**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**

**DO AMAZONAS**

**DEPARTAMENTO DE ENSINO SUPERIOR**

**CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**Ronei Macedo Gualberto**

SISTEMA DE NOTIFICAÇÃO DE ACIDENTES POR ARRAIAS

Manaus, Amazonas

Maio, 2014

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**

**DO AMAZONAS**

**Ronei Macedo Gualberto**

SISTEMA DE NOTIFICAÇÃO DE ACIDENTES POR ARRAIAS

Trabalho de Conclusão de Curso apresentado à banca examinadora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciências e Tecnologia do Amazonas (IFAM), Campus Manaus Centro, como requisito para a conclusão de curso.

**Profa. Dra. Andréa Pereira Mendonça - IFAM**

**Profa. Dra. Maria Cristina dos Santos - UFAM**

(Orientadoras)

Manaus – Amazonas

© Ronei Macedo Gualberto – Maio, 2014

SISTEMA DE NOTIFICAÇÃO DE ACIDENTES POR ARRAIAS

**Aprovado em Maio, 2014**

BANCA EXAMINADORA

Profa. Dra. Andréa Pereira Mendonça

(Orientador Acadêmico)

Profa. MSc. Elloá Barreto Guedes da Costa

Prof. Esp. Rogério Carminé

**Dedicatória**

*Dedico à minha família , amigos e professores.*

**Agradecimentos**

A Deus por me ajudar, todos os dias, a me manter firme no meus objetivos de vida, mostrando-me que é possível vencer os desafios que aparecem ao longo do cotidiano.

Agradeço, também, à minha família, em especial. à minha mãe que sempre acreditou em mim, não medindo esforços para que eu pudesse seguir em frente nos meus estudos, ensinando-me os caminhos para sempre ser uma pessoa digna e justa. Ao meu irmão, que apesar de muitas vezes nos desentendermos, sempre foi uma inspiração para mim.

Agradeço à professora Andrea Mendonça pela ajuda ao longo da orientação deste trabalho, possibilitando a minha evolução através das disciplinas ministradas e dos conselhos sempre bem realistas. Também, ao professor Rogério Carminé, pelas aulas divertidas de programação em Java, em que pude aprender e aperfeiçoar meu modo de programar, sempre enfatizando as melhores práticas.

Aos amigos que fiz ao longo desse curso,especialmente, minha equipe inseparável: Francisca Dantas, Érika Silva, Isaac Santos e Ovídio Santos. Juntos compartilhamos alegrias e dificuldades ao longo da graduação. Agradeço ao Ovídio Santos por proporcionar esses momentos inesquecíveis em sua casa onde aprendemos muito. Agradeço a Francisca Dantas por sempre conseguir gerenciar essa nossa equipe, sempre resolvendo os conflitos entre agente. Agradeço à Érika Silva por toda a ajuda que me deu para minha formação profissional e pessoal, ensinando-me o sentido real de uma amizade verdadeira. Agradeço a Isaac Santos por sempre estar disposto a me ouvir e conversar nos momentos de dificuldade, sempre encontrando um solução simples, salvando a pátria.

Aos colegas de trabalho da PRODAM José Silas e Geison Nascimento pela troca de idéias que foram muito importantes para melhoria deste trabalho.

Ao professor Elloá Barreto e Rogério Carminé por participarem da banca de avaliações pelas sugestões de melhoria deste trabalho.

E outros colegas e professores do curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

**Epígrafe**

*Sem passado, nem futuro, eu vivo um dia de cada vez*

*Humberto Gessinger*

**Resumo**

Animais peçonhentos são responsáveis por acidentes que podem provocar sequelas ou levar a vítima ao óbito, representando uma questão de saúde pública. No Brasil, existem sistemas implantados para notificação de tais acidentes, principalmente voltados para casos de agravos ocasionados por ofídios, escorpiões e aracnídeos. O estudo sobre acidentes envolvendo animais aquáticos apresenta informações escassas; por vezes, pouco conclusivas a respeito do controle e prevenção dos acidentes. As arraias estão incluídas no grupo de animais aquáticos que provocam acidentes graves. Devido ao fato destes acidentes,geralmente, ocorrerem em áreas remotas e de difícil acesso aos serviços de saúde, são, comumente, adotados procedimentos populares de tratamento, levando assim a não notificação da ocorrência. O Amazonas, por exemplo, abriga uma variedade de espécies de arraias que podem causar acidentes, sendo que as principais vítimas são populações ribeirinhas que realizam atividades no meio aquático,tais como, a pescaria. Há poucas informações oficiais sobre as ocorrências de acidentes por arraias no estado do Amazonas, devido principalmente ao fato de não existir um mecanismo unificado para notificar os acidentes ocorridos ao longo do território amazonense. Em consequência da falta de notificações dos acidentes, não há também o devido conhecimento da população mais atingida, dos sintomas apresentados e dos tipos de tratamentos efetivados, dificultando a tomadas de medidas para prevenção e controle destes acidentes. A fim de mitigar os problemas apresentados, este trabalho tem como resultado final um sistema web que foi desenvolvido com a finalidade de permitir a comunidade em geral e aos profissionais de saúde notificarem os acidentes por arraia no estado do Amazonas. O sistema provê recursos para as notificações destes acidentes, identificação do animal causador, da localidade geográfica de ocorrência do acidente, assim como dos procedimentos de tratamento. O sistema disponibiliza também um conjunto de relatórios com dados estatísticos que poderão subsidiar estudos epidemiológicos, pesquisas sobre os tipos de ocorrência e gravidade desses acidentes no estado do Amazonas, assim como ajudar na tomada de decisão quanto à administração de tratamentos para este tipo de acidente.No desenvolvimento do sistema , foi adotado o Processo Unificado Ágil (AUP), a linguagem de modelagem unificada (UML), plataforma Java e o sistema gerenciador de banco de dados MySQL.

**Palavras-Chave**. Animais Peçonhentos;Arraias; Sistema de Notificação de Acidentes.

**Lista de Figuras**

[Figura 1 - Ciclo de vida do AUP. 8](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094839)

[Figura 2 - Exemplo de Diagrama de Caso de Uso. 12](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094840)

[Figura 3 - Estrutura de pastas de um projeto em Maven 19](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094841)

[Figura 4 - Módulos do Sistema 23](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094842)

[Figura 5 - Módulo de gerenciamento do sistema. 25](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094843)

[Figura 6 - Módulo de notificação e acompanhamento de acidentes por unidade de saúde. 27](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094844)

[Figura 7 - Módulo de notificação de acidentes não-oficiais. 28](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094845)

[Figura 8 - Módulo de consultas e geração de relatórios de acidentes 29](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094846)

[Figura 9 - Tela Inicial do Sistema 30](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094847)

[Figura 10 - Solicitação de criação conta 31](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094848)

[Figura 11 - Validar Conta de Usuário 31](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094849)

[Figura 12 - Cadastrar Profissional de Saúde 32](#_Toc388094850)

[Figura 13 - Pesquisar Pacientes 33](#_Toc388094851)

[Figura 14 - Visualizar Paciente 33](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094852)

[Figura 15 - Dados do Paciente 34](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094853)

[Figura 16 - Dados do acidente 34](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094854)

[Figura 17 - Conclusão do cadastro do acidente 35](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094855)

[Figura 18 - Dados do paciente 36](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094856)

[Figura 19 - Dados do acidente 36](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094857)

[Figura 20 - Dados da Arraia 37](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094858)

[Figura 21 - Identificar o local anatômico da picada 37](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094859)

[Figura 22 - Editar prontuário 38](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094860)

[Figura 23 - Editar Tratamento 39](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094861)

[Figura 24 - Concluir acidente 40](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094862)

[Figura 25 - Consulta de dados estatísticos 41](#_Toc388094863)

[Figura 26 - Relatório emitido pelo sistema 42](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094864)

[Figura 27 - Diagrama de classe na visão de análise 43](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094865)

[Figura 28 - Diagrama de classes na visão de projeto 44](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094866)

[Figura 29 - Diagrama de entidade-relacionamento 45](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094867)

[Figura 30 - Padrão Arquitetural MVC 46](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094868)

[Figura 31 - Modelo com várias visões 47](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094869)

[Figura 33 - Página inicial do sistema 67](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094870)

[Figura34 - Solicitação de criação conta 68](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094871)

[Figura 35 - Validar Conta de Usuário 69](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094872)

[Figura 36 - Cadastrar profissional de saúde 70](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094873)

[Figura 37 - Tela inicial do perfil de profissional de saúde 71](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094874)

[Figura 38 - Visualizar Acidente 72](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094875)

[Figura 39 - Cadastrar Acidente 73](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094876)

[Figura 40 - Dados do Paciente 74](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094877)

[Figura 41 - Dado do acidente 75](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094878)

[Figura 42 - Dados da arraia 76](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094879)

[Figura 43 - Editar prontuário 77](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094880)

[Figura 44 - Marcar local da picada 78](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094881)

[Figura 45 - Editar Tratamento 79](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094882)

[Figura 46 - Concluir Acidente 80](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094883)

[Figura 47 - Consulta de dados estatísticos 81](file:///C:\tcc2\workspace-svn\snaa\docs\TCC%20I_Ronei_vFinal.docx#_Toc388094884)

**Lista de Quadros**

[Quadro 1 - Recursos utilizados no desenvolvimento do SNAA 7](#_Toc388094885)

[Quadro 2 - Artefatos gerados ao longo das fases do AUP 10](#_Toc388094886)

[Quadro 3 - Exemplo de diagrama de caso de uso 11](#_Toc388094887)

[Quadro 4 - Arquivo pom.xml 19](#_Toc388094888)

[Quadro 5 - Código implementado no início do desenvolvimento do SNAA 48](#_Toc388094889)

[Quadro 6 - Código após refatorações 48](#_Toc388094890)

[Quadro 7 - Caso de Uso CDU01: Solicitar Criação de Conta. 56](#_Toc388094891)

[Quadro 8 - Caso de Uso CDU02: Validar Solicitação 58](#_Toc388094892)

[Quadro 9 - Caso de Uso CDU03: Cadastrar profissional de saúde. 58](#_Toc388094893)

[Quadro 10 - Caso de Uso CDU 04: Pesquisar Paciente. 60](#_Toc388094894)

[Quadro 11 - Caso de Uso CDU05: Cadastrar Acidente. 61](#_Toc388094895)

[Quadro 12 - Caso de Uso CDU06 :Cadastrar Paciente 62](#_Toc388094896)

[Quadro 13 - Caso de Uso CDU07 : Identificar Arraia 62](#_Toc388094897)

[Quadro 14 - Caso de Uso CDU08 :Cadastrar Prontuário 63](#_Toc388094898)

[Quadro 15 - Caso de Uso CDU09: Cadastrar Tratamento 64](#_Toc388094899)

[Quadro 16 - Caso de Uso CDU10 : Concluir Acidente 64](#_Toc388094900)

[Quadro 17 - Caso de Uso CDU11: Reativar acidente 65](#_Toc388094901)

[Quadro 18 - Caso de Uso CDU12: Consultar Dados Estatísticos. 66](#_Toc388094902)

**Sumário**

[Capítulo 1 1](#_Toc387566623)

[Introdução 1](#_Toc387566624)

[Capítulo 2 4](#_Toc387566625)

[Fundamentação Teórica 4](#_Toc387566626)

[2.1 Acidentes por Arraia 4](#_Toc387566627)

[2.2 Notificação de acidentes por arraia 5](#_Toc387566628)

[2.3 Recursos utilizados no desenvolvimento do SNAA 7](#_Toc387566629)

[2.3.1 AUP – *Agile Unified Process* 7](#_Toc387566630)

[2.3.3 UML (*Unified Modeling Language*) 11](#_Toc387566631)

[2.3.4 Plataforma Java 13](#_Toc387566632)

[2.3.5 JSF 14](#_Toc387566633)

[2.3.6 Ferramenta ORM, Hibernate e JPA. 15](#_Toc387566634)

[2.3.4 CDI 15](#_Toc387566635)

[2.3.5 MySQL 16](#_Toc387566636)

[2.3.6 Demoiselle Framework 17](#_Toc387566637)

[2.3.7 JBoss AS 18](#_Toc387566638)

[2.3.8 Apache Maven 18](#_Toc387566639)

[2.3.8 Jasper Reports e IReport 19](#_Toc387566640)

[2.3.8 Astah Community 20](#_Toc387566641)

[Capítulo 3 21](#_Toc387566642)

[SNAA 21](#_Toc387566643)

[3.1 Concepção 21](#_Toc387566644)

[3.1.1 Descrição geral do software 21](#_Toc387566645)

[3.1.2 Diagramas de Casos de Uso 22](#_Toc387566646)

[3.2.1 Módulo de gerenciamento do sistema 23](#_Toc387566647)

[3.1.3 Interface dos usuários 30](#_Toc387566648)

[3.1.4 Diagramas de classes 43](#_Toc387566649)

[3.1.5 Diagramas de entidades e relacionamentos 45](#_Toc387566650)

[3.1.5 Elaboração 46](#_Toc387566651)

[3.1.7 Construção 48](#_Toc387566652)

[3.1.7 Implantação 49](#_Toc387566653)

[Capítulo 4 50](#_Toc387566654)

[Considerações Finais 50](#_Toc387566655)

[Referências Bibliográficas 52](#_Toc387566656)

[Anexo I – Ficha de notificação individual de acidentes por animais peçonhentos 54](#_Toc387566657)

[Apêndice 1 – Especificação de Casos de Uso do Módulo de Gerenciamento do Sistema 56](#_Toc387566660)

[Apêndice 2 – Especificação de Casos de Uso do Módulo de Notificação e Acompanhamento de acidentes por unidade de saúde 60](#_Toc387566661)

[Apêndice 3 – Especificação de Casos de Uso do Módulo de consultas e geração de relatórios de acidentes 66](#_Toc387566662)

[Apêndice 4 - Interfaces do Usuário (Protótipos Iniciais) 67](#_Toc387566663)

# Capítulo 1

# Introdução

Animais peçonhentos são reconhecidos como aqueles que produzem veneno e possuem algum aparato para injetá-lo na sua presa ou predador, ou vítima (BRASIL, 2013). Serpentes, escorpiões, aranhas, lagartas, abelhas, sapos, arraias, dentre outros (SOUZA, 2010) são exemplos de animais peçonhentos e algumas vezes, responsáveis por acidentes que podem evoluir ao óbito (DORNELES, 2009). O Brasil apresenta fauna peçonhenta muito diversa, incluindo a maior parte dos animais apontados acima e que, anualmente, causam grande número de tais envenenamentos, constituindo um quadro devastador de doença negligenciada (SOUZA, 2010).

Segundo MINTON Jr (1974) citado em SALLES e MENEGUCI DA CUNHA (2007), os acidentes por animais peçonhentos de interesse médico são considerados um problema de saúde pública, dada à incidência, a gravidade e as sequelas deixadas nas vítimas. Somente no Brasil, em média temos, ao ano, cerca de 20.000 casos de acidentes com serpentes, 5.000 com aranhas e 8.000 por escorpiões. O número médio anual de acidentes por animais peçonhentos calculado para o período de 1990 a 1995,foi de aproximadamente 32.000 casos/ano, e este número só foi menor do que o encontrado para os casos de Malária (todas as formas) (546.398 /ano) e de Tuberculose (todas as formas) (81.290/ano) (BOCHNER, 2003).

No território brasileiro, os animais peçonhentos habitam os mais diversos tipos de ambiente. No meio terrestre, os acidentes ofídicos são um dos mais frequentes e mais graves, ocorrendo em todos os estados brasileiros, seguido por acidentes ocasionados por escorpiões e aranhas. Dados do Sistema de Informações de Mortalidade(SIM), citados em (BOCHNER, 2003), apontam que, para o período de 1990 a 1995, a média anual de óbitos foi de 132 óbitos ocasionados por serpentes, 6 por aranhas e 38 por escorpiões. No meio aquático, os acidentes por peixes peçonhentos são causados principalmente por bagres, tanto no mar como em rios, arraias (vários gêneros) em água doce ou salgada, peixe-escorpião (*Scorpaema)* encontrado em água salgada nas praias rochosas, peixe-sapo (*Thalassophryne)*, moreias(*Gimnothorax*) e cações encontrados na água salgada (DORNELES, 2009).

No contexto do estado do Amazonas, os casos de acidentes por serpentes continuam sendo os mais frequentes.Entretanto, segundo dados do Butantan (2009), os casos envolvendo arraias estão no topo do ranking de acidentes por animais peçonhentos, junto com outros bichos perigosos, e mais conhecidos, como escorpiões e jararacas, sendo responsáveis por causar um dos acidentes mais graves de envenenamento por animal aquático. Apesar da baixa letalidade, os acidentes são responsáveis por uma elevada morbidade caracterizada por sequelas permanentes e casos de incapacidade temporária para o trabalho. Geralmente, as ocorrências dos acidentes manifestam-se quando as arraias são pisadas ou manuseadas de maneira inadequada (GARRONE NETO e HADDAD JUNIOR, 2009).

No Brasil, o monitoramento e catalogação dos casos de acidentes causados por este grande número de espécies venenosas vêm sendo continuamente registrados pelo Sistema de Informação de Agravos de Notificação (SINAN) da Secretaria de Vigilância em Saúde do Ministério da Saúde(SOUZA, 2010), sendo que a maior ênfase de notificação está relacionada aos acidentes causados por serpentes,escorpiões e aranhas.Em contrapartida, os acidentes causados por animais aquáticos apresentam comunicações esparsas e pouco conclusivas em termos de epidemiologia, relatos dos sinais e sintomas e medidas terapêuticas empregadas nesses envenenamentos (HADDAD JUNIOR, 2003). Segundo NETO & HADDAD (2009), de modo diferente do que ocorreu com os sistemas de notificação para acidentes ofídicos e, posteriormente, para os causados de aranhas e escorpiões, os estudo sobre acidentes provocados por animais aquáticos sempre foram relegados a um segundo plano, negligenciado pelos órgãos responsáveis pelo conhecimento, controle e prevenção de agravos envolvendo animais peçonhentos.

No Amazonas, por exemplo, os acidentes provocados por arraias são de difícil notificação devido ao aspecto geográfico de seus municípios,isto é, locais de difícil acesso aos serviços de saúde pública, o que favorece a adoção dos mais variados tipos de tratamentos populares em vez da procura por auxílio médico, contribuindo para sua não-notificação e ocorrência silenciosa (GARRONE NETO e HADDAD JUNIOR, 2009).

A fim de mitigar os problemas referentes a falta de notificação, este trabalho tem por objetivo disponibilizar um sistema web para possibilitar que a comunidade em geral e os profissionais de saúde notifiquem os acidentes por arraias no estado do Amazonas,contribuindo assim para uma melhoria da saúde no território amazonense.

Este sistema web denomina-se SNAA e teve seu desenvolvimento guiado pelo Processo Unificado Ágil (AUP). Foi implementado em Java e utilizou-se o banco de dados MySQL.

O sistema proposto provê recursos para as notificações destes acidentes, identificação do animal causador, da localidade geográfica de ocorrência do acidente, assim como dos procedimentos adotados no tratamento. Com base nessas informações, o sistema disponibiliza também um conjunto de relatórios com dados estatísticos que poderão subsidiar estudos epidemiológicos, pesquisas de ocorrência e gravidade desses acidentes no estado do Amazonas.

Afim de apresentar os aspectos teóricos e técnicos que fundamentam este trabalho, bem como o detalhamento das fases do desenvolvimento do SNAA, esta monografia está organizada em quatro capítulos. No capítulo 2 são apresentados a fundamentação teórica e os trabalhos relacionados. A análise e projeto SNAA são apresentados no Capítulo 3, seguida pelas considerações finais no Capítulo 4.

# Capítulo 2

# Fundamentação Teórica

Neste capítulo são abordados os conceitos que fundamentam de um sistema de notificação de acidentes. Inicialmente, para contextualização do problema tratado neste trabalho, serão descritas algumas características biológicas das arraias e aspectos do envenenamento causados por este animal. Posteriormente, são apresentados os sistemas de notificação de acidentes por arraias existentes a nível nacional e estadual. Ao final do capítulo, são apresentadas as recursos tecnológicos adotados no desenvolvimento do SNAA.

## 2.1 Acidentes por Arraia

Raias (ou arraias) são animais da classe *Chondrythies*, um grupo predominantemente marinho, cujo nome faz referência à característica distintiva mais marcante desses animais: o esqueleto formado por tecido cartilaginoso. Junto com os tubarões, as arraias constituem a subclasse *Elasmobranchii*, diferindo destes por apresentar o corpo achatado dorsiventralmente, com avantajadas nadadeiras peitorais; olhos e espiráculos localizados no topo da cabeça além da boca e das fendas branquiais posicionais na região ventral do corpo (GARRONE NETO e HADDAD JUNIOR, 2009).

Segundo GARRONE NETO e HADDAD JUNIOR (2009), a superordem *Batoidea* é quem abriga o maior número de espécies (aproximadamente 630), divididas em seis ordens, sendo a ordem das *Myliobatiformes* a que possui maior importância médica, pois compreende as "arraia de ferrão", cuja característica principal da maioria das 180 espécies é possuir ferrões retrosserilhados na porção dorsal da cauda.

No Brasil, a ordem *Myliobatiformes* é representada por cerca de 50 espécies, distribuídas entre sete famílias (GARRONE NETO e HADDAD JUNIOR, 2009), sendo que apenas a *Potamotrygonidae* é restrita a ambientes fluviais, compondo-se de 20 espécies distribuídas em três gêneros: *Potamotrygon*, *Paratrygon* e *Plesiotrygon*. Os três gêneros são encontrados na bacia amazônica, sendo *Plesiotrygon* a esta bacia, enquanto que *Paratrygon* também ocorre no rio Orinoco na Venezuela(GENTIL VASCONCELOS e SÁ DE OLIVEIRA, 2011).

As arraias vivem no fundo do mar e dos rios, escondem-se em covas rasas escavadas em locais arenosos ou lodosos. São carnívoras, alimentando-se de pequenos peixes, caramujos,crustáceos e larvas de insetos. As arraias de água doce têm um tamanho que varia de 30 centímetros até, aproximadamente, um metro de diâmetro; algumas arraias marinhas podem atingir um diâmetro de cinco metros, quando adultas (PARANÁ, 2012).

Apesar de as arraias serem consideradas dóceis, os acidentes são muito comuns entre pescadores do litoral, na Bacia Amazônica e na região do Pantanal. Pelo hábito de se manterem no fundo arenoso ou lodoso das praias, podem ser inadvertidamente pisoteadas por banhistas. Os acidentes ocorrem quando a região dorsal da arraia é tocada ou pressionada; isso provoca uma resposta muscular com flexão da cauda para cima “apontando” o ferrão para o local estimulado, atingindo geralmente o tornozelo ou o pé da vítima (PARANÁ, 2012).

Os principais aspectos clínicos do envenenamento por arraias são a dor e necrose na área da picada. Os ferimentos perfuro-cortantes resultantes podem ser de extensão e profundidade variáveis, dependendo da região anatômica atingida (PARANÁ, 2012).

Ao perfurar a pele da vítima, o epitélio que recobre o ferrão da arraia é danificado, expondo as células glandulares e liberando as toxinas. Na tentativa de retirada do ferrão, por ser retroserrilhado, o ferimento pode se ampliar pelo dilaceramento dos tecidos, facilitando a absorção de toxinas e acentuando os sinais e sintomas(PARANÁ, 2012).

Em relação ao tratamento de vítimas de acidentes por arraia, NETO & HADDAD (2009) afirmam que existe uma rotina de tratamento em que é recomendado a imediata imersão do ponto comprometido pela picada em água quente (em torno de 60 graus), pois experimentos e atendimentos a vítimas têm demonstrado grande redução da dor com esta medida. No entanto, em regiões ribeirinhas é comum que as vítimas utilizem-se de medidas caseiras, crendices e lendas para tratar do acidente. Tais procedimentos, acabam acarretando maiores danos à saúde da vítima e problemas relacionados a notificação desse tipo de agravo, já que as vítimas não procuram a ajuda em órgãos responsáveis pelo controle e prevenção desses tipos de acidentes. Maiores informações sobre os sistemas de notificação de acidentes por arraias são fornecidas na seção seguinte.

## 2.2 Notificação de acidentes por arraia

De modo diferente com o que ocorreu com os sistemas de notificação implantados para o dimensionamento dos acidentes ofídicos e, posteriormente, para os casos de araneísmo e escorpionismo, o estudo dos acidentes provocados por animais aquáticos sempre foi relegado a um segundo plano, negligenciado pelos órgãos responsáveis pelo conhecimento, controle e prevenção de agravos envolvendo animais peçonhentos (GARRONE NETO e HADDAD JUNIOR, 2009).

Segundo NETO & HADDAD (2009),os fatores responsáveis por essa deficiência são:

* a baixa letalidade observada nos acidentes, embora sua elevada morbidade muitas vezes seja responsável por sequelas permanentes e caso de incapacidade temporária para o trabalho;
* o fato de a ocorrência destes acidentes, normalmente, estar relacionada a áreas de localização remota, onde o acesso aos serviços de saúde pública é restrito, o que favorece a adoção dos mais variados tipos de tratamentos populares em vez da procura por auxílio médico, contribuindo para a não-notificação e, consequentemente, ocorrência silenciosa.

NETO & HADDAD (2009)afirmam também que as poucas informações disponíveis de acidentes com arraias são fruto de realização de estudos transversais, por procura ativa de casos, ou da consulta a prontuários previamente elaborados e deixados para preenchimento em unidades de saúde da região norte do País.

Tal afirmação pode-se constatar através da investigação da ficha de notificação individual de acidentes por animais peçonhentos (Anexo I), disponibilizada pelo Sistema de Informação de Agravos de Notificação (SINAN).

O SINAN é mantido pelo governo federal e tem como objetivo coletar,transmitir e disseminar dados gerados rotineiramente pelo Sistema de Vigilância Epidemiológica das três esferas de governo (SAÚDE, MINISTÉRIO DA, 2007). Segundo (BOCHNER e STRUCHINER, 2002), o SINAN é o único sistema que possui um módulo específico para tratar acidentes por animais peçonhentos.

A ficha apresenta uma seção denominada Dados do acidente, no qual deve-se preencher informações sobre o tipo de acidente, oferecendo opções para diversos tipos de acidentes: por exemplo serpentes, aranhas,lagartas.No entanto, não oferece opções para preenchimento de informações específicas de acidentes causados por arraias. Na parte inferior do formulário, tem-se uma tabela que descreve as manifestações clínicas, classificação e soroterapia apenas para os seguintes tipos de acidente: ofidismo, escorpionismo, araneísmo e lonomia.

No site do DATASUS[[1]](#footnote-2) (Departamento de Informática do SUS) é fornecido uma consulta dos dados estatísticos das notificações registradas no SINAN, oferecendo alguns filtros para a consulta do tipo de acidente.Dentre eles, existe as opções: Todas as categorias, Em Branco, Serpente, Aranha, Escorpião e Lagarta.

Diante dessas evidências, pode-se verificar que o SINAN não oferece de forma satisfatória informações a respeito de acidentes causados por arraias, já que na própria ficha de notificação não oferece campos para preenchimento sobre esse tipo de agravo. Tal realidade evidencia a importância do trabalho ora proposto.

## 2.3 Recursos utilizados no desenvolvimento do SNAA

Esta seção apresenta as tecnologias adotadas no processo, modelagem e implementação do SNAA - Sistema de Notificação de Acidentes por Arraias.

Quadro 1 - Recursos utilizados no desenvolvimento do SNAA

## 2.3.1 AUP – *Agile Unified Process*

O Processo Unificado (PU),em inglês *Unified Process,* é um modelo de processo de desenvolvimento de software iterativo e incremental que promove práticas amplamente reconhecidas na indústria de software. Baseado neste modelo de processo, a *Rational Software Corporation* lançou o processo Unificado da Rational (RUP), em inglês *Rational Unified Process*. O RUP é um modelo de processo clássico, preferencialmente, aplicável a grandes projetos que envolvem uma grande equipe de desenvolvimento.

A fim de adaptar as boas práticas do RUP a pequenos projetos e que atendam a uma filosofia ágil, Scott Ambler criou o AUP (*Agile Unified Process*). O AUP descreve uma abordagem simples e de fácil compreensão para o desenvolvimento de software usando técnicas e conceitos ágeis,ainda assim permanecendo fiel as práticas do RUP (AMBLER, 2005).

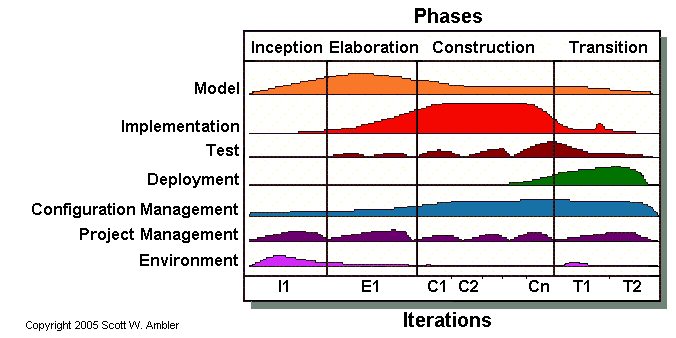
Em uma abordagem de desenvolvimento iterativo e incremental, um projeto é organizado em um série de pequenas iterações de duração fixa, gerando, ao final, um sistema parcial (ou módulo) , executável, testável e integrável.

O AUP adota a filosofia ágil descrita no Manifesto Ágil, um conjunto de práticas que todo método ágil deve seguir. Tal manifesto foi elaborado por um grupo de interessados em métodos iterativos e ágeis que se reuniram para criarem a Aliança Ágil (Agile Alliance); declarando, assim, os princípios para captação do espírito do método ágeis.

Dentre os principais colaboradores do Manifesto Ágil podemos, destacar: Ken Beck, Martin Fowler, Robert C. Martin, Dave Thomas. Para mais detalhes (BECK, BEEDLE, *et al.*, 2001)

Na Figura 1 é apresentada as fases, disciplinas e iterações que um projeto de software utilizando AUP deve considerar.

Figura 1 - Ciclo de vida do AUP.



Como pode ser observada na Figura 1, o desenvolvimento de software deve obedecer a quatro fases,a saber:

* **Concepção (*Inception*)**: O objetivo é identificar o escopo inicial do projeto e a arquitetura potencial do sistema a fim de gerar estimativas financeiras e prover informações que subsidiem a aceitação pelas partes interessadas;
* **Elaboração (*Elaboration*):**O objetivo é prover uma arquitetura central, resolução dos altos riscos e definição de estimativas mais realistas;
* **Construção(*Construction*)**: O objetivo é construir o software, trabalhando em uma base incremental regular, que atende às necessidades de maior prioridade das partes interessadas no projeto;
* **Transição(*Transition*)**: O objetivo é validar e implantar o sistema em seu ambiente de produção.

É importante destacar que as fases do AUP não segue um processo em seqüencial ( ou ciclo de vida em cascata), no qual primeiramente precisamos definir todos os requisitos para , então, fazer todo ou a maioria do projeto. Pelo contrário, a cada iteração, podem existir atividades de fases distintas sendo executadas em paralelo.

Ainda conforme a Figura 1, o AUP possui sete disciplinas: Modelagem (*Model*), Implementação (*Implementation*), Teste(*Test*)**,** Implantação (*Deployment*), Gerenciamento de Configuração (*Configuration Management*), Gerenciamento de Projetos(*Project Management*) e Ambiente (*Environmen*t). As disciplinas definem um conjunto de atividades (e artefatos relacionados) que são realizadas ao longo das iterações de desenvolvimento. Em síntese, as disciplinas são definidas como segue:

* **Modelagem (*Model*)**: O objetivo desta disciplina é promover o entendimento do negócio da organização, o domínio do problema a ser abordado pelo projeto, e a identificação de uma solução viável para resolver o domínio do problema;
* **Implementação(*Implementation*)**:O objetivo desta disciplina é o de transformar o(s) modelo(s) em código executável e executar um nível básico de testes, nomeadamente os testes de unidade;
* **Teste(*Test*):** O objetivo desta disciplina é a realização de uma avaliação objetiva para garantir a qualidade. Isto inclui as atividades de validação e verificação dos requisitos;
* **Implantação (*Deployment*)**. O objetivo desta disciplina é planejar a entrega do sistema e executar este plano a fim de tornar o sistema disponível para os usuários finais;
* **Gerenciamento de Configuração (*Configuration Management*)**. O objetivo desta disciplina é o de gerir o acesso aos artefatos criados durante todas as fases do desenvolvimento do software. Isso inclui não apenas controlar versões de artefatos ao longo do tempo, mas também controlar e gerenciar mudanças;
* **Gerenciamento de Projetos(*Project Management*)**. O objetivo desta disciplina é a de dirigir as atividades que ocorrem no projeto. Isto inclui a gestão de riscos, a coordenação de pessoas e controle de prazos e orçamento;
* **Ambiente (*Environment*):** O objetivo desta disciplina é o de apoiar o resto do esforço, garantindo que o processo adequado, orientação (normas e diretrizes) e ferramentas (hardware, software, etc.) estão disponíveis para a equipe quando necessário

Durante uma iteração o trabalho prossegue na maioria ou em todas as disciplinas. Entretanto, o esforço relativo no decorrer destas disciplinas mudam ao longo do tempo. A iterações iniciais tendem a dar uma ênfase maior aos requisitos e ao projeto, enquanto que as últimas disciplinas dão a esses itens uma ênfase maior, à medida que os requisitos e o projeto central se estabilizam por meio de um processo de realimentação e adaptação.

Neste trabalho, O AUP foi escolhido por ser um processo ágil, cujas fases e disciplinas são de conhecimento do autor. Cabe destacar, no entanto, que por tratar-se de um trabalho desenvolvido por uma única pessoa, algumas disciplinas do AUP não serão adotadas, são elas: gerência de projetos e ambiente. Destaca-se também que a gerência de configurações e testes serão realizados de forma *ad hoc,* pelo próprio desenvolvedor, não obedecendo a uma metodologia específica. No caso dos testes, serão realizados apenas testes unitários, os quais serão efetivados em paralelo com a implementação. Mais detalhes sobre o AUP podem ser encontrados em (AMBLER, 2005).

Ao longo do desenvolvimento desse trabalho, foram gerados um conjunto de artefatos de software, conforme listado no Quadro:

Quadro 2 - Artefatos gerados ao longo das fases do AUP

|  |  |
| --- | --- |
| **Fases** | A**rtefatos** |
| ***Inception* (Concepção)** | Descrição geral do software (escopo);  Diagramas de casos de uso;  Especificação dos casos de uso (versão inicial);  Prototipação das interfaces dos usuários |
| **Elaboration (Elaboração)** | - Diagramas de classes (modelo de análise)  - Diagramas de classes (modelo de projeto)  - Especificação dos casos de uso (atualizações);  - Diagrama de entidades e relacionamento (MER)  - Descrição da arquitetura do software; |
| **Construction (Construção)** | - Código fonte; |
| **Transaction (Implantação)** | - Plano de Implantação |

## 2.3.3 UML (*Unified Modeling Language*)

A linguagem de Modelagem Unificada (do inglês: *Unified Modeling Language - UML*) é uma linguagem visual para especificar, construir e documentar os artefatos dos sistemas (LARMAN, 2007).

A UML, através de sua estrutura, conduz à criação e leitura de seus modelos, mas não determina quais e nem quando esses modelos precisam ser criados.Essa é uma responsabilidade do processo de desenvolvimento.Desta forma, pode-se usar qualquer processo com a UML (MELO, 2010).

A UML encontra-se na versão 2.4 sendo composta por treze (13) diagramas,listados no Quadro 1(FOWLER, 2005).

Quadro 3 - Exemplo de diagrama de caso de uso

|  |  |
| --- | --- |
| Diagramas Estruturais | Diagramas Comportamentais |
| Diagrama de classes | Diagrama de atividades |
| Diagrama de componentes | Diagrama de casos de uso |
| Diagrama de estruturas compostas | Diagrama de máquina de estados |
| Diagrama de implantação | Diagrama de sequência |
| Diagrama de objetos | Diagrama de comunicação |
| Diagrama de pacotes | Diagrama de visão geral da interação |
|  | Diagrama de sincronização |

É importante destacar que não é necessário que todos os diagramas da UML sejam elaborados ao longo do desenvolvimento de um software. Recomenda-se apenas criar aqueles que irão colaborar para um melhor entendimento e visão das partes do sistema. Larman (2007) afirma que os principais diagrama que devem ser produzidos em desenvolvimento iterativo utilizando UML são:

* **Diagrama de Casos de Uso:**Um caso de uso descreve uma sequência de ações que representam um cenário principal(perfeito) e cenários alternativos, com o objetivo de demonstrar o comportamento de um sistema, através de interações com atores(MELO, 2010). Um ator representa um conjunto de papéis exercido por um usuário do sistema ao interagir com um determinado caso de uso (MELO, 2010). Um caso de uso pode se relacionar com atores ou com outros casos de uso. A Figura 2 ilustra um exemplo de um diagrama de caso de uso. Neste exemplo, tem-se um ator(*público*) se relacionando com o caso de uso *Notificar acidentes não-oficiais*. Outros diagramas de casos de uso serão apresentados no Capítulo 3;
* **Diagramas de Classes:** Exibe uma visão estática das classes (seus atributos e métodos), interfaces e seus relacionamentos. Neste trabalho, os diagramas de classes serão criados na fase de elaboração e darão suporte na definição da arquitetura do sistema;
* **Diagramas de Sequência:** É um dos diagramas de interação da UML e enfatiza à ordem temporal das trocas de mensagens entre os objetos que participam de determinada interação. Neste trabalho, diagramas de sequência serão adotados apenas nos casos em que sua modelagem contribuir significativamente para o entendimento do sistema.

No Capítulo 3, os diagramas de caso de uso serão utilizados para demonstrar as funcionalidades do Sistema de Notificação de Acidentes por Arraia, assim como definir a organização dos módulos e atores do sistema.

Informações mais detalhadas sobre os diagramas de UML podem ser encontradas em (Fowler,2005;Larman,2007;Melo,2010).

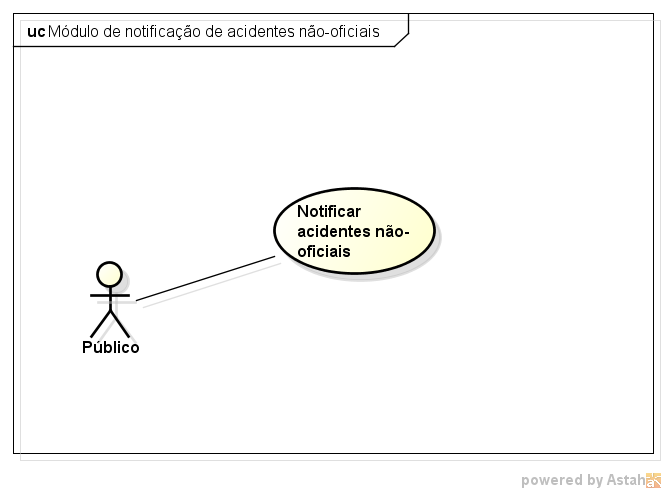


Figura 2 - Exemplo de Diagrama de Caso de Uso.

## 

## 

## 2.3.4 Plataforma Java

É uma plataforma de desenvolvimento de software criada pela *Sun Microsystems*, que inclui linguagem de programação orientada a objetos, uma coleção de APIs (classes,componentes e *frameworks*) para desenvolvimento de aplicações multi-plataforma e um ambiente de execução presente em *browsers*, *mainframes*, Sistemas Operacionais, celulares,*tablets*, cartões inteligentes,etc. O Quadro 2 lista as principais bibliotecas que o Java possui.

Quadro 3 - Bibliotecas da Plataforma Java.

|  |  |
| --- | --- |
| **Bibliotecas da Plataforma Java** | |
| JAVA SE (*Java Platform, Standard Edition*) | Inclui o ambiente de execução e as bibliotecas comuns. |
| JAVA EE (*Java Platform Enterprise Edition*) | Inclui bibliotecas voltadas para o desenvolvimento de aplicações distribuídas. |
| JAVA ME (*Java Micro Edition*) | Inclui bibliotecas para o desenvolvimento de aplicações para dispositivos móveis e embarcados. |
| JAVA CARD | Tecnologia que permite que pequenos [aplicativos](http://pt.wikipedia.org/wiki/Aplicativo) ([applets](http://pt.wikipedia.org/wiki/Applet)) baseados em [plataforma Java](http://pt.wikipedia.org/wiki/Plataforma_Java) sejam executados com segurança em [smartcards](http://pt.wikipedia.org/wiki/Smart_card) e dispositivos similares com limitações de processamento e armazenamento. |
| JAVA FX | Incluir bibliotecas para desenvolvimento de aplicações multimídia em desktop/web (JavaFX Script) e dispositivos móveis (JavaFX Mobile). |

Em 2009 a Oracle comprou a Sun, fortalecendo a marca. A Oracle sempre foi, junto com a IBM, uma das empresas que mais investiram e fizeram negócios através do uso da plataforma Java (CAELUM, 2013). Atualmente, existem grandes empresas utilizando Java, tais como Google, RedHat,HP, Nokia, entre outras.

Segundo o site do JAVA(www.java.com)[[2]](#footnote-3), ele é executado em mais de 850 milhões de computadores pessoais e em bilhões de dispositivos em todo o mundo, inclusive telefones celulares e dispositivos de televisão.

Portanto, a plataforma Java para o desenvolvimento de aplicações Web é, atualmente, uma das melhores opções, devido as suas características de independência de arquitetura de hardware e de portabilidade entre sistemas operacionais. Além disso,disponibiliza uma variedade de ferramentas de apoio, possui grande adesão na comunidade de desenvolvedores e na indústria de software.

## 2.3.5 JSF

*Java Server Faces* (JSF) é um framework de componentes de interface de usuário do lado do servidor para aplicações web baseadas em tecnologia Java (ORACLE, 2010). Essencialmente, o JSF permite-nos criar aplicações web utilizando componentes, fazendo com o que desenvolvedor se preocupe o mínimo possível com as tecnologias Java Script e HTML. O JSF é fortemente baseado nos padrões MVC e Front Controller.

O MVC (*Model-View-Controller*) é um padrão arquitetural que tem por objetivo isolar a lógica de negócio da lógica uma aplicação. Esse padrão (ou alguma variação) é amplamente adotado nas principais plataformas de desenvolvimento atuais.

A abordagem MVC separa a Visão e Modelos pelo estabelecimento de um protocolo do tipo inserção/notificação (*subscribe/notify)* entre eles. Uma visão deve garantir que sua aparência reflita o estado do modelo. Sempre que os dados do modelo mudam, o modelo notifica as visões que dependem dele. (GAMMA, HELM, *et al.*, 2000).

O padrão MVC divide uma aplicação em três tipos de componentes: modelo, visão e controlador.

* **Modelo (*Model*):** encapsula os dados e as funcionalidades da aplicação;
* **Visão (*View*):** é responsável pela exibição de informações, cujos dados são obtidos do modelo;
* **Controlador (*Controller*):** recebe as requisições do usuário e aciona o modelo e/ou a visão.

No padrão *Front Controller*, existe um controlador que processa todas as requisições que chegam ao web site . Dessa forma, esse padrão consolida toda a manipulação de pedidos através de um único objeto manipulador, possibilitando realizar o comportamento comum que pode ser modificado em tempo de execução.

## 2.3.6 Ferramenta ORM, Hibernate e JPA.

Aplicações orientadas a objetos e bancos de dados relacionais têm diferentes mecanismos para estruturar dados . Diversas partes de um objeto , como coleções e herança , não estão presentes em bancos de dados relacionais . Quando você constrói um modelo de objetos com lógicas de negócios é importante utilizar estes mecanismos para organizar melhor os dados e o seu comportamento. Desta forma, o modelo de objetos e o esquema relacional não combinam. (FOWLER, 2002)

Para facilitar a comunicação entre aplicações que seguem o modelo orientado a objetos e bancos de dados que seguem o modelo relacional são, comumente, utilizadas ferramentas que automatizam a transição de dados entre as aplicações e os SGDBs. Essas ferramentas são conhecidas como ferramentas ORM - *Object Relational Mapping.*

As ferramentas ORM oferecem mecanismos de consultas independentes da linguagem SQL .Dessa forma, o acoplamento entre as aplicações e os SGDBs diminui drasticamente. A principal ferramenta ORM para Java, utilizada no mercado, é o Hibernate. Mas, existem outras que possuem o mesmo objetivo, tais como Eclipse Link e Open JPA.

Devido ao sucesso da utilização do Hibernate, a especificação JPA (*Java Persistence API*) foi criada com o objetivo de padronizar as ferramentas ORM para aplicações Java e consequentemente diminuir a complexidade do desenvolvimento. Atualmente, essa especificação está na sua segunda versão

Neste trabalho será utilizado a implementação JPA do Hibernate devido ao fato dele ser o mais antigo e amplamente utilizado pelas empresas e corporações. As outras implementações de JPA mais conhecidas são EclipseLink e OpenJPA.

## 2.3.4 CDI

Contextos e Dependency Injection (CDI) é uma especificação para a plataforma Java EE 6 que possui características que ajudam a unir a camada web e a camada transacional da plataforma Java EE. CDI é um conjunto de serviços que, usados ​​em conjunto, tornar mais fácil para os desenvolvedores a usar objetos corporativos, juntamente com a tecnologia Java Server Faces em aplicações web.

Os serviços mais fundamentais fornecidos pelo CDI são os seguintes:

* **Contextos**: a capacidade de vincular o ciclo de vida e interações de componentes stateful aos contextos do ciclo de vida bem definidos, mas extensíveis
* **Injeção de dependência**: a capacidade de injetar componentes em uma aplicação de uma maneira typesafe, incluindo a capacidade de escolher no momento da implantação, que a implementação de uma interface específica para injetar

Além disso, o CDI oferece os seguintes serviços:

* A integração com o Expression Language (EL), que permite que qualquer componente a ser usado diretamente em uma Java Server Faces página ou uma página de Java Server Pages (JBOSS, 2014)
* A capacidade de decorar componentes injetados
* A capacidade de associar interceptores com componentes usando typesafe bindings de interceptadores
* Um modelo de notificação de eventos

A documentação dessa especificação pode ser obtida em (JCP, 2009)

## 2.3.5 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto, desenvolvido,distribuído e mantido pela *Oracle Corporation* (MYSQL, 2013).

O MySQL basea-se na SQL (*Structured Query Language*), linguagem padronizada utilizada para acessar bancos de dados. Dependendo do ambiente de programação, pode-se utilizar a SQL diretamente (por exemplo, para gerar relatórios),através de declarações de comandos SQL, ou usar uma API específica da linguagem que esconde a sintaxe SQL (MYSQL, 2013).

O MySQL é software de código aberto, pois permite a qualquer um usar e estudar o código fonte e modificá-lo para atender às suas necessidades. O MySQL usa a GPL (*GNU General Public Licence*) que define o que é possível fazer com o software em diferentes situações.

O MySQL foi adotado neste trabalho devido as características de ser de código de fonte aberto, ser multi-plataforma, possuir uma boa performance e ter uma ampla comunidade de desenvolvimento, auxiliando na melhoria e evolução da ferramenta.

## 2.3.6 Demoiselle Framework

O Demoiselle Framework é projeto iniciado pelo SERPRO (Serviço Federal de Processamento de Dados) , disponibilizado como software livre, sob a licença LGPL version 3, que implementa o conceito de framework integrador.

Seu objetivo é facilitar a construção de aplicações minimizando tempo dedicado à escolha e integração de frameworks especialistas, o que resulta no aumento da produtividade e garante a manutenibilidade dos sistema. O Demoiselle possui uma divisão baseada em funcionalidades. Isto significa que o framework não é monolítico, no qual todas as funcionalidades estão contidas em único pacote. Essencialmente, o Demoiselle é dividido em 3 (três) partes: Core, Extensões e Componentes

* **Core:** Contém as implementações das funções básicas para o funcionamento do Framework: suas classes utilitárias,controle de mensagens, exceções,etc., utiliza-se bastante dos recursos do CDI (Contexts and Dependency Injection). É também onde se encontram as definições das anotações para os estereótipos das camadas, pois o Demoiselle recomenda o uso do padrão arquitetural MVC (Modelo, Visão e Controle) no desenvolvimento de aplicações
* **Extensões:** São funcionalidades dependentes do Core, mas de uso opcional, que implementam facilitadores para utilização de tecnologias aderentes ao JEE 6 e dessa forma estendem os recursos básicos do Core. Estão disponíveis extensões para as tecnologias JSF 2, JPA 2, JTA e Java Standard Edition (SE).

Nesse trabalho, as extensões que foram usadas foram: ***Demoiselle-JPA e Demoiselle-JSF.***

* **Componentes:**  recurso opcionais que podem não depender de nenhuma característica do Demoiselle, mas que são utilizados para agregar funcionalidades de implementações de terceiros, e por isso em muitos casos são dependentes de fornecedores, e são comumente úteis para o desenvolvimento de aplicações com Demoiselle.

Neste trabalho, foi utilizado os seguintes componentes opcionais do demoiselle:

* ***Demoiselle-Report***: baseado no JasperReports, é um facilitador para manipulação de relatórios.
* ***Demoiselle-Email:*** Facilitador para envio de e-mail, baseado na API Java Email. Tem como diferencial a disponibilização de uma DSL (Domain Specific Language) que simplifica ainda mais a manipulação do componente.

A motivação para se utilizar o Demoiselle , nesse trabalho, foi justamente a capacidade de integração com as principais especificações JEE, comumente utilizadas em projeto web, tais como: JSF, JPA e CDI.

O desenvolvimento do Demoiselle é sustentado por meio da comunidade que foi criada em torno do projeto. O SERPRO, como fundador e primeiro patrocinador, mantém uma equipe de desenvolvedores dedicados à evolução da ferramenta que também é o padrão de desenvolvimento de sistemas do próprio SERPRO.

## 2.3.7 JBoss AS

O JBoss Application Server é um servidor de aplicação de código aberto baseado na plataforma JEE e implementado completamente na linguagem de programação Java.

Os modelos de componentes de aplicação incluem especificações, como Enterprise Java Beans (EJBs), Java Server Pages (JSP) e Servlets. Alguns exemplos pf serviços Java EE que estão disponíveis para estes componentes incluem comunicação remota, segurança, gerenciamento de transações, persistência, messaging, pool de recursos, controle de concorrência, nomeando e serviços de diretório, e implantação.

A versão utilizada nesse trabalho foi a versão 7.01.Final. Para mais informações consulte (JBOSS, 2014)

## 2.3.8 Apache Maven

O Apache Maven é uma ferramenta de Gerenciamento de Projetos que tenta resolver alguns problemas inerentes ao desenvolvimento de um projeto de software, tais como:

* Gerenciamento de dependências: Após baixar o projeto, é comum faltar dependências (JARs), o projeto não compila.
* Gerenciamento de construção (build): Deseja-se automatizar a construção do projeto (geração de JAR, WAR e EAR), testes, deploys etc.
* Execução de testes automáticos;
* Montar relatórios e documentação

Nesse trabalho, o maven foi utilizado para gerenciamento das dependências do projeto, em que o projeto possui um arquivo xml que contem as informações necessárias para buscar as dependências. No quadro apresenta um configuração de um arquivo pom.xml.

Quadro 4 - Arquivo pom.xml

<project>

   <modelVersion>4.0.0</modelVersion>

   <groupId>br.com.bc</groupId>

   <artifactId>meu-projeto</artifactId>

   <version>0.0.1</version>

   <packaging>war</packaging>

   <dependencies>

      <dependency>

         <groupId>junit</groupId>

         <artifactId>junit</artifactId>

         <version>4.8.2</version>

         <scope>test</scope>

      </dependency>

      <dependency>

         <groupId>org.jboss.resteasy</groupId>

         <artifactId>resteasy-jaxrs</artifactId>

         <version>2.2.1.GA</version>

      </dependency>

   </dependencies>

</project>

O maven possui uma padronização de pastas do projeto, conforme ilustra a figura, de modo que fica mais fácil a compreensão da estrutura dos projetos.

Figura 3 - Estrutura de pastas de um projeto em Maven

## http://4.bp.blogspot.com/-9ulBIh4KDYQ/TngSvhTMxdI/AAAAAAAAAng/7QhhzRloqSA/s320/maven_estrutura2.png

## 2.3.8 Jasper Reports e IReport

JasperReports é uma biblioteca para geração de relatórios de código aberto. É totalmente escrito em Java e é capaz de usar dados provenientes de qualquer tipo de fonte de dados e produzir documentos que podem ser visualizados, impressos ou exportados em uma variedade de formatos de documentos, incluindo HTML, PDF, Excel, OpenOffice e Word .

No entanto, trabalhar diretamente com o Jasper Reports é uma atividade custosa pois, basicamente, o formato dos arquivos do Jasper Reports é xml. Para contornar esse dificuldade, existem ferramentas no mercado que facilitam a confecção dos relatórios. Nesse trabalho, utilizou-se a ferramenta IReport Designer como IDE de desenvolvimento dos relatórios.

## 2.3.8 Astah Community

É uma ferramenta desenvolvida na plataforma Java para modelagem de diagramas UML. Nesse trabalho, todos os diagramas criados foram com auxílio dessa ferramenta, em que permite a exportação de imagens no formato .jpeg e .png.

# ­Capítulo 3

# SNAA

Neste capítulo são apresentados os artefatos produzidos ao longo do desenvolvimento do SNAA. Os artefatos são apresentados de acordos com as fases do *Agile Unified Process* (AUP), conforme explanado no Capítulo 2.

## 3.1 Concepção

No AUP, a fase de concepção é um passo inicial curto, para estabelecer uma visão comum e o escopo básico do projeto. Nesta fase os seguintes artefatos podem ser iniciados: descrição geral do software (escopo). diagramas e especificações de caso de uso, interfaces do usuários (prototipação), diagrama de classes (modelo de análise e de projeto) e por último o digrama de entidades e relacionamentos (modelo de banco de dados). Cada um desses artefatos serão apresentados nas próximas seções.Case ressaltar que tais artefatos são apenas parcialmente completados nesta fase.Eles foram iterativamente refinados nas iterações e fases subsequentes.

## 3.1.1 Descrição geral do software

O objetivo deste trabalho é desenvolver um sistema web para notificação de acidentes por arraias e o disponibilizar aos profissionais de saúde e à comunidade em geral e para que possam notificar as ocorrências desse tipo de acidente no estado do Amazonas.

O sistema disponibilizará os seguintes perfis de acesso: administrador,representante de unidade de saúde,profissional de saúde e público.

O *administrador* será responsável pelo módulo de gerenciamento do sistema, no qual será possível manter os cadastros básicos, necessários para o correto funcionamento dos outros módulos do sistema; criar contas de usuário; validar as solicitações para criação de conta de representante de unidade de saúde e alterar *status* dos usuários(ativo ou inativo).

O usuário com perfil de *representante de unidade de saúde* é responsável por gerenciar as contas de profissionais de saúde que estão vinculados a unidade de saúde ao qual ele é o representante. Essencialmente, a principal funcionalidade utilizada por este usuário é a criação das contas de usuário para os profissionais de saúde. Para o perfil de representante de unidade de saúde é obrigatória a criação de uma conta de usuário no sistema, a qual será solicitada por meio do preenchimento de um formulário, disponível no sistema.

O usuário com perfil de *profissional de saúde*, por sua vez, poderá registrar informações sobre acidentes,tais como, a identificação do paciente, identificação dos dados do acidente e da arraia que causou o agravo. Ao profissional da saúde, também, será disponibilizado a opção de cadastro de prontuário e tratamento do paciente. Para o profissional de saúde é obrigatório a criação de uma conta de usuário no sistema, a qual será criada e mantida pelo representante de sua unidade de saúde.

As ocorrências registradas pelos profissionais da saúde irão compor a base de dados de acidentes notificados por unidades de saúde. Estes dados, serão utilizados, por exemplo, para a geração de dados estatísticos de acidentes, os quais serão disponibilizados ao público em geral. É importante destacar que nenhuma informação privada dos pacientes será disponibilizada.Portanto, o sistema proposto deverá ser um mecanismo de auxílio aos órgãos responsáveis pelo controle e prevenção de acidentes causados por arraias nos municípios do estado do Amazonas.

Para o perfil *público*, será disponibilizado a opção de notificar acidentes não-oficiais. Neste caso, a notificação é semelhante à realizada pelo profissional de saúde: será disponibilizado um formulário para cadastro da notificação do acidente, porém sem as opções para registro de prontuário e tratamento do paciente, que são inerentes ao profissional de saúde. Cabe salientar, que para este perfil não há necessidade de criação de uma conta no sistema.

As notificações cadastradas pelo público em geral serão categorizadas como acidentes não-oficiais que, embora sem comprovação, deverão ajudar na construção de um histórico destes acidentes no Estado. Neste caso, a veracidade dos fatos precisa ser apurada com o devido acompanhamento.

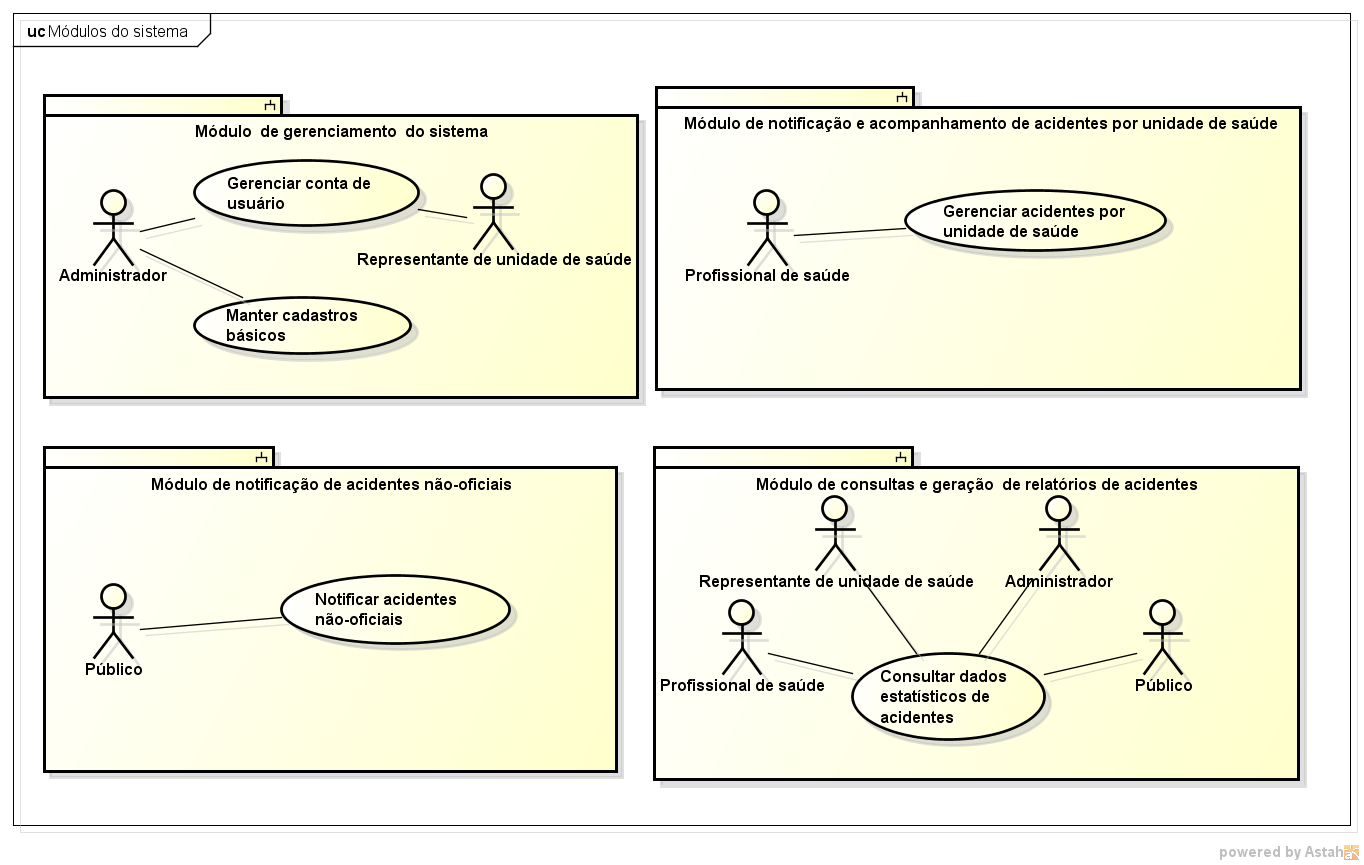
## 3.1.2 Diagramas de Casos de Uso

Conforme a definição do escopo do sistema, as funcionalidades foram organizadas em diagramas de casos de uso de uso acordo com o perfil do usuário Neste trabalho, foram criados dois tipos de casos de uso: O primeiro apresenta uma visão geral , baseada em módulos do sistema; outra apresenta um detalhamento dos casos de uso de cada módulo do sistema.

A aplicação será composta por quatro módulos descritos a seguir e ilustrados na Figura 1.

* **Gerenciamento do sistema:** engloba as funcionalidades de cadastrar e gerenciar as contas de usuário e de manutenção dos cadastros básicos do sistema;
* **Notificação e acompanhamento de acidentes por unidade de saúde:** disponibiliza aos profissionais de saúde vinculados a uma unidade de saúde as funcionalidades de pesquisar e registrar notificações de acidentes por arraias, possibilitando-lhes o registro de prontuário e procedimentos de tratamentos do paciente que sofreu o acidente;
* **Notificação de acidentes não-oficiais:** permite ao público em geral notificar acidentes que por algum motivo não foram notificados em nenhum órgão de saúde competente;
* **Módulo de consultas e geração de relatórios de acidentes:** disponibiliza consultas e emissão de relatórios dos acidentes por arraias ocorridos no estado e registrados no sistema, sejam eles notificados por alguma unidade de saúde ou pelo público em geral.

Figura 4 - Módulos do Sistema



## 3.2.1 Módulo de gerenciamento do sistema

O módulo de gerenciamento do sistema oferece duas funcionalidades: gerenciar contas de usuários e manter cadastros básicos, cujo detalhamento do diagrama de casos de uso está ilustrado na Figura 4. O detalhamento dos casos de uso para a funcionalidade gerenciar contas de usuário permite ao administrador criar novas contas de administrador ou aprovar contas para representantes de unidades de saúde.Os representantes de unidade de saúde, por sua vez, ficam responsável por criar as contas dos profissionais de saúde que trabalham em sua unidade de saúde.

Para criar um usuário com conta de administrador, deverá ser preenchido um formulário informando os seguintes dados do novo usuário: nome, CPF,RG,e-mail e número de telefone. Após o envio do formulário, o usuário cadastrado deverá ser informado através de uma mensagem de e-mail sobre a efetivação da conta.

A criação de conta de representante de unidade de saúde somente é criada através do envio de solicitação de criação de conta por um representante de uma unidade de saúde. Neste caso, o sistema disponibilizará um formulário on-line para que representantes de unidades de saúde possam efetuar essa solicitação. No formulário será necessário informar: nome da unidade de saúde, cidade em que está localizada,endereço, login e senha. Após a submissão do formulário, o administrador avaliará a solicitação podendo efetivar ou não a criação da conta. Será enviado,por email, ao solicitante a resposta com a decisão do administrador.

Os representantes de unidade de saúde têm a responsabilidade de gerenciar as contas de profissionais de saúde, isto é, cabe a eles a criação das contas dos profissionais de saúde no sistema. Para criação da conta de profissional de saúde, é necessário informar as seguintes informações: o nome,CRM (Conselho Regional de Medicina), CPF,e-mail, telefone, login e senha.

A funcionalidade manter cadastros básicos disponibiliza ao administrador recursos para manutenção dos cadastros básicos do sistema. Por exemplo, cadastro de municípios, localidades, imagens de arraias, etc.

A descrição detalhada dos casos de uso pertencentes a este módulo encontram-se no Apêndice 2.

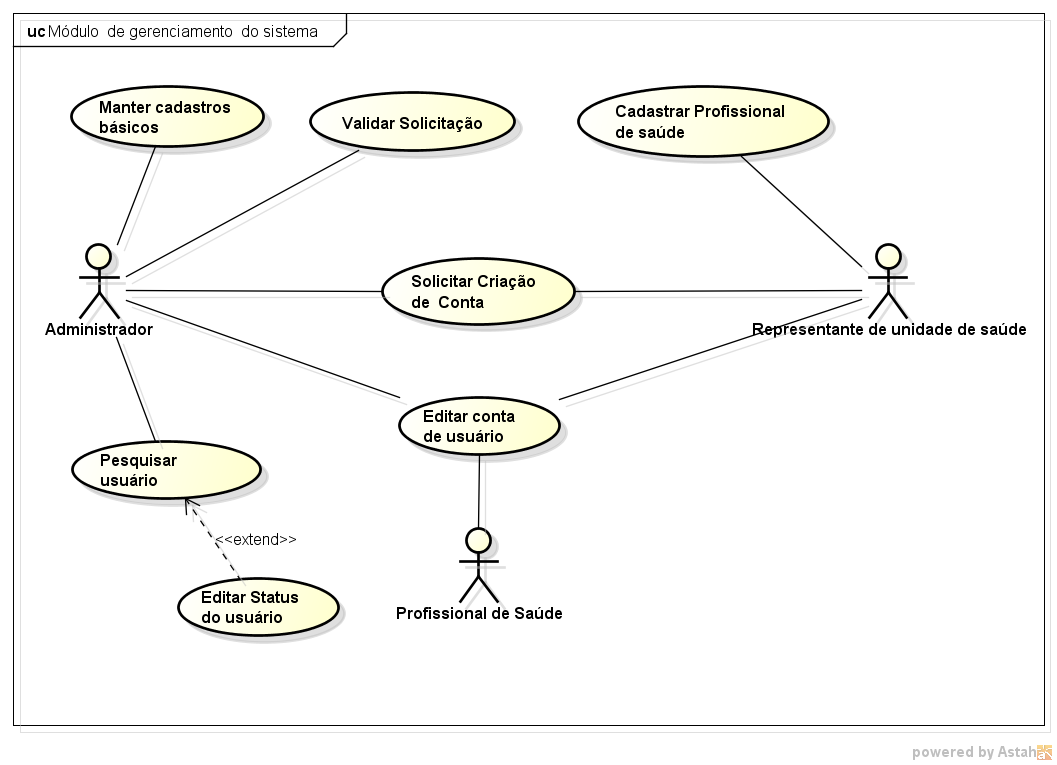


Figura 5 - Módulo de gerenciamento do sistema.

O módulo de notificação e acompanhamento de acidentes por unidade de saúde é o mais importante do sistema de notificação de acidentes por arraias e o digrama de casos de uso para este módulo é apresentado na Figura 5.

Como pode ser observado na Figura 5, é neste módulo que os profissionais de saúde irão registrar e manter dados de acidentes, além de poderem consultar dados de pacientes.

No cadastro de um acidente, o profissional de saúde deverá informar os dados pessoais do paciente, os dados do acidente,prontuário e procedimentos de tratamento do paciente.

Quanto aos dados pessoais do paciente, as informações requeridas são : CPF,RG, nome,data de nascimento,sexo, escolaridade, profissão, naturalidade e nacionalidade.

Os dados do acidente referem-se às informações sobre a localização do acidente (município, tipo de local, cor da água, etc.),informações temporais (data do acidente, horário do acidente e período ano).

No preenchimento do prontuário, o profissional de saúde deverá informara possível causa do acidente e em que parte do corpo do paciente ocorreu a ferrada da arraia. Neste último caso, a identificação ocorrerá por meio de indicação em uma imagem da parte do corpo humano no qual ocorreu a ferrada da arraia. Além disso, deverá ser registrado quais foram os sinais e sintomas do paciente, tanto locais quanto sistêmicos e também qual é a categoria de vítima(banhista, pesquisador, pescador,etc.) e qual atividade ela estava fazendo no momento do acidente (nadando, pescando, etc.).

Para registro das informações do tratamento do paciente, será necessário informar quais foram os medicamentos (com efeitos sistêmicos e locais) prescritos para o paciente e também se o paciente realizou algum procedimento caseiro na tentativa de melhorar o ferimento.No formulário de tratamento conterá informações sobre a evolução do paciente e a possibilidade de sinalizar se o paciente ficou internado ou não, e se ficou internado indicar por quanto tempo. Ao final é necessário informar qual é o quadro clínico de saída do paciente.

Quanto as consultas, serão disponibilizados para o profissional de saúde recursos para pesquisa de pacientes registrados no sistema e seus respectivos acidentes. Para efetivar uma consulta, o profissional de saúde deverá informar os dados pessoais do paciente.A partir dos resultados da consulta, o profissional de saúde poderá efetuar atualizações das informações do acidente.

Esse módulo possuirá uma funcionalidade na qual será possível sinalizar através da escolha de imagens qual foi o animal causador do acidente, levando em consideração uma escala de certeza.

No Apêndice 1, 2 e 3 é apresentada uma descrição detalhada dos casos de uso apresentados na Figura 6.

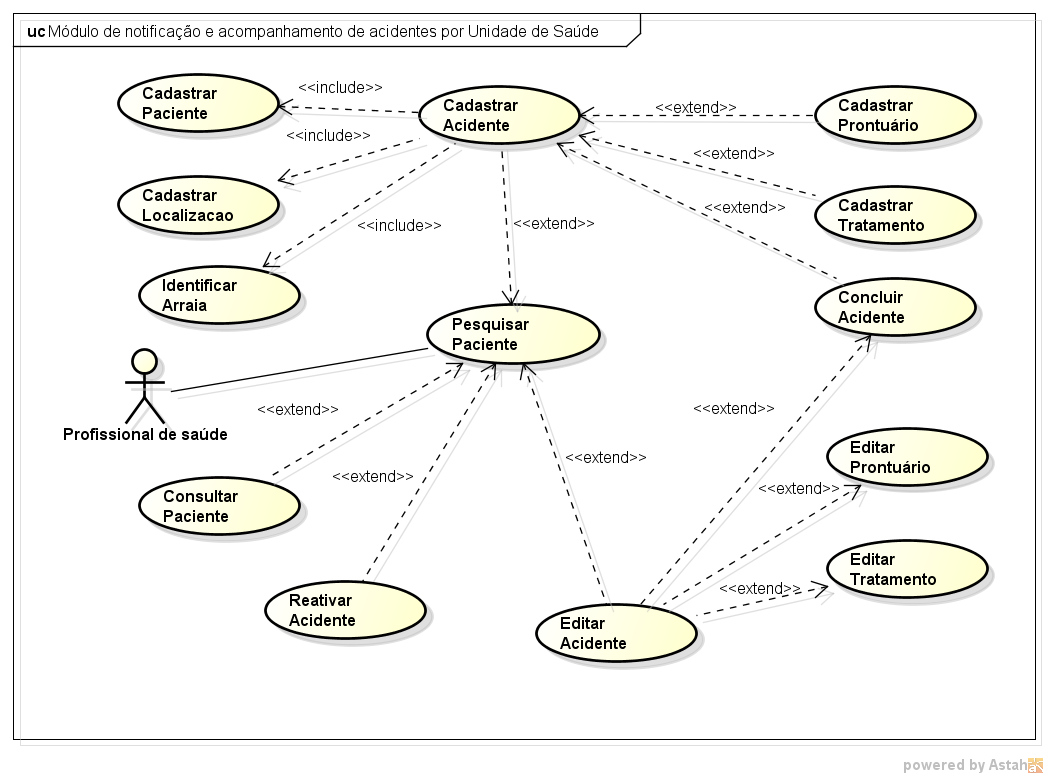
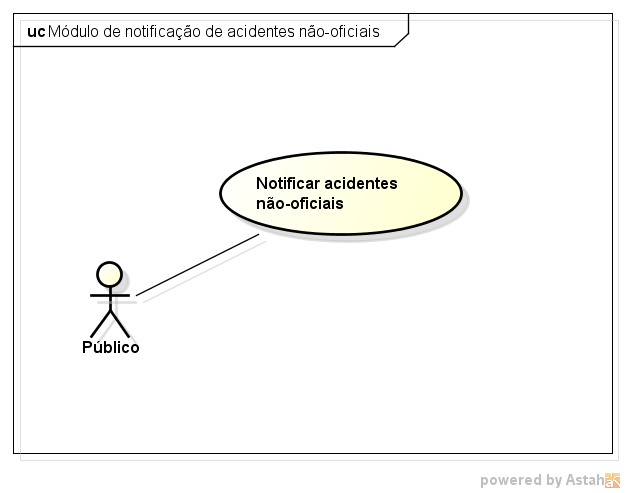


Figura 6 - Módulo de notificação e acompanhamento de acidentes por unidade de saúde.

O sistema disponibilizará um formulário para que seja possível registrar acidentes não-oficiais, isto é, aqueles acidentes que por algum motivo não foram registrados por órgãos competentes de saúde no momento em que ocorreram. Basicamente, será semelhante ao formulário que os profissionais da saúde realizam para a notificação de acidentes oficiais; exceto pelo fato de que é a própria vítima do acidente (na Figura 6 identificada pelo ator Público) que faz a notificação e, por isso, não há o preenchimento de formulário de prontuário e tratamento.

Figura 7 - Módulo de notificação de acidentes não-oficiais.

­­­­­

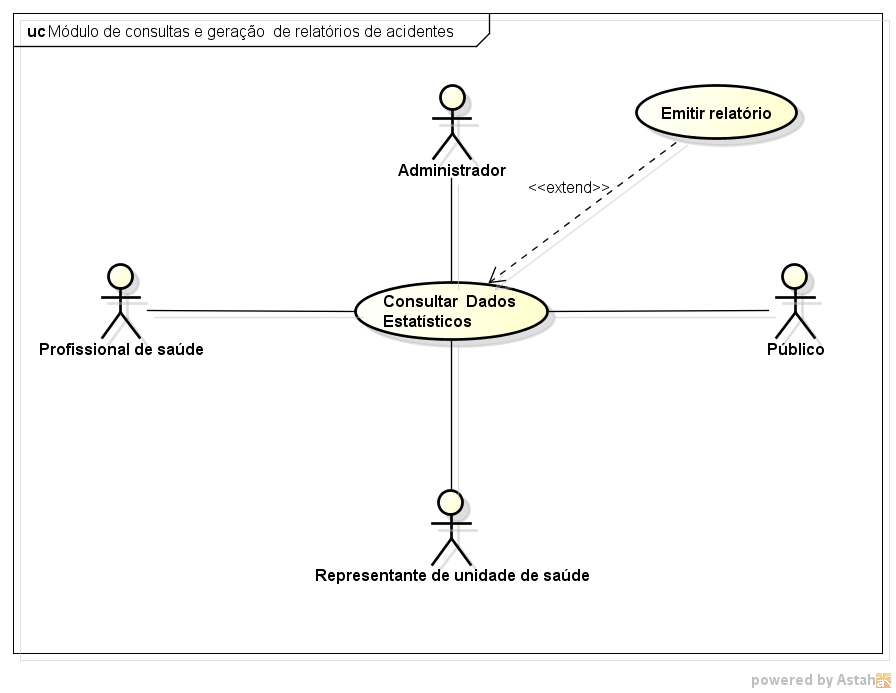
 ­O módulo de consultas e geração de relatórios de acidentes disponibilizará tanto ao público em geral quanto aos profissionais de saúde informações estatísticas sobre os acidentes ocorridos no estado. O sistema disponibilizará um conjunto pré-definido de opções de consultas (por exemplo, consulta de município, tipo de arraia, local anatômico da ferrada, etc.). Os resultados das consultas serão exibidos na forma de relatórios. Estes relatórios mostrarão dados quantitativos dos acidentes de acordo com as opções de consultas selecionadas. Dentre os relatórios básicos a serem emitidos pelo sistema estão: quantidade de acidentes em cada município do estado do Amazonas.

Figura 8 - Módulo de consultas e geração de relatórios de acidentes

## 3.1.3 Interface dos usuários

Esse seção apresenta as interfaces do usuário baseadas na funcionalidades existentes de cada usuário no sistema. Cabe ressaltar que as interfaces mostradas nessa seção já são as implementadas no sistema, pois, devido ao fato da utilização de um processo iterativo e incremental, os esboços iniciais de interface sofreram modificações ao longo do desenvolvimento para que a apresentação das informações fossem mais claras para usuário final. No entanto, no Apêndice 4 pode-se visualizar os esboços iniciais das interfaces.

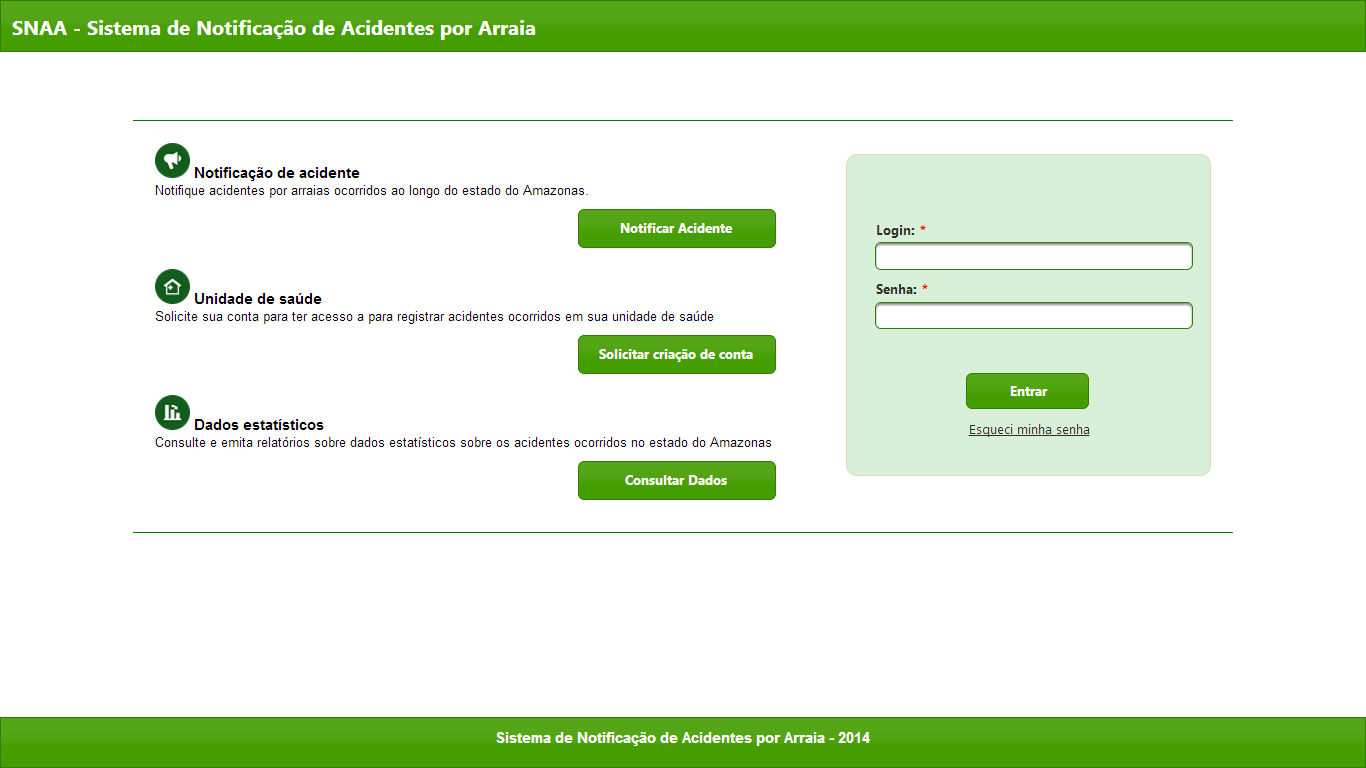
 Inicialmente, é necessário existir uma conta de representante de unidade de saúde, o qual terá a responsabilidade de criar as contas dos profissionais de saúde. A conta de representante de unidade de saúde somente pode ser adquirida através da opção *Solicitar Criação de Conta* na página inicial do sistema (Figura 9). A partir desta opção será exibido um formulário de envio de solicitação para criação da conta (Figura 10).

Figura 9 - Tela Inicial do Sistema

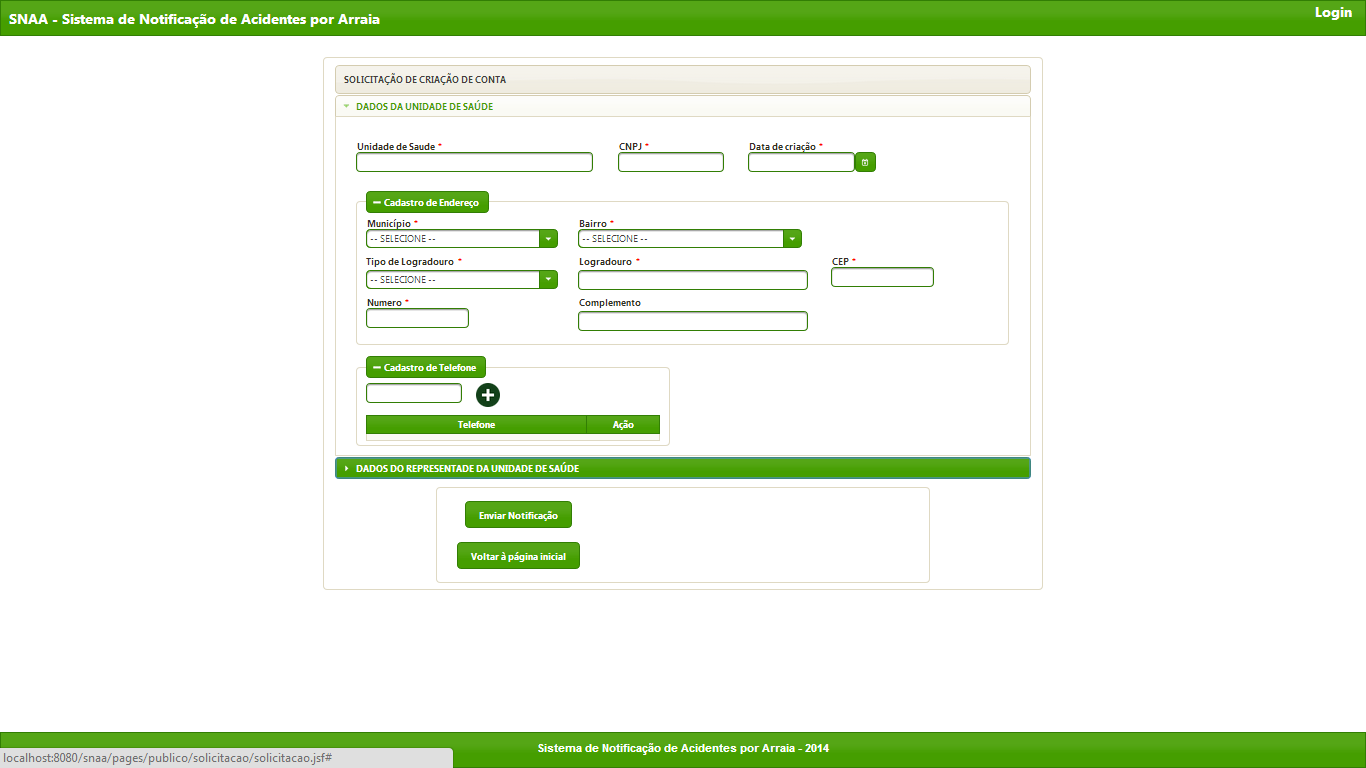
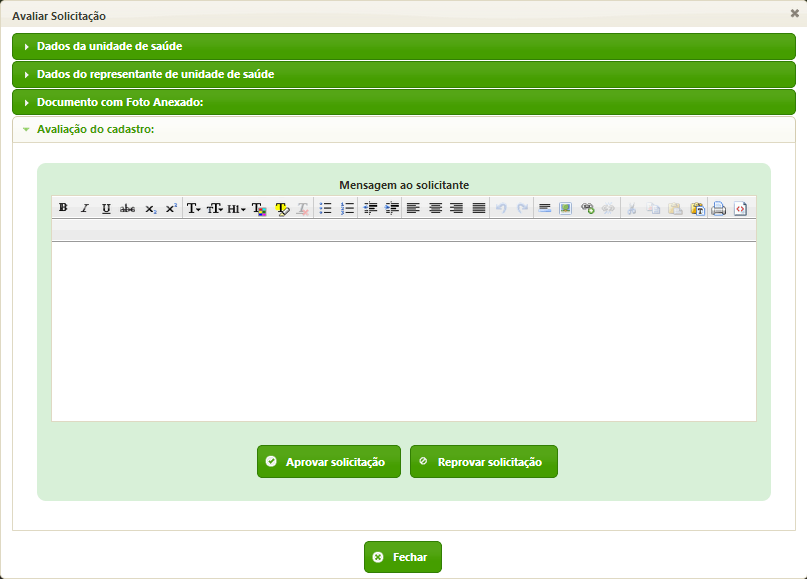


Figura 10 - Solicitação de criação conta

Após o envio da solicitação, o administrador deverá validá-la, enviando sua resposta através de uma mensagem de e-mail (Figura 11). Se a resposta for positiva para a criação da conta, a conta será criada e o representante da unidade poderá efetuar login na página inicial do sistema e criar as contas dos profissionais de saúde.

Figura 11 - Validar Conta de Usuário



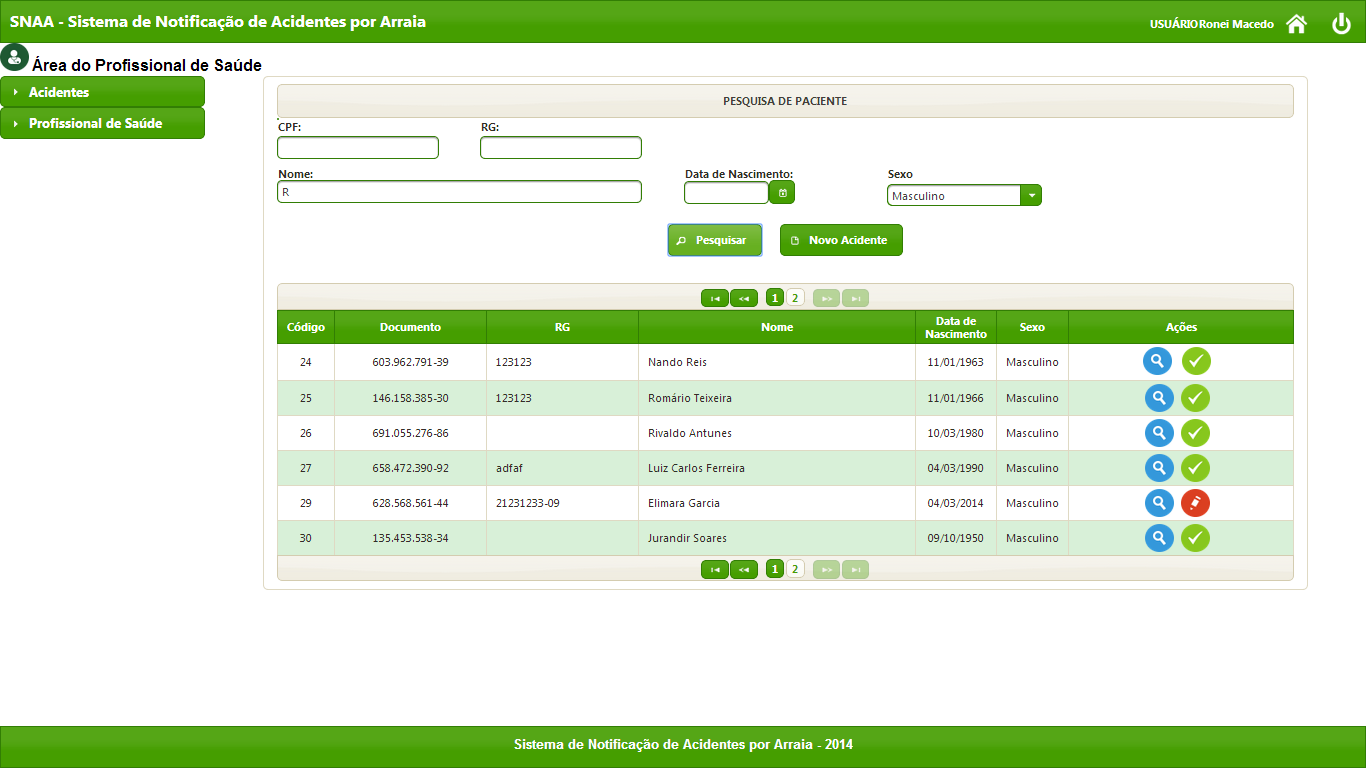
Ao acessar a página inicial de seu perfil, o representante de unidade de saúde poderá criar as contas de seus profissionais de saúde através da opção *Cadastrar Profissional de saúde* (Figura 12). Para a criação da conta de profissional de saúde, é necessário informar as seguintes dados:

Figura 12 - Cadastrar Profissional de Saúde



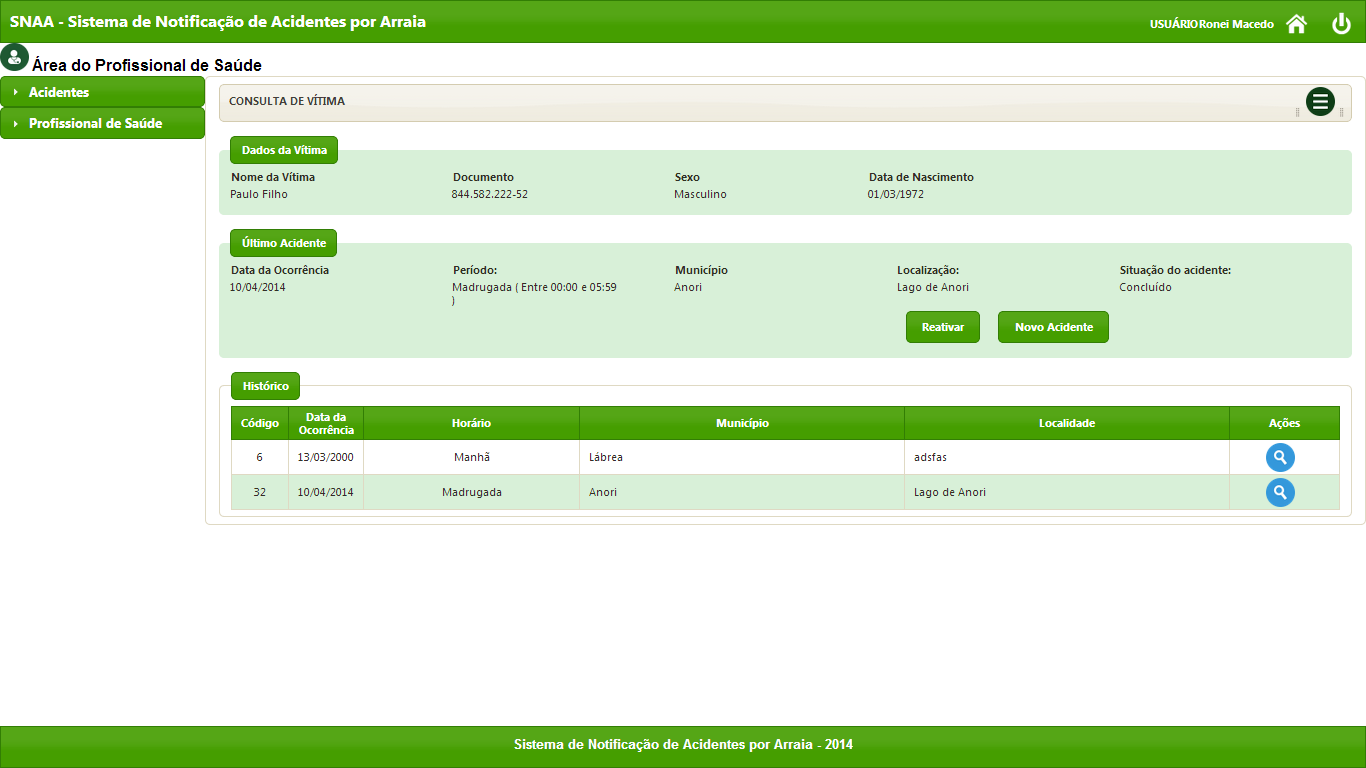
Após o profissional de saúde efetuar no login no sistema, será apresentada a tela inicial de seu perfil (Figura 13). Poderá efetuar pesquisas de pacientes, no qual lhe permitirá escolher as opções de *Visualizar Paciente, Alterar Acidente ou Reativar Acidente*. O profissional também poderá criar uma nova notificação de acidente por arraia através da opção *Cadastrar acidente*.

Figura 13 - Pesquisar Pacientes



Na opção *Visualizar Paciente* será mostrado os dados cadastrais do paciente e a listagem do histórico dos acidentes que ele sofreu(Figura 14).

Figura 14 - Visualizar Paciente



Para o cadastro de um novo acidente será exibido com o formulário no qual deve ser informado os dados do paciente (Figura 15) , dados do acidente (Figura 16) . Após efetivar o cadastro (Figura 17), o sistema redirecionará para tela de edição no qual será possível registrar informações específicas do acidente.

Figura 15 - Dados do Paciente

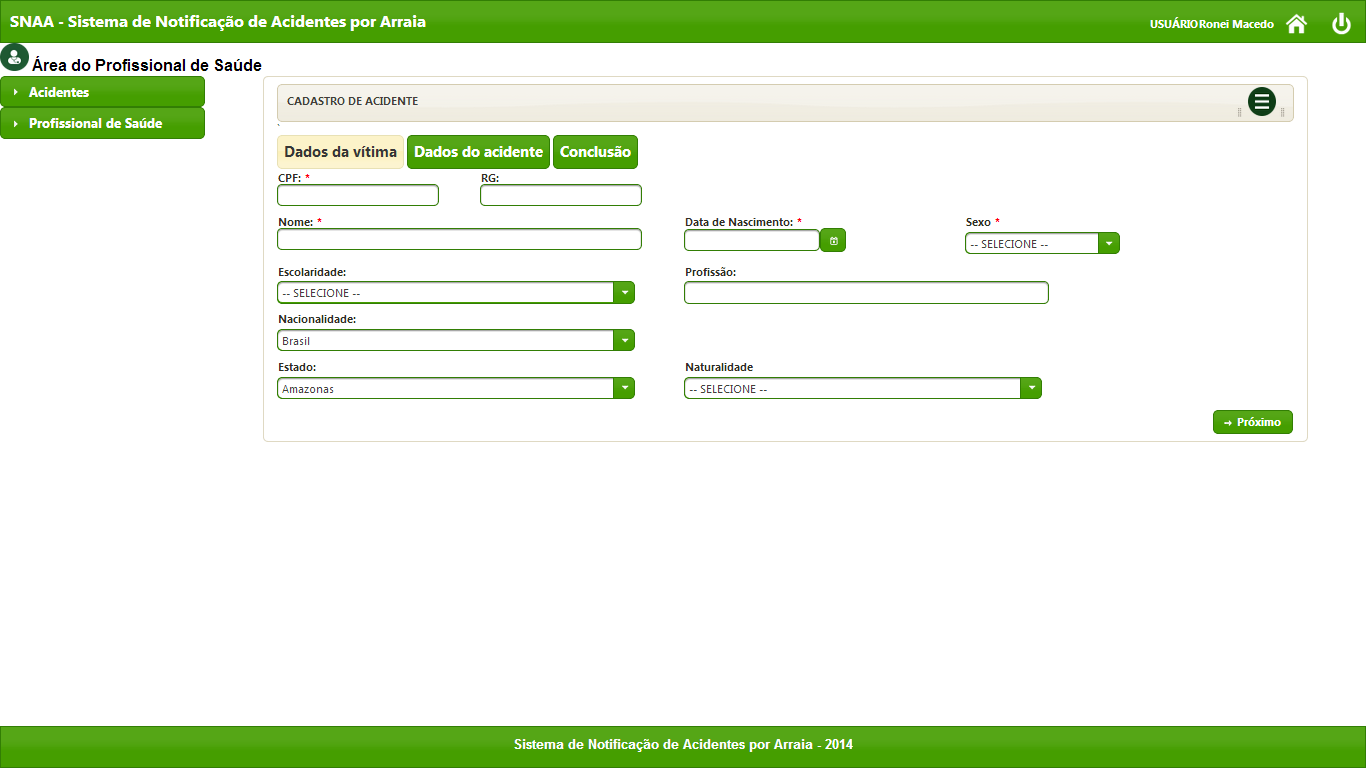


Figura 16 - Dados do acidente



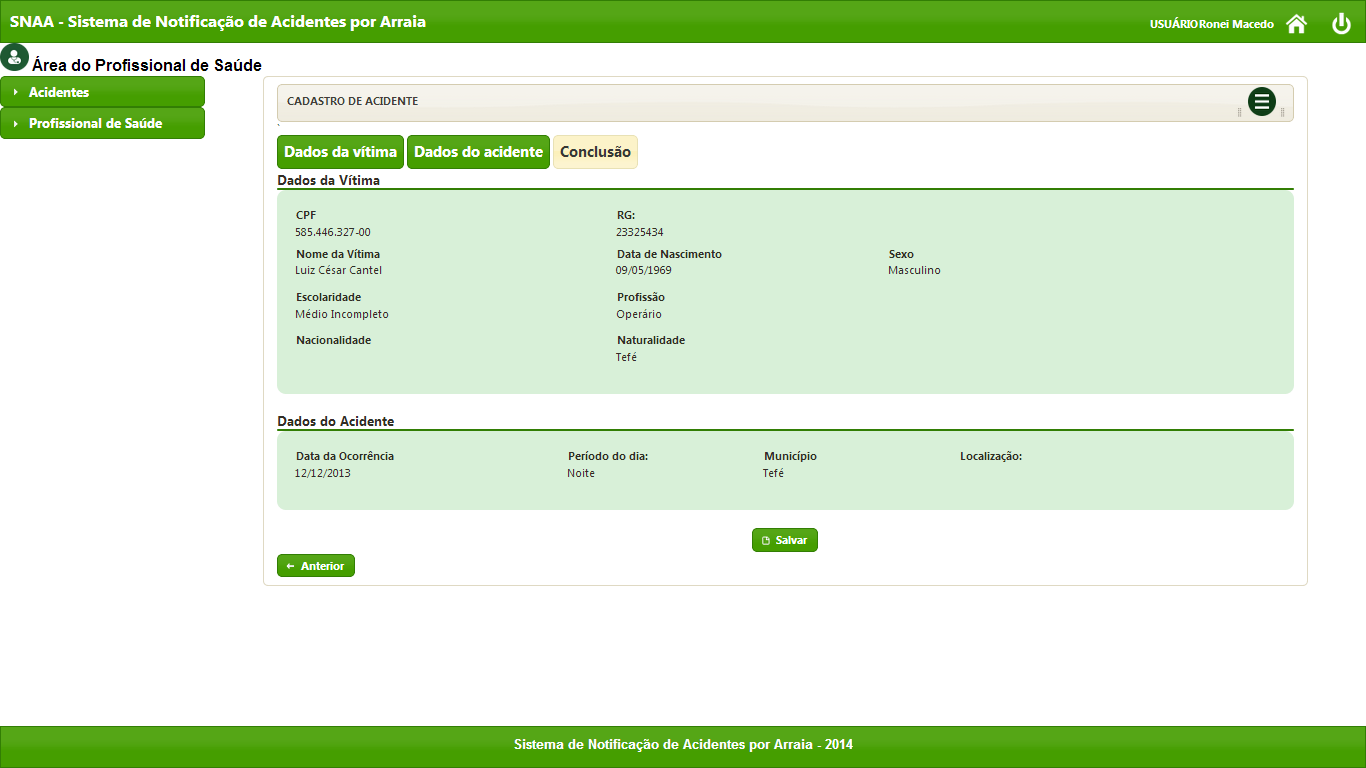


Figura 17 - Conclusão do cadastro do acidente

A opção *Editar Acidente* exibirá um painel com os principais dados do acidente e disponibilizará as opções de inclusão de informações sobre o acidente dividas em abas logo abaixo do painel. As opções são:

* *Dados do Paciente -* (Figura 18) ;
* Dados do acidente - (Figura 19)
* *Dados da arraia -* (Figura 20)*;*
* *Editar prontuário-* (Figura 21)*;*
* *Editar Tratamento-* (Figura 22)*;*
* *Concluir Notificação do acidente-* (Figura 23).

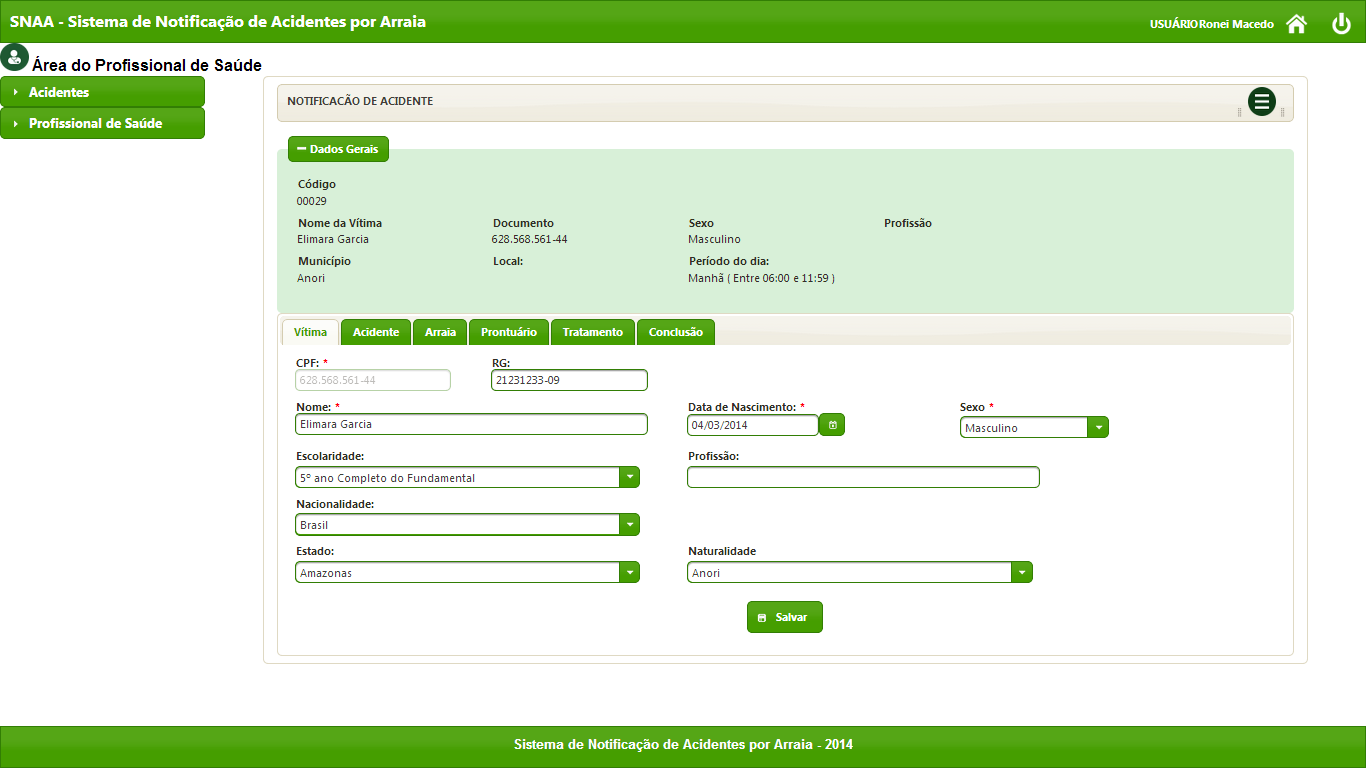


Figura 18 - Dados do paciente

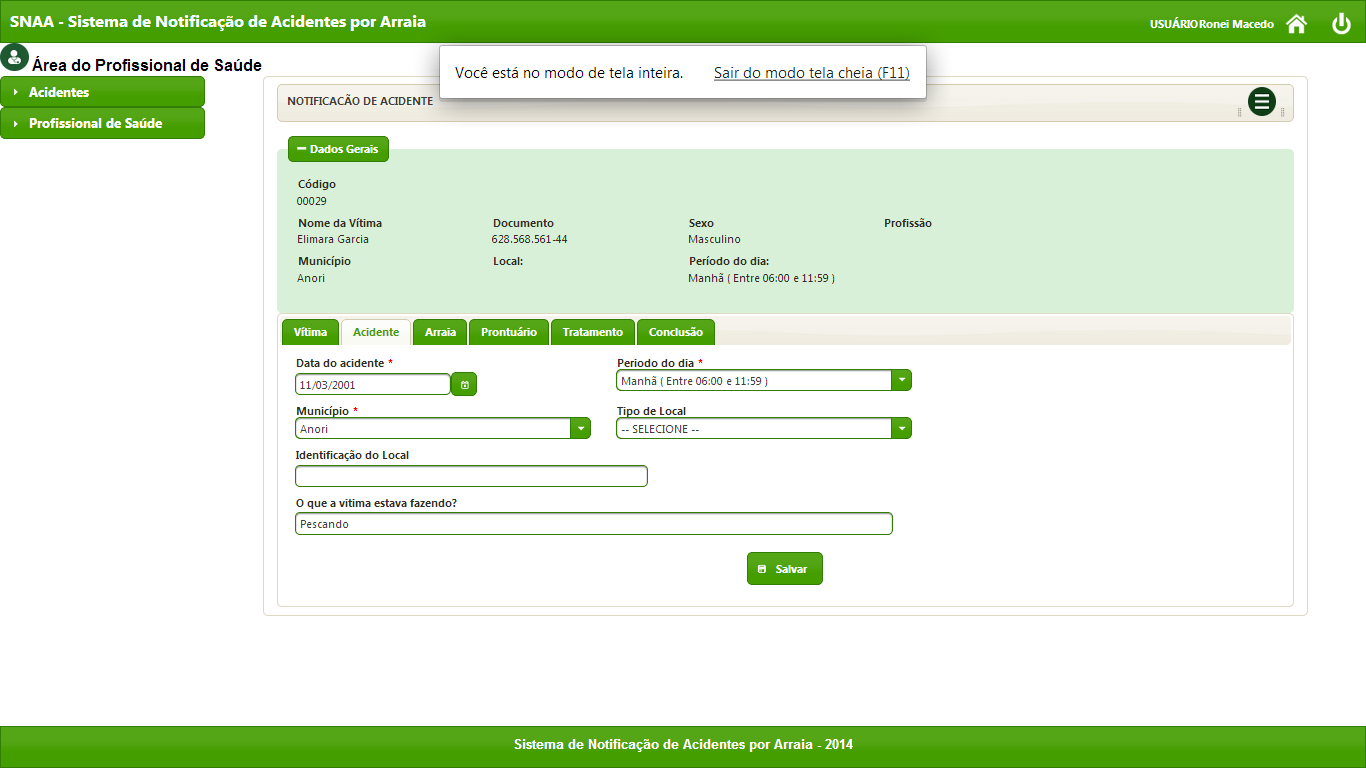


Figura 19 - Dados do acidente

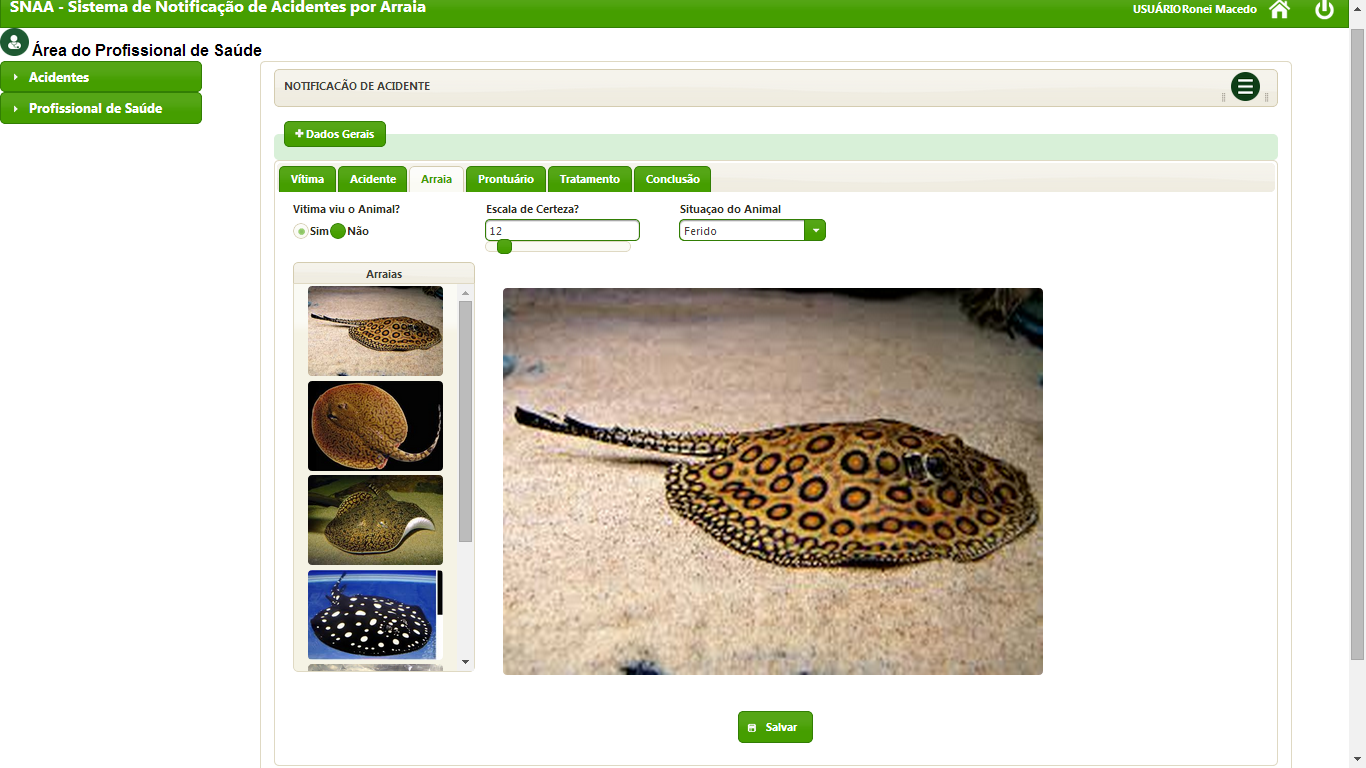
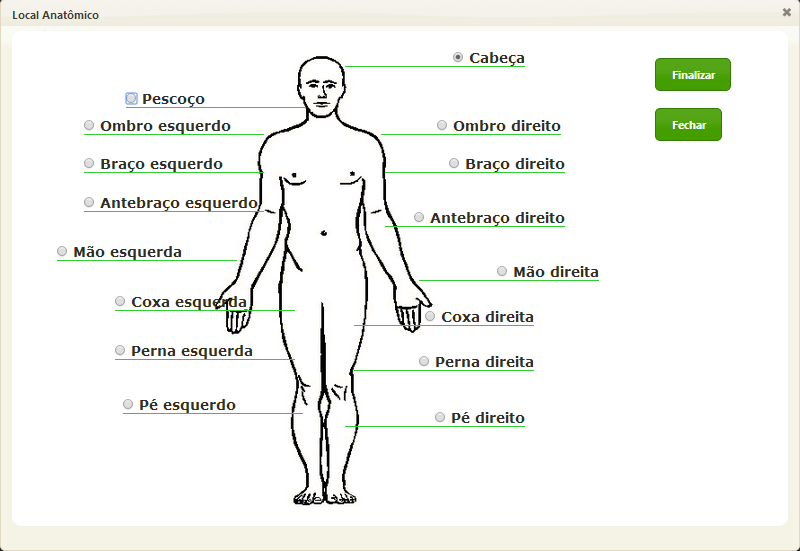


Figura 20 - Dados da Arraia

A opção *Editar prontuário* disponibilizará ao profissional de saúde um formulário para registrar as informações do sinais e sintomas do paciente tanto locais quanto sistêmicos ao chegar em sua unidade de saúde. Possibilitará a identificação do local anatômico da picada de forma interativa através de apontamento da parte do corpo em imagens (Figura 21).

Figura 21 - Identificar o local anatômico da picada



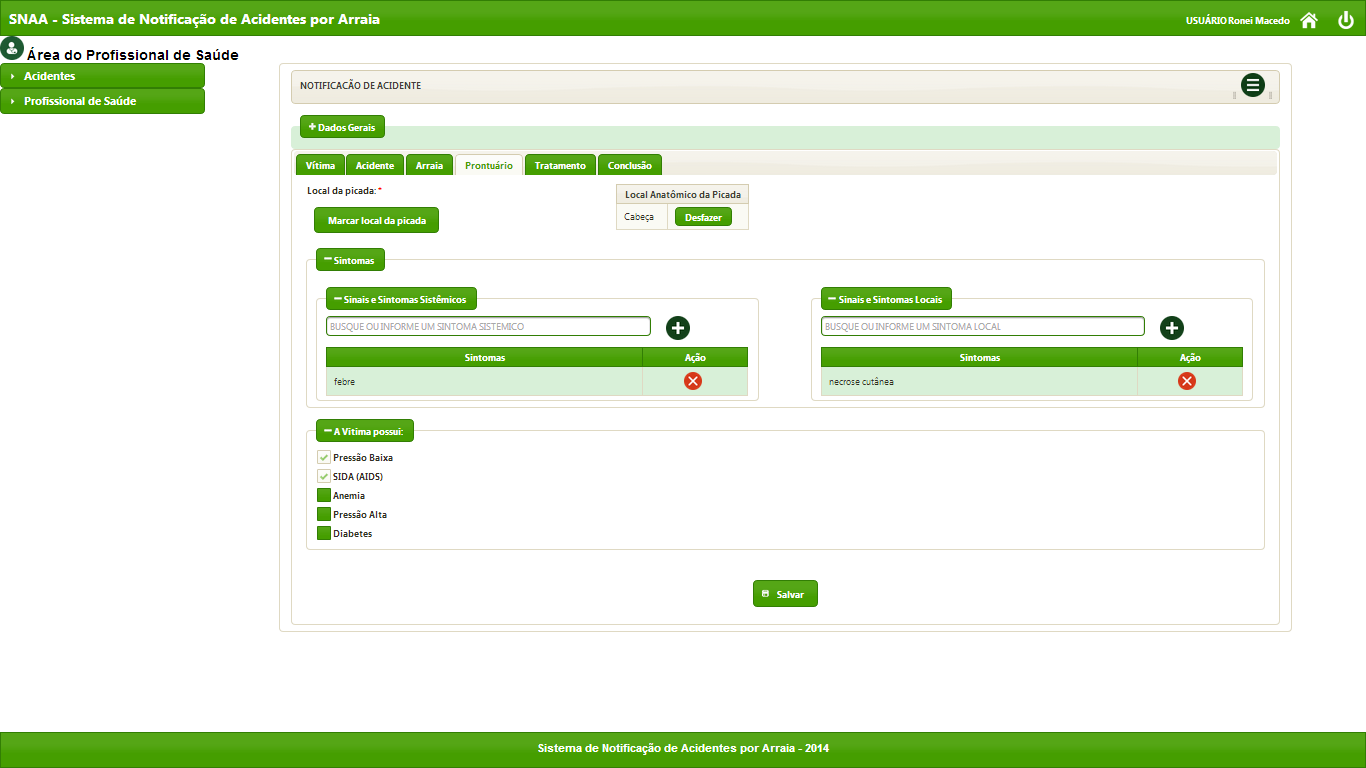


Figura 22 - Editar prontuário

A opção *Editar tratamento* permitirá registrar as informações dos procedimentos de tratamento a que o paciente foi submetido seja eles receitados por um médico ou simplesmente utilizado pelo paciente sem prescrição médica.O Profissional de saúde também informará algumas informações sobre exames laboratoriais.

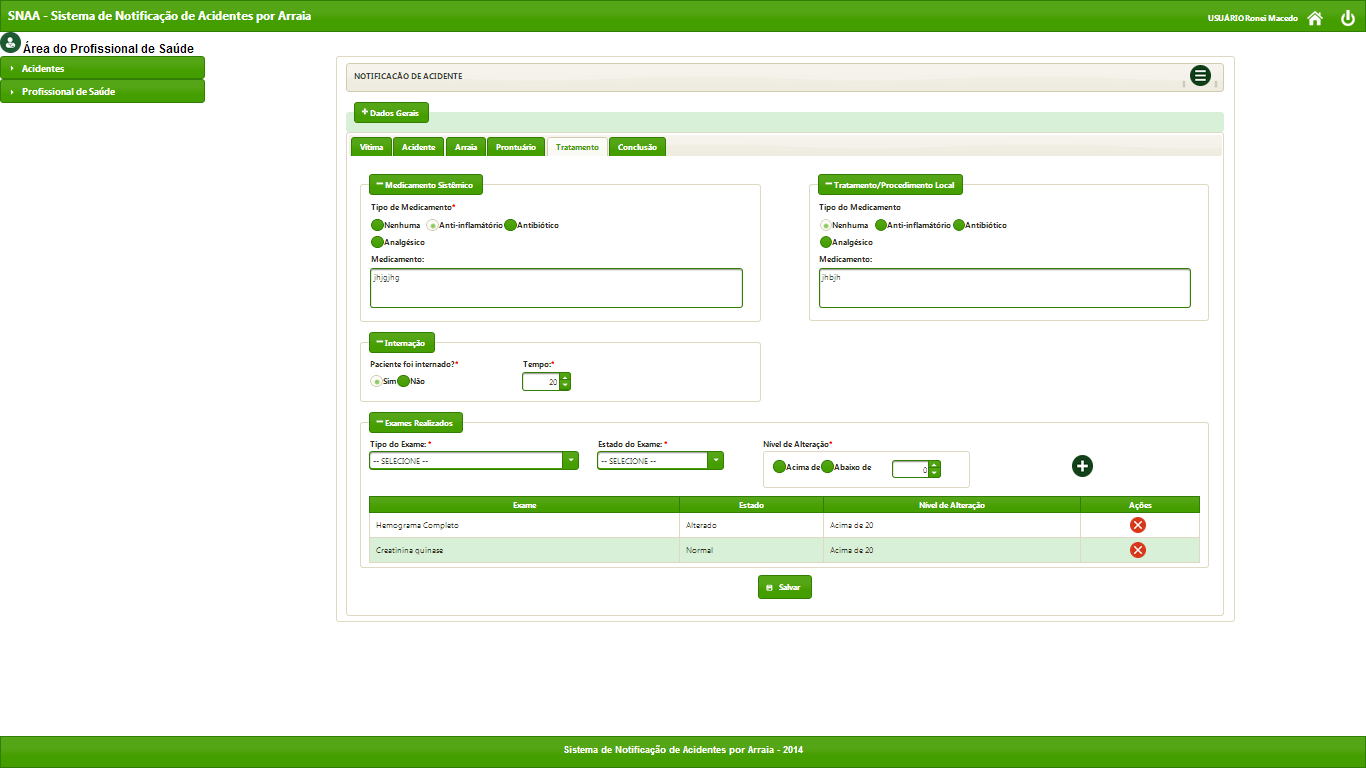
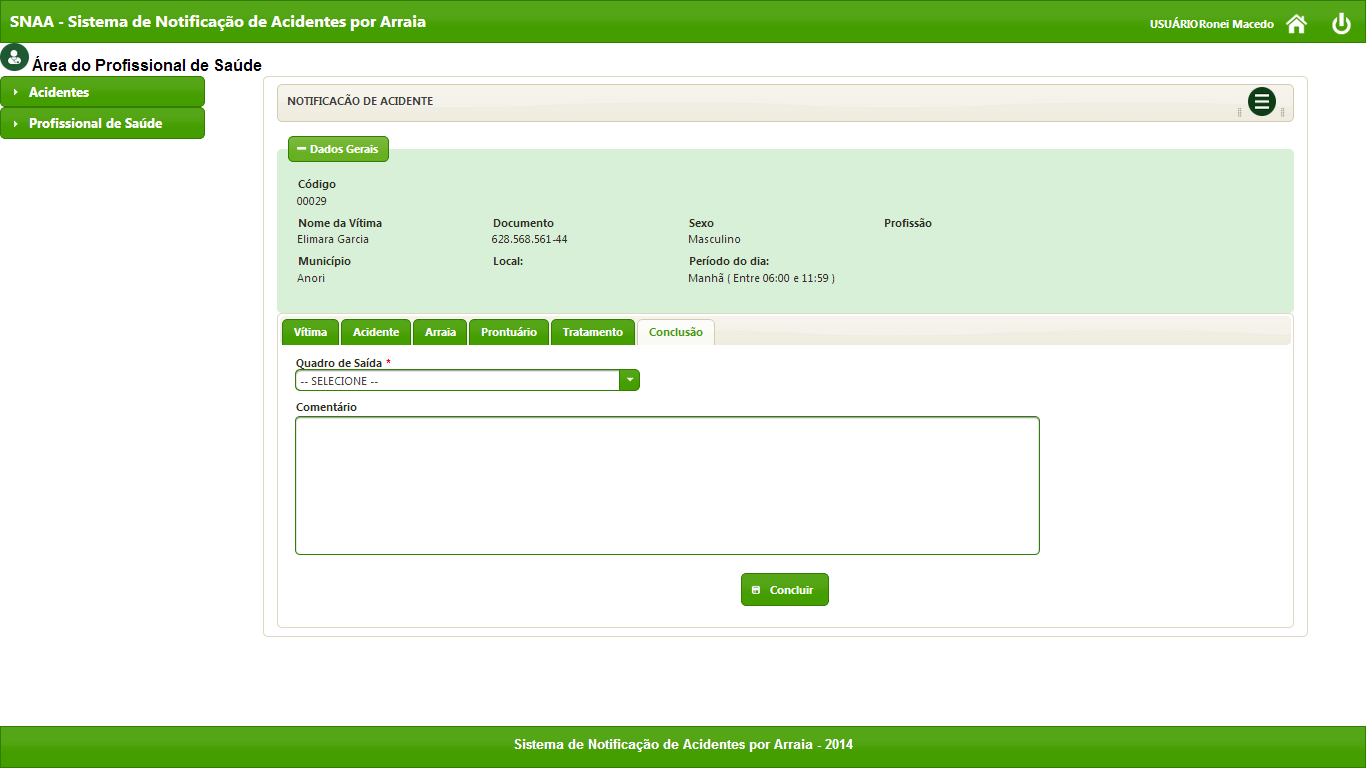


Figura 23 - Editar Tratamento

Na opção *Concluir acidente* deverá ser informado qual foi e quadro de saída do paciente e o profissional informará um comentário se achar relevante. Ao realizar conclusão o acidente passa a ser um acidente concluído.

Figura 24 - Concluir acidente



O profissional de saúde, assim como os outros atores que interagem com o sistema, pode consultar dados estatístico dos acidentes ocorridos ao longo do tempo no estado do Amazonas.

O profissional pode consultar através da opção Consultar Dados Estatísticos na página inicial do sistema.Será mostrado um formulário de pesquisa (Figura 22) com diversos filtros podem ser combinados para gerar a consulta quantitativa dos acidentes. Após gerar a consulta irá aparecer opção *Emitir Relatório,* em que irá emitir a consulta realizada em formato PDF (Figura 23).

Figura 25 - Consulta de dados estatísticos

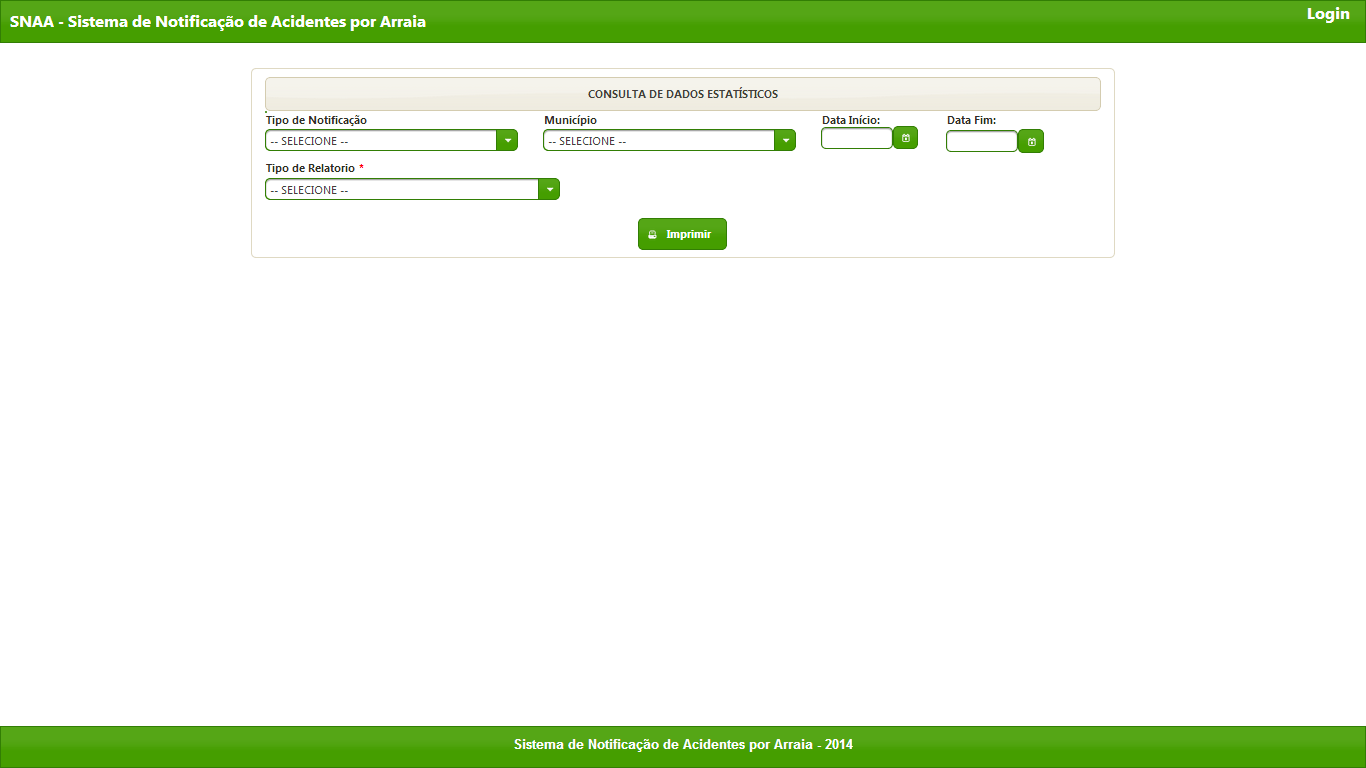
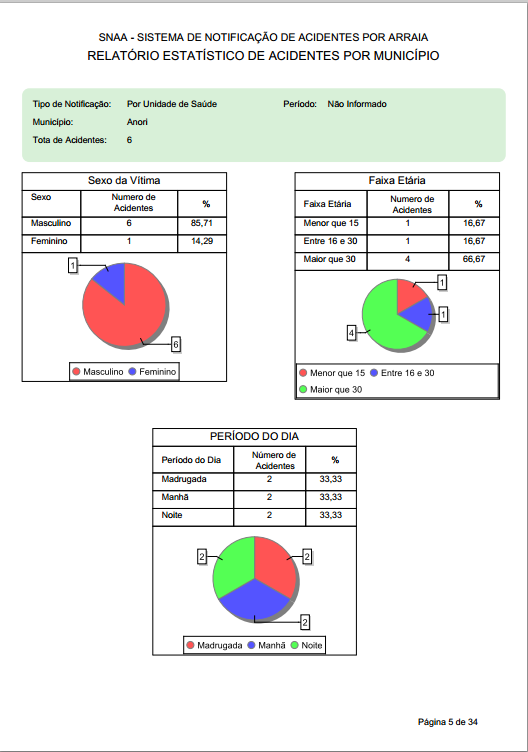


Figura 26 - Relatório emitido pelo sistema



## 3.1.4 Diagramas de classes

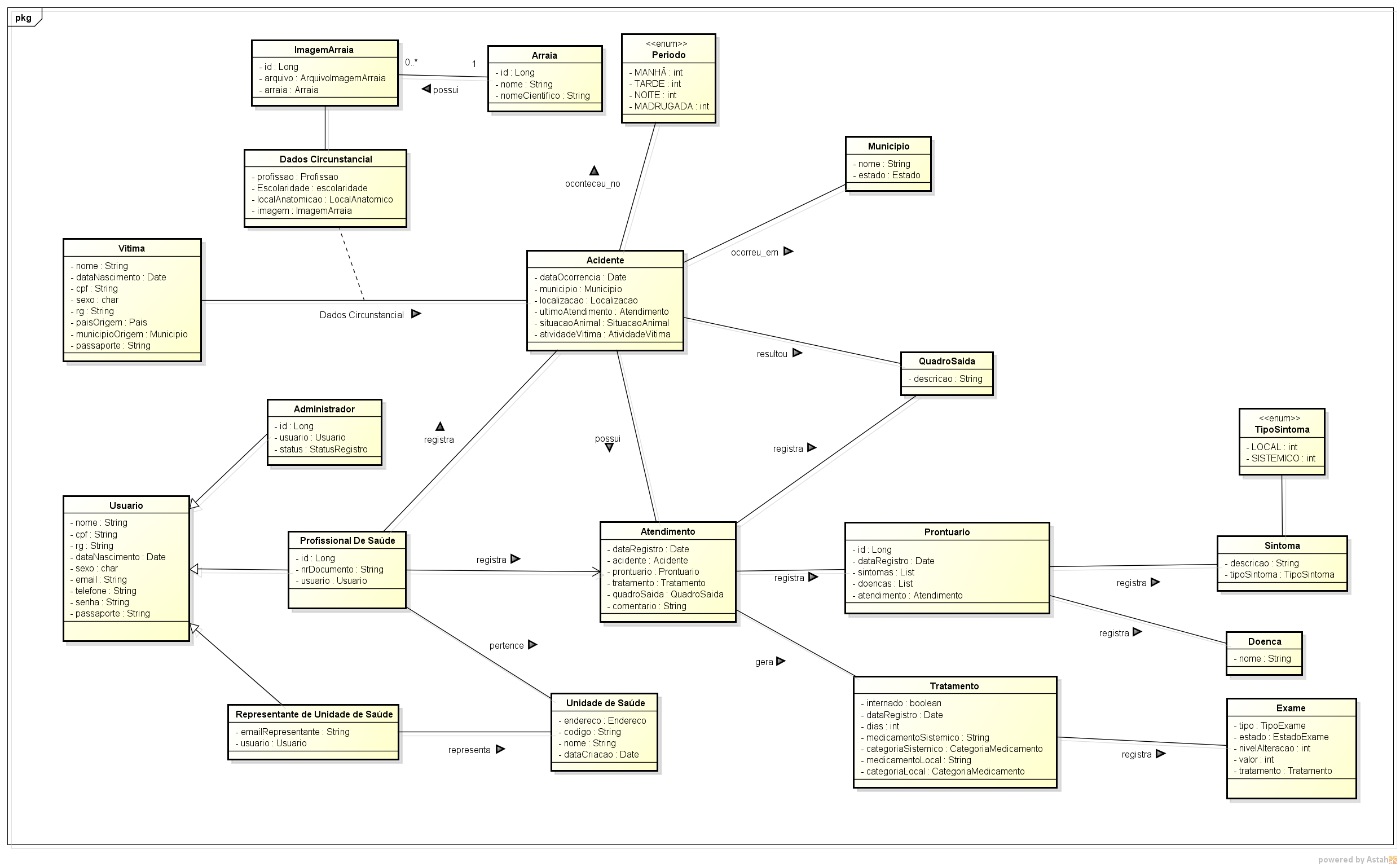
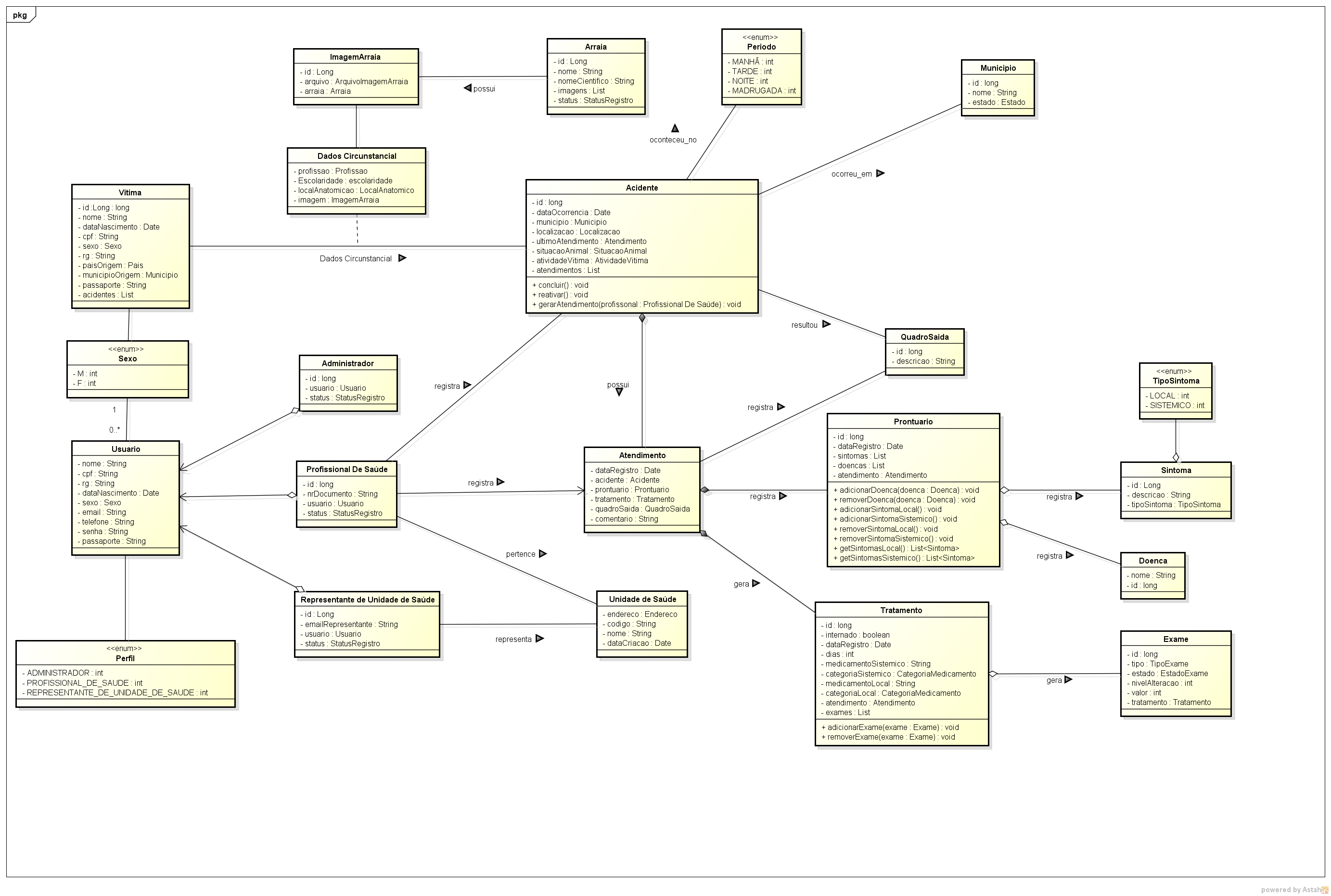
Nesta seção são apresentados os diagramas de classes em duas visões: a de análise e a de projeto: A de análise (modelo de domínio) permite observar a modelagem dos conceitos e atributos relevantes ao domínio do problema. Portanto, ilustra importantes conceitos em um determinado domínio. O diagrama de classes na visão de análise também é conhecido por *perspectiva conceitual.*

Figura 27 - Diagrama de classe na visão de análise

O diagrama de classes na visão de projeto (Figura 28), por sua vez, demonstra aspectos de implementação, levando em consideração a tecnologia subjacente.

Figura 28 - Diagrama de classes na visão de projeto



Observando os dois diagramas na disposição e na quantidade de classes. Essa diferença é ocasionada pela utilização das tecnologias de implementação e padrões de projetos empregados como: framework de segurança (Demoiselle Framework) , mapeamento objeto-relacional (Hibernate) e padrão de projeto Value Object.

## 3.1.5 Diagramas de entidades e relacionamentos

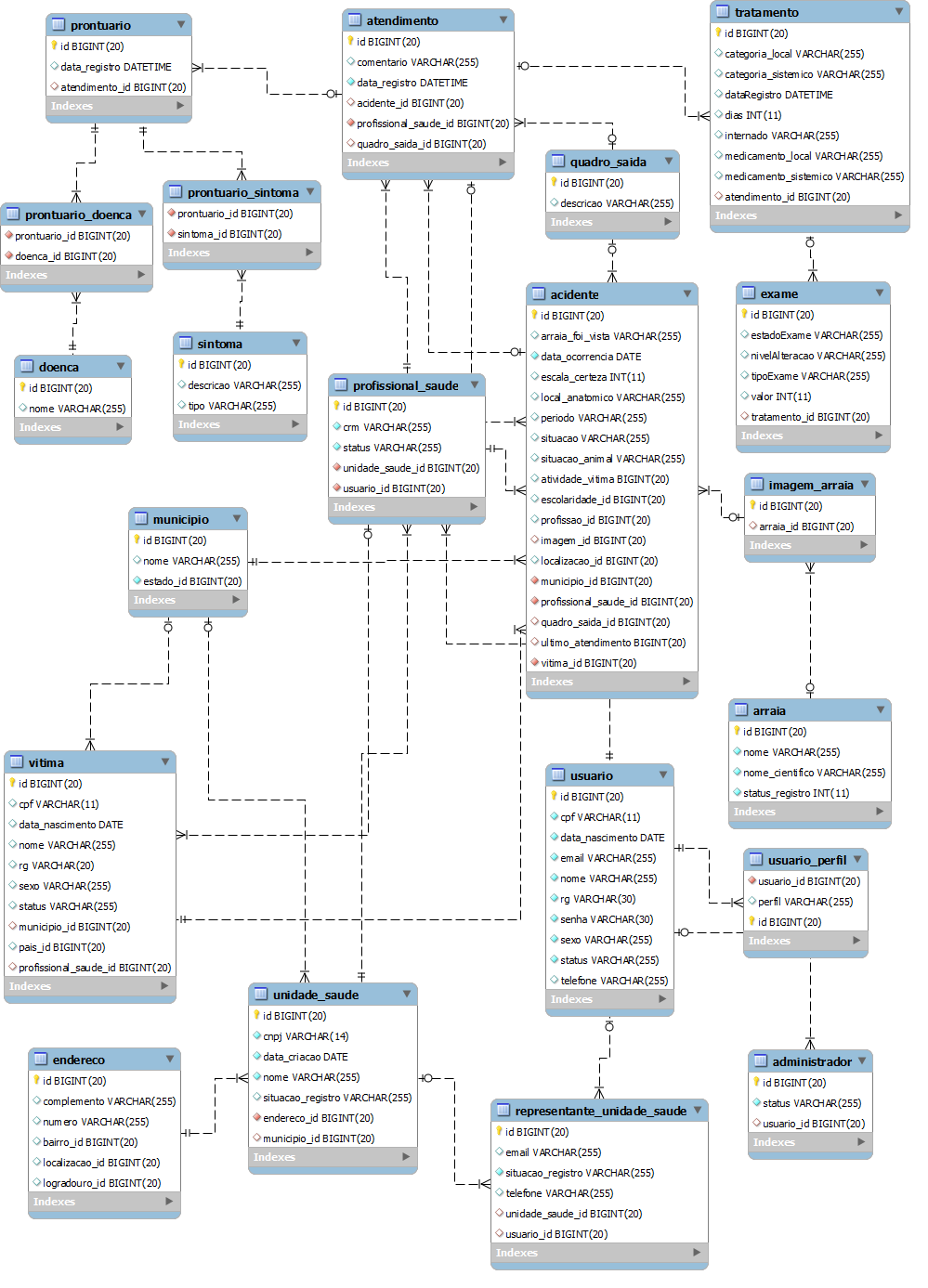
O diagrama de entidades e relacionamentos permite visualizar o esquema do banco de dados do SNAA. Em relação ao diagrama de classes, pode-se perceber que cada entidade do banco de dados possui um correspondente classe no diagrama de classe na visão do projeto. Essa correspondência é devido ao mapeamento objeto-relacional realizado pelo *framework JPA*, ou seja, o framework é responsável por gerar as entidades no banco de dados. O banco de dados foi modelado para seguir a normalização denominada a terceira forma normal, reduzindo a redundância e possíveis chances deles se tornarem dados inconsistentes.

Figura 29 - Diagrama de entidade-relacionamento

## 3.1.5 Elaboração

Em concordância com o que foi anteriormente descrito, a fase de elaboração inicia a construção da arquitetura do software. A arquitetura adotada neste trabalho está baseada no padrão arquitetural denominado MVC (Model-View-Controller).

Figura 30 - Padrão Arquitetural MVC



A abordagem MVC é composta por três tipos de objetos. O Modelo é o objeto de aplicação, A Visão é apresentação na tela e o Controlador é o que define a maneira como a interface do usuário reage às entradas do mesmo. O padrão MVC separa esses objetos em camadas para aumentar a flexibilidade e reutilização.

Abordagem MVC separa Visão e Modelos pelo estabelecimento de um protocolo do tipo de inserção/notificação (*subscribe/notify*) entre eles. Uma visão deve garantir que a sua reflita o estado do modelo (GAMMA, HELM, *et al.*, 2000). Sempre que os dados do modelo mudam, o modelo notifica as visões que dependem dele. Em resposta, cada visão tem a oportunidade de atualizar-se. Esta abordagem permite ligar múltiplas visões a um modelo para fornecer diferentes apresentações. A figura 14 exemplifica tal conceito, em que mostrar um modelo que contém alguns valores e suas visões são: uma planilha, um histograma e um gráfico de pizza.

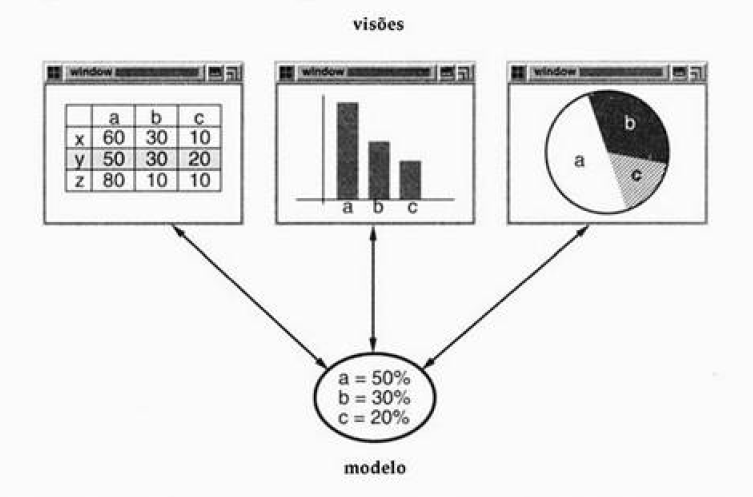


Figura 31 - Modelo com várias visões

A arquitetura MVC adotado foi a baseada no Demoiselle Framework, em que utiliza-se de controladores para identificar cada camada da arquitetura da aplicação. É comum que as aplicações utilizem apenas três camadas: visão, negócio e persistência. Existem aplicações que utilizam fachadas. Por esse motivo, foram implementados nessa versão do framework quatro controladores:

|  |  |
| --- | --- |
| Elemento | Descrição |
| Controller | Possibilita a criação de outros controladores de camada além dos pré-estabelecidos pelo framework |
| ViewController | Define a classe como um controlador da camada de apresentação |
| FacadeController | Define a classe como um controlador de fachada |
| BusinessController | Define a classe como um controlador da camada de negócio |
| PersistenceController | Define a classe como um controlador da camada de persistência |

## 3.1.7 Construção

A fase de construção corresponde à implementação do código propriamente dito.Nessa fase foi realizada a codificação dos seguintes artefatos: classes, páginas web, arquivos de mensagens,etc. Devido ao caráter iterativo e incremental do AUP, quando necessário o código foi melhorado através de refatorações para a melhoria da qualidade e legibilidade do código. Para exemplificar, o (Quadro 4) mostrar um código de um método que carrega os dados do acidente na edição de acidente. Podemos verificar que o método está meio confuso de entender e faz várias lógicas que poderiam ser desmembradas em outros métodos menores. No (Quadro 5) apresenta o código foi refatorado de modo que foi criado métodos menores para representar cada informação inerente ao acidente.

Quadro 5 - Código implementado no início do desenvolvimento do SNAA

@PostConstruct

**public** **void** carregarDadosDoAcidente() {

**if** (getBean().getVitima() == **null**) {

Vitima vitima = **new** Vitima();

**if** (idVitima.getValue() != **null**) {

vitima = vitimaBC.load(idVitima.getValue());

estado = vitima.getMunicipioOrigem().getEstado();

} **else** {

vitima.setPaisOrigem(snaaContext.getPais());

estado = snaaContext.getEstado();

}

getBean().setVitima(vitima);

**if** (getBean().getMunicipio() == **null**) {

getBean().setMunicipio(**new** Municipio());

}

} **else** {

Long idEstado = getBean().getVitima().getMunicipioOrigem()

.getEstado().getId();

**this**.estado = **new** Estado(idEstado);

}

**if** (getBean().getLocalizacao() != **null**) {

Long idTipoLocalizacao = getBean().getLocalizacao()

.getTipoLocalizacao().getId();

**this**.tipoLocalizacao = **this**.tipoLocalizacaoBC

.load(idTipoLocalizacao);

}

}

Quadro 6 - Código após refatorações

@PostConstruct

**public** **void** carregarDadosDoAcidente() {

carregarVitima();

carregarPais();

carregarEstado();

carregarLocalizacao();

}

## 3.1.7 Implantação

Esta subseção define as informações para implantação do SNAA. Para implantá-lo é necessário um servidor com sistema operacional Windows Server 2008 ou Linux, como no mínimo 4 Gigabyte de memória RAM e Disco de 500 Gigabyte. Nesse servidor, deve existir um JBoss AS e banco de dados MySQL executando.

|  |  |
| --- | --- |
| **PLANO DE IMPLANTAÇÃO DO SNAA** | |
| Servidor | Windows Server 2008 ou Linux, Disco de Rígido de 500 Gigabyte |
| Servidor de aplicação; | JBoss AS |
| Banco de dados: | MySQL |

Para realizar o implantação é necessário acessar o repositório da aplicação no seguinte endereço <url a definir>. é importante ressaltar que existe um versão implantada no seguinte endereço <url a definir>.

# Capítulo 4

# Considerações Finais

Este trabalho teve como objetivo disponibilizar um sistema de notificação de acidentes por arraia para o estado do Amazonas. Esse sistema foi denominado *SNAA* e o seu desenvolvimento seguiu uma abordagem de desenvolvimento ágil, iterativa e incremental, guiado pelo Processo Unificado Ágil (AUP). Na processo implementação do *SNAA*, foi utilizado a plataforma Java e o banco de dados MySQL. O código fonte do SNAA está disponível sob licença software livre em <url a definir> e também está implantado em <url a definir>.

O resultado final deste trabalho relevância em vários aspectos da biologia e da saúde pública. Primeiramente, permitirá a captura de informações dos acidentes acontecidos no estado do Amazonas,independente do período em que tenham ocorrido,permitindo o conhecimento e consolidação de um histórico sobre este tipo de acidente e, por conseguinte, auxiliando biólogos e outros estudiosos da área.Os pesquisadores, interessados em biodiversidade, poderão obter conhecimentos sobre a distribuição geográficas das espécies de arraias, existentes no Estado do Amazonas.O sistema auxiliará os órgãos competentes na tomada decisões acerca do controle e prevenção dos acidentes.

O desenvolvimento deste trabalho possibilitou a integração dos conhecimentos obtidos nas diferentes disciplinas cursadas no curso de Tecnologia em Análise e Desenvolvimento de Sistemas, como por exemplo,análise de sistemas, processo de desenvolvimento, linguagens de programação e banco de dados. Possibilitou, também, conhecer novos conceitos da área desenvolvimento de sistemas, tais como: padrões de projeto. utilização de frameworks e como aplicar segurança em uma aplicação WEB.

Ao longo do desenvolvimento do SNAA, foram surgindo algumas dificuldades que poderiam gerar riscos para a correta concretização deste trabalho. Os mais críticos foram: o gerenciamento do tempo para que se cumprisse o cronograma estabelecida; a redação científica e o emprego das normas da ABNT, de tal maneira que o texto da monografia estivesse redigido de forma apropriada para avaliação da banca examinadora.

Em relação às limitações deste trabalho, alguns requisitos não-funcionais ainda não foram implementados, tais como o mecanismo de auditoria.No entanto em versões, posteriores, do sistema tal requisito já estará disponível. Espera-se também que SNAA integre-se com o algum serviço de geolocalização (tal como,Google Maps) para que se possa definir as coordenadas o local geográfico onde ocorrem os acidentes.

Quanto a trabalhos futuros, espera-se que o *SNAA* possa se integrar a outros sistemas do governo e que o seu desenvolvimento continue através da colaboração da comunidade de desenvolvimento de software, trazendo o resultado esperado por este trabalho de maneira que o sistema evolua para entender novas exigências.

# Referências Bibliográficas

AMBLER, S. The Agile Unified Process (AUP). **The Agile Unified Process (AUP)**, 2005. Disponivel em: <http://www.ambysoft.com/unifiedprocess/agileUP.html>. Acesso em: 30 jun. 2013.

BECK, K. et al. http://manifestoagil.com.br/. **Manifesto para o desenvolvimento ágil de software**, 2001.

BOCHNER, R. Acidentes por Animais Peçonhentos: Aspectos Históricos, Epidemiológicos, Ambientais e Sócio-Econômicos, 2003.

BOCHNER, R.; STRUCHINER, C. J. Acidentes por animais peçonhentos e sistemas nacionais de informação, Rio de Janeiro, Maio 2002.

BRASIL. Acidentes por Animais Peçonhentos. **portal.saude.gov.br**, 2013. Disponivel em: <http://portal.saude.gov.br/portal/saude/profissional/visualizar\_texto.cfm?idtxt=31479&janela=1>. Acesso em: 12 Abril 2013.

CAELUM. Apostila do curso FJ-11 Java e Orientação a Objetos. **http:** //www.caelum.com.br/, 2013. Disponivel em: <http://www.caelum.com.br/apostila-java-orientacao-objetos/>. Acesso em: 30 jun. 2013.

DORNELES, A. L. Frequência de acidentes por animais peçonhentos ocorridos no Rio Grande do Sul, 2001 - 2006, Porto Alegre - RS, 2009.

FOWLER, M. **Patterns of Enterprise Application Architecture**. 1. ed. [S.l.]: [s.n.], 2002.

FOWLER, M. **UML Essencial:** um breve guia para a linguagem padrão de modelagem de negócios. 3. ed. [S.l.]: Bookman, 2005.

GAMMA, E. et al. **Padrões de Projeto Soluções reutilizáveis de software orientado a objetos**. Porto Alegre: bookman, 2000.

GARRONE NETO, D.; HADDAD JUNIOR, V. Acidentes por raias. In: CARDOSO, J. L. C., et al. **Animais peçonhentos no Brasil:** biologia, clínica e terapêutica dos acidentes. 2 Edição. ed. São Paulo,Brasil: Sarvier, 2009. Cap. 30, p. 295-313.

GENTIL VASCONCELOS, H. C.; SÁ DE OLIVEIRA, J. C. Alimentação de Potamotrygon motoro (CHONDRICHTHYES, POTAMOTRYGONIDAE) na planície de inundação do APA do Rio Curiaú, Macapá-Amapá-Brasil, Macapá-Amapá, v. 1, n. 2, 2011.

HADDAD JUNIOR, V. Animais aquáticos de importância médica no Brasil. **Revista da Sociedade Brasileira de Medicina Tropical**, São Paulo, Outubro 2003.

JBOSS. http://jbossas.jboss.org/. **JBoss Application Server 7**, 2014. Disponivel em: <http://jbossas.jboss.org/>. Acesso em: 01 Maio 2014.

JCP. JSR 299: Contexts and Dependency Injection for the JavaTM EE platform. **Java Community Process**, 2009. Disponivel em: <https://jcp.org/en/jsr/detail?id=299>. Acesso em: 01 Maio 2014.

K19. DesenvolvimentoWeb com JSF2 e JPA2. **K19 Treinamentos**, 2013. Disponivel em: <http://www.k19.com.br/downloads/apostilas/java/k19-k12-desenvolvimento-web-com-jsf2-e-jpa2>. Acesso em: 30 jun. 2013.

LARMAN, C. **Utilizando UML e Padrões Uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3. ed. [S.l.]: Bookman, 2007.

MELO, A. C. **Desenvolvendo aplicações com UML 2.2:** Do conceitual à implementação. 3. ed. Rio de Janeiro: Brasport, 2010.

MYSQL. Documentação do MySQL, 2013. Disponivel em: <http://downloads.mysql.com/docs/refman-5.6-en.a4.pdf>. Acesso em: 30 jun. 2013.

ORACLE. The Java EE 5 Tutorial. **http:** //docs.oracle.com, 2010. Disponivel em: <http://docs.oracle.com/javaee/5/tutorial/doc/bnaph.html>. Acesso em: 30 jun. 2013.

PARANÁ. ACIDENTES COM ANIMAIS AQUÁTICOS – ARRAIAS E PEIXES COM FERRÃO, 2012. Disponivel em: <http://sesaeventos.saude.ws/zoonose/html/arquivos/NT%2002\_2012%20ARRAIAS%20E%20PEIXES.pdf>.

SALLES, R. D. O. L.; MENEGUCI DA CUNHA, A. Biologia, Prevenção e Primeiros socorrros em acidentes com animais peçonhentos: um trabalho com turmas do ensino fundamental, Outubro 2007.

SAÚDE, MINISTÉRIO DA. Sistema de Informação de Agravos de Notificação Normas e Técnicas, 2007. Disponivel em: <http://bvsms.saude.gov.br/bvs/publicacoes/07\_0098\_M.pdf>. Acesso em: 30 jun. 2013.

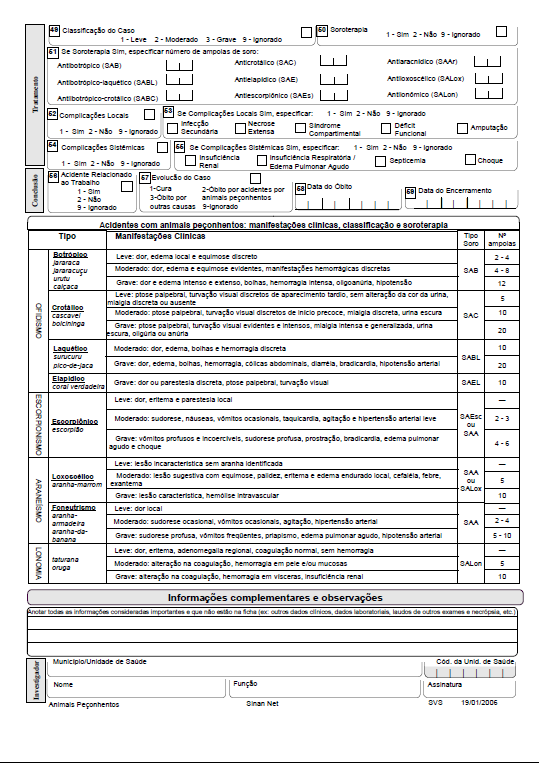
SOUZA, W. D. Doenças negligenciadas. Rio de Janeiro: Academia Brasileira de Ciências, 2010. Cap. 5, p. 36-38.

# Anexo I – Ficha de notificação individual de acidentes por animais peçonhentos

# Untitled.png

# ­

# ­­­



# Apêndice 1 – Especificação de Casos de Uso do Módulo de Gerenciamento do Sistema

Quadro 7 - Caso de Uso CDU01: Solicitar Criação de Conta.

|  |
| --- |
| **Ator principal:** Representante de unidade de saúde.  **Interessados e Interesses:**  - Representante de unidade de saúde: Deseja ter seu pedido de criação de conta efetivado pelo administrador do sistema e ser notificado através de e-mail se sua solicitação foi aceita ou não.  - Administrador: deseja analisar e validar a solicitação de criação de conta de usuário, informando ao solicitante a sua resposta.  **Pré-condições:** O representante de unidade não poderá ter uma conta já vinculada ao sistema.  **Pós-condições (Garantia de Sucesso):**O representante receberá uma notificação por e-mail sobre a situação de cadastro (se a solicitação foi aceita ou não).  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O representante escolhe a opção *Solicitar Conta de Usuário* na página inicial do sistema; 2. Será exibido um formulário no qual deverá ser informado as informações necessárias para a solicitação. 3. Informa os dados de sua unidade:  * Nome da unidade de saúde * Código da unidade de saúde; * Data de criação * Endereço (Município,Bairro,Logradouro,CEP,Numero e Complemento)  1. Informa seus dados de representante de unidade:  * Nome; * Data de Nascimento; * Sexo; * CPF (Caso for estrangeiro, informar o número de passaporte); * RG (Registro Geral); * E-mail * Telefone  1. O usuário também deverá realizar upload de arquivo representando seus documentos digitalizado, para fins de validação pelo administrador; 2. O representante deverá informar os dados de autenticação:  * Senha e confirmação de senha;  1. Escolhe a opção *Enviar Notificação*; 2. O solicitação é registrada e enviada ao Administrador. 3. O sistema mostra ­­ usuário a mensagem "*Solicitação registrada com sucesso*"   **Extensões (ou Fluxo Alternativo):**   1. **Campos obrigatórios não preenchidos:** Caso algum campo obrigatório do formulário não tenha sido preenchido, o sistema notificará o usuário, informando-lhe quais o campos que não foram preenchidos. 2. **Conta de usuário existente:** Caso o profissional informe um login que já exista no sistema, o sistema notificará o usuário emitindo a mensagem "*Conta de usuário existente*". 3. **Formato de arquivo inválido** Caso o for informado um arquivo para upload que não esteja no padrão de codificação especificado na Regra de Negócio **RN01,** o sistema emitirá a mensagem: *"O formato do arquivo informado não é válido"* 4. **Senha e confirmação não coincidem:** Caso o representante informe uma senha que não coincida com a confirmação de senha, o sistema deverá emitir a mensagem: "Senha e Confirmação não coincidem"   **Regras de negócio:**  **RN01:** O formato de arquivo permitido para upload são: .png, .jpeg; |

Quadro 8 - Caso de Uso CDU02: Validar Solicitação

|  |
| --- |
| **Ator principal:** Administrador  **Interessados e Interesses:**  - Administrador: deseja identificar e validar a solicitação de criação conta de usuário.  -Representante de unidade: deseja receber um e-mail informando-lhe se sua solicitação foi aceita ou não  **Pré-condições:** O representante deverá ter previamente enviado uma solicitação ao administrador  **Pós-condições (Garantia de Sucesso):** Envio da confirmação ao representante de unidade de saúde através de uma mensagem por e-mail informando-lhe se a solicitação foi aceita ou não.  Caso o administrador aceite a solicitação , a conta de representante de saúde será criada e ativada no sistema, podendo o representante efetuar login no sistema.  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O administrador seleciona a opção "Solicitações pendentes" 2. O sistema mostrará a listagem de todas as solicitações não validadas. 3. O administrador escolhe em alguma registro de solicitação a opção *Avaliar Solicitação* 4. Será exibido um diálogo no qual exibido os dados do solicitante e o documento digitalizado. 5. O administrador informará se aceita ou não a solicitação, descrevendo uma mensagem explicando sua decisão, que será enviada no corpo da mensagem de e-mail para o solicitante. 6. O administrador escolhe a opção *Aprovar Solicitação*  ou *Reprovar Solicitação.* 7. O sistema emite mensagem: "*Avaliação efetivada com sucesso*" |

Quadro 9 - Caso de Uso CDU03: Cadastrar profissional de saúde.

|  |
| --- |
| **Ator principal:** Representante de unidade de saúde;  **Interessados e Interesses:**  - Representante de unidade de saúde: deseja as contas dos profissionais de saúde de sua de unidade.  **Pré-condições:** Estar autenticado no sistema  **Pós-condições (Garantia de Sucesso):**Efetivação da criação das contas dos profissionais de saúde.  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O representante escolherá a opção *Cadastrar Profissional*; 2. Será mostrado um formulário no qual deverá ser informado as informações necessárias para a solicitação 3. O representante informará as seguintes informações do profissional de saúde:  * CPF; * Número do Documento de registro de profissão ; * RG; * Nome * Data de Nascimento; * Sexo; * Telefone; * E-mail;  1. O representante deverá informar os dados de autenticação para o profissional de saúde:  * Senha e confirmação de senha;  1. Escolhe a opção *Salvar*; 2. O sistema efetiva a criação de conta; 3. O sistema mostra ­­ usuário a mensagem "*Conta de profissional de saúde criada com sucesso*"   **Extensões (ou Fluxo Alternativo):**   1. **Campos obrigatórios não preenchidos:** Caso algum campo do formulário não tenha sido preenchido, o sistema notificará o usuário, informando-lhe quais o campos que não foram preenchidos. 2. **Conta de usuário existente:** Caso o profissional informe um login que já exista no sistema, o sistema notificará o usuário emitindo a mensagem "*Conta de usuário existente*". 3. **Senha e confirmação não coincidem:** Caso o representante informe uma senha que não coincida com a confirmação de senha, o sistema deverá emitir a mensagem: "Senha e Confirmação não coincidem" |

# Apêndice 2 – Especificação de Casos de Uso do Módulo de Notificação e Acompanhamento de acidentes por unidade de saúde

Quadro 10 - Caso de Uso CDU 04: Pesquisar Paciente.

|  |
| --- |
| **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja buscar um determinado registro de paciente.  **Pré-condições:** O profissional deverá está autenticado no sistema.  **Pós-condições (Garantia de Sucesso):**Resultado da pesquisa apresentado ao profissional de saúde.  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O profissional de saúde seleciona a opção pesquisar acidente; 2. Preenche nenhum ou mais campos de pesquisa ; 3. Escolhe a opção pesquisar*;* 4. O sistema processa e apresenta o resultado da pesquisa para o profissional;   **Extensões (ou Fluxo Alternativo):**   1. **Nenhum filtro informado:** Caso nenhum filtro seja informado, o sistema retornará todos os pacientes que o profissional de saúde registrou. 2. **Nenhum registro encontrada:** Caso a consulta realizada pelo usuário não contenha nenhuma registro, o sistema emitirá a mensagem: "Nenhum registro encontrado" 3. **Visualizar Acidente:** O usuário pode selecionar em algum registro pesquisado a opção *Visualizar Paciente*. **estender Visualizar Acidente.** 4. **Editar Acidente:** O usuário pode selecionar em algum registro pesquisado a opção *Editar Acidente*, se esta opção estiver disponível. **estender Editar Acidente.** 5. **Reativar Acidente :**O usuário pode selecionar em algum registro pesquisado a opção *Reativar Acidente*, caso esta opção estiver disponível. **estender Reativar Acidente.** |

Quadro 11 - Caso de Uso CDU05: Cadastrar Acidente.

|  |
| --- |
| **Caso de uso CDU6 - Cadastrar acidentes**  **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja registrar um novo acidente por arraia.  **Pré-condições:** O profissional deverá está autenticado no sistema.  **Pós-condições (Garantia de Sucesso):**Cadastro do acidente efetivado e exibição da mensagem informando que foi realizado cadastro do acidente.  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O profissional de saúde seleciona a opção *Cadastrar Acidente;* 2. O profissional informa os dados pessoais do paciente **include Cadastrar Paciente** 3. Preenche o formulário de cadastro de acidente, registrando as seguintes informações:  * Data do acidente; * Período do dia que ocorreu o acidente; * Município da ocorrência; * Tipo de local (rio, lago,igarapé,etc.); * Identificação do local; * Informação da atividade que o vítima estava fazendo no momento do acidente;  1. É apresentado o resumo dos dados do acidente e do paciente informados no formulário *;* 2. O usuário escolhe a opção concluir 3. O Usuário será redirecionado para tela de edição de acidentes,em que poderá ser cadastradas informações mais detalhadas do acidente, tais como:  * Cadastro de Prontuário (**estender Cadastrar Prontuário)** * Identificar a arraia (**estender Identificar Arraia**) * Cadastro de Tratamento (**estender Cadastro de Tratamento)** * Conclusão do registro do acidente (**estender Concluir Acidente)**   **Extensões (ou Fluxo Alternativo):**   1. **Campos obrigatórios não preenchidos:** Caso algum campo obrigatório do formulário não tenha sido preenchido, o sistema notificará o usuário, informando-lhe quais o campos que não foram preenchidos. |

Quadro 12 - Caso de Uso CDU06 :Cadastrar Paciente

|  |
| --- |
| **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja registrar um novo paciente por arraia.  **Pré-condições:** O profissional deverá está autenticado no sistema.  **Pós-condições (Garantia de Sucesso):**Cadastro do acidente efetivado e exibição da mensagem informando que foi realizado.  **Cenário de Sucesso Principal (ou Fluxo Básico)**   1. Preenche o formulário de cadastro de paciente, registrando as seguintes informações:  * CPF; * RG * Nome; * Data de Nascimento; * Sexo; * Escolaridade; * Profissão; * Nacionalidade * Estado e Naturalidade (Caso o paciente for brasileiro)      1. O profissional retorna ao fluxo principal do caso de uso de **CDU05 - Cadastrar Acidente**   **Extensões (ou Fluxo Alternativo):**  **Campos obrigatórios não preenchidos:** Caso algum campo obrigatório do formulário não tenha sido preenchido, o sistema notificará o usuário, informando-lhe quais o campos que não foram preenchidos.  **Paciente existente no sistema:** Caso o cpf informado já exista no sistema, será mostrado a mensagem: "CPF já existe no sistema " |

Quadro 13 - Caso de Uso CDU07 : Identificar Arraia

|  |
| --- |
| **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja registrar os dados sobre arraia causadora do acidente.  **Pré-condições:**   * O profissional deverá está autenticado no sistema. * O usuário deverá ter iniciado previamente o caso de uso **CDU05 - Cadastrar Acidente**   **Pós-condições (Garantia de Sucesso):**  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O Sistema exibe o formulário com uma lista de imagens de arraias. 2. O usuário escolhe uma imagem baseada na semelhança com as características informadas pelo paciente 3. O usuário escolhe valor na escala de certeza; 4. O usuário informa a situação do animal no acidente; 5. O usuário retorna ao fluxo principal do caso de uso **CDU05 - Cadastrar Acidente** |

Quadro 14 - Caso de Uso CDU08 :Cadastrar Prontuário

|  |
| --- |
| **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja registrar os dados sobre o paciente;  **Pré-condições:**   * O profissional deverá está autenticado no sistema. * O usuário deverá ter escolhido a opção *Editar Acidente.*   **Pós-condições (Garantia de Sucesso):**  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O profissional de saúde seleciona a aba *Prontuário;* 2. O Sistema exibe o formulário com para preenchimento dos dados do prontuário do paciente. 3. O Profissional seleciona a opção *Marcar local da picada;* 4. O profissional seleciona a através de imagem em que parte do corpo foi o local da picada 5. O profissional insere os sinais e sintomas locais e sistêmicos que o paciente apresenta; 6. O profissional informa a causa do acidente; 7. O profissional identifica o que o paciente estava fazendo no momento do acidente; 8. *C*licar na opção *Salvar;* 9. O sistema exibe a mensagem "*Prontuário incluído com sucesso*"; |

Quadro 15 - Caso de Uso CDU09: Cadastrar Tratamento

|  |
| --- |
| **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja registrar os dados sobre o tratamento do paciente;  **Pré-condições:**   * O profissional deverá está autenticado no sistema. * O usuário deverá ter escolhido a opção *Editar Acidente.*   **Pós-condições (Garantia de Sucesso):**  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O profissional de saúde seleciona a aba *Tratamento;* 2. O Sistema exibe o formulário com para preenchimento dos dados do tratamento do paciente. 3. O Profissional informa qual foi o medicamento tanto local como sistêmico que o paciente foi submetido; 4. O profissional informa qual foi o tratamento ou procedimento local do que o paciente foi submetido; 5. O profissional informa se o paciente ficou e quanto dias ficou; 6. O profissional informa os dados de exames laboratoriais do paciente 7. *C*licar na opção *Salvar;* 8. O sistema exibe a mensagem "*Tratamento incluído com sucesso*"; |

Quadro 16 - Caso de Uso CDU10 : Concluir Acidente

|  |
| --- |
| **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja concluir a notificação do acidente  **Pré-condições:**   * O profissional deverá está autenticado no sistema. * O usuário deverá ter escolhido a opção *Editar Acidente.*   **Pós-condições (Garantia de Sucesso):**O registro do acidente passa a ter status de CONCLUÍDO  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O profissional de saúde seleciona a aba *Conclusão;* 2. O Sistema exibe o formulário efetivar a conclusão do acidente; 3. O profissional seleciona o Quadro de saída *;* 4. O profissional insere um comentário se achar relevante; 5. *C*licar na opção *Salvar;* 6. O sistema exibe a mensagem "*Acidente concluído com sucesso*"; |

Quadro 17 - Caso de Uso CDU11: Reativar acidente

|  |
| --- |
| **Ator principal:** Profissional de saúde  **Interessados e Interesses:**  - Profissional de saúde: deseja reabrir para edição um acidente que esta com status de CONCLUÍDO.  **Pré-condições:** O profissional deverá está autenticado no sistema.  **Pós-condições (Garantia de Sucesso):**Após a reativação do registro o profissional poderá a editar os dados do acidente, no entanto ele irá gerar os históricos das mudanças nos dados do prontuário e de tratamento  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O profissional de saúde seleciona a opção Reativar Acidente em algum registro na tela pesquisa de pacientes 2. O sistema altera o estado do registro para PENDENTE; 3. O sistema exibe a mensagem "*Acidente reativado com sucesso*";   **Regras Negócio:**  **RN01 -** Somente registro de acidentes com situação igual a CONCLUÍDO poderão possuir essa opção; |

# Apêndice 3 – Especificação de Casos de Uso do Módulo de consultas e geração de relatórios de acidentes

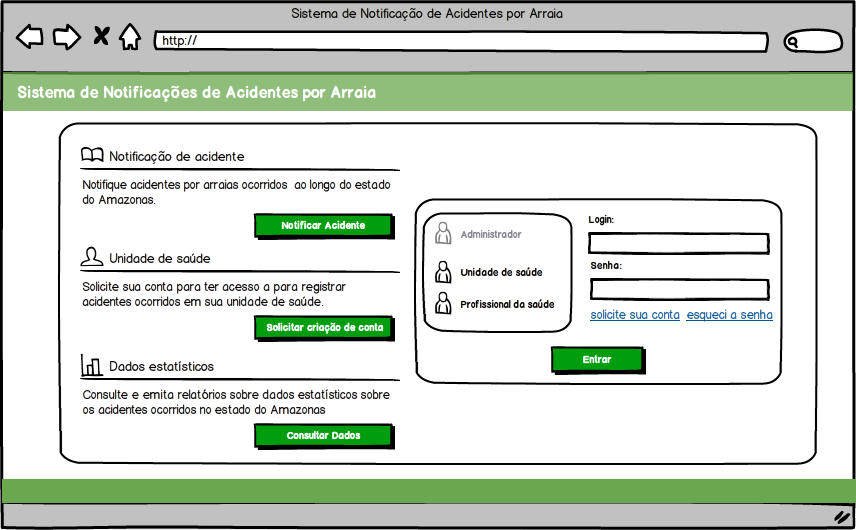
Quadro 18 - Caso de Uso CDU12: Consultar Dados Estatísticos.

|  |
| --- |
| **Ator principal:** Representante de unidade de saúde, Profissional de saúde, Administrador e Público  **Interessados e Interesses:**  - Todos os atores do sistema, que desejam consultar e emitir relatórios de dados estatísticos do acidentes  **Pré-condições:**  **Pós-condições (Garantia de Sucesso):**Resultado da consulta apresentado ao usuário.  **Cenário de Sucesso Principal (ou Fluxo Básico):**   1. O usuário seleciona a opção *Consultar Dados Estatísticos;* 2. Preenche zero ou mais campos de pesquisa ; 3. Escolhe o tipo de relatório que pode ser os seguintes:  * Relatório de Acidentes por Município; * Ranking de acidentes por Município; * Ranking de acidentes por Sexo; * Ranking de acidentes por Faixa Etária; * Ranking de acidentes por Sintomas; * Ranking de acidentes por Quadro de Saída  1. Usuário informa o código de verificação mostrado no captcha. 2. Escolhe a opção *Emitir relatório;* 3. O sistema processa e emite o relatório em uma nova aba do navegador;   **Extensões (ou Fluxo Alternativo):**   1. **Nenhum registro encontrada:** Caso a consulta realizada pelo usuário não nenhuma registro, o sistema emitirá a mensagem: "Nenhum registro encontrado" |

# Apêndice 4 - Interfaces do Usuário (Protótipos Iniciais)

Este apêndice apresenta as interfaces do usuário para o perfil de *profissional de saúde* ao executar o fluxo principal do sistema. As interfaces foram construídas com o auxílio da ferramenta Balsamiq Mockup.

Figura 33 - Página inicial do sistema



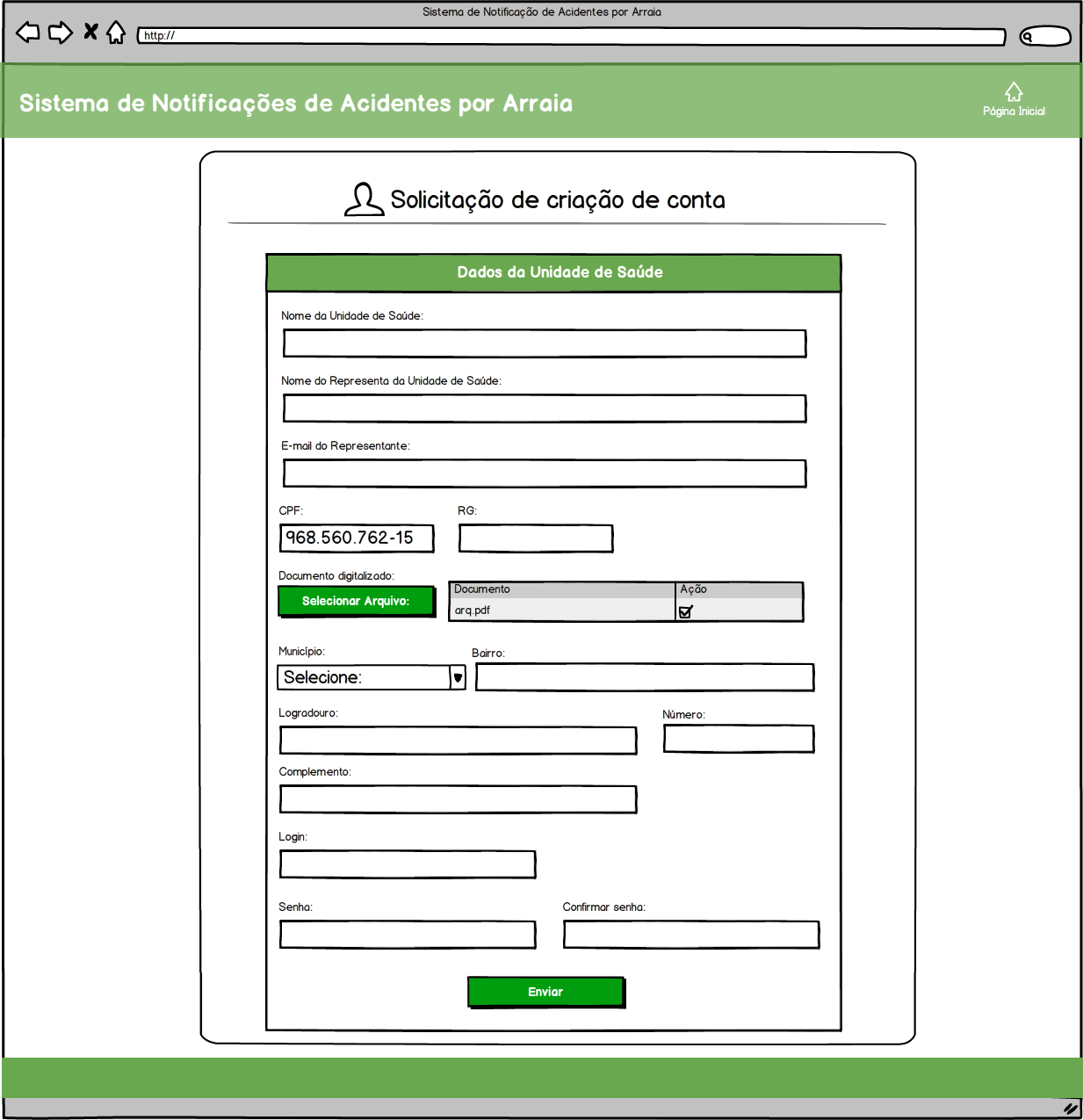


Figura34 - Solicitação de criação conta

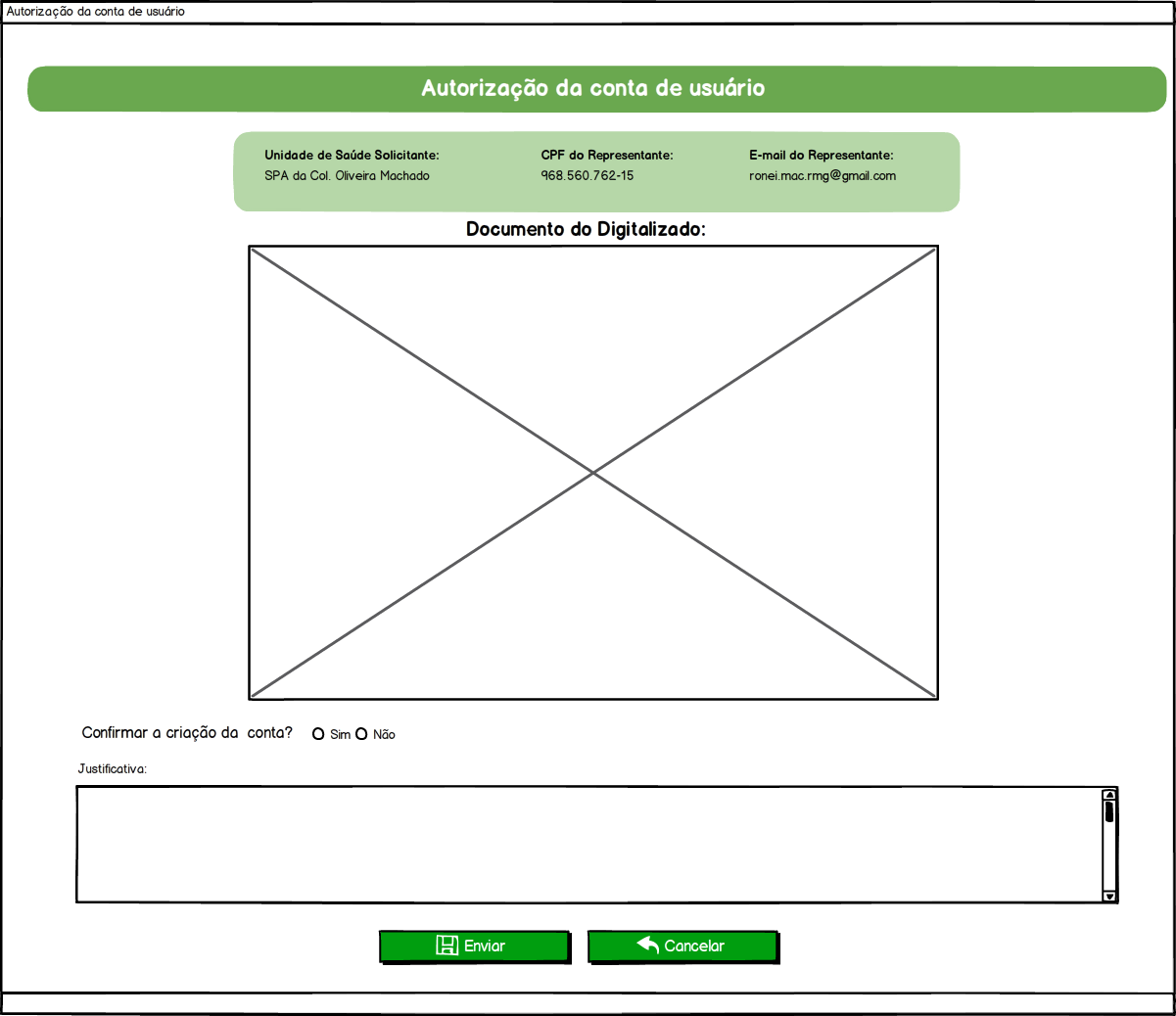


Figura 35 - Validar Conta de Usuário

Figura 36 - Cadastrar profissional de saúde



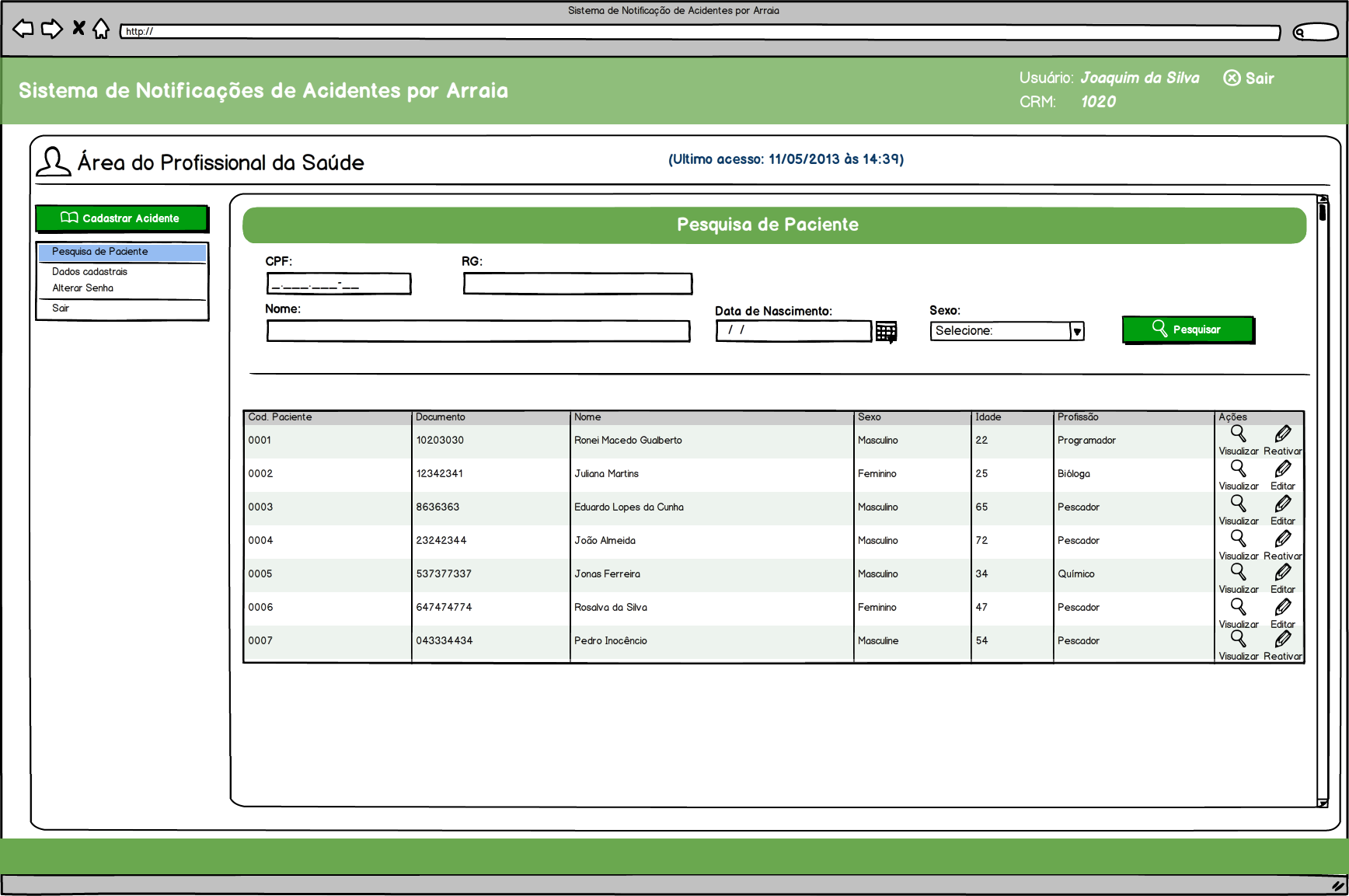
d

Figura 37 - Tela inicial do perfil de profissional de saúde

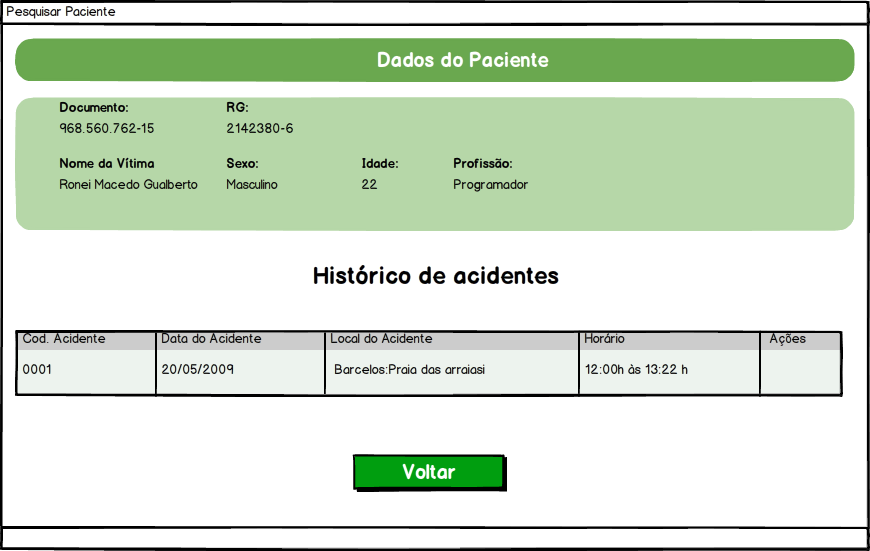


Figura 38 - Visualizar Acidente

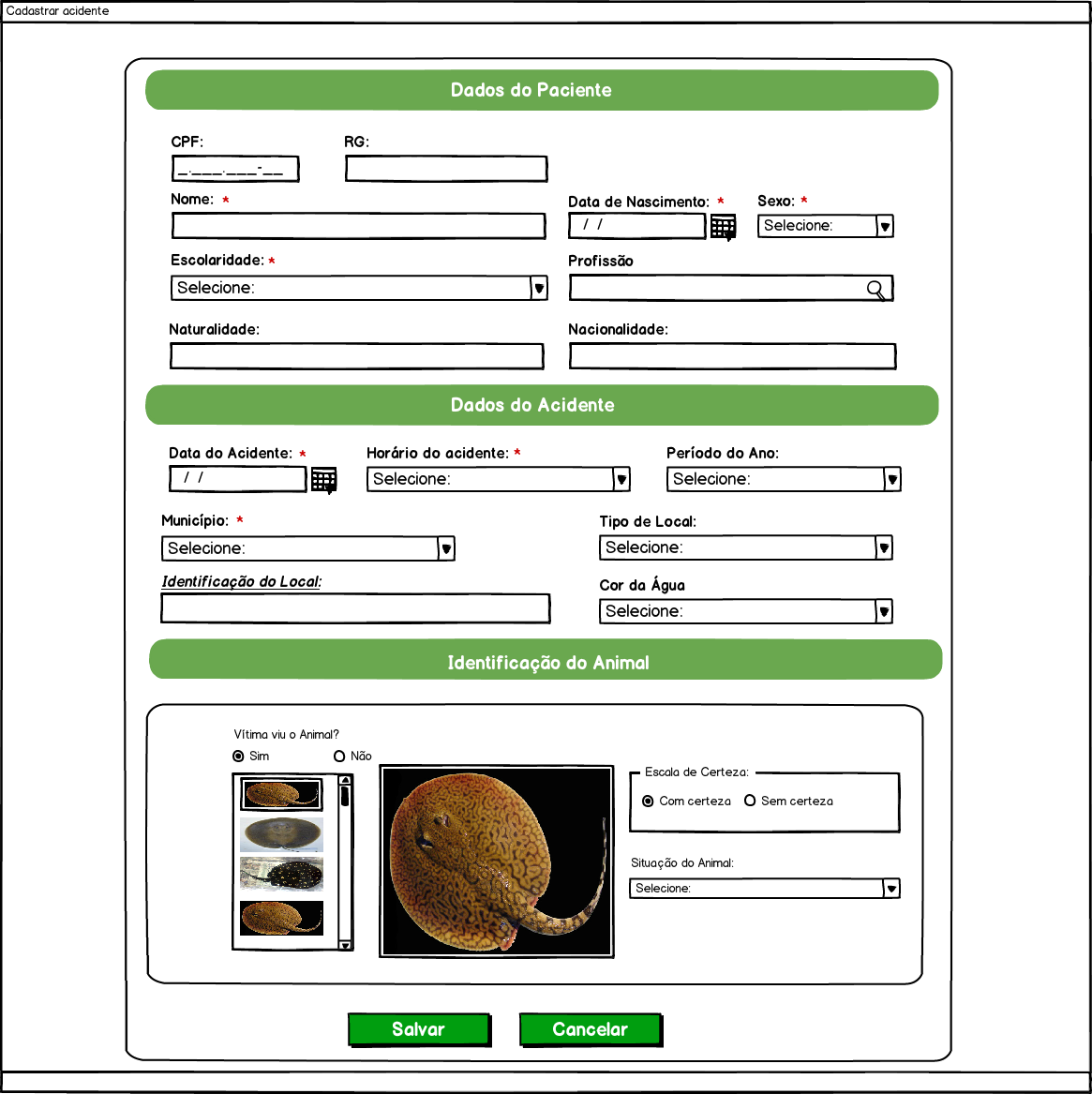


Figura 39 - Cadastrar Acidente

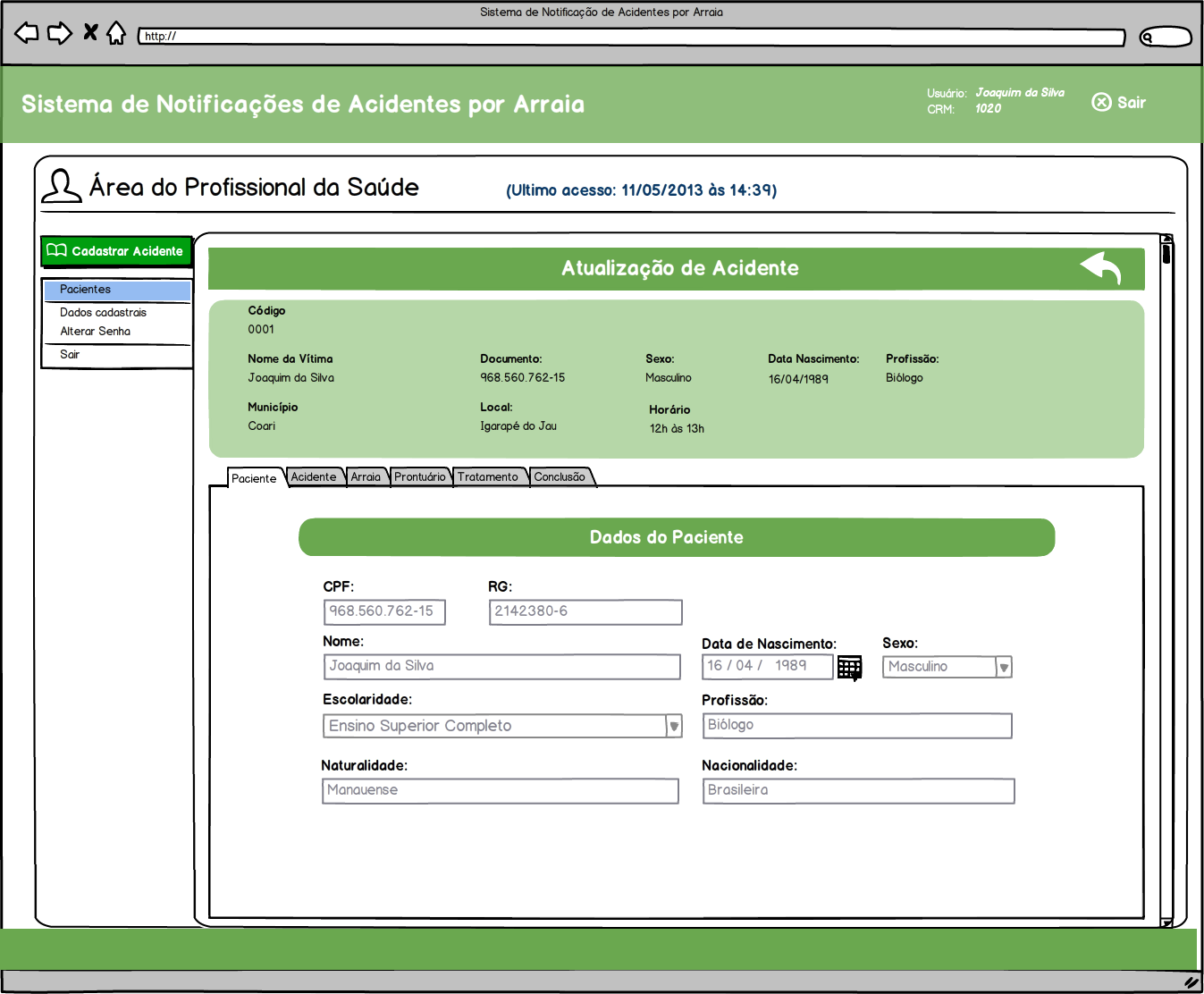


Figura 40 - Dados do Paciente

Figura 41 - Dado do acidente

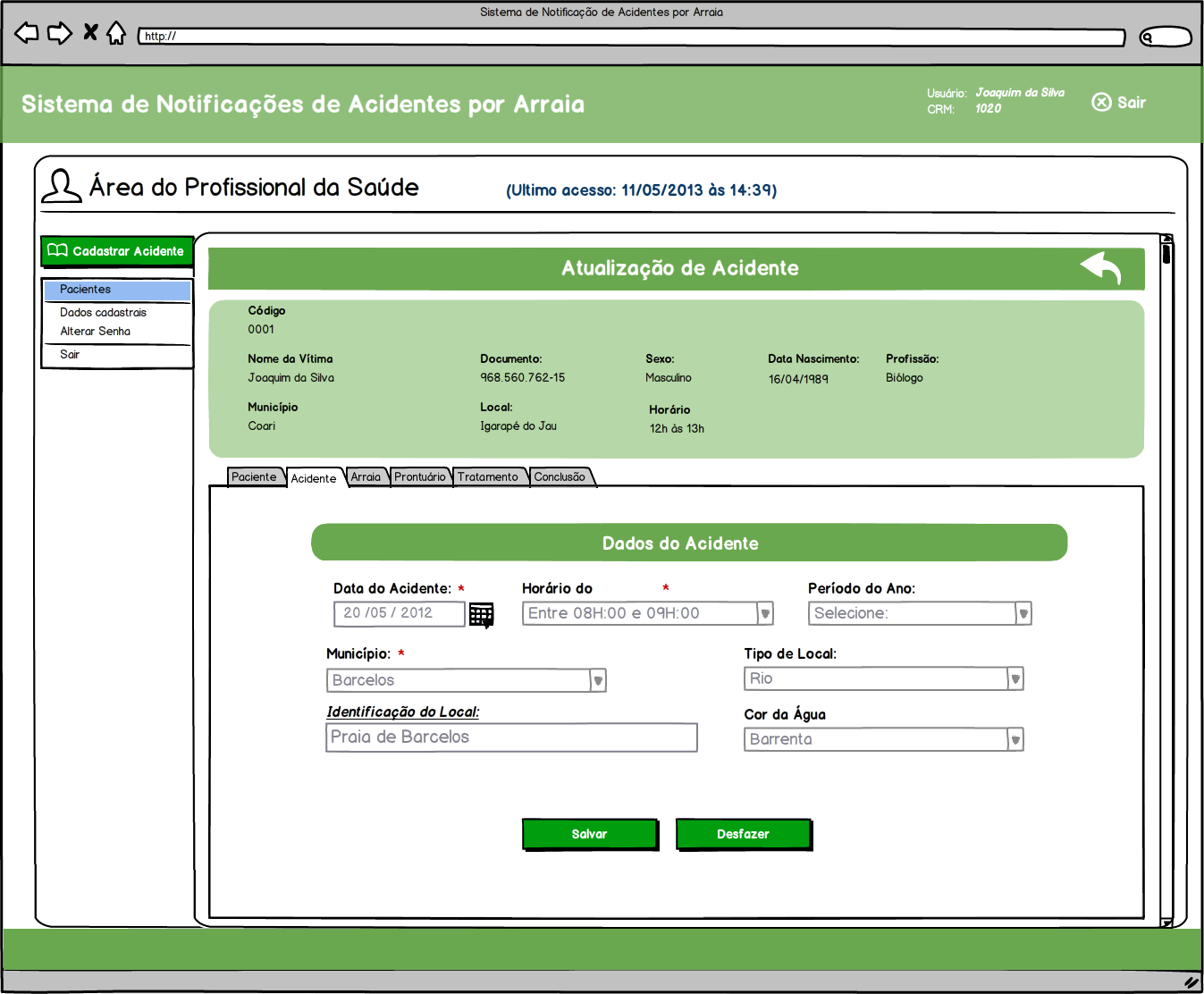


Figura 42 - Dados da arraia

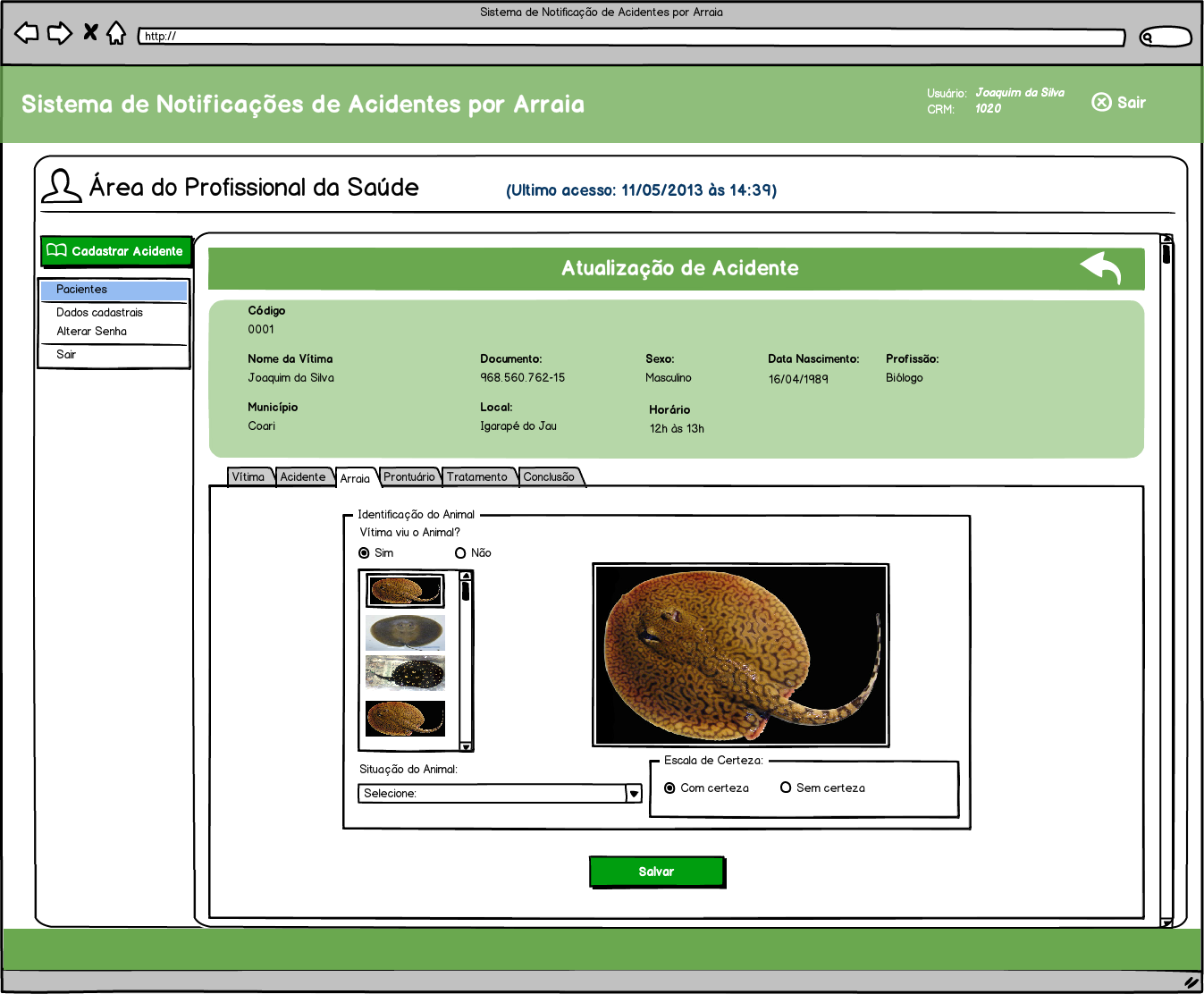


Figura 43 - Editar prontuário

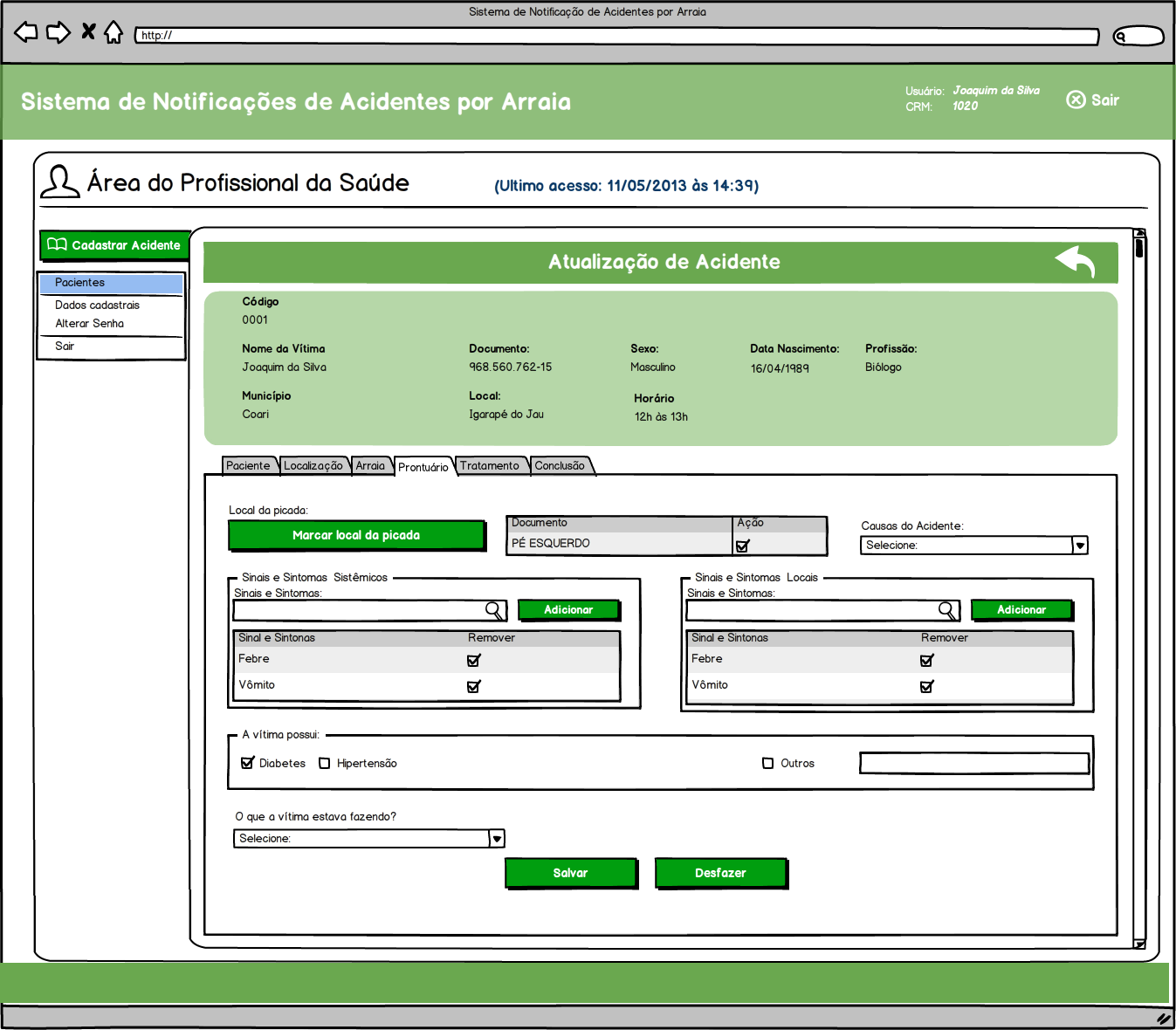
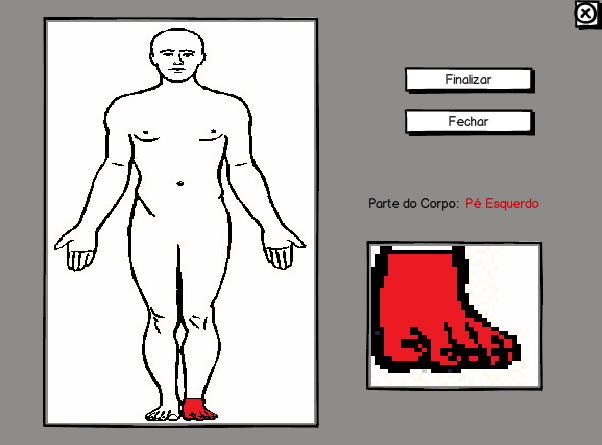


Figura 44 - Marcar local da picada



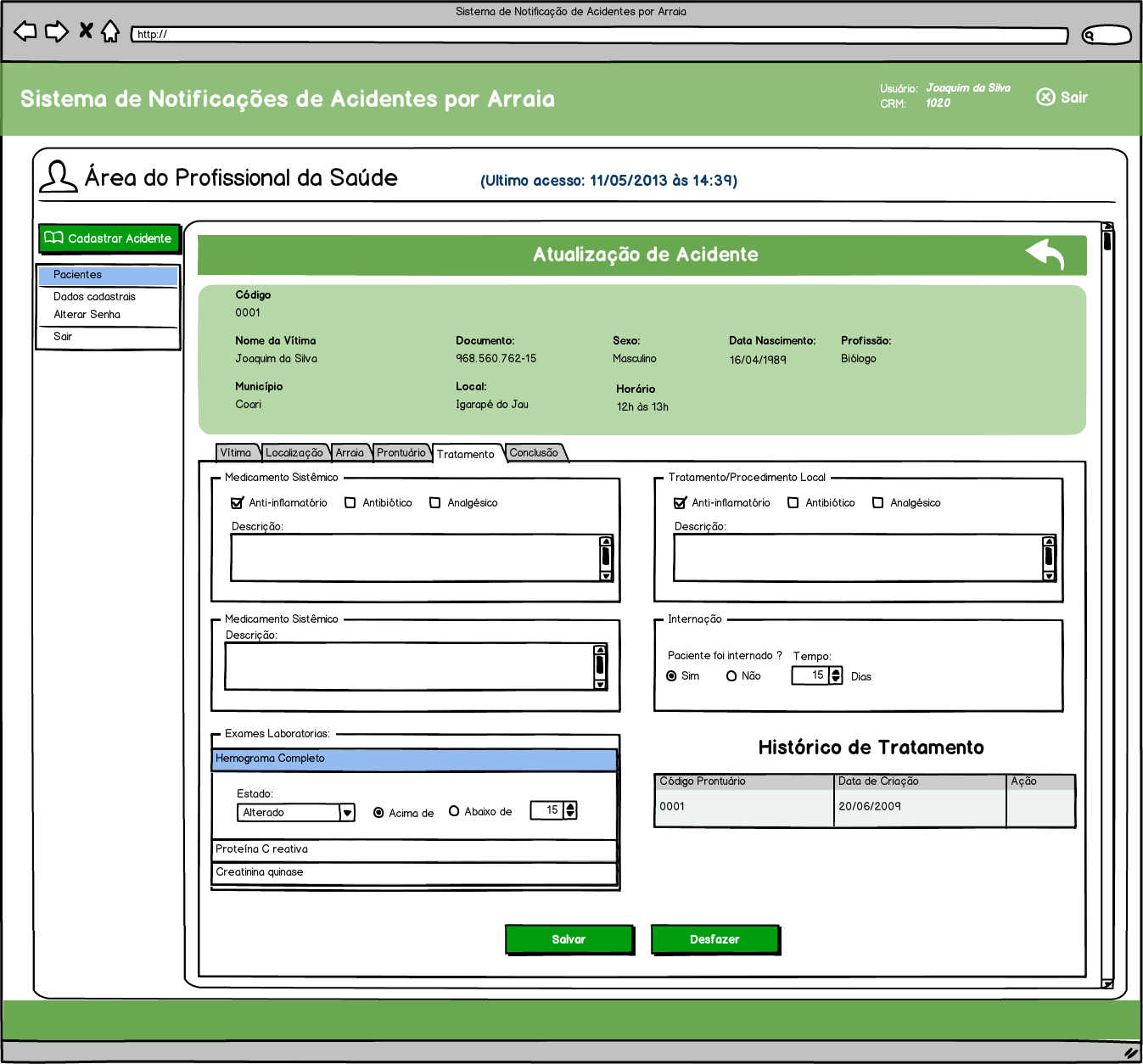


Figura 45 - Editar Tratamento

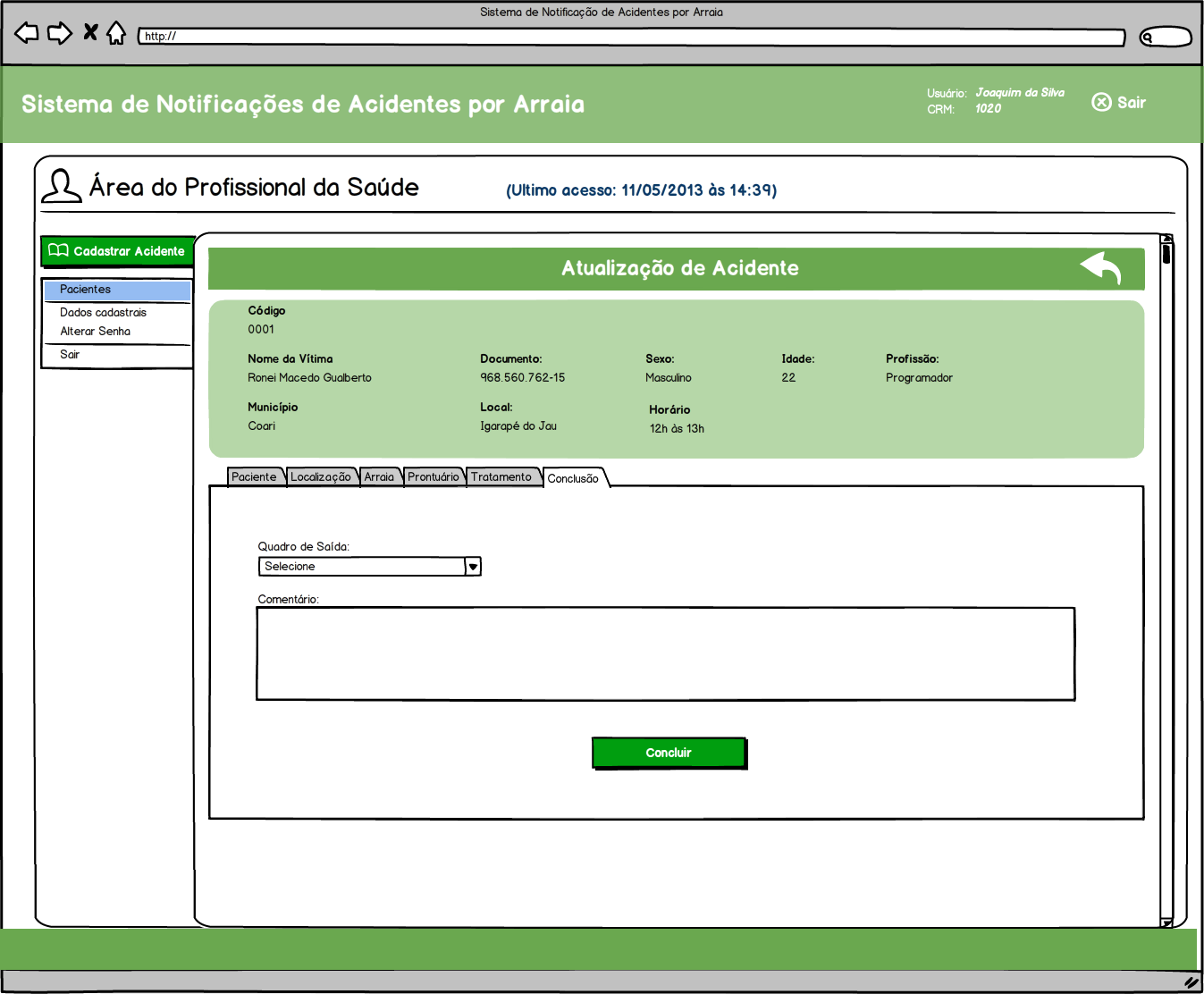


Figura 46 - Concluir Acidente



Figura 47 - Consulta de dados estatísticos

1. Disponível em <http://dtr2004.saude.gov.br/sinanweb/tabnet/dh?sinannet/animaisp/bases/animaisbrnet.def> [↑](#footnote-ref-2)
2. Acessado em 29/06/2013 [↑](#footnote-ref-3)