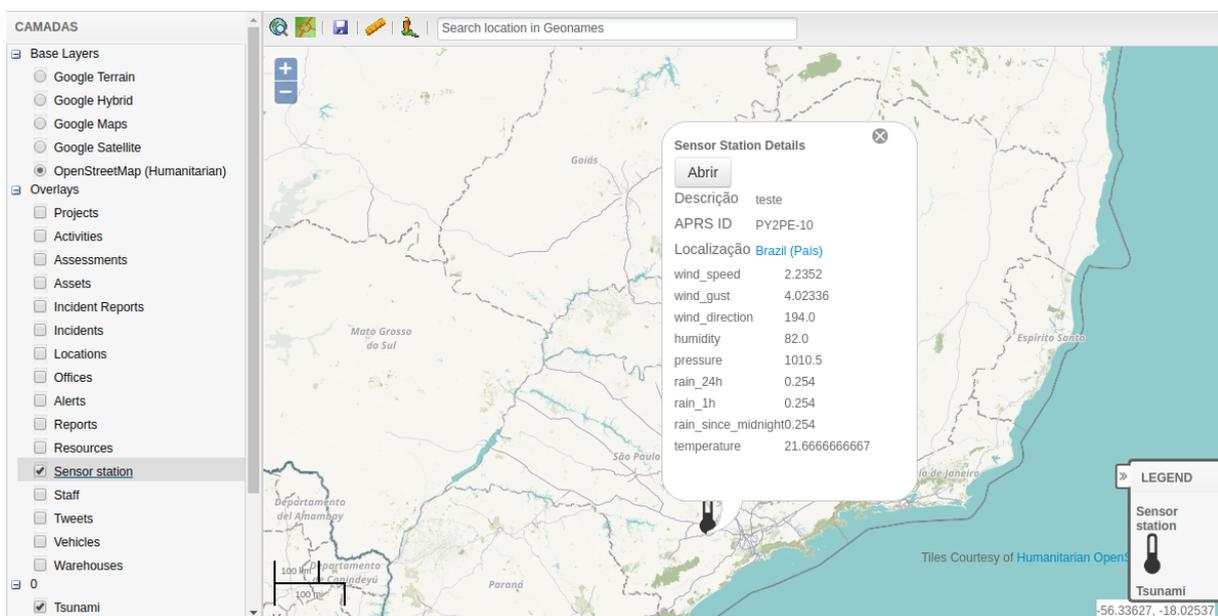


Figura 26 – Exemplo de *popup* com informações dos sensores.

Como pode ser visto é exibido um *popup* que permite a visualização dos dados do último

registro de leitura. Com isto fecha-se o ciclo metodológico para alcançar o resultado na qual este trabalho propõe-se.

Finalizada a criação da funcionalidade de rastreamento dos sensores o sistema foi colocado em produção e criado um *script* para que fosse feito a inserção de dados reais provenientes dos planos de contingência do município de Santa Maria Madalena do estado do Rio de Janeiro.

4.3 Instalação em produção e inserção de dados

A fim de verificar como o sistema *Sahana Eden* se comportava em um ambiente de produção, foi feito o *deploy* do mesmo em servidor com especificações conforme a seguir:

- 2 núcleos de processamento
- 2GB de memória RAM
- 20GB de HD
- Debian 8.7

Os mantenedores do projeto disponibilizam no endereço scripts para realizar o *deploy* do sistema em um ambiente de produção de maneira mais fácil. Para este projeto foi adotado o set de configurações que utilizam o servidor web 'Apache' e o sistema gerenciador de banco de dados 'MYSQL'.

Feito a instalação, com o sistema apto a ser utilizado, foi realizada a entrada de dados de ocorrências de incidentes georreferenciados. Para tal foi utilizados dados reais provenientes da defesa civil do município de Santa maria Madalena situado no estado do Rio de Janeiro. Os dados utilizados foram retirados dos planos de contingência de proteção e defesa civil. Tais documentos contém pontos georreferenciados de alguns tipos de ocorrências que são de total importância para tomar decisões em meio a um desastre, tais como: pontos de deslizamentos, enxurradas, inundações e etc.

O *Sahana Eden* possui a funcionalidade de importar arquivos do tipo 'kml', contendo os pontos de interesse. Diante disto com o intuito de agilizar a entrada de dados foi criado um *script*, pode ser conferido no Código 4.10, o *script* gera um arquivo do tipo 'kml' a partir dos dados extraídos dos planos de contingência de proteção e defesa civil.

```
1 import re
2 import utm
3 import simplekml
4
5 def normalize_coordinate(coordinate):
6     return int(str(int(coordinate))[:6])
```

```

7
8 def names(text):
9     return re.findall('Endere o (.*) CEP', text, re.DOTALL)
10
11 def neighborhood(text):
12     return re.findall(r'Bairro (.*) Criticidade', text, re.DOTALL)
13
14 def point_of_reference(text):
15     return re.findall(r'Ponto de Refer ncia (.*) Criticidade', text, re.
16     DOTALL)
17
18 def description(name, point_of_reference):
19     return "Endere o: " + name + " <br/> " + "Ponto de Refer ncia: " +
20     point_of_reference
21
22 def coordinates(text):
23     c1 = re.findall(r'UTM(.*) E /', text, re.DOTALL)
24     c2 = re.findall(r' E /(.*) S ', text, re.DOTALL)
25     c3 = re.findall(r' Fuso (.*)K UTM', text, re.DOTALL)
26     return zip(c1, c2, c3)
27
28 def kml_generate(name, icon_url='http://maps.google.com/mapfiles/kml/shapes
29 /falling_rocks.png'):
30     text = open(name + '.txt').readlines()[0]
31     latlong = []
32     for coordinate in coordinates(text):
33         try:
34             latlong.append(utm.to_latlon(normalize_coordinate(int(
35             coordinate[0])), int(coordinate[1]), int(coordinate[2]), 'K'))
36         except Exception as e:
37             latlong.append(utm.to_latlon(normalize_coordinate(coordinate
38             [0]), int(coordinate[1]), int(coordinate[2]), 'K'))
39     kml=simplekml.Kml()
40     names_in_text = names(text)
41     descriptions = [description(data[0], data[1]) for data in zip(names(
42     text), point_of_reference(text))]
43     for i, coordinate in enumerate(latlong):
44         print(i, coordinate)
45         pnt = kml.newpoint(name=names_in_text[i], description = descriptions[
46         i], coords=[(str(coordinate[1]), str(coordinate[0]))])
47         pnt.style.iconstyle.icon.href = icon_url
48     kml.save(name + '_santa_maria_madalena.kml')
49 kml_generate('rolamentos', 'http://maps.google.com/mapfiles/kml/shapes /
50 falling_rocks.png')
51 kml_generate('deslizamentos', 'http://maps.google.com/mapfiles/kml/shapes /
52 target.png')
53 kml_generate('incendios_florestais', 'http://maps.google.com/mapfiles/kml/

```

```

shapes/firedept.png')
45 kml_generate('enxurradas', 'http://maps.google.com/mapfiles/ms/micons/
waterfalls.png')
46 kml_generate('inundacoes', 'http://maps.google.com/mapfiles/kml/shapes/
water.png')

```

Código 4.10 – Script para geração de de kml contendo pontos de interesse

O *script* proposto, quando executado criará os arquivos 'kml' com os pontos de interesse de acordo com os planos de contingência de proteção e defesa civil. Vale ressaltar que este *script* só funciona com o formato de dados disposto nos o planos de contingência de proteção e defesa civil de Santa Maria Madalena, porém ele pode ser adaptado para outros formatos de apresentação dados.

Após a geração dos arquivos 'kml' os mesmos foram importados para a plataforma *Sahana Eden*, de tal modo que os pontos de interesse sejam plotados no mapa, conforme a figura 27.

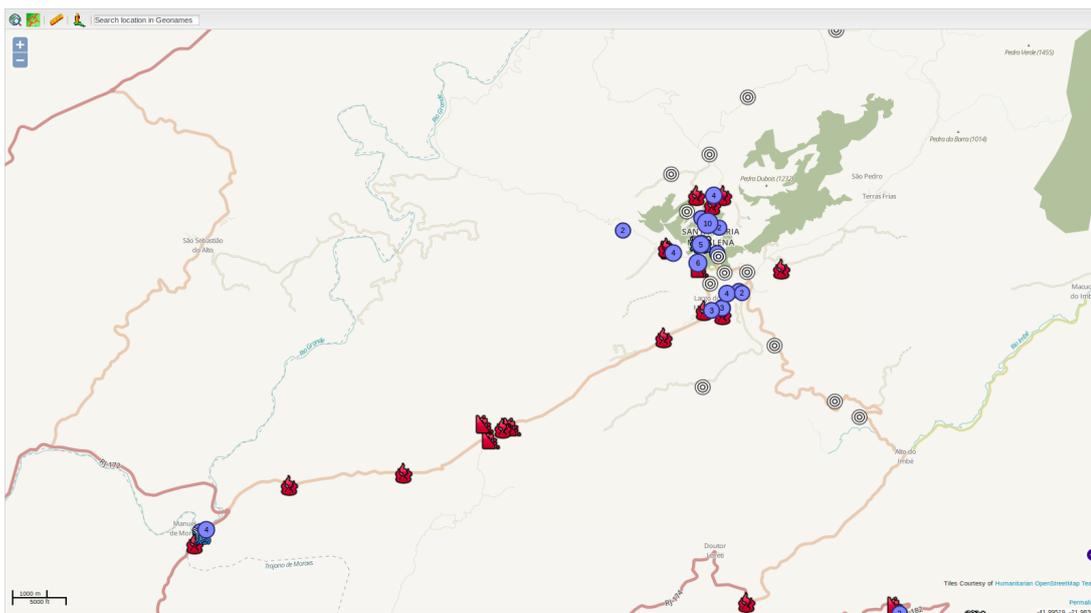


Figura 27 – Pontos de interesse mostrados no mapa do *Sahana Eden*

A informação com os pontos mais vulneráveis historicamente ou mais propícios a ocorrência de um determinado evento é de total importância para a tomada de decisão dos responsáveis pela coordenação da situação. Além disso o *script* 4.10 insere informações extras, com a descrição do ponto de interesse. Estas informações são também extraídos do plano de contingência de Santa Maria Madalena. As informações são inseridas no arquivo 'kml' que podem ser vistas clicando no ponto de interesse conforme ilustrado na figura 28.

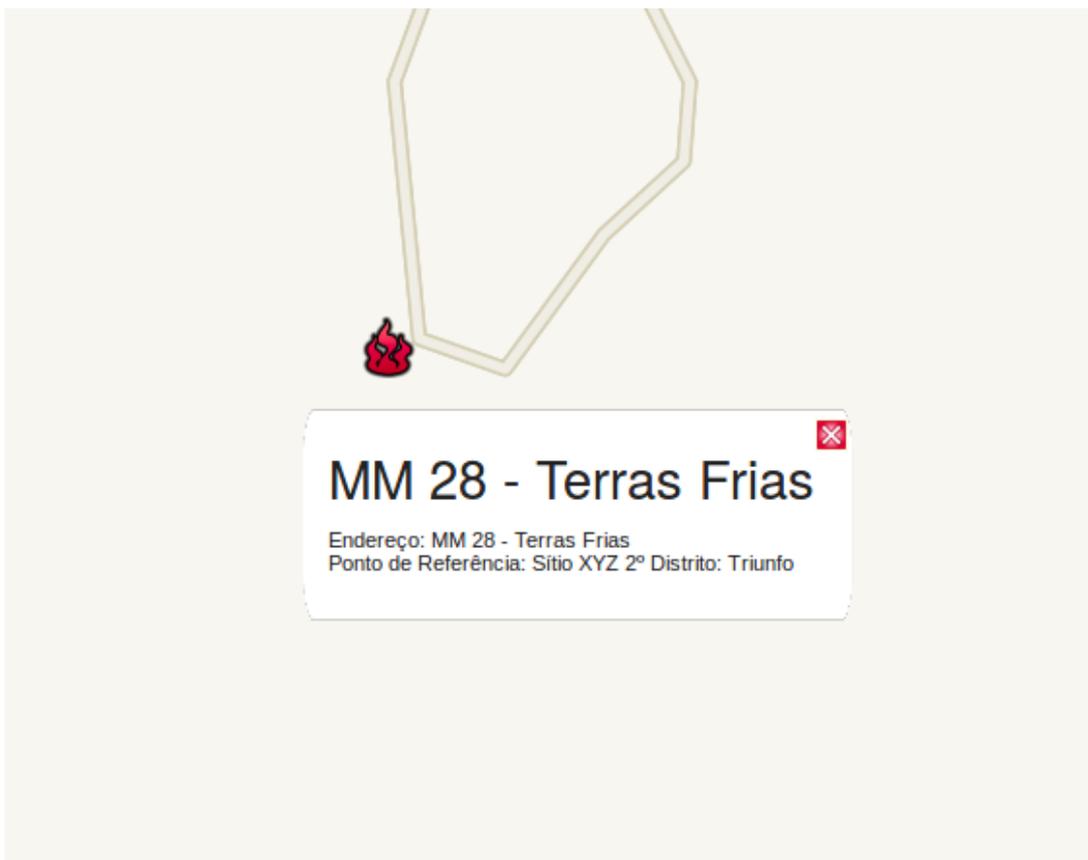


Figura 28 – Detalhe de um ponto de interesse mostrado no mapa do *Sahana Eden*

4.4 Estudo de caso

A utilização do sistema APRS para rastreamento de unidades móveis e estações de sensores, tende a gerar um ganho significativo na aporte de dados a plataforma *Sahana Eden*, favorecendo a articulação de respostas por parte do tomador decisão em meio a desastres, bem como forma de prevenção. Contudo faz-se necessário analisar na prática o uso dos dados provenientes do sistema APRS.

Esta seção apresentará de forma exemplificada possíveis utilizações das novas funcionalidades geradas a partir deste trabalho. A seção será dividida em duas subseções, que tratarão acerca dos rastreamento de unidades móveis e rastreamento de estações de sensores respectivamente.

4.4.1 Rastreamento de unidades móveis

Tratando-se de gerência de desastres a ciência por parte do tomador de decisão de onde estão localizados seus recursos móveis é de total importância para coordenar os atendimentos a população afetada de forma eficiente. A disponibilização das informações de localização de unidades móveis atreladas a outras informações, possibilita ao tomador de decisão uma leitura mais ampla da situação tendendo a gerar respostas mais eficientes nas decisões tomadas.

Esta subsecção apresentará algumas possibilidades geradas pela interação entre as informações que são possivelmente cadastradas na plataforma *Sahana Eden* e a aquisição de dados do sistema APRS, para rastreamento de unidades móveis.

Como citado anteriormente a plataforma *Sahana Eden* possui uma funcionalidade de mapeamento que permite que todos os dados baseados em geolocalização sejam condensados e visualizados em um mapa. Os mapas São uma excelente ferramenta para fornecer uma consciência situacional, que é essencial quando se planeja preparar-se para uma catástrofe.

Dentre várias características a plataforma *Sahana Eden* possui o registro de incidente. Assim que é feito o cadastro por algum agente este passa a ser visualizado no mapa. Permitindo que o tomador de decisões tome alguma medida para resolver ou mitigar o problema. A Figura 29 exemplifica o registro de um incidente no mapa.

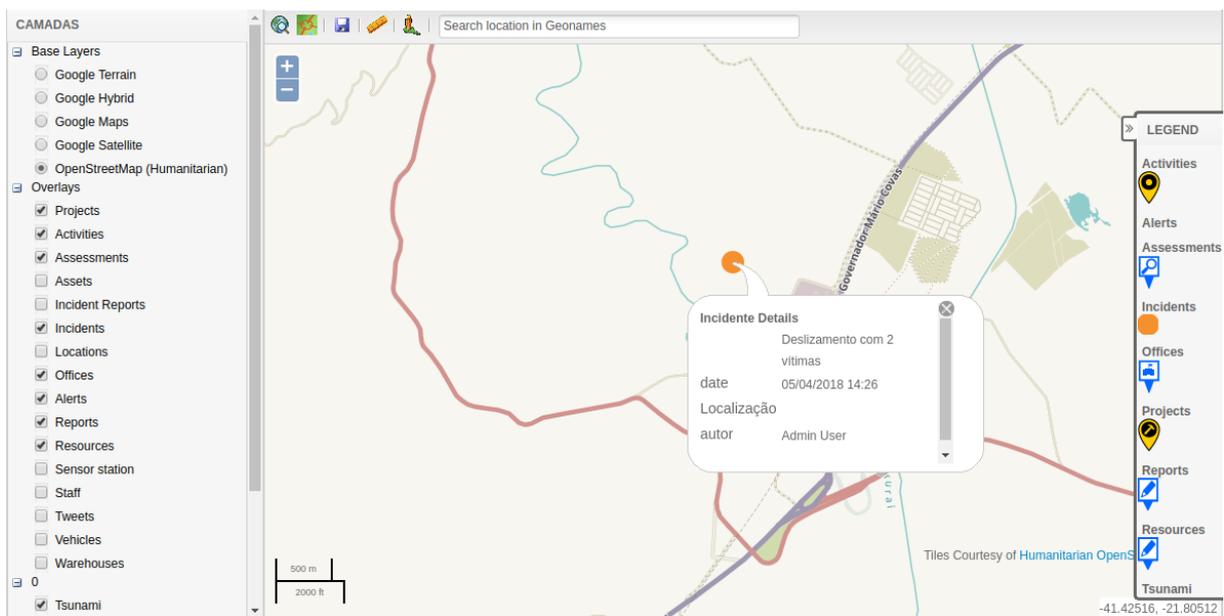


Figura 29 – Exemplo de registro de incidente no mapa.

Uma vez que é registrado um incidente e o tomador de decisão identifica este incidente, faz-se necessário a mobilização de recursos para socorrer possíveis envolvidos no incidente. Neste momento a utilização de unidades móveis como ambulâncias e carros da defesa civil, é imprescindível para dar rapidez no transporte tanto de recursos materiais, como recursos humanos.

Na mobilização das unidades móveis, a ciência da localização de cada unidade é de total importância, uma vez que pode ser priorizado o deslocamento da unidade que esteja mais

próximo ao local do ocorrido. A Figura 30 exemplifica o registro de um incidente e duas unidades móveis.

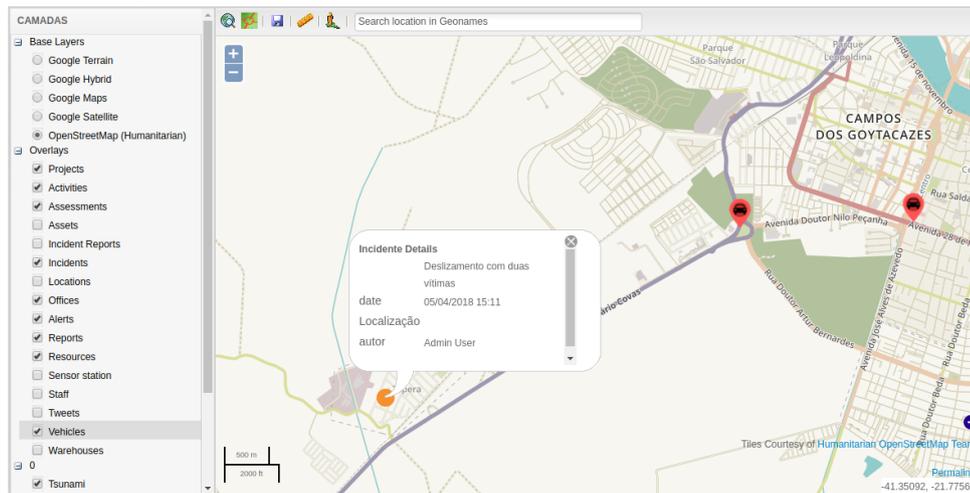


Figura 30 – Exemplo de unidades móveis próximo ao local de um registro de incidente no mapa.

A imagem mostra a visão do tomador de decisão em relação ao mapa contendo alguns elementos. É possível ver um registro de incidente e duas unidades móveis. Neste momento o responsável pela coordenação dos recursos tem a visão completa dos seus recursos alocáveis de forma georreferenciada além de poder ver detalhes das unidades móveis clicando sobre o ícone. Isto permite que ele mobilize ao local do incidente a unidade que considera mais adequada considerando a distância e características da unidade móvel.

4.4.2 Rastreamento de estações de sensores

Diante de catástrofes ou mesmo como forma de prevenção, é de suma importância que a pessoa que esteja a frente da tomada de decisão, tenha a sua disposição informações que o permita tomar decisões de maneira mais eficientes e assertivas possíveis.

Dentre as informações mais importantes estão as referentes a elementos da natureza, como: Velocidade do vento, temperatura, chuva e etc. Ter este tipo de informação atrelados as outras informações já existente na plataforma *Sahana Eden* amplia de forma significativa o poder de reação às adversidades.

Esta subseção apresentará algumas possibilidades geradas pela interação entre as informações que são possivelmente cadastradas na plataforma *Sahana Eden*, principalmente os dados históricos de incidentes, que foram inseridos na plataforma, conforme demonstrado na seção 4.3, e as informações das estações de sensores provenientes do sistema APRS.

A plataforma *Sahana Eden* possibilita a visualização dos dados sobre diversos tipos de mapas. As características do terreno do lugar analisado é um fator de grande importância e impacto, tanto na intensidade da catástrofes, quanto no manejo das respostas à catástrofes.

Diante disto a plataforma *Sahana Eden* conta com um mapa que apresenta o relevo do terreno. Este mapa é fornecido pelo 'google' e tem sua aparência como mostrado na Figura 31

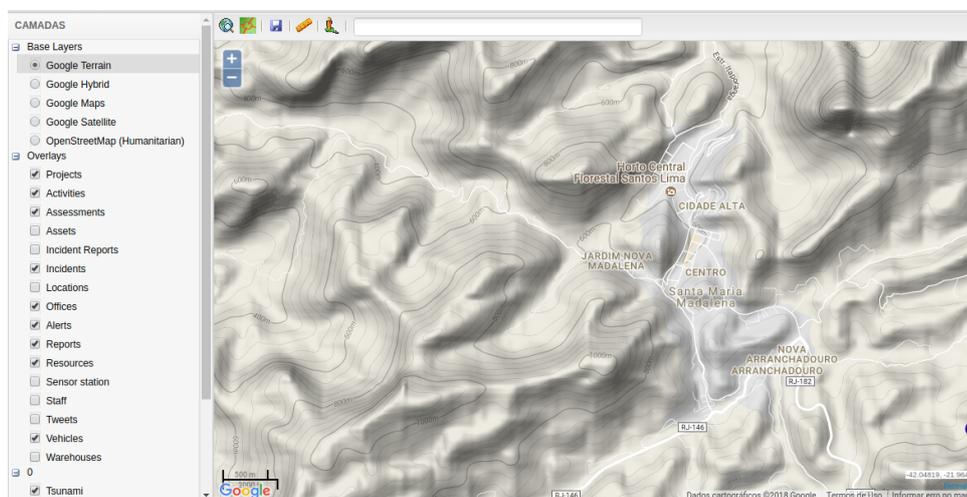


Figura 31 – Mapa com relevos.

Conforme mostrado na imagem o mapa apresenta o relevo do terreno, apresentando a altitude dos relevos. Esta informação atreladas a outras informações como: Dados históricos de incidentes e dados provenientes de estações de sensores, conferem ao tomador de decisões tomar ações preventivas, de modo a preservar a vida de possíveis afetados. A Figura 32 demonstra um exemplo de interações entre dados para prover uma leitura situacional ao tomador de decisão para que o mesmo possa tomar medidas cabíveis.

A imagem apresenta uma possível representação de dados, onde há uma interação entre representação de relevos, estações de sensores e dados históricos de deslizamento. Com isto o tomador de decisão poderia visualizar as informações dos sensores como mostrado na Figura 26 e caso o nível de precipitação no alto do morro atinja um nível que ele considere perigoso, ele pode de forma preventiva evacuar a área na base do morro, pois é uma área onde historicamente ocorre incidentes com deslizamento de terra.

A interação de informações na plataforma *Sahana Eden*, permite ao tomador de decisão uma visão ampla da situação. Como o cadastro de sensores foi feita de maneira flexível, sendo possível o armazenamento de dados de sensores da mais variadas fontes, a situação ilustrada é apenas uma diante da infinidade de situações onde a leitura de dados de sensores podem ser empregadas.

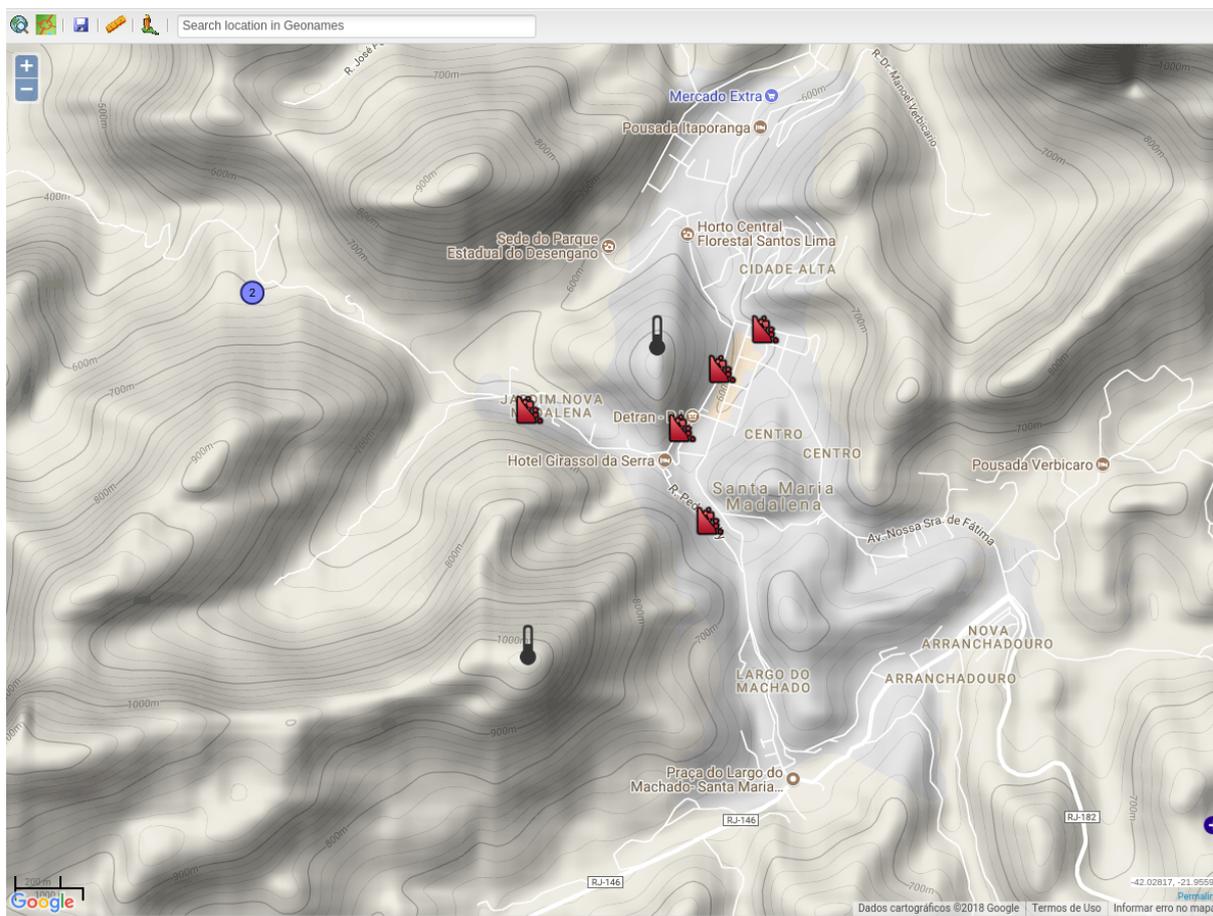


Figura 32 – Mapa com relevos, estações de sensores e dados históricos.

A implementação da integração da plataforma *Sahana Eden* com a tecnologia APRS abre uma infinidade de possibilidades no que diz respeito ao fluxo de informações. A Figura 33 apresenta uma variação da Figura 3, contemplando o fluxo de informação a partir da integração da plataforma *Sahana Eden* com o sistema APRS.

Como pode ser visto a adição do sistema APRS a plataforma *Sahana Eden*, confere aos operadores a possibilidade de fluxos de dados diferentes dos apresentados na Figura 3. As informações de veículos e sensores, que anteriormente eram inseridas manualmente, agora são inseridas de forma automática.

As informações de geolocalização de unidades móveis, bem como informações referentes a estações telemétricas são transmitidas via rádio até que as ondas cheguem a um ??, que é uma estação de rádio ligado a um computador com conexão com a internet. Uma vez que a estação recebe os sinais de rádio ela repassa esses dados ao computador, que envia os dados ao

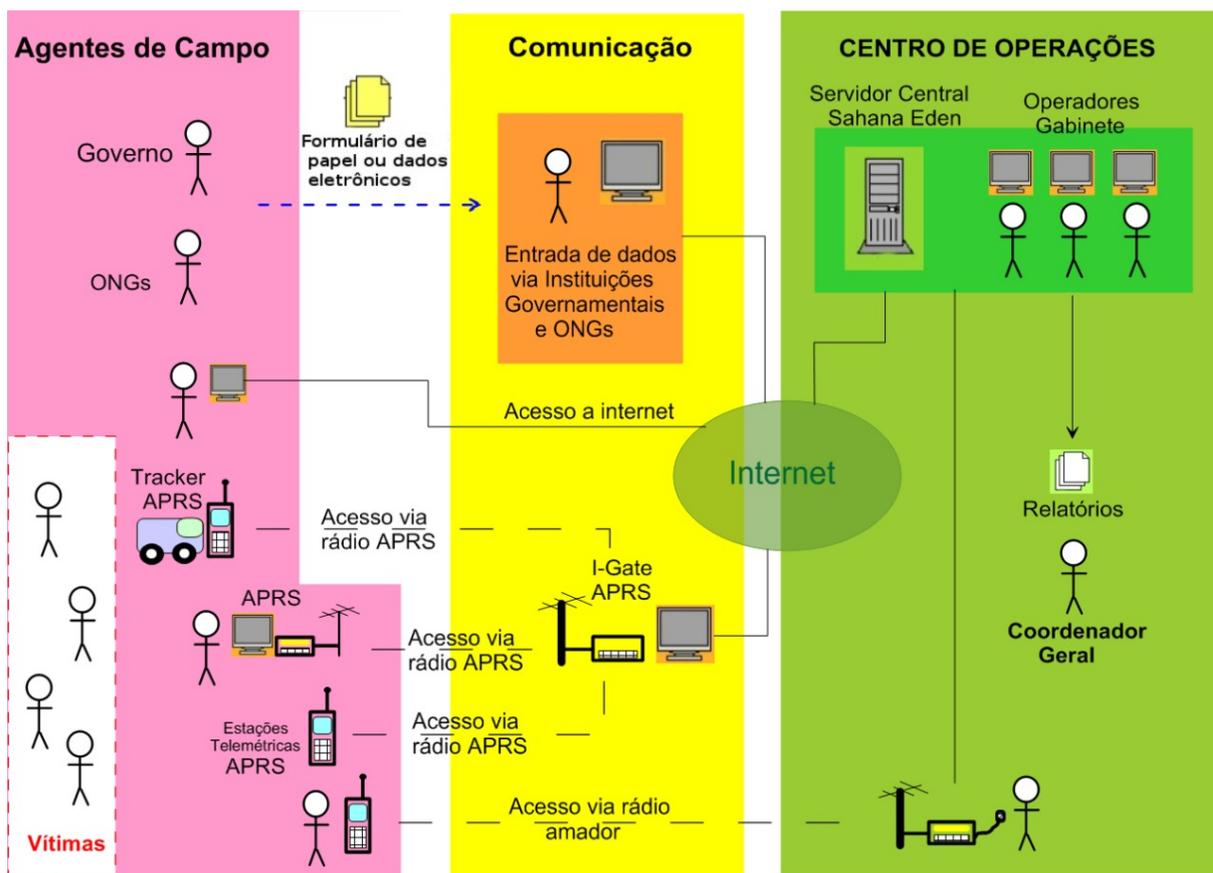


Figura 33 – Proposta de estrutura de comunicação e gestão com a inserção do APRS.

servidor APRS via internet.

Uma vez que os dados das unidades móveis e estações telemétricas estão nos servidores APRS o *Sahana Eden* pode acessar esta informação e persistir em seu banco de dados, tornando as informações disponíveis aos usuários.

Uma vez criada a infraestrutura para suportar as operações de rádio, abre-se a possibilidade da transmissão de dados de voz via rádio amador. É possível que agentes que estejam na área do desastre possa transmitir informações via rádio amador que poderão ser inseridas na plataforma *Sahana Eden* por um operador que esteja no centro de operações.

4.5 Contribuições Geradas

Como anteriormente citado este projeto tende a gerar contribuições efetivas para a comunidade, seja através contribuições de código para o sistema *Sahana Eden* em si, ou criação de novas ferramentas. A seguir serão relatadas algumas das contribuições geradas até o momento.

Assim que foram iniciadas a utilização do sistema *Sahana Eden*, foi constatado que a utilização do sistema *Sahana Eden* ficava inutilizável quando idiomas contendo '-' em sua abreviação como por exemplo 'pt-Br', 'zh-cn' e etc, eram configurados. Após constatado o erro

foi iniciada uma investigação, a fim de encontrar o código que gerava este erro. Após encontrado o trecho de código que gerava o problema, o mesmo foi solucionado e enviado um pedido de correção para os mantenedores do *Sahana Eden*, para que a correção fosse incorporado ao código principal. A solicitação foi recebida e aceita pelos mantenedores, conforme mostra a figura 34.



Figura 34 – Commit da correção de erro na mudança de idiomas.

Outra coisa muito importante e que pode gerar dificuldades no uso do sistema é o idioma da interface do mesmo. O Sistema *Sahana Eden* por padrão tem os termos da sua interface escrito em inglês, tendo assim alguns termos não traduzidos para o português do Brasil. Diante disto uma parte dos esforços deste trabalho visa atender a questão da tradução de termos para o português brasileiro. Foram feitas inúmeras contribuições neste sentido, conforme mostrado na figura 35

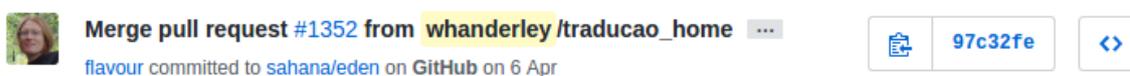


Figura 35 – Commit para tradução de termos presentes na página inicial.

Assim como o *Sahana Eden* o APRS também é um importante objeto de estudo deste trabalho. Assim a fim de extrair informações do APRS para serem exibidas na plataforma *Sahana Eden*, foi criada uma biblioteca *python* para acessar a API do site . O site *aprs.fi* funciona como um concentrador de dados da rede APRS, permitindo o rastreamento dos mais diversos tipos de elemento, como por exemplo: veículos, estações meteorológicas e etc. Além de exibir os dados no mapa conforme a figura 36 demonstra.

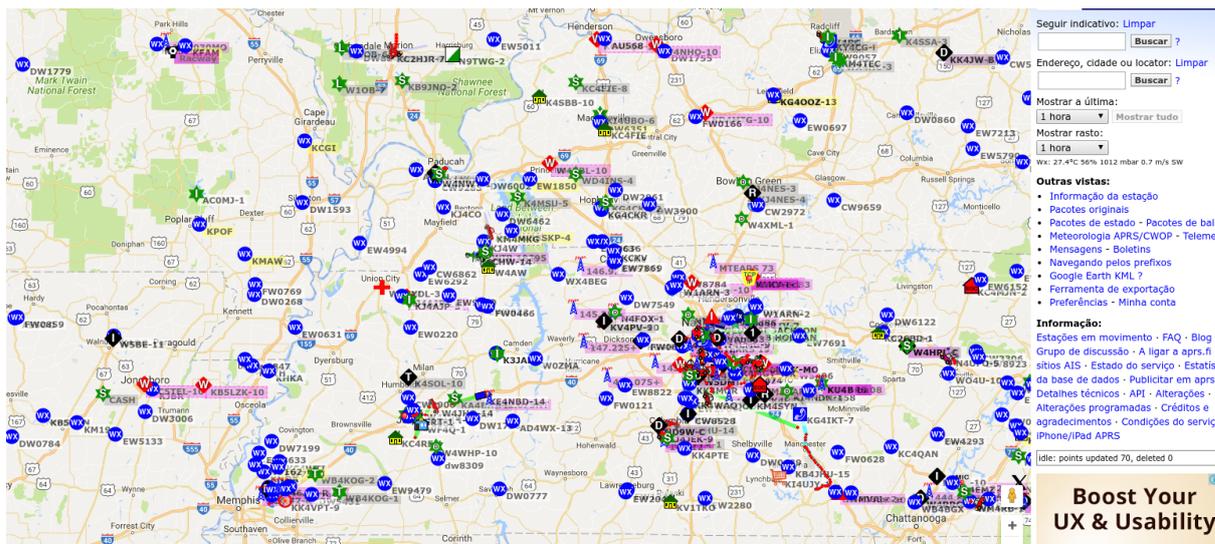


Figura 36 – Mapa do site aprs.fi

A biblioteca funciona acessando a API do site 'aprs.fi' e extraindo informações a cerca do ponto de interesse, seja veículo ou estação meteorológica. Entretanto como o site é privado e não tem garantia de disponibilidade optou-se por não utilizar esta biblioteca para a solução final deste trabalho.

A biblioteca é chamada de 'python-aprsfi' e está disponível para instalação do repositório padrão de bibliotecas python(pypy). O código fonte está disponível em <https://github.com/SAEG-DSE/python-aprsfi>, bem como instruções de uso.

5 CONSIDERAÇÕES FINAIS

É possível notar que os fenômenos naturais são potencialmente destrutivos, podendo causar desastres quando altera o fluxo normal de um determinado povo. Este pode causar prejuízos monetários ou em casos mais graves a perda de vidas. Além desses há também os desastres de origem não natural, decorrentes de atividades humanas, que podem ser igualmente destrutivos.

Com os estudo acerca dos conceitos envolvidos, nota-se que a gestão de desastres tem evoluído deixando de ser apenas um conjunto de medidas reativas, para se tornar um conceito maior que envolve o controle dos riscos na qual as pessoas estão expostas. Além disso nas últimas décadas vem se intensificando o debate a nível global sobre meios de tornar as sociedades mais resistentes a catástrofes e uma vez afetada por um fenômeno consiga retornar ao seu estado normal o mais rápido possível.

O trabalho apresenta a plataforma de ajuda humanitária *Sahana Eden*, que já foi diversas vezes utilizada em momentos de crise. Além da apresentação da plataforma o trabalho vislumbra a integração da mesma com a tecnologia APRS, que tende a aumentar a velocidade e precisão na qual as informações acerca da geolocalização de pontos móveis são abarcadas a plataforma *Shana Eden*.

Além do produto final, a integração da plataforma *Sahana Eden* e o APRS, este trabalho tende a gerar subprodutos durante o processo de desenvolvimento, que vão desde contribuições de código ao projeto do *Sahana Eden*, como também o desenvolvimento de uma ferramenta para acesso as informações do site <https://aprs.fi>, como os apresentados na seção 4.5.

O objetivo do trabalho foram atingidos, uma vez que foi implementada a aquisição de dados provenientes dos servidores APRS e integrados a plataforma *Sahana Eden* para georreferenciação de unidades móveis, quanto para o rastreamento de estação de sensores. Os conceitos aplicados podem ser expandidos para outras aplicações na plataforma. Espera-se que a integração da tecnologia APRS mostre-se como um aliado na gestão de desastres, otimizando tempo e recursos, que na prática pode representar a preservação de vidas.

Referências

- AITSI-SELMI, A. et al. The sendai framework for disaster risk reduction: Renewing the global commitment to people's resilience, health, and well-being. *International Journal of Disaster Risk Science*, Springer, v. 6, n. 2, p. 164–176, 2015. Citado na página 13.
- APRS Gateway mit Direwolf/RPi. 2016. Disponível em: <<http://wp.andreas.bieri.name/myblog/2016/12/21/aprs-gateway-mit-direwolf/rpi/>>. Citado 2 vezes nas páginas 1 e 20.
- BRUNINGA, B. *APRS: Automatic Packet Reporting System*. Disponível em: <<http://www.aprs.org/>>. Citado na página 7.
- CENAD. *Anuário brasileiro de desastres naturais: 2013*. [S.l.]: Cenad Brasília, 2014. Citado 2 vezes nas páginas 3 e 6.
- CHAIYASOONTHORN, S.; HONGYIM, N.; MITATHA, S. Building automatic packet report system to report position and radiation data for autonomous robot in the disaster area. In: IEEE. *Control, Automation and Systems (ICCAS), 2015 15th International Conference on*. [S.l.], 2015. p. 85–88. Citado na página 19.
- CORDEIRO, K. d. F.; CAMPOS, M. L. M.; BORGES, M. R. aDApTA: Adaptive approach to information integration in dynamic environments. *Computers in Industry*, v. 71, p. 88–102, ago. 2015. ISSN 01663615. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0166361515000470>>. Citado 2 vezes nas páginas 7 e 14.
- GROUP, A. W. et al. Aprs protocol reference version 1.0. *Tucson Amateur Radio Corp*, 2000. Citado na página 19.
- HANSSEN, Ø. Position tracking in voluntary search and rescue operation. ISCRAM, 2015. Citado 2 vezes nas páginas 1 e 19.
- ISDR, U. Hyogo framework for action 2005-2015: building the resilience of nations and communities to disasters. In: *Extract from the final report of the World Conference on Disaster Reduction (A/CONF. 206/6)*. [S.l.: s.n.], 2005. v. 380. Citado na página 12.
- KOHLI, S. et al. Sahana eden essential guide. *Google Summer of Code Documentation Summit, Mountain View, USA, 18-20 October 2011*, 2011. Citado 3 vezes nas páginas 8, 13 e 14.
- MATTEDI, M. A.; BUTZKE, I. C.; others. A relação entre o social e o natural nas abordagens de hazards e de desastres. *Ambiente & Sociedade*, v. 4, n. 9, p. 1–22, 2001. Disponível em: <<http://www.scielo.br/pdf/asoc/v4n9/16877.pdf>>. Citado na página 7.
- NGO, D. K. *Relief Planning Management Systems-Investigation of the Geospatial Components*. 2013. Citado na página 13.
- NURI S.T., M. M. H. Z. *Disaster Management Information System*. 2016. Disponível em: <<http://slideplayer.com/slide/6654882/>>. Citado 2 vezes nas páginas 1 e 15.
- PERDIKARIS, J. *Physical Security and Environmental Protection*. [S.l.]: CRC Press, 2014. Citado na página 11.

PEREIRA, P. C.; JÚNIOR, A. L. R. Uso de aprs na monitorização de unidades móveis na área de saúde. 2006. Citado na página 19.

PINKOWSKI, J. *Disaster management handbook*. [S.l.]: CRC press, 2008. Citado 2 vezes nas páginas 11 e 12.

POTERIE, A. T. de la; BAUDOIN, M.-A. From yokohama to sendai: Approaches to participation in international disaster risk reduction frameworks. *International Journal of Disaster Risk Science*, Springer, v. 6, n. 2, p. 128–139, 2015. Citado na página 13.

PRUSTALIS, M.; SILVA, C. D. The sahana free and open source disaster management system in haiti. *ICT for Disaster Risk Reduction, United Nations Asian and Pacific Training Centre for Information and Communication Technology for Development (UN-APCICT/ESCAP), ICTD Case Study*, v. 2, 2010. Citado na página 14.

REDUCTION, I. S. for D. *Living with risk: A global review of disaster reduction initiatives*. [S.l.]: United Nations Publications, 2004. Citado na página 12.

TEMNIKOVA¹, I.; BIYIKLI, D.; BOON, F. First steps towards implementing a sahana eden social media dashboard. 2013. Citado na página 13.

THIEREN, M. Health information systems in humanitarian emergencies. *Bulletin of the World Health Organization*, v. 83, n. 8, p. 584–589, 2005. Disponível em: <http://www.scielo.org/scielo.php?pid=S0042-96862005000800011script=sci_arttextlng=pt>. Citado na página 7.

TOBIN, G. A. *Natural hazards: explanation and integration*. Guilford Press, 1997. Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=id=RUGSoNvbMSECo&fndpg=PA1&dq=Natural+hazards:+explanation+and+integration&ots=KIA9U&sig=OzkrtnOIZePSbv9d8SaJGF4w0Qw>>. Citado na página 6.

TONDER, H. P. V. *Improving Automatic Position Reporting System (APRS) Throughput and Reliability*. Tese (Doutorado) — Stellenbosch: University of Stellenbosch, 2005. Citado na página 19.

UNISDR. *World Conference on Disaster Reduction - 18-22 January 2005, Kobe Hyogo, Japan*. 2005. Disponível em: <<http://www.unisdr.org/2005/wcdr/wcdr-index.htm>>. Citado na página 12.

UNISDR, M. Unisdr terminology for disaster risk reduction. *United Nations International Strategy for Disaster Reduction (UNISDR) Geneva, Switzerland*, 2009. Citado 2 vezes nas páginas 11 e 12.

UNISDR, M. Working Background Text on Terminology for disaster Risk Reduction. *United Nations International Strategy for Disaster Reduction (UNISDR) Geneva, Switzerland*, 2015. Citado na página 12.

WAHLSTROM, M.; GUHA-SAPIR, D. The human cost of weather-related disasters 1995-2015. *Geneva, Switzerland: UNISDR*, 2015. Citado na página 6.