

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

Faculdade de Tecnologia da Baixada Santista “Rubens Lara”

Curso Superior de Tecnologia em Sistemas para Internet

**SISTEMA PARA ANÁLISE DE VULNERABILIDADES EM APLICAÇÕES
WEB**

SANTOS

JUNHO/2015

ANA ROSA ROCHA SANTOS

DENISE DA SILVA FERREIRA

WLADIMIR GOMES RODRIGUES FILHO

**SISTEMA PARA ANÁLISE DE VULNERABILIDADES EM APLICAÇÕES
WEB**

Trabalho de Conclusão de Curso
apresentado à Faculdade de Tecnologia
Baixada Santista “Rubens Lara”, como
exigência parcial para obtenção do título de
Tecnólogo em Sistemas para Internet.

Orientador: Prof. Jorge Luiz Chiara

SANTOS

JUNHO/2015

FERREIRA, Denise da S.; RODRIGUES FILHO, Wladimir G.; SANTOS, Ana R. R.

Sistema para análise de vulnerabilidades em aplicações web / Ana Rosa

Rocha Santos, Denise da Silva Ferreira, Wladimir Gomes Rodrigues Filho;
orientador: Jorge Luiz Chiara – Santos, 2015

75f.

Trabalho de Conclusão de Curso – Centro de Educação Tecnológica Paula Souza,
Faculdade de Tecnologia da Baixada Santista “Rubens Lara”, Curso de Sistemas para
Internet.

1.Segurança da Informação, 2. Aplicações Web, 3. Vulnerabilidades

ANA ROSA ROCHA SANTOS

DENISE DA SILVA FERREIRA

WLADIMIR GOMES RODRIGUES FILHO

**SISTEMA PARA ANÁLISE DE VULNERABILIDADES EM APLICAÇÕES
WEB**

Trabalho de Conclusão de Curso
apresentado à Faculdade de Tecnologia da
Baixada Santista “Rubens Lara”, como
exigência parcial para obtenção do título de
tecnólogo em Sistemas para Internet.

Santos, 27 de junho de 2015.

Banca Examinadora

Prof. Jorge Luiz Chiara – orientador

Faculdade de Tecnologia da Baixada Santista “Rubens Lara”

Presidente

Profa. Ma. Rosana Cammarosano

Faculdade de Tecnologia da Baixada Santista “Rubens Lara”

Prof. Luiz Alexandre da Costa Araujo

Faculdade de Tecnologia da Baixada Santista “Rubens Lara”

Aos nossos pais, pelo carinho, proteção, paciência e dedicação incondicional. Aos nossos professores por dedicarem suas vidas a educação, nos inspirando como indivíduos e profissionais.

AGRADECIMENTOS

Aos nossos pais, pelo carinho, proteção, paciência e dedicação incondicional que nos motiva a dar o nosso melhor em tudo aquilo que nos envolvemos, ainda que não pareça possível retribuir todo o esforço em nós depositado.

Aos nossos professores, que nos inspiram diariamente, construindo e compartilhando conhecimento e valores enquanto nos prepararam para o mercado profissional.

Ao nosso orientador, pelo papel fundamental que desempenhou em todas as etapas deste projeto, fornecendo sugestões, críticas construtivas, compartilhando conhecimento e, principalmente, estimulando cada integrante a dedicar o seu melhor.

A única segurança verdadeira na vida provém de saber que a cada dia você melhora de alguma maneira. -Anthony Robbins.

RESUMO

FERREIRA, Denise da S.; RODRIGUES FILHO, Wladimir G.; SANTOS, Ana R. R. Sistema para análise de vulnerabilidades em aplicações web. 2015. 75 páginas. Trabalho de Conclusão de Curso de Graduação de Tecnólogo em Sistemas para Internet, Centro Estadual de Educação Tecnológica Paula Souza, Faculdade de Tecnologia da Baixada Santista “Rubens Lara”, Santos, 2015.

A informação encontra-se em um cenário caracterizado pela crescente utilização de sistemas informatizados. Estes sistemas estão, cada vez mais, operando via internet, o que torna a informação um bem sujeito a um extenso número de ameaças e dos mais variados tipos. Devido aos avanços alcançados por parte dos atacantes, ou a lançamentos de novas tecnologias que apresentam novas vulnerabilidades, surge a necessidade de desenvolver mecanismos capazes de auxiliar na análise das vulnerabilidades presentes em sistemas *web*. A aplicação desenvolvida durante este projeto tem como base a identificação de uma ameaça relevante e de grande incidência em sistemas web na atualidade, chamada Injeção de Código.

Palavras-chave: Informação. *Internet*. Ameaça. Vulnerabilidade. Aplicação *web*.

ABSTRACT

FERREIRA, Denise da S.; RODRIGUES FILHO, Wladimir G.; SANTOS, Ana R. R. System for Vulnerabilities Analysis in Web Applications. 2015. 75 pages. Trabalho de Conclusão de Curso de Graduação de Tecnólogo em Sistemas para Internet, Centro Estadual de Educação Tecnológica Paula Souza, Faculdade de Tecnologia da Baixada Santista "Rubens Lara", Santos, 2015.

Nowadays, information is in a scenario marked by the increasing utilization of computerized systems. These systems are increasingly operating through internet, making information a property vulnerable to a large number of all kinds of threats. Due to advances achieved by the attackers, or releases of new technologies that contain new vulnerabilities, the need to develop new mechanisms to assist in vulnerabilities analysis in web systems emerges. The application developed in this project has as its foundation the identification of a relevant and of high incidence threat when it comes to web systems nowadays, called SQL Injection.

Keywords: Information. Internet. Threats. Vulnerability. Web Application.

LISTA DE ABREVIATURAS DE SIGLAS

AJAX	Asynchronous Javascript and XML (Javascript Assíncrono e XML)
API	Application Programming Interface (Interface de Programação de Aplicativos)
CSS	Cascading Style Sheets (Folha de Estilos)
HTML	HyperText Markup Language (Linguagem de Marcação de Hipertexto)
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)
ID	Identity (Identidade)
INTERNET	International Network (Rede Internacional)
IP	Internet Protocol (Protocolo de Internet)
ISP	Internet Service Provider (Provedor de Serviço de Internet)
JDBC	Java Database Connectivity (Conectividade de Banco de Dados Java)
JDK	Java Development Kit (Coleção de Desenvolvimento Java)
JSON	JavaScript Object Notation (Notação de Objeto Javascript)
JVM	Java Virtual Machine (Máquina Virtual Java)
ME	Relação das Mensagens do Sistema
ORM	Object-relational mapping (Mapeamento de Objeto Relacional)
OWASP	Open Web Application Security Project
PENTEST	Teste de Penetração
PHP	Hypertext Preprocessor (Pré-processamento de Hipertexto)
POM	Project Object Model (Projeto de Modelo de Objeto)
RIU	Requisitos de Interface com o Usuário
RN	Regras de negócio
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language (Linguagem Query Estruturada)
TCP	Transmission Control Protocol (Protocolo de Controle de Transmissão)

UD	Diagrama de Casos de Uso
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)
URL	Uniform Resource Locator (Localizador Padrão de Recursos)
USC	Modelo de Casos de Uso
W3C	World Wide Web Consortium (Consórcio World Wide Web)
XHTML	Extensible HyperText Markup Language (Linguagem Extensível para Marcação de Hipertexto)
XML	Extensible Markup Language (Linguagem Extensível de Marcação Genérica)
XSS	Cross-site scripting

LISTA DE FIGURAS

Figura 1 – Modelo Cascata	37
Figura 2 – O processo de desenvolvimento de Protótipo	38
Figura 3 – Modelo Espiral.....	39
Figura 4 – Diagrama de Caso de Uso	53
Figura 5 – Diagrama de Classes	63
Figura 6 – Estrutura da Base de Dados	64
Figura 7 – Página Principal	65
Figura 8 – Tela de Consulta	65
Figura 9 – Página de Autenticação	66
Figura 10 – Página Inicial do Administrador	66
Figura 11 – Página de Templates	67
Figura 12 – Página Novo Template	67
Figura 13 – Página Editar Template	68
Figura 14 – Página Excluir Template	68

LISTA DE TABELAS

Tabela 1 - Primeira iteração do processo de desenvolvimento	45
Tabela 2 - Segunda iteração do processo de desenvolvimento	46
Tabela 3 - Terceira iteração do processo de desenvolvimento	46
Tabela 4 - Quarta iteração do processo de desenvolvimento	47
Tabela 5 – Quinta iteração do processo de desenvolvimento	48
Tabela 6 – Relação das Regras de negócio (RN)	50
Tabela 7 – Relação dos Requisitos Funcionais (RF)	50
Tabela 8 – Relação dos Requisitos não Funcionais (RNF)	51
Tabela 9 – Relação dos Requisitos de interface com o usuário (RIU)	52
Tabela 10 – Relação das Mensagens do Sistema (ME)	52
Tabela 11 – Relação de Atores do Sistema	52
Tabela 12 – Relação dos Casos de Uso	53
Tabela 13 – Caso de Uso Analisar Domínio.....	54
Tabela 14 – Caso de Uso Consultar Domínio	56
Tabela 15 – Caso de Uso Autenticar Usuário	57
Tabela 16 – Caso de Uso Criar Template	58
Tabela 17 – Caso de Uso Editar Template	60
Tabela 18 – Caso de Uso Excluir	61

SUMÁRIO

1 INTRODUÇÃO	16
1.1 JUSTIFICATIVA DO TEMA	16
1.2 PROBLEMA DA PESQUISA	17
1.2.1 Hipóteses ou suposições	17
1.3 OBJETIVOS	17
1.3.1 Objetivo Geral	17
1.3.2 Objetivos Específicos	17
1.4 PROCEDIMENTOS METODOLÓGICOS	17
1.5 ORGANIZAÇÃO DA MONOGRAFIA	18
2 INFORMAÇÃO	20
2.1 COMUNICAÇÃO	22
2.2 ATIVO	23
2.3 CLASSIFICAÇÃO DA INFORMAÇÃO	23
3 SEGURANÇA DA INFORMAÇÃO	26
3.1 AMEAÇAS	27
3.2 VULNERABILIDADES	28
3.3 MEDIDAS DE SEGURANÇA	28
3.4 IMPACTO E INCIDENTE	29
3.5 RISCOS	29
4. DESENVOLVIMENTO WEB	31

4.1 INTERNET	31
4.2 WEB	32
4.3 LINGUAGENS DE PROGRAMAÇÃO	33
4.3.1 Html	33
4.3.2 Xhtml	33
4.3.3 Css	34
4.3.4 Xml	34
4.3.5 Javascript	34
4.3.6 Java	35
4.3.7 Php	35
4.4 PROJETO DE SISTEMA	35
4.4.1 A Engenharia de Software	36
4.4.1.1 Modelo Cascata	36
4.4.1.2 Modelo Prototipação	37
4.4.1.3 Modelo Espiral	38
4.4.2 Gerenciamento, controle e persistência de dados	40
4.4.2.1 Maven	40
4.4.2.2 Banco de dados	41
4.4.2.3 Hibernate ORM	42
4.4.2.4 Whois	42
5 PENTEST	43

5.1 SQL INJECTION	43
6 O SISTEMA.....	45
6.1 PROCESSO DE DESENVOLVIMENTO	45
6.2 MODELAGEM DO SISTEMA	48
6.2.1 Diagrama de Casos de Uso	49
6.2.1.1 Escopo	49
6.2.1.2 Regras de Negócio.....	50
6.2.1.3 Requisitos.....	50
6.2.1.3.1 <i>Requisitos Funcionais (RF)</i>	50
6.2.1.3.2 <i>Requisitos Não Funcionais (RNF)</i>	51
6.2.1.3.3 <i>Requisitos de Interface com o Usuário (RIU)</i>	52
6.2.1.3.4 <i>Mensagens do Sistema</i>	52
6.2.1.4 Modelo de Casos de Uso (USC)	52
6.2.1.4.1 <i>Relação de Atores</i>	52
6.2.1.4.2 <i>Lista de Casos de uso</i>	53
6.2.1.4.3 <i>Diagrama de Caso de Uso</i>	53
6.2.1.4.4 <i>Detalhamento de Caso de Uso</i>	54
6.2.2 Diagrama de Classes	63
6.2.3 Estrutura da Base de Dados.....	63
6.2.4 Funcionamento do Sistema.....	64
6.2.4.1 Página principal.....	64

6.2.4.2 Página de consulta	65
6.2.4.3 Página de autenticação do usuário	66
6.2.4.4 Página Inicial da Área do Administrador.....	66
6.2.4.5 Página de templates.....	67
6.2.4.6 Página de criação de novo template	67
6.2.4.7 Página de edição de template	68
6.2.4.8 Página exclusão de template	68
7 CONSIDERAÇÕES FINAIS	69
REFERÊNCIAS.....	70
GLOSSÁRIO.....	73

1 INTRODUÇÃO

A informação é um dos bens mais preciosos de uma organização. Há algum tempo, esse bem poderia ser facilmente protegido dentro de um local físico, como uma gaveta. Sabe-se que no contexto organizacional, a informação tem sua preservação exercida através da existência de políticas de segurança, que descrevem a filosofia e procedimentos fundamentais para sua utilização através da elaboração e definição de um conjunto de estratégias, regras e padrões.

Com a utilização cada vez mais crescente de sistemas informatizados para as mais variadas finalidades, atualmente, a informação encontra-se sujeita a um extenso número de ameaças e dos mais variados tipos. Tais ameaças se manifestam, por exemplo, no formato de ladrões cibernéticos, vírus capazes de destruir arquivos no disco rígido, fraudes diversificadas ou em incidentes de origem natural.

1.1 JUSTIFICATIVA DO TEMA

A variação constante no cenário de ameaças para a segurança de aplicações, devido aos avanços alcançados por parte dos atacantes ou a lançamentos de novas tecnologias que apresentam novas vulnerabilidades, exige medidas de defesa também atualizadas. Em razão disso, a *Open Web Application Security Project* (OWASP), uma organização sem fins lucrativos, cuja maior parte dos associados são voluntários, incluindo a Direção da OWASP, os Comitês Globais, os Líderes dos Capítulos, os Líderes de Projetos e os membros dos projetos, tem como objetivo promover a segurança nas aplicações, através da disponibilização de recursos grátis e abertos a todos os interessados.

Em 2013 foi divulgada, pela OWASP, uma lista com 10 riscos de Segurança em Aplicações (OWASP TOP 10), uma versão atualizada da mesma lista divulgada em 2010. A elaboração desta se baseia na gravidade dos riscos que incidem em um grande número de organizações, e acompanha informações sobre probabilidade de ocorrência e impacto técnico desses riscos. A indicação é que cada risco seja analisado com foco no ambiente de negócio da empresa em questão.

1.2 PROBLEMA DA PESQUISA

Atualmente, analisar os riscos presentes em uma aplicação *web* além de ser possível somente através de uma consultoria, apresenta custo elevado, inviabilizando a prática para muitas organizações. De que forma é possível fornecer um serviço além de gratuito, *online* e aberto ao público, para apurar vulnerabilidades de sistemas *web*?

1.2.1 Hipóteses ou suposições

Algumas das vulnerabilidades mais presentes em aplicações *web* foram listadas pela organização OWASP em um *top 10* publicado em 2013. A aplicação desenvolvida durante este projeto visa identificar o risco de maior relevância destes, chamado *Sql Injection*.

1.3 OBJETIVOS

Desenvolvimento de uma aplicação de realização de testes em outras aplicações *web*.

1.3.1 Objetivo Geral

Criação de uma ferramenta para realização de testes em sistemas *web* para análise de vulnerabilidade dos mesmos.

1.3.2 Objetivos Específicos

A aplicação objetiva verificar se é possível estabelecer uma conexão via HTTP, recuperar as informações do respectivo servidor autoritativo, mapear os *links* de acesso e testar as vulnerabilidades apresentadas pela aplicação.

1.4 PROCEDIMENTOS METODOLÓGICOS

O desenvolvimento do projeto é fundamentado em pesquisas bibliográficas e aplicadas. Os procedimentos adotados percorrem desde a delimitação do tema, até a conclusão do projeto, seguindo os passos:

1. Delimitação do Tema: Nesta etapa foi feita a escolha do tema e definido os objetivos que se pretende alcançar com o estudo;

2. Levantamento da bibliografia: Após a delimitação do tema foi realizado o levantamento de livros, artigos científicos, revistas e sites relacionados ao tema proposto;
3. Leitura e documentação: Pesquisa e seleção do conteúdo apropriado e relevante para elaboração do trabalho, a partir do material levantado na etapa anterior;
4. Construção Textual: Desenvolvimento da estrutura textual do trabalho, elaboração do conteúdo com embasamento teórico dentro dos padrões estabelecidos;
5. Projeto e desenvolvimento: Criação da aplicação com base na documentação já desenvolvida.
6. Conclusão: Apresentação da aplicação para testes de vulnerabilidades, juntamente com uma crítica sobre os resultados obtidos e as possibilidades de implementações futuras a partir desses resultados.

1.5 ORGANIZAÇÃO DA MONOGRAFIA

O capítulo 2 define o conceito de Informação, evidenciando sua importância no cenário de negócios atual. Inicialmente, diferencia os conceitos de dado e informação, posteriormente, introduz outros, fundamentais: comunicação, conhecimento e inteligência. Segue com o tratamento da informação como um ativo nas organizações e, por fim, aborda sua classificação.

O capítulo 3 aborda definições acerca da Segurança da Informação, conceituando elementos fundamentais para a compreensão do tema e da aplicação proposta. Tais como ameaças, vulnerabilidades, medidas de segurança, impacto e incidente e risco.

O capítulo 4, a respeito de Desenvolvimento *Web*, aborda conceitos relacionados às tecnologias utilizadas para o funcionamento da *internet*, bem como definições das principais características de uma página *web* e um *site web*. São abordadas também, as definições de algumas linguagens de programação, modelos de processo de engenharia de software, e ferramentas de gerenciamento, controle e persistência de dados adotadas no projeto.

O capítulo 5 trata sobre a técnica *Pentest*, conceituando e apontando o cenário de utilização. No primeiro e único subcapítulo, chamado “5.1 *SQL Injection*”, é definido o método de ataque de maior incidência atualmente, conhecido em português como Injeção de Código, explanando como este opera, e como pode ser identificado.

O capítulo 6 aborda o sistema como um todo, desde a definição simplificada a detalhes do desenvolvimento e da execução. É iniciado com uma breve descrição do sistema, exemplificando a área de atuação, abordando os elementos gráficos que compõe a interface da aplicação e explicando a maneira como pode ser acessado. Nos demais subcapítulos, há o levantamento de informações acerca do processo de desenvolvimento da aplicação, contando, no primeiro subcapítulo, com a descrição em tabelas de todas as iterações presentes. No segundo subcapítulo, denominado “6.2 Modelagem do Sistema”, há uma breve definição do que se trata uma Modelagem de Sistema seguido do conceito e descrição de todos os componentes presentes na aplicação. Em “6.3 Funcionamento do Sistema”, são listadas as funcionalidades da aplicação, com capturas de todas as telas exibidas pelo sistema junto de uma breve descrição de cada uma delas, apontando o que oferecem ao usuário ou ao administrador.

O capítulo 7, por ser o último capítulo, discorre a respeito dos objetivos da aplicação, o que foi alcançado, a utilidade do sistema no âmbito acadêmico e objetivos em relação ao futuro da aplicação, como aperfeiçoamento de funcionalidades.

2 INFORMAÇÃO

O mundo industrializado enfrenta, desde a década de 70, um período de transição definido pela caminhada em direção a um novo modelo de economia. Este modelo apresenta como principal característica a sustentação na informação, e seu sucesso é determinado por aquilo que se sabe, ou seja, por tudo aquilo que se é conhecido, em detrimento daquilo que se possui, como por exemplo, capital e terras (MCGEE, 1993).

Atualmente vivemos na chamada “Era da informação”, em que a informação é considerada um importantíssimo recurso, afirma Chiavenato (2010), capaz de auxiliar no processo de decisão e no alcance de objetivos no âmbito organizacional, aponta Oliveira (1993) e, por essa razão, de crucial importância no cenário de negócios vigente.

Segundo Chiavenato (2010), a informação pode ser definida como um conjunto de dados que apresenta um significado específico, uma ferramenta que reduz a incerteza em relação a algo cuja pretensão é ser compreendido, conhecido. É importante a discussão de três conceitos relativos à informação para uma melhor compreensão do conceito. São eles: dado, informação e comunicação.

De acordo com Chiavenato (2010, p. 52),

Dado: a palavra dado significa um fato. Assim, os dados são fatos que constituem a matéria-prima básica da informação. Mas não são a informação. Os dados são representados por símbolos, como letras, números, quantidades, códigos, cores, sinais, etc.

No contexto da informática, ciência responsável por estudar o tratamento da informação em meios digitais, segundo Silva (2015), os dados podem ser divididos em três tipos primitivos: dados numéricos, representados por números inteiros ou reais, dados caracteres e dados lógicos. Os dados numéricos inteiros são todos os números positivos ou negativos, como 10, 0 e 3. Dados numéricos do tipo reais são dados fracionários, como, por exemplo, 7,5 e 1,3 e também podem ser positivos ou negativos. Em se tratando de dados do tipo caracteres (ou alfanuméricos), são dados que contém letras, símbolos especiais e números, também conhecidos como *strings*, literal ou cadeia de caracteres. Por fim, os dados de tipos lógicos, ou *booleanos*, como também são conhecidos, são aqueles que apresentam os valores

de verdadeiro ou falso. Capron e Johson (2004) definem dado, como a matéria-prima a ser processada pelo computador, de forma a, uma vez atribuída importância, tornar-se informação.

Segundo Chiavenato (2010, p. 52),

Informação: é o conhecimento relevante produzido como resultado de um conjunto de dados ou de um processamento de dados. Ela ocorre quando os dados são analisados e processados e passam a ganhar um significado. A informação é constituída por um conjunto de dados que contém algum significado ou que reduz a incerteza a respeito de alguma coisa. O dado em si não possui significado, a informação sim. Os números 1, 16 e 1946 são dados que, em si, não têm significado algum. Contudo, se arranjados da seguinte maneira: 16/1/1946, representam uma informação, por exemplo, a data de aniversário de uma pessoa muito querida.

Informações e dados, quando analisados e interpretados quanto a sua relevância, confiabilidade e importância, oferecem a oportunidade de identificar uma situação. A esta oportunidade é designada o conceito de conhecimento. Uma vez identificada e construída uma situação, é possível optar pelas melhores decisões, caracterizando o conceito de Inteligência, cujo objetivo é alcançar o êxito esperado de uma determinada situação (CARDOSO JÚNIOR apud DANTAS, 2011).

Uma vez aliada aos conceitos de conhecimento e inteligência, a informação oferece oportunidades, criando valor para organizações e, dessa maneira, tornando-se sinônimo de vantagem competitiva, conjunto de características que tornam uma empresa economicamente superior. Por essa razão, a informação apresenta-se como elemento motriz no que se diz respeito ao ambiente competitivo corrente, e, representa também, alta influência corporativa, afirma Dantas (2011), seja na realização de acordos de sucesso entre pessoas, empreendimentos, blocos econômicos, povos e nações. Dessa forma, é natural que o mundo moderno dedique significativa atenção à informação.

É importante, aponta Chiavenato (2010), enfatizar que o conceito de informação reflete importância na compreensão de um sistema, uma vez que os componentes deste estão interligados entre si, constituindo uma rede de informação. Um sistema de informação, define Ralph (2002 apud DANTAS, 2011), é um conjunto de elementos ou componentes que estão relacionados e coletam, manipulam e disseminam dados e a informação, buscando atender a um objetivo.

Sabe-se que o volume de informação sobre um sistema, significa o conhecimento que se tem do mesmo; “Para se conhecer um sistema, a primeira coisa a se fazer é obter informações sobre ele” (CHIAVENATO; IDALBERTO, p.52). Dessa forma, a informação atua na redução de incertezas a respeito de um sistema, sendo o funcionamento do mesmo intimamente ligado a canais de informação e seu desempenho (CHIAVENATO, 2010).

Por ser tratar de um conjunto de dados que apresentam um significado, a informação tem, como característica principal, a capacidade de transmitir uma mensagem, objeto central de um determinado tipo de comunicação. A informação pode ser transmitida através de processos transacionais, que envolvem, por exemplo, operações relacionadas à transferência de valores monetários, ou através do ato de comunicação entre indivíduos e máquinas, em que a informação é passada de um para o outro (SÊMOLA, 2003).

2.1 COMUNICAÇÃO

A respeito do conceito de Comunicação, Chiavenato (2010), define como uma informação a ser transmitida por alguém e, além de recebida, interpretada por outrem. Toda a informação, a menos que seja de caráter estritamente confidencial, deve ser comunicada. “Comunicação significa tornar comum uma informação”, afirma Chiavenato (2010, p.53). O modelo comunicativo de Roman Jakobson (apud GUIMARÃES, 2012) evidencia seis elementos presentes em qualquer ato de comunicação humana, sendo eles:

- a. Mensagem: o que será transmitido, o conjunto de informações a serem enviadas,
- b. Emissor ou remetente: quem está enviando a mensagem,
- c. Receptor ou destinatário: a quem se destina a mensagem,
- d. O código: sistema de signos que ambos, receptor e emissor, precisam compartilhar a fim de estabelecer compreensão,
- e. O canal ou contato: meio físico que tornará possível a comunicação entre o emissor e o receptor,
- f. O referente ou contexto: o assunto da mensagem.

O processo de comunicação é influenciado por diversos fatores, externos e internos, que atrapalham a sua eficiência. De acordo com Chiavenato (2010), uma mensagem que sofre perdas, interferências ou ruídos, representa uma falha na comunicação. Um projeto de sistema de comunicação eficiente considera a noção de conteúdo de informação e a sua transferência (YOUNG, 2006).

2.2 ATIVO

Um ativo, segundo ISO/IEC13335-1:2004 (apud DANTAS, 2011), é definido como qualquer coisa que apresente valor para uma organização. Ou seja, tudo aquilo que faz parte dos processos responsáveis por manipular e processar a informação, hoje o ativo principal para organizações, afirma Dantas (2011), pode ser classificado como um ativo. A exemplo, além da própria informação, tem-se: o meio em que a informação se encontra e os equipamentos utilizados para seu manuseio, transporte e descarte (SÊMOLA, 2003).

Entre as muitas formas de divisão e agrupamento de ativos para facilitar seu tratamento, há a que os separa em: equipamentos, aplicações, usuários, ambientes, informações e processos (FONTES, 2008).

Através de um inventário de ativos há a possibilidade de classificar as informações das quais se tem posse, bem como seus respectivos proprietários. Tais informações podem se apresentar de diferentes formas, como por exemplo: transmitidas verbalmente, através de documentos eletrônicos, documentos em papel, *e-mail*, sistemas de informação/bases de dados e mídias de armazenamento, como cartões de memória (KOSUTIC, 2014).

2.3 CLASSIFICAÇÃO DA INFORMAÇÃO

De acordo com Fontes (2008), a principal razão que leva a necessidade de se classificar informações, está na variação de níveis de confidencialidade e nas interpretações pessoais de cada indivíduo dentro de uma organização quanto ao grau de confidencialidade das mesmas. Para definir a classificação da informação, é necessário levantar alguns pontos.

O responsável por definir o nível de classificação em organizações, é, segundo Fontes (2008), geralmente, o gestor da informação. Ele tem como

designação, definir o grau de sigilo de informações estruturadas, que são as informações acessadas através de transações. Tudo aquilo gerado pelo usuário, como planilhas e relatórios, por sua vez, deve ser classificado por este, através de análises e avaliações de riscos, determinando sua relevância a partir da gravidade das consequências de uma quebra de confidencialidade das referentes informações (KOSUTIC, 2014).

O nível de confidencialidade, afirma Fontes (2008), pode sofrer variação com o passar do tempo, passando de nível altamente confidencial, a de conhecimento geral. A informação pode ser classificada em diferentes níveis de importância, variando de acordo com a necessidade de cada organização (KOSUTIC, 2014).

Geralmente, a classificação acontece através da identificação do grau de sigilo, porém, também pode ser classificada de outras maneiras, como por exemplo, de acordo com o seu custo diante ou gestor, tempo de retenção em cópias de segurança e também através das providências tomadas em relação ao seu descarte mediante encerramento da organização (FONTES, 2008).

O número de níveis estimado no mercado atual varia entre três a cinco, de acordo com Fontes (2008), sendo, geralmente:

- a. Confidencial: representa o mais alto nível de confidencialidade, sendo de suma importância proteger as informações que aqui se enquadram de acesso externo. A violação destas pode levar a organização à perda de competitividade e a prejuízos financeiros. São exemplos de informações confidenciais: senhas, dados pessoais de clientes, salários,
- b. Restrita: estão situadas no nível médio de confidencialidade. São as informações que independente de sua natureza precisam estar restritas a um número determinado de pessoas, sendo necessárias medidas específicas para protegê-las,
- c. Interna: informações deste nível representam o mais baixo nível de confidencialidade, ou seja, se forem acessadas não causarão grande impacto. Exemplo: agenda telefônica,
- d. Pública: informações de acesso público não apresentam a necessidade de sigilo uma vez que todos podem visualiza-las. Não é

necessário investir nestas para torna-las seguras. São exemplos de informações públicas: folders e teste de sistemas e serviços.

A implementação das informações deve acontecer de forma gradual e, para cada nível de confidencialidade, deve ser definido sua guarda física, transmissão por correio eletrônico, autorização por cópias, transporte no malote interno, envio via correio convencional, transmissão via fax e forma de destruição física.

Em relação ao nível de granularidade da informação, que classifica a informação como um campo, arquivo, ou conjunto de campos, por exemplo, de acordo com Fontes (2008, p.194):

O nível de granularidade da informação a ser utilizada para a classificação em relação ao sigilo deve ser um nível que o gestor entenda e seja significativo para ele. No ambiente computacional considerado para efeito do gestor, um bom conjunto de referência é: transações, telas geradas por transações, relatórios gerados por transações. Para o ambiente computacional mais técnico, podemos considerar que o nível de sigilo de um arquivo deve ser igual ao maior nível de sigilo do conjunto de transações que tomam por base esse arquivo.

A classificação da informação tem como objetivo principal, garantir um nível de proteção adequado com o propósito de preservar o que é priorizado pela organização, seja ela pública ou privada (KOSUTIC, 2014).

3 SEGURANÇA DA INFORMAÇÃO

Em sua definição, por Sêmola (2003), a Segurança da Informação é considerada uma área de conhecimento que objetiva a adoção de medidas que impeçam acessos e alterações não autorizados, bem como a indisponibilidade, dos ativos da informação.

Dependendo do contexto em que estiver inserida, pode ser considerada um “meio” ou um “fim” de todo o processo, conforme afirmação de Sêmola (2003, p. 40):

A expressão “Segurança da Informação” é, por si só, um termo ambíguo, podendo assumir dupla interpretação:

1. Segurança como uma prática adotada para tornar um ambiente seguro (atividade, ação, preservação dos princípios), de caráter interdisciplinar, composta de um conjunto de metodologias e aplicações que visam estabelecer: controles de segurança (por exemplo: de autenticação, autorização e auditoria) dos elementos constituintes de uma rede de comunicação e/ou que manipulem a informação; e procedimentos para garantir a continuidade de negócios na ocorrência de incidentes.
2. Resultado da prática adotada, objetivo a ser alcançado. É a característica que a informação adquire ao ser alvo de uma prática de segurança (segura – adjetivo, objetivo da prática).

De acordo com Dantas (2011), para que a informação possa ser utilizada, é necessário garantir a preservação de seus três princípios básicos:

- a. Integridade – Garantir a integridade é impedir a modificação, alteração ou destruição da informação, sem que estas ações estejam autorizadas.
- b. Disponibilidade – É a garantia de que o acesso à informação e aos ativos correspondentes a ela estejam sempre disponíveis para as pessoas autorizadas.
- c. Confidencialidade – É a garantia de que somente as pessoas autorizadas podem acessar determinada informação.

Além disso, dependendo do que se pretende alcançar, alguns aspectos são considerados essenciais na prática da segurança da informação, afirma Sêmola (2003), são eles:

- a. Autenticação – É o processo de controle de identificação e reconhecimento dos elementos que fazem parte da transação eletrônica, possibilitando o acesso à informação.

b. Legalidade – É o atributo da informação que se encontra de acordo com o estabelecido em contrato ou legislação vigente.

Dantas (2011) acrescenta outras propriedades que também podem estar envolvidas no processo:

a. Autenticidade – Garantia de que a informação é procedente da fonte indicada.

b. Responsabilidade – É a participação conjunta de responsabilidades, por todos que participam do ciclo de vida da informação.

c. Não Repúdio – Garantia de que a informação chegará ao seu destino.

d. Confiabilidade – Garantia da veracidade da informação, se a procedência é verdadeira e, portanto, confiável.

Elas são definidas, sucintamente, por Dantas (2011, p. 15):

Dessa forma, a autenticidade do emissor é a garantia de que quem se apresenta como remetente é realmente quem diz ser. A confiabilidade é a garantia de que a informação está completa e igual à sua forma original quando do envio pelo remetente, e expressa uma verdade. O não repúdio é a garantia de que o emissor ou receptor não tem como alegar que a comunicação não ocorreu, e a responsabilidade diz respeito aos deveres e proibições entre remetente e destinatário.

3.1 AMEAÇAS

Segundo Sêmola (2003), as ameaças são responsáveis por causar danos à informação e seus ativos, agindo através da exploração de suas vulnerabilidades, resultando em perdas e impacto aos seus conceitos. Elas podem ser classificadas de acordo com seu intuito:

a. Naturais - que decorrem de fenômenos da natureza.

b. Voluntárias - que são causadas por agentes humanos, propositalmente.

c. Involuntárias - que são causadas, na maioria das vezes, inconscientemente, por desconhecimento.

Para Dantas (2011), a mecanização dos sistemas que processam e armazenam a informação contribui para maior acessibilidade e disponibilidade da mesma, deixando-a mais propensa às ameaças.

3.2 VULNERABILIDADES

As vulnerabilidades são elementos inertes que necessitam de elementos causadores (ameaças) para gerar incidentes. Estão relacionadas com as fragilidades que podem estar contidas nos processos, políticas, equipamentos e nos recursos humanos, define Dantas (2011).

Sêmola (2003) as classifica da seguinte forma:

- a. Físicas – Inadequação de instalações prediais, como falha no planejamento de salas e ausência de equipamentos de segurança.
- b. Naturais – Incêndios, terremotos, tempestades, falta de energia, etc.
- c. *Hardware* – Desgaste, obsolescência, má utilização ou má instalação dos recursos tecnológicos.
- d. *Software* – Má instalação ou configuração, que resultem em acessos indevidos, vazamento de informações, perda de dados ou indisponibilidade dos mesmos.
- e. Mídias – Perdas e danos de fitas, discos, relatórios e impressos.
- f. Comunicação – Acessos não autorizados ou perda de comunicação.
- g. Humanas – Falta de treinamento, falta de sigilo com informações confidenciais, falta de prática das rotinas de segurança, ações de vandalismo, sabotagem, greve, etc.

3.3 MEDIDAS DE SEGURANÇA

As medidas de segurança são consideradas a união dos esforços para proteção da informação, com o objetivo de reduzir as vulnerabilidades, de impedir que elas sejam exploradas pelas ameaças, limitar os impactos ou diminuir os riscos de alguma forma. Com essa afirmação, Sêmola (2003) as considera controles que podem ter as seguintes características:

- a. Preventivas – Objetivam a prevenção na ocorrência de incidentes, visando a manutenção da segurança já implementada através de mecanismos estabelecidos pela instituição.
- b. Detectáveis – Visam identificar os causadores de ameaças, a fim de evitar que as vulnerabilidades sejam exploradas por elas.

c. Corretivas – Tem como finalidade a adaptação às condições de segurança estabelecidas pela instituição, ou redução de impactos, através da correção das estruturas tecnológicas e humanas.

As medidas de segurança podem possuir mais de uma característica. Dessa forma, a classificação serve apenas para identificar o foco do trabalho proposto. (SÊMOLA, 2003)

3.4 IMPACTO E INCIDENTE

Sêmola (2003) considera os impactos como a compreensão dos prejuízos gerados por um incidente de segurança nos processos de negócios de uma organização. Ainda segundo ele, um incidente é consequência das ameaças que agem sobre as vulnerabilidades, e sua gravidade pode ser medida pelo impacto que produz.

A perda dos requisitos de segurança, para Dantas (2011), "está diretamente relacionada com os seguintes impactos: divulgação (confidencialidade); modificação (integridade); perda, destruição e interrupção (disponibilidade)."

3.5 RISCOS

A probabilidade de um incidente ocorrer, causando a perda dos princípios básicos da segurança da informação e um possível impacto nos negócios, é definida como risco. (SÊMOLA, 2003)

Para Dantas (2011), o risco é representado pela combinação de dois elementos: a consequência e a probabilidade. Ele afirma ainda que, considerando a probabilidade de uma ação se concretizar, suas consequências são negativas.

Os elementos citados por Dantas (2011) são as variáveis utilizadas na equação do risco que obtém como resultado o produto do impacto pela frequência em que determinado evento ocorre, e é representada pela fórmula:

$$R (\text{risco}) = C (\text{consequência}) \times P (\text{frequência})$$

A medida da frequência pode ser baseada em dados históricos, através de pesquisas; erros e falhas (humanas e de equipamentos e sistemas); ou na

combinação de eventos externos, que se mede pelo cálculo do produto das ameaças pelas vulnerabilidades (DANTAS, 2011). Já as consequências, ainda segundo Dantas (2011), são representadas pela soma dos possíveis efeitos da concretização de um evento, e podem ser calculadas por níveis ou financeiramente.

Para Sêmola (2003), as variáveis do cálculo do risco dependem de cada negócio e a identificação destas é o primeiro passo da operação. Ele inclui na equação, além dos elementos utilizados por Dantas (2011), as medidas de segurança. Tais medidas devem ser consideradas por interferirem na redução ou no aumento do risco.

Portanto, a equação de risco de segurança da informação é definida por Sêmola (2003) como o produto das vulnerabilidades, pelas ameaças e impactos, divididos pelas medidas de segurança, e sua fórmula é representada por:

$$R = (V \times A \times I) / M$$

Sêmola (2003) atenta ao fato de não haver segurança total e, portanto, à necessidade de concentrar os esforços na preparação para as mudanças comportamentais que as variáveis estão sujeitas, influenciando no cálculo do risco, cujo resultado sofre variação. Dessa forma, ele pode ser ajustado com agilidade aos padrões estabelecidos pela organização.

4 DESENVOLVIMENTO WEB

Com o passar das décadas, a *web* foi abraçada por milhares de empresas, por se tratar de um canal econômico para comunicar informações e realizar transações com clientes. Ela provê uma forma dos marqueteiros, profissionais responsáveis pela aplicação do marketing, conhecerem as pessoas que visitam seus sites, tornando-os capazes de identificar abordagens eficientes com as mesmas.

4.1 INTERNET

A *internet* pública, também conhecida somente como *internet*, é a rede global de computadores que interconecta milhões de equipamentos de computação ao redor do mundo. São chamados de hospedeiros ou sistemas finais, todos os equipamentos utilizados para a navegação na *internet*, desde computadores pessoais ou *PCs* como também são conhecidos, a TVs, computadores portáteis, telas de fotos, telefones celulares, automóveis, câmeras *web* entre outros (KUROSE; ROSS, 2006).

Um provedor de serviços de *internet*, ou ISP (*Internet Service Provider*), provê o computador servidor e o *software* para a realização da conexão com a *internet* (CAPRON; JOHNSON, 2004). *ISPs* podem ser residenciais, corporativos, de universidades ou como o *T-Mobile*, que fornece acesso em locais públicos como *shoppings*, aeroportos e hotéis (KUROSE; ROSS, 2006).

Segundo Capron e Johnson (2004), o protocolo mais importante e que torna a *internet* universalmente possível é denominado TCP/IP (*Transmission Control Protocol* – Protocolo de Controle de Transmissão, *Internet Protocol* – Protocolo da Internet), e permite a qualquer computador comunicar-se com a *internet*.

A chamada *Intranet*, como é denominada a *internet* privada, é uma rede que pode ser de caráter corporativo ou governamental em que os hospedeiros ou *hosts* não trocam mensagens com outros hospedeiros se estes se encontrarem fora dela (KUROSE; ROSS, 2006).

A *internet* possibilita que aplicações distribuídas, quando executadas em seus sistemas finais, troquem dados entre si, estando entre essas aplicações: mensagens instantâneas, áudios, vídeos, telefonia, jogos distribuídos, compartilhamento de

arquivos, correio eletrônico, navegação *web* entre diversas outras. Novas aplicações são criadas e disponibilizadas todos os dias, sendo a *internet* uma infraestrutura guiada pelos avanços na tecnologia (KUROSE; ROSS, 2006).

4.2 WEB

A *web*, palavra de origem inglesa e que significa “teia” ou “rede”, é constituída de quatro componentes fundamentais, de acordo com Kurose e Ross (2006): HTML, HTTP, servidor para a *web* e *browser*, ou como também é conhecido, navegador.

Um *site web* pode ser definido como um portal de informações de acesso a nível mundial ou a um número específico de pessoas, sendo destinado aos mais variados assuntos e oferecendo diversos tipos de informações. A ideia é colocar informações à disposição do mundo inteiro ou de um grupo selecionado de pessoas de maneira rápida, tornando-as disponíveis facilmente aos clientes e a outros usuários nelas interessados. Os *sites web* podem conter quaisquer tipos de informações, desde dados de uma empresa a *clips* de vídeos de Guerra nas Estrelas (SHARMA; VIVEK; RAJIV, 2000).

É através de uma página *web* que um *site web* tem suas informações disponibilizadas na *internet*. A disponibilização acontece por meio da representação de um tipo que pode ser entendida por navegadores, também conhecidos como *browsers* ou *web browsers*. Capron e Johnson (2004) definem como navegador, um *software* de interface utilizado para explorar a *internet*.

Os navegadores possibilitam aos usuários interagirem com documentos hospedados em um servidor *web*. O servidor *web*, *software* que interpreta as solicitações do cliente, executa a ação apropriada a fim de possibilitar a leitura de tais documentos. Servidores *web* e clientes precisam de um protocolo para se comunicar, conhecido como HTTP (*Hyper-Text Transfer Protocol*), que determina a possibilidade de comunicação entre os participantes. (SHARMA; VIVEK; RAJIV, 2000).

Uma página *web* é acessada através de um caminho, chamado de URL (*Uniform Resource Locator*), como pode ser definido o endereço de um arquivo suportado pelo HTTP, como uma página ou uma imagem, na *internet*. (SHARMA; VIVEK; RAJIV, 2000).

Uma aplicação *web*, também conhecida como *web app*, é definida por Rouse (2011) como um programa de aplicação armazenado em um servidor remoto e oferecido na *internet* através de um navegador. Aniceto (2009) define uma aplicação *web* como um sistema de informática cujo acesso se dá através de um navegador, utilizando a *internet* ou a *intranet*. Seu desenvolvimento parte da necessidade de simplificar a utilização e manutenção de um sistema, sendo o código fonte mantido em um local em que pode ser acessado por diferentes usuários.

Segundo Conallen (2003) aplicações *web* se utilizam de tecnologias que facilitam a criação de conteúdo dinâmico e que permitem usuários do sistema afetar a lógica de negócios no servidor, sendo essa característica a principal diferença entre um *web site* e uma aplicação *web*.

4.3 LINGUAGENS DE PROGRAMAÇÃO

Uma linguagem de programação, de acordo com Puga e Rissetti (2009), é formada de palavras, denominadas palavras-chave, que agrupadas em frases, chamadas estruturas de programação, produzem um determinado significado. Sendo assim, um programa consiste de palavras-chave e estruturas de programação, organizadas de forma compreensível ao ser humano. São diversos os tipos de linguagem utilizados para o desenvolvimento de aplicações, variando de acordo com a tarefa que se pretende executar.

4.3.1 Html

Os documentos apresentados em uma página *web* encontram-se no formato de texto HTML (*Hyper-Text Markup Language*) que é uma linguagem simples, com uma série específica de *tags* que permitem a formatação de dados de forma que possam ser visualmente atrativos quando visualizados através de um navegador (SHARMA; VIVEK; RAJIV, 2000).

4.3.2 Xhtml

A linguagem de marcação XHTML, ou *Extensible HyperText Markup Language*, especifica o formato de texto em uma página *web*, sendo baseada na HTML e diferenciando-se desta pela possibilidade de incorporação de uma folha de estilos externa.

4.3.3 Css

A CSS, acrônimo de *Cascading Style Sheets*, ou, em português, folha de estilos, é uma tecnologia do W3C, ou *World Wide Web Consortium*, responsável por desenvolver padrões *web*, que permite a especificação e apresentação dos elementos em uma página *web*, como por exemplo, fontes, cores, espaçamentos, entre outros. Uma folha de estilos pode ser incorporada fora da XHTML ou na HTML, diretamente na seção *head*, que corresponde ao cabeçalho da página, utilizando o elemento *style*, responsável por definir características relacionadas ao estilo da página (DEITEL, 2008).

4.3.4 Xml

A XML não é propriamente uma linguagem, afirma Sharma, Vivek e Rajiv, (2000), sendo considerada uma série de especificações que servem de base para a definição de linguagens. Em outras palavras, é possível definir a sua própria linguagem com base nas regras XML, criando um conjunto de *tags* próprio para o fornecimento de informações sobre informações, sendo a XML orientada unicamente no sentido de dar informações, sem tratar dos aspectos de apresentação. Graves (2003), define que a linguagem XML representa dados como uma *string* de texto, incluindo uma marcação intercalada, que permite que o texto seja intercalado de acordo com seu conteúdo e forma, com o propósito de descrever as propriedades dos dados.

4.3.5 Javascript

Para a criação de páginas *web* dinâmicas, segundo Deitel (2008), são utilizados *scripts*, sendo a linguagem responsável pela criação destes chamada de *javascript*. A linguagem apresenta como característica principal sua alta portabilidade, contando com duas finalidades: introduzir a criação de *script* no lado do cliente, tornando as páginas *web* mais dinâmicas, e servir de base para a criação de *scripts* no lado servidor. Jorge (2004) define a *javascript* como uma linguagem de fácil assimilação e aprendizado, que permite o usuário final controlar seus programas de maneira eficaz, e aponta *scripts* como os responsáveis por preencher uma lacuna deixada pelas linguagens de programação.

4.3.6 Java

Java é uma linguagem de programação e uma plataforma de computação lançada em 1995 pela *Sun Microsystems*, companhia que vendia computadores, componentes, programas de computadores e serviços de TI.

É especializada na criação de pequenos programas e permite a criação de uma aplicação que não depende de outras, apresentando todos os recursos de uma linguagem que se destina a desenvolver aplicações, tal como a linguagem C (ASCENCIO; CAMPOS, 2007).

Além disso, fornece suporte ao paradigma orientado a objeto, afirma Jorge (2004), sendo a todos os programas, obrigatório o uso de classes, espécie de dado que armazena informações, denominadas atributos, e funções, chamadas de métodos.

Existem, atualmente, inúmeras aplicações e *websites* que não funcionam, a menos que o programa Java esteja instalado, se fazendo necessário e presente em *laptops, datacenters, celulares, internet* e muitas outras plataformas. (JAVA, 2015)

4.3.7 Php

A respeito da linguagem PHP ou *Hypertext Preprocessor*, criada por Rasmus Lerdorf, segundo Deitel (2008, p.462),

O PHP é uma tecnologia de código aberto que conta com o apoio de uma grande comunidade de usuários e desenvolvedores. É independente de plataforma, havendo implementações para todos os principais sistemas operacionais, como UNIX, Linux, Mac e Windows, e também aceita muitos bancos de dados, como o MySQL.

4.4 PROJETO DE SISTEMA

O projeto, segundo Pfleeger (2004), é o processo de transformação do problema em uma solução. Ele afirma, ainda, que a descrição de uma solução é também definida como projeto.

“A atividade do projeto de *software* engloba o conjunto de princípios, conceitos e práticas que levam ao desenvolvimento de um sistema ou produto com alta qualidade.” (PRESSMAN, 2011)

4.4.1 A Engenharia de Software

A engenharia de *software*, segundo Pfleeger (2004), está relacionada a resolução de problemas, necessariamente ou desejavelmente, com o uso de um sistema computacional. Este princípio é compartilhado por Pressman (2011), que acrescenta a utilização de processos, métodos e ferramentas na construção desses sistemas, processos estes englobados por: comunicação, planejamento, modelagem, construção e empregos.

Sommerville (2011, p. 6) reafirma o seu foco na produção do *software*, e inclui a manutenção do mesmo no processo, quando este já está em uso. Ele utiliza as seguintes expressões para definir a engenharia de *software*:

1. Disciplina de engenharia. Engenheiros fazem as coisas funcionarem. Eles aplicam teorias, métodos e ferramentas onde for apropriado. No entanto, eles os usam seletivamente e sempre tentam descobrir as soluções para os problemas, mesmo quando não há teorias e métodos aplicáveis. Os engenheiros também reconhecem que devem trabalhar de acordo com as restrições organizacionais e financeiras, então buscam soluções dentro dessas restrições.
2. Todos os aspectos da produção de *software*. A engenharia de *software* não se preocupa apenas com os processos técnicos do desenvolvimento de *software*. Ela também inclui atividades como gerenciamento de projeto de *software* e desenvolvimento de ferramentas, métodos e teorias para apoiar a produção do *software*.

Os processos de *software* seguem modelos, “alguns são receitas do caminho que o desenvolvimento de *software* deveria seguir e outros são descrições do modo como o desenvolvimento do *software* é realmente feito.” (PFLEEGER, 2004)

Para Sommerville (2011), cada modelo fornece informações parciais sobre o processo. Um modelo de processo genérico é uma abstração que explica o desenvolvimento, permitindo a ampliação e adaptação do mesmo. Pressman (2011) define como a forma em que os métodos, ações e tarefas são organizados cronologicamente e sequencialmente.

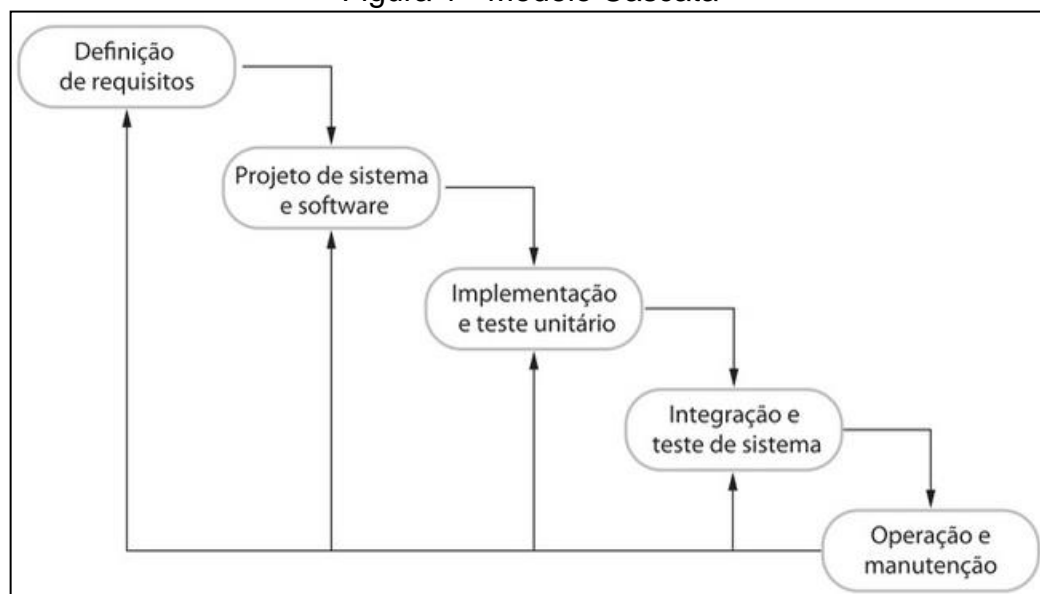
4.4.1.1 Modelo Cascata

O modelo cascata é um dos primeiros modelos propostos, segundo Pfleeger (2004), nele, cada etapa do processo deve ser terminada antes de uma próxima iniciar.

Sommerville (2011) o exemplifica como um processo dirigido a planos, afirmando que as atividades deve ser todas planejadas antes de serem colocadas em prática.

Para Pressman (2011), este modelo é também conhecido como ciclo de vida clássico e segue uma abordagem sistemática e sequencial para o desenvolvimento de *software*, que se inicia no levantamento de requisitos e segue até o suporte após sua conclusão, conformes o fluxo mostrado na figura 1.

Figura 1 - Modelo Cascata



Fonte: SOMMERVILLE, 2011.

4.4.1.2 Modelo Prototipação

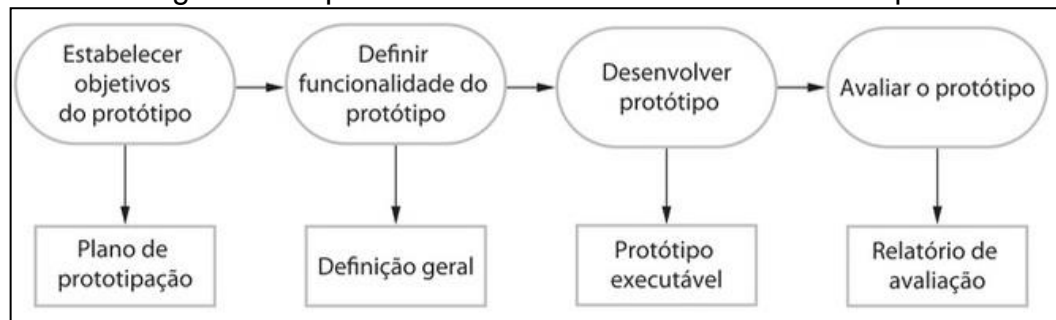
A prototipação pode ser usada como um modelo de processo separadamente, ou como uma técnica que pode ser inserida em outros modelos e, seu foco principal é auxiliar as partes envolvidas no projeto a entender o que será desenvolvido. (PRESSMAN, 2011)

Plfeeger (2004) justifica sua utilização como um modelo efetivo, afirmando que a prototipação permite uma construção rápida do sistema, ou de parte dele. Dessa forma, questões mal esclarecidas, nos requisitos ou no projeto, são expostas e discutidas entre desenvolvedor, usuário e cliente para que se chegue a um consenso sobre proposta e necessidade.

Sucintamente, Sommerville (2011) define um protótipo como a primeira versão de um sistema de *software*, em que são demonstrados os conceitos, testadas as opções do projeto, identificados os problemas e sugeridas as soluções.

De acordo com Pfleeger (2004), o desenvolvimento do sistema pode ser iniciado pelo levantamento simples de requisitos com os clientes e usuários. As alternativas devem ser estudadas a partir deste resultado e ele deve ser usado de base para que as decisões acerca do que se pretende obter sejam tomadas. Após esta fase, os requisitos são revisados e, de forma consensual, é decidido como eles deveriam ser. O desenvolvedor reconsidera e altera a especificação para, por fim, codificar o sistema e discutir as alternativas novamente.

Figura 2 - O processo de desenvolvimento de Protótipo



Fonte: SOMMERVILLE, 2011

4.4.1.3 Modelo Espiral

Sommerville (2011) define o modelo espiral como um *framework* de processo de *software* dirigido a riscos proposto por Boehm (1988), onde esse processo é apresentado como um espiral e cada volta representa uma de suas fases. A volta mais interna está relacionada com a viabilidade do sistema, o ciclo posterior, com a definição de requisitos, o seguinte, com o projeto do sistema, e assim por diante. Ele afirma que “o modelo em espiral combina prevenção e tolerância a mudanças, assume que mudanças são um resultado de riscos de projeto e inclui atividades explícitas de gerenciamento de riscos para sua redução”.

A cada iteração, as alternativas são ponderadas de acordo com os requisitos e restrições, e a decisão por alguma delas depende da viabilidade e adequação. (PRESSMAN, 2011)

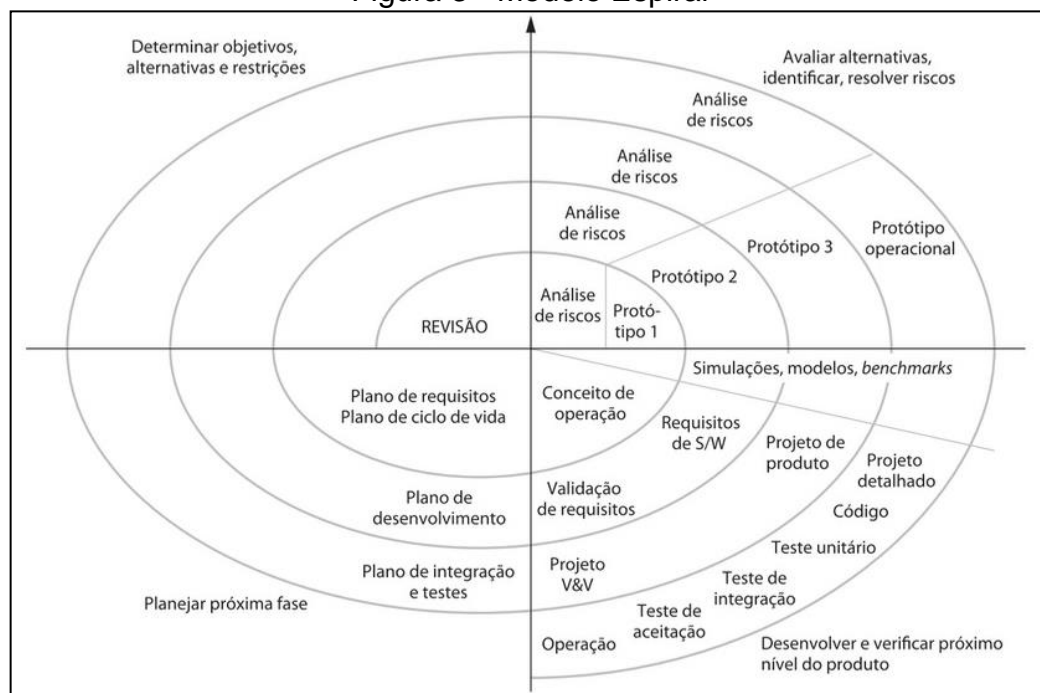
Como forma de exemplificar, Pfleeger (2004) afirma que os projetistas podem ficar inseguros ao apresentar uma interface “x” ou “y” para o usuário, portanto, são

criados protótipos de cada uma delas para que sejam testados e, só então, escolhido o que será utilizado no projeto. Dessa forma, reduzem-se os riscos de frustração na escolha antecipada da interface.

As voltas no espiral descritas por Sommerville (2011), e ilustradas na figura 3, se dividem em quatro setores:

1. Definição de objetivos: Nessa fase do projeto são definidos os objetivos específicos, identificadas as restrições, elaborado um plano de gerenciamento, identificados os riscos e, eventualmente, planejadas as estratégias em função desses riscos.
2. Avaliação e redução de riscos: Após a identificação dos riscos, e o detalhamento de cada um deles, formam-se medidas para sua redução.
3. Desenvolvimento e validação: Nessa etapa é escolhido o melhor modelo de desenvolvimento, com base na avaliação dos riscos da etapa anterior.
4. Planejamento: Este setor se dedica à revisão do projeto, com o objetivo de decidir a continuidade do modelo. A partir dessa decisão elaboram-se planos para a próxima volta do espiral.

Figura 3 - Modelo Espiral



Fonte: SOMMERVILLE, 2011.

4.4.2 Gerenciamento, controle e persistência de dados

O gerenciamento e controle do projeto envolvem diversas tarefas que podem ser realizadas com auxílio de mecanismos automatizados. Essas ferramentas facilitam na compilação, execução de testes, geração de artefatos, controle de versão, entre outras atividades.

Em termos de Java, a persistência de dados, significa que se pretende manter o estado dos objetos além do escopo JVM (*Java Virtual Machine*), para que o mesmo esteja disponível mais tarde.

4.4.2.1 Maven

Maven é uma ferramenta utilizada para construção e gerenciamento de projetos baseados em Java. Seu objetivo é facilitar a compreensão do projeto pelos desenvolvedores, no menor período de tempo possível (MAVEN, 2015). Para isso, concentra sua preocupação nos seguintes aspectos:

- a. Tornar o processo de compilação fácil: Maven não fornece muita blindagem dos detalhes;
- b. Fornecer um sistema de construção uniforme: Maven permite a construção de um projeto usando seu *project object model (POM)* e um conjunto de *plugins* que são compartilhados por todos os projetos usando Maven, proporcionando um sistema de construção uniforme. Os desenvolvedores familiarizados com a forma de construção de um projeto Maven sabem automaticamente como todos os projetos Maven são construídos, economizando tempo na navegação entre muitos projetos;
- c. Fornecer informações sobre a qualidade do projeto: Maven fornece muita informação útil do projeto, parte retirada do seu POM e parte gerada a partir de fontes do seu projeto, como por exemplo: Alterar documento de registro criado diretamente do controle de origem, atravessar fontes referenciadas, listas de discussão, lista de dependência e relatórios de teste de unidade, incluindo a segurança;
- d. Fornecer orientações para desenvolvimento de melhores práticas: Maven tem como objetivo reunir princípios atuais para o desenvolvimento de melhores práticas, para facilitar a condução do projeto;

- e. Permitir a migração transparente para novas funcionalidades: fornece meios de atualização fácil, permitindo que os clientes possam usufruir de todas as mudanças feitas na ferramenta.

4.4.2.2 Banco de dados

Um banco de dados pode ser definido como uma coleção de dados relacionados de diferentes tamanhos e complexidade variada. Pode ser gerado e mantido manualmente, ou pode ser automatizado, que por sua vez, é criado e mantido tanto por um grupo de aplicativos, quanto por um sistema gerenciador de banco de dados, também conhecido como *SGBD*. (ELMASRI; NAVATHE, 2005)

Em se tratando de um SGBD, Elmasri e Navathe (2005) descreve como uma coleção de programas que torna possível a criação e manutenção de um banco de dados, ou, ainda, como um sistema de *software* que atende a um determinado propósito, facilitando os processos de construção, definição, manipulação e compartilhamento de bancos de dados. O processo de armazenar os dados em determinada mídia controlada pelo SGBD é denominado construção. A definição implica em especificar os tipos, estruturas e restrições para os dados que serão armazenados em um banco de dados. A manipulação apresenta funções como pesquisas em bancos de dados e atualização. Por fim, o compartilhamento permite o acesso de múltiplos usuários ao banco de dados. Segundo Sharma (2001), em um banco de dados relacional, os dados são organizados através de tabelas, formadas por colunas e linhas. Para especificar as tabelas, informações e os dados em um banco é necessário utilizar comandos SQL.

A SQL, ou, *Structured Query Language*, em português, Linguagem de Consulta Estruturada, é a linguagem padrão e mais conhecida no mundo para interagir com um banco de dados. A tecnologia Java fornece uma maneira padrão para realizar a conexão com um banco de dados e *front ends*, e enviar instruções SQL, através de uma API chamada JDBC. Uma API, acrônimo de *Application Programming Interface*, é uma interface de programa de aplicativos, sendo a JDBC caracterizada por apresentar uma série de interfaces, que possibilitam a conexão e a manipulação de um banco de dados, tornando dessa forma possível inserir, atualizar e acessar dados. (SHARMA, 2001)

O MySQL é o banco de dados de código aberto mais popular no mundo, devido a sua confiabilidade e bom desempenho, sendo o banco de dados mais utilizado em se tratando de aplicações baseadas na *web*. (MYSQL, 2015)

4.4.2.3 Hibernate ORM

O Hibernate ORM é um *framework*, abstração que une códigos entre projetos de *software* a fim de prover uma funcionalidade genérica, que possibilita aos desenvolvedores escrever mais facilmente aplicações cujos dados permanecem por mais tempo que o processo de aplicação. Por ser um *framework* de Objeto/Mapeamento Relacional (ORM), o Hibernate tem como preocupação principal a persistência de dados, uma vez que isso se aplica a bancos de dados relacionais, através de um JDBC. (HIBERNATE, 2015)

O Hibernate tem como uma de suas características principais a persistência idiomática, que permite desenvolver classes persistentes seguindo idiomas orientados a objeto, incluindo: herança, polimorfismo, associação, composição e o *framework* de coleções Java. Em razão de dispensar interfaces ou classes base para classes persistentes, o *framework* possibilita que qualquer classe ou estrutura de dado seja persistente. Outra característica do Hibernate é a alta performance, suportando inicializações lentas, por exemplo. Entre outras características, do Hibernate, está a escalabilidade, servindo a milhões de usuários, e a extensibilidade, sendo altamente configurável. (HIBERNATE, 2015)

4.4.2.4 Whois

O serviço WHOIS, em inglês, *who is* (quem é), provê o registro de domínios a partir da identificação e informações de contato como nome, endereço, *e-mail*, número de telefone a milhões de empresas, organizações, negócios e indivíduos. (WHOIS, 2015)

O WHOIS não é um banco de dados único e centralizado, sendo os dados de registros administrados por entidades independentes conhecidas como *registrars* e *registries*. O protocolo WHOIS é público, ou seja, possibilita que qualquer pessoa tenha acesso aos dados e possa identificar o detentor de um nome registrado ou *registrant* de um domínio. (WHOIS, 2015)

5 PENTEST

O *pentest*, teste de penetração ou teste de invasão simula ataques controlados em busca de vulnerabilidades em redes, em sistemas operacionais e em aplicações. “Os especialistas em segurança da informação utilizam técnicas de testes de invasão para avaliar as defesas de uma empresa.” (WEIDMAN, 2014)

Segundo Domingues (2012), os testes são realizados por especialistas, e devem ser iniciados pelo levantamento de informações do ambiente a ser testado, seguir com o planejamento, preparação e, posteriormente, a execução.

Lanna (2010) afirma que deve ser considerado o seguinte processo:

- a. O objetivo de um Pen Test é comprovar o grau de impacto de um ataque feito a partir de uma vulnerabilidade encontrada no sistema;
- b. O Pen Test é recomendado para sistemas com controles de segurança mais maduros, e não encontra nem explora todas as falhas encontradas;
- c. O alcance de um Pen Test é limitado e dirigido a um sistema alvo, pois trata-se de uma tarefa de pesquisa que pode se aprofundar a partir dos primeiros resultados;
- d. Trata-se de uma tarefa de análise manual realizada por um Consultor Especialista, que pode ser suportada por ferramentas de testes;
- e. Testar vulnerabilidades é o primeiro passo para orientar e priorizar as atividades de pesquisa de um Pen Test;
- f. Não substitui processos preventivos automatizados que avaliam os controles de segurança periodicamente;
- g. O Gerenciamento de Vulnerabilidades provê documentação histórica importante para orientar e otimizar recursos em um Pen Test.

5.1 SQL INJECTION

“*SQL Injection* é um conhecido método utilizado em ataques a banco de dados através de formulários que contenham campos de entradas de dados do tipo texto” (NETPOINT, 2008 apud ALMEIDA *et al.*, 2010).

As falhas de Injeção acontecem quando um comando ou uma consulta é feita por um invasor, enviando dados não confiáveis para o interpretador. Dessa forma, o interpretador pode ser enganado, permitindo o acesso a dados não autorizados. (OWASP, 2013)

Para identificar se uma aplicação está vulnerável à injeção podem ser realizados alguns procedimentos, como verificar o código, se os usos dos interpretadores separam os dados suspeitos do comando ou consulta. Para isso,

podem ser utilizadas ferramentas de análise de código, e os testes de invasão podem validar os resultados através de *exploits* que confirmam a vulnerabilidade. (OWASP, 2013)

Segundo Almeida *et al.* (2010), o ataque possui uma defesa simples, mas devido a falta de informação por parte dos desenvolvedores é uma das práticas mais comuns na *internet*. Um exemplo pode ser dado pela seguinte *string* de conexão via SQL entre a aplicação e a base de dados: `SELECT id, password FROM user WHERE id = 'UsuarioExemplo' and password = '123456';`.

No exemplo, a *string* retorna se o usuário UsuarioExemplo possui a senha '123456'. Em caso positivo, localiza e retorna os dados referentes a este usuário. No entanto, a entrada de texto não possui tratamento que impeça a inserção de caracteres especiais, possibilitando que o usuário insira um *id* de 'Usuario'Exemplo'. Com esta entrada, o comando requisitado retorna um erro, pois não pode ser interpretado.

A aplicação, ao permitir a entrada texto com o caractere 'aspas', se torna vulnerável ao ataque às informações da base de dados, exemplificada na seguinte *string* de conexão alterada pelo atacante: `SELECT id, password FROM user WHERE id = ' ' or 1 = 1 and password = '123456';`.

O OWASP (2013), afirma que os “ataques mais perigosos poderiam modificar dados ou até mesmo chamar procedimentos armazenados”.

6 O SISTEMA

O sistema desenvolvido durante este projeto consiste em uma aplicação *web* que identifica a possibilidade de existência de risco do ataque *SQL Injection* em *sites* e sistemas *web*. A aplicação atua, primeiramente, verificando se é possível estabelecer uma comunicação via HTTP. Em seguida, recupera as informações do respectivo servidor autoritativo, servidor responsável por deter os direitos e informações de determinadas faixas de domínio, mapeia os *links* de acesso e testa as vulnerabilidades identificadas na aplicação.

A aplicação é denominada “ivi”, nome inspirado nas iniciais das palavras identificação, vulnerabilidade e informação. O logotipo da aplicação, símbolo de identificação de uma marca, consiste na grafia “ivi”, junto da ilustração de uma coruja, animal conhecido por enxergar no escuro, elemento que faz analogia com a aplicação, uma vez que esta busca por vulnerabilidades que não estão visíveis em *sites* e sistemas *web*.

A aplicação é estruturada na plataforma *web*, sendo suas páginas fornecidas para usuários através do acesso em *PCs*, *tablets* e demais dispositivos com conexão com a *internet*, capazes de visualizar o conteúdo da aplicação em um navegador *web*.

6.1 PROCESSO DE DESENVOLVIMENTO

Com base no modelo espiral de desenvolvimento de *software*, seu o ciclo de vida é composto de um conjunto de 5 iterações, conforme ilustram as tabelas a seguir.

Tabela 2 - Primeira iteração do processo de desenvolvimento

Iteração 1	
Objetivo	Desenvolver uma arquitetura de projeto.
Restrições	Uso mínimo de dependências.
Alternativas	Minimizar o uso de <i>frameworks</i> .
	Manter consultas em cache.
Riscos	Nenhum
Contra Medidas	Nenhuma
Resultados	Aplicação de contra medidas não foi necessária

Planos	Próxima iteração é estabelecer conexão <i>http</i> para análise dos dados.
Compromisso	4 semanas

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 3 - Segunda iteração do processo de desenvolvimento

Iteração 2	
Objetivo	Manter uma conexão <i>http</i> e implementar o mapeamento dos <i>links</i> .
Restrições	Abrir uma conexão única pode afetar a performance da aplicação.
Alternativas	Tratar pequenas requisições, para análise individual.
Riscos	Exceder a memória do servidor e queda de conexão.
Contra Medidas	Filtros das pesquisas e resultados restringidos a uma amostragem aleatória.
Resultados	Diminuir o número de requisições e estabelecer conexões individuais contribuirá para uma melhor performance.
Planos	Associar domínio a proprietário e responsável.
Compromisso	4 semanas

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 4 - Terceira iteração do processo de desenvolvimento

Iteração 3	
Objetivo	Coletar e tratar informações provenientes do <i>WHOIS</i> .
Restrições	Tratar diferentes <i>templates</i> (modelos que tratam os campos definidos pelos servidores autoritativos) de retorno.
Alternativas	Criar um cadastro de configurações de

	<i>templates.</i>
Riscos	<i>Whois server</i> é descentralizado, não existe uma regra para tratar domínio em outros países.
Contra Medidas	Usar o próprio <i>framework</i> da <i>Whois</i> .
Resultados	Atendeu as expectativas.
Planos	Testar a amostragem para vulnerabilidade de <i>sql injection</i> .
Compromisso	2 semanas

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 5 - Quarta iteração do processo de desenvolvimento

Iteração 4	
Objetivo	Testar a injeção de código malicioso em uma amostragem de <i>links</i> do domínio testado e analisar o retorno buscando indícios de vulnerabilidade.
Restrições	Comportamento diferenciado para cada erro em banco de dados distintos.
Alternativas	Criar um <i>template</i> de erros genéricos que se enquadram em todos os casos.
Riscos	Teste em duas fases para viés confirmativo pode comprometer a performance da aplicação e esgotar recursos do servidor.
Contra Medidas	Restringir o teste a uma fase.
Resultados	Restrição da fase de teste para 1 contribuiu para que a análise consuma apenas um tempo razoável, sem esgotar recursos do servidor.
Planos	<i>Layout</i> e <i>Design</i> do <i>front end</i>
Compromisso	4 semanas

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 6 - Quinta iteração do processo de desenvolvimento

Iteração 5	
Objetivo	Desenhar a disposição dos elementos e implementar um <i>layout</i> parcialmente responsivo, que se adeque com a usabilidade e facilidade ao acessar os recursos disponíveis pela aplicação.
Restrições	Pouco conteúdo para o formato padrão de telas para cada funcionalidade.
Alternativas	Implementar um <i>layout</i> de uma tela com todas as funcionalidades a disposição.
Riscos	Mudar o formato de exibição atual das telas, para se adequar a requisições <i>ajax</i> .
Contra Medidas	Telas com saída no formato de <i>Json</i> para interpretação dos dados por requisições via <i>ajax</i> .
Resultados	A adaptação do retorno das informações para <i>Json</i> possibilitou que as requisições fossem exibidas por uma tela única.
Planos	Última iteração.
Compromisso	0 semanas

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2 MODELAGEM DO SISTEMA

A UML, abreviatura de *Unified Modeling Language*, é a linguagem mais utilizada em se tratando de projeto de *software* no contexto de orientação a objeto, servindo de instrumento para a modelagem visual de um *software*. (YOSHIMA, 2005)

A etapa inicial na modelagem de um sistema é a definição do escopo, e segundo Oliveira (2008), nela são identificados os conjuntos de Casos de Uso, bem como definido a função do sistema. O Diagrama de Casos de Uso (*UD*) é utilizado

durante o levantamento e análise de requisitos do sistema, composto, segundo Yoshima (2005) de atores (usuários) e casos de uso (ações).

Entre as outras etapas que constituem uma modelagem de sistema está a delimitação de Regras de Negócio, segundo Bell (1990, apud MOURA; TONIETO 2003), etapa em que são definidas as diretrizes e restrições relacionadas a estados e processos organizacionais.

Através de diagramas a UML permite simplificar a comunicação de ideias relacionadas ao projeto, facilitando a visualização e promovendo a identificação de soluções em equipe. (YOSHIMA, 2005)

6.2.1 Diagrama de Casos de Uso

O Diagrama de Casos de Uso (UD) é o diagrama mais geral da UML, e normalmente nota-se a sua utilização nas etapas de levantamento e análise de requisitos do sistema. (OLIVEIRA, 2008)

O Diagrama é composto de um Ator, representado por um boneco, e um Caso de Uso, representado usualmente por uma elipse. O Caso de Uso é conectado ao Ator por uma linha, e representa o comportamento do sistema do ponto de vista do Ator, responsável pela utilização do *software* ou serviço. Sendo assim, o nome do Caso de Uso define o objetivo do Ator, ou seja, o que ele deve fazer no sistema. (YOSHIMA, 2005)

6.2.1.1 Escopo

Este projeto tem por objetivo criar uma aplicação que forneça uma análise básica de vulnerabilidades que podem comprometer o sistema ou dados dos *sites* testados. A fim de informar empresas de pequeno porte sobre a insegurança de suas aplicações.

Segue uma breve descrição do escopo do projeto:

- a. Análise de um domínio específico;
- b. Consulta de domínios previamente analisados;
- c. Autenticação do usuário administrativo;

- d. Inclusão de um *template* teste (diferenciado por extensão de domínio, identifica o proprietário, responsável, e-mail de contato, última atualização do registro e país de origem.);
- e. Edição de um *template* de teste;
- f. Exclusão de um *template* de teste.

6.2.1.2 Regras de Negócio

A seguir, são exibidas as regras para utilização dos casos de uso.

Tabela 7 - Relação das Regras de Negócio (RN)

RN-01	É obrigatório o preenchimento do campo do domínio
RN-02	Ao realizar a análise o sistema armazena o domínio, proprietário, responsável, e-mail de contato e última atualização.
RN-03	Apenas usuários autorizados podem acessar a área de administração do sistema.

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.3 Requisitos

Nesta etapa ocorre o levantamento de todos os requisitos necessários para a elaboração de Casos de Uso, sendo a documentação responsável por sustentar a construção do *software*. (MEDEIROS, 2004)

6.2.1.3.1 Requisitos Funcionais (RF)

Os requisitos funcionais descrevem as funções que o sistema deve executar.

Tabela 8 - Relação dos Requisitos Funcionais (RF)

RF-001	<u>Analisar domínio</u> O sistema deve permitir que o usuário insira um domínio para análise. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RF-002	<u>Consultar domínio</u> O sistema deve permitir que o usuário acesse as informações dos domínios já analisados. Prioridade: <input type="checkbox"/> Essencial <input checked="" type="checkbox"/> Importante <input type="checkbox"/> Desejável
RF-003	<u>Autenticar usuário</u> Este requisito faz a autenticação do administrador através de seu login e senha e, em seguida, exibe as opções, com as funcionalidades permitidas a ele. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RF-004	<u>Criar template</u> O sistema deve permitir que o administrador inclua um novo template no sistema Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RF-005	<u>Editar template</u> O sistema deve permitir que o administrador edite um template existente

	Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RF-006	<u>Excluir template</u> O sistema deve permitir que o administrador exclua um template Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.3.2 Requisitos Não Funcionais (RNF)

Os requisitos não funcionais englobam os aspectos gerais do sistema, as condições que ele deve atender.

Tabela 9 - Relação dos Requisitos não Funcionais (RNF)

RNF-001	<u>Portabilidade</u> O sistema deve ser compatível com os navegadores todos os navegadores. Prioridade: <input type="checkbox"/> Essencial <input checked="" type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-002	<u>Desempenho</u> O sistema deve garantir que o tempo de retorno não seja maior que 20 segundos. Prioridade: <input type="checkbox"/> Essencial <input checked="" type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-003	<u>Segurança</u> O sistema deve dispor de mecanismos de segurança para a autenticação do administrador e controle de acesso a conteúdos e funcionalidades do sistema. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-004	<u>Usabilidade</u> O sistema deve prover o usuário com interface simples e intuitiva, de fácil navegação para facilitar o uso do mesmo por parte dos usuários. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-005	<u>Apresentação de interface gráfica</u> O sistema deve fazer uso, exclusivamente, da língua Portuguesa para todo e qualquer texto apresentado no portal de conteúdos. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-006	<u>Arquitetura de software</u> A implementação do sistema deve empregar uma arquitetura de 3 (três) camadas: apresentação, negócio e dados. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-007	<u>Linguagem de programação adotada</u> A implementação do sistema deve utilizar a linguagem Java com JDK. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-008	<u>Disponibilidade</u> O sistema deverá estar disponível aos usuários 24 horas por dia e 7 dias por semana. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável
RNF-009	<u>Banco de dados</u> A implementação do sistema deve empregar o MySql como servidor de banco de dados. Prioridade: <input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.3.3 Requisitos de Interface com o Usuário (RIU)

Estes requisitos determinam as características que devem ser implementadas na interface com o usuário.

Tabela 10 - Relação dos Requisitos de Interface com o Usuário (RIU)

RIU-001	O sistema deverá apresentar uma tela com informações sobre a aplicação
RIU-002	O sistema deverá apresentar uma caixa de texto para digitação do domínio a ser analisado.
RIU-003	O sistema deverá apresentar uma tela com uma lista de domínios analisados.
RIU-004	O sistema deverá apresentar uma tela de login para administrador.
RIU-005	O sistema deve apresentar uma tela de templates com opção de criar, editar e excluir

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.3.4 Mensagens do Sistema

Na tabela a seguir são definidas todas as mensagens que o sistema deve emitir para o usuário.

Tabela 11 - Relação das Mensagens do Sistema (ME)

ME-001	O domínio está inacessível no momento!
ME-002	Domínio fora do escopo de teste! Consulte no rodapé.
ME-003	Não há páginas suficientes para teste!
ME-004	Não é possível coletar informações sobre este domínio!
ME-005	Domínio inválido!

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.4 Modelo de Casos de Uso (USC)

Um modelo de caso de uso é um modelo que descreve como diferentes tipos de usuários interagem com o sistema para resolver um problema. Como tal, ele descreve as metas dos usuários, as interações entre os usuários e o sistema, bem como o comportamento necessário do sistema para satisfazer estas metas.

6.2.1.4.1 Relação de Atores

Um ator é quem executa um caso de uso. Na tabela 11, estão listados os atores do sistema e os processos em que eles estão envolvidos.

Tabela 12 - Relação de Atores do Sistema

ATOR	PROCESSO
Usuário	Analisar domínio
	Consultar domínio
Administrador	Autenticar Usuário

	Criar template
	Editar template
	Remover template

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.4.2 Lista de Casos de uso

Segue a tabela com a lista dos casos de uso do sistema.

Tabela 13 - Relação dos Casos de Uso

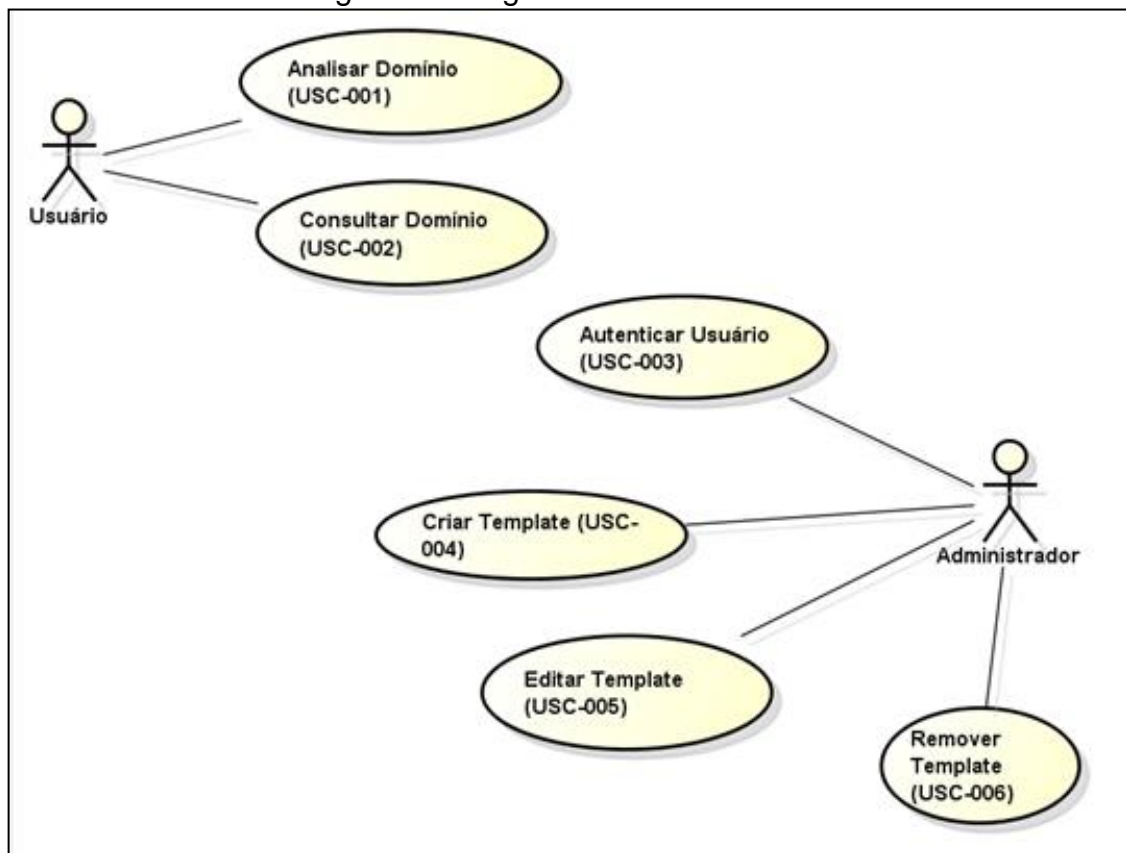
USC-001	Analisar domínio
USC-002	Consultar domínio
USC-003	Autenticar Usuário
USC-004	Criar template
USC-005	Editar template
USC-006	Remover template

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.4.3 Diagrama de Caso de Uso

O diagrama de caso de uso descreve as funcionalidades do sistema, na visão do usuário, conforme ilustra a figura 4.

Figura 4 - Diagrama de Caso de Uso



Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.1.4.4 Detalhamento de Caso de Uso

Os casos de uso são detalhados nas tabelas a seguir, para cada um deles é escrito um cenário, com base nos requisitos do sistema.

Tabela 14 - Caso de Uso Analisar Domínio

1. Caso de Uso	USC-001: Analisar Domínio
1.1 Objetivo	O objetivo deste caso de uso é analisar o domínio inserido pelo ator (usuário)
1.2 Pré-Condições	Ator informa o domínio que deve ser analisado
1.3 Condição final de sucesso	Ator visualiza as informações do domínio e resultado da análise
1.4 Condição final de Falha	Ator não consegue acessar as informações do domínio
1.5 Ator primário	Usuário
1.6 Ator secundário	Não tem
1.7 Requisito Funcional	RF-001
1.8 Requisito de Interface do usuário	RIU-001 e RIU-002
1.9 Evento	O ator seleciona Analisar domínio
1.10 Fluxo Principal	1. Sistema apresenta uma tela com o campo contendo o domínio e a opção Analisar 2. Ator(usuário) informa o domínio 3) Ator(usuário) seleciona a opção Analisar 4) Sistema verifica se o domínio é válido. [E1.1] [E1.2] [E1.3] [E1.4] [E1.5] 5) Sistema apresenta a tela com as informações do domínio e da análise 6) Caso de uso é encerrado.
1.11 Fluxos Alternativos	N/A
1.12 Fluxos de Exceção	[E1.1] Domínio inacessível. 1. Sistema apresenta a mensagem ME-001 (O domínio está inacessível no momento!) 2. Sistema apresenta a opção OK. 3. Ator seleciona a opção OK. 4. Caso de uso retorna para o passo 2 do fluxo principal. [E1.2] Domínio fora do escopo 1. Sistema apresenta a mensagem ME-002 (Domínio fora do

	<p>escopo de teste! Consulte no rodapé.)</p> <p>2. Sistema apresenta a opção OK.</p> <p>3. Ator seleciona a opção OK.</p> <p>4. Caso de uso retorna para o passo 2 do fluxo principal.</p> <p>[E1.3] Páginas insuficientes para teste</p> <p>1. Sistema apresenta a mensagem ME-003 (Não há páginas suficientes para teste!)</p> <p>2. Sistema apresenta a opção OK.</p> <p>3. Ator seleciona a opção OK.</p> <p>4. Caso de uso retorna para o passo 2 do fluxo principal.</p> <p>[E1.4] Falta de informações sobre o domínio</p> <p>1. Sistema apresenta a mensagem ME-004 (Não é possível coletar informações sobre este domínio!)</p> <p>2. Sistema apresenta a opção OK.</p> <p>3. Ator seleciona a opção OK.</p> <p>4. Caso de uso retorna para o passo 2 do fluxo principal.</p> <p>[E1.5] Domínio inválido</p> <p>1. Sistema apresenta a mensagem ME-005 (Domínio inválido!)</p> <p>2. Sistema apresenta a opção OK.</p> <p>3. Ator seleciona a opção OK.</p> <p>4. Caso de uso retorna para o passo 2 do fluxo principal.</p>
1.13 Pontos de Inclusão (Include)	N/A
1.14 Pontos de Extensão (Extended)	N/A
1.15 Prioridade	Alta
1.16 Complexidade	Média
1.17 Frequência	Este USC é utilizado em média 1 vez por dia. Em caso de nova análise de um mesmo domínio é necessário um intervalo de uma semana entre elas.
1.18 Requisitos Não Funcionais	RNF-001, RNF-002, RNF-004, RNF-005 e RNF-008
1.19 Regras de Negócio (RN)	RN-01 e RN-02
1.20 Informações complementares	N/A
1.21 Pendências	N/A

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 15 - Caso de Uso Consultar Domínio

2. Caso de Uso	USC-002: Consultar Domínio
2.1 Objetivo	O objetivo deste caso de uso é consultar as informações de um domínio já analisado.
2.2 Pré-Condições	Ator seleciona em uma lista o domínio desejado.
2.3 Condição final de sucesso	Ator visualiza as informações do domínio e resultado da análise.
2.4 Condição final de Falha	Ator não consegue acessar as informações do domínio
2.5 Ator primário	Usuário
2.6 Ator secundário	Não tem
2.7 Requisito Funcional	RF-002
2.8 Requisito de Interface do usuário	RIU-001 e RIU-003
2.9 Evento	O ator seleciona uma opção de domínio disponível.
2.10 Fluxo Principal	1. Sistema apresenta uma tela com uma lista contendo domínios analisados. 2. Ator(usuário) seleciona o domínio. 5) Sistema apresenta a tela com as informações do domínio e da análise. 6) Caso de uso é encerrado.
2.11 Fluxos Alternativos	N/A
2.12 Fluxos de Exceção	N/A
2.13 Pontos de Inclusão (Include)	N/A
2.14 Pontos de Extensão (Extended)	N/A
2.15 Prioridade	Média
2.16 Complexidade	Baixa
2.17 Frequência	Este USC é utilizado em média 1 vez por dia
2.18 Requisitos	RNF-001, RNF-002, RNF-004, RNF-005 e RNF-008

Não Funcionais	
2.19 Regras de Negócio (RN)	RN-02
2.20 Informações complementares	N/A
2.21 Pendências	N/A

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 16 - Caso de Uso Autenticar Usuário

3. Caso de Uso	USC-003: Autenticar Usuário
3.1 Objetivo	O objetivo deste caso de uso é efetuar o login do ator (administrador) no sistema, exibindo as telas permitidas.
3.2 Pré-Condições	O ator deve ser o administrador do sistema.
3.3 Condição final de sucesso	Ação da área de administrador do sistema.
3.4 Condição final de Falha	Sistema bloqueado.
3.5 Ator primário	Administrador
3.6 Ator secundário	Não tem
3.7 Requisito Funcional	RF-003
3.8 Requisito de Interface do usuário	RIU-004
3.9 Evento	O ator acessa a área de administrador
3.10 Fluxo Principal	<ol style="list-style-type: none"> 1. Sistema apresenta uma janela com os campos contendo as informações: <ol style="list-style-type: none"> 1.1. Login 1.2. Senha 1.3. As opções: <ol style="list-style-type: none"> 1.3.1. Entrar 2. Ator(administrador) informa o Login e a senha 3) Ator(administrador) seleciona a opção Entrar [A3.1] 4) Sistema verifica se usuário e senha são válidos. [E3.1] 5) Sistema autentica o administrador e apresenta a tela com as opções que ele pode acessar. 6) Caso de uso é encerrado.
3.11 Fluxos	<p>[A3.1] Ator seleciona a opção Sair.</p> <ol style="list-style-type: none"> 1. Caso de Uso é encerrado.

Alternativos	
3.12 Fluxos de Exceção	[E3.1] Senha inválida 1. Caso de uso retorna para o passo 2 do fluxo principal.
3.13 Pontos de Inclusão (Include)	N/A
3.14 Pontos de Extensão (Extended)	N/A
3.15 Prioridade	Alta
3.16 Complexidade	Baixa
3.17 Frequência	N/A
3.18 Requisitos Não Funcionais	RNF-004, RNF-005 e RNF-008
3.19 Regras de Negócio (RN)	RN-03
3.20 Informações complementares	N/A
3.21 Pendências	N/A

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 17 - Caso de Uso Criar Template

4. Caso de Uso	USC-004: Criar Template
4.1 Objetivo	O objetivo deste caso de uso é criar um template de análise de domínios
4.2 Pré-Condições	Ator(administrador) informa a intenção de criar um template
4.3 Condição final de sucesso	Ator consegue fazer a inclusão de um novo template.
4.4 Condição final de Falha	Ator não consegue fazer a inclusão de um novo template.
4.5 Ator primário	Administrador
4.6 Ator secundário	Não tem
4.7 Requisito Funcional	RF-004
4.8 Requisito de	RIU-005

Interface do usuário	
4.9 Evento	O ator seleciona a opção criar template
4.10 Fluxo Principal	<ol style="list-style-type: none"> 1. O ator deverá fazer a autenticação (Include USC-003 Autenticar Usuário) 2. Sistema apresenta uma tela com a lista dos templates e as opções: <ol style="list-style-type: none"> 2.1 novo 2.2 editar [A4.1] 2.3 excluir [A4.2] 3. Ator(administrador) seleciona a opção novo template. 4. Sistema apresenta a tela com campos de preenchimento habilitados e oferece as opções: <ol style="list-style-type: none"> 4.1 Enviar 4.2 Voltar [A4.3] 5. Ator preenche os campos e seleciona a opção salvar. 6. Sistema salva as informação do novo template. 7. Caso de uso é encerrado.
4.11 Fluxos Alternativos	<p>[A4.1] Ator seleciona a opção Editar</p> <ol style="list-style-type: none"> 1. O sistema direciona para o USC-005 (Editar Template) <p>[A4.2] Ator seleciona a opção Excluir</p> <ol style="list-style-type: none"> 1. O sistema direciona para o USC-006 (Excluir Template) <p>[A4.3] Ator seleciona a opção Voltar</p> <ol style="list-style-type: none"> 1. Sistema retorna ao passo 2 do fluxo principal. <p>Caso de uso é encerrado</p>
4.12 Fluxos de Exceção	N/A
4.13 Pontos de Inclusão (Include)	Fluxo de Evento: Principal (Básico): Include USC-003: Autenticar Usuário Passo: 1
4.14 Pontos de Extensão (Extended)	N/A
4.15 Prioridade	Média
4.16 Complexidade	Baixa
4.17 Frequência	Este USC é utilizado em média 1 vez a cada 5 meses.
4.18 Requisitos Não Funcionais	RNF-001, RNF-002, RNF-004, RNF-005 e RNF-008
4.19 Regras de Negócio (RN)	RN-03
4.20 Informações complementares	N/A

4.21 Pendências	N/A
-----------------	-----

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 18 - Caso de Uso Editar Template

5. Caso de Uso	USC-005: Editar Template
5.1 Objetivo	O objetivo deste caso de uso é editar um template de análise de domínios
5.2 Pré-Condições	Ator(administrador) informa a intenção de editar um template
5.3 Condição final de sucesso	Ator consegue fazer a edição de um template.
5.4 Condição final de Falha	Ator não consegue fazer a edição de um template.
5.5 Ator primário	Administrador
5.6 Ator secundário	Não tem
5.7 Requisito Funcional	RF-005
5.8 Requisito de Interface do usuário	RIU-005
5.9 Evento	O ator seleciona a opção editar template
5.10 Fluxo Principal	<ol style="list-style-type: none"> 1. O ator deverá fazer a autenticação (Include USC-003 Autenticar Usuário) 2. Sistema apresenta uma tela com a lista dos templates e as opções: <ol style="list-style-type: none"> 2.1 novo [A5.1] 2.2 editar 2.3 excluir [A5.2] 3. Ator(administrador) seleciona a opção editar template. 4. Sistema apresenta a tela com campos habilitados e oferece as opções: <ol style="list-style-type: none"> 4.1 Salvar 4.2 Voltar [A5.3] 5. Ator edita os campos e seleciona a opção salvar. 6. Sistema salva as informação do template atualizado. 7. Caso de uso é encerrado.
5.11 Fluxos Alternativos	<ol style="list-style-type: none"> [A5.1] Ator seleciona a opção Novo <ol style="list-style-type: none"> 2. O sistema direciona para o USC-004 (Criar Template) [A5.2] Ator seleciona a opção Excluir <ol style="list-style-type: none"> 2. O sistema direciona para o USC-006 (Excluir Template) [A5.3] Ator seleciona a opção Voltar <ol style="list-style-type: none"> 2. Sistema retorna ao passo 2 do fluxo principal. 3. Caso de uso é encerrado

5.12 Fluxos de Exceção	N/A
5.13 Pontos de Inclusão (Include)	Fluxo de Evento: Principal (Básico): Include USC-003: Autenticar Usuário Passo: 1
5.14 Pontos de Extensão (Extended)	N/A
5.15 Prioridade	Média
5.16 Complexidade	Baixa
5.17 Frequência	Este USC é utilizado em média 1 vez a cada 5 meses.
5.18 Requisitos Não Funcionais	RNF-001, RNF-002, RNF-004, RNF-005 e RNF-008
5.19 Regras de Negócio (RN)	RN-03
5.20 Informações complementares	N/A
5.21 Pendências	N/A

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

Tabela 19 - Caso de Uso Excluir Template

6. Caso de Uso	USC-006: Excluir Template
6.1 Objetivo	O objetivo deste caso de uso é excluir um template de análise de domínios
6.2 Pré-Condições	Ator(administrador) informa a intenção de excluir um template
6.3 Condição final de sucesso	Ator consegue fazer a exclusão de um template.
6.4 Condição final de Falha	Ator não consegue fazer a exclusão de um template.
6.5 Ator primário	Administrador
6.6 Ator secundário	Não tem
6.7 Requisito Funcional	RF-006
6.8 Requisito de Interface do usuário	RIU-005

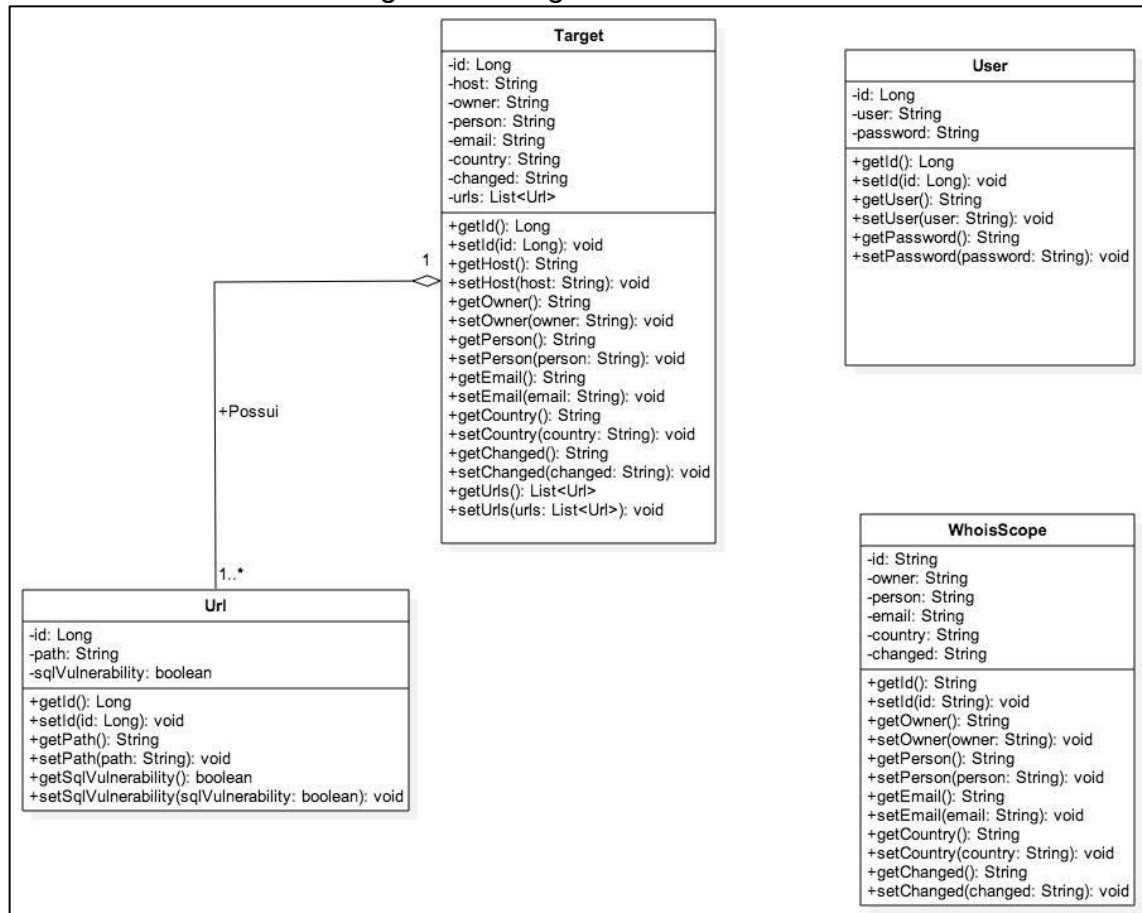
6.9 Evento	O ator seleciona a opção excluir template
6.10 Fluxo Principal	<ol style="list-style-type: none"> 1. O ator deverá fazer a autenticação (Include USC-003 Autenticar Usuário) 2. Sistema apresenta uma tela com a lista dos templates e as opções: <ol style="list-style-type: none"> 2.1 novo [A5.1] 2.2 editar [A5.2] 2.3 excluir 3. Ator(administrador) seleciona a opção excluir template. 4. O sistema apresenta as opções <ol style="list-style-type: none"> 4.1 Sim, deletar 4.2 Voltar [E6.1] 5. ator seleciona a opção "Sim, deletar!" 4. Sistema exclui o template e suas respectivas informações 5. Caso de uso é encerrado.
6.11 Fluxos Alternativos	[A5.1] Ator seleciona a opção Novo <ol style="list-style-type: none"> 1. O sistema direciona para o USC-004 (Criar Template) [A5.2] Ator seleciona a opção Editar <ol style="list-style-type: none"> 1. O sistema direciona para o USC-005 (Editar Template)
6.12 Fluxos de Exceção	[E6.1] Voltar <ol style="list-style-type: none"> 1. Caso de uso volta para o passo 2 do fluxo principal.
6.13 Pontos de Inclusão (Include)	Fluxo de Evento: Principal (Básico): Include USC-003: Autenticar Usuário Passo: 1
6.14 Pontos de Extensão (Extended)	N/A
6.15 Prioridade	Média
6.16 Complexidade	Baixa
6.17 Frequência	Este USC é utilizado em média 1 vez a cada 5 meses.
6.18 Requisitos Não Funcionais	RNF-001, RNF-002, RNF-004, RNF-005 e RNF-008
6.19 Regras de Negócio (RN)	RN-03
6.20 Informações complementares	N/A
6.21 Pendências	N/A

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.2 Diagrama de Classes

A figura 5 faz a representação da estrutura do sistema, com base em suas classes, bem como o relacionamento entre elas.

Figura 5 - Diagrama de Classes

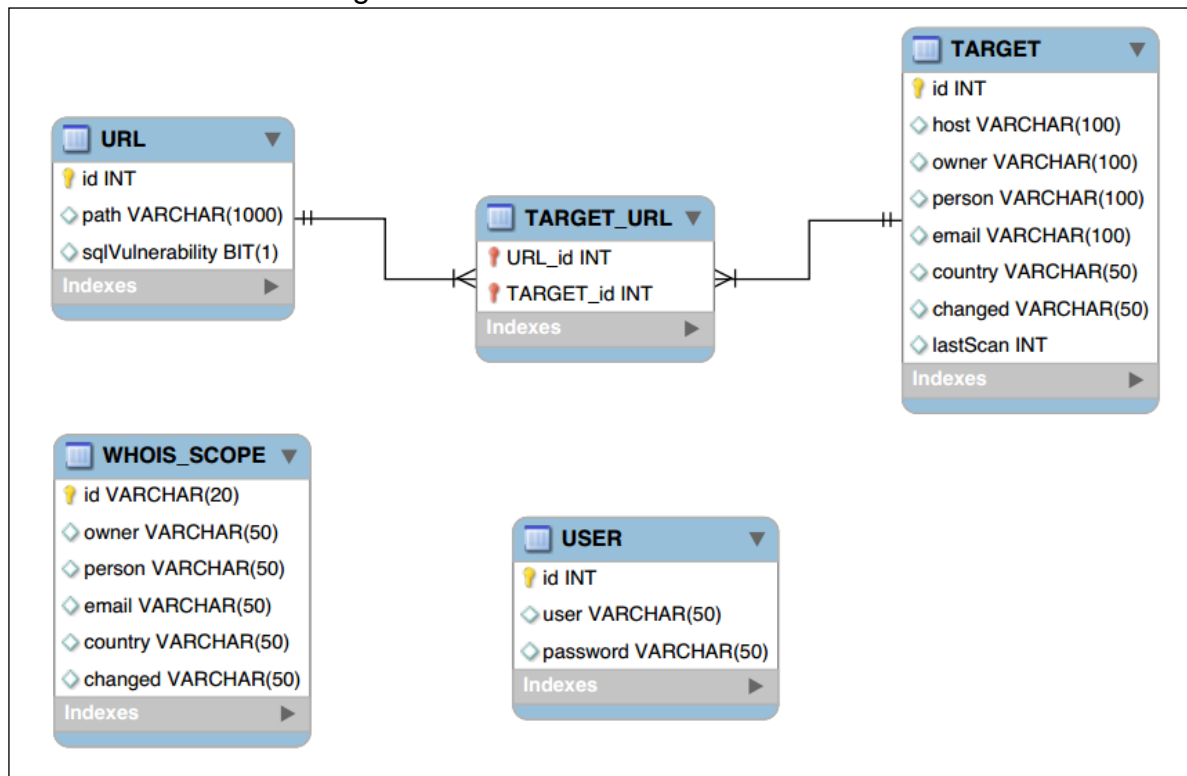


Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.3 Estrutura da Base de Dados

O sistema gerenciador de banco de dados utilizado no sistema foi o software *MySQL*, cuja modelagem dos dados está ilustrada na figura 6.

Figura 6 - Estrutura da Base de Dados



Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

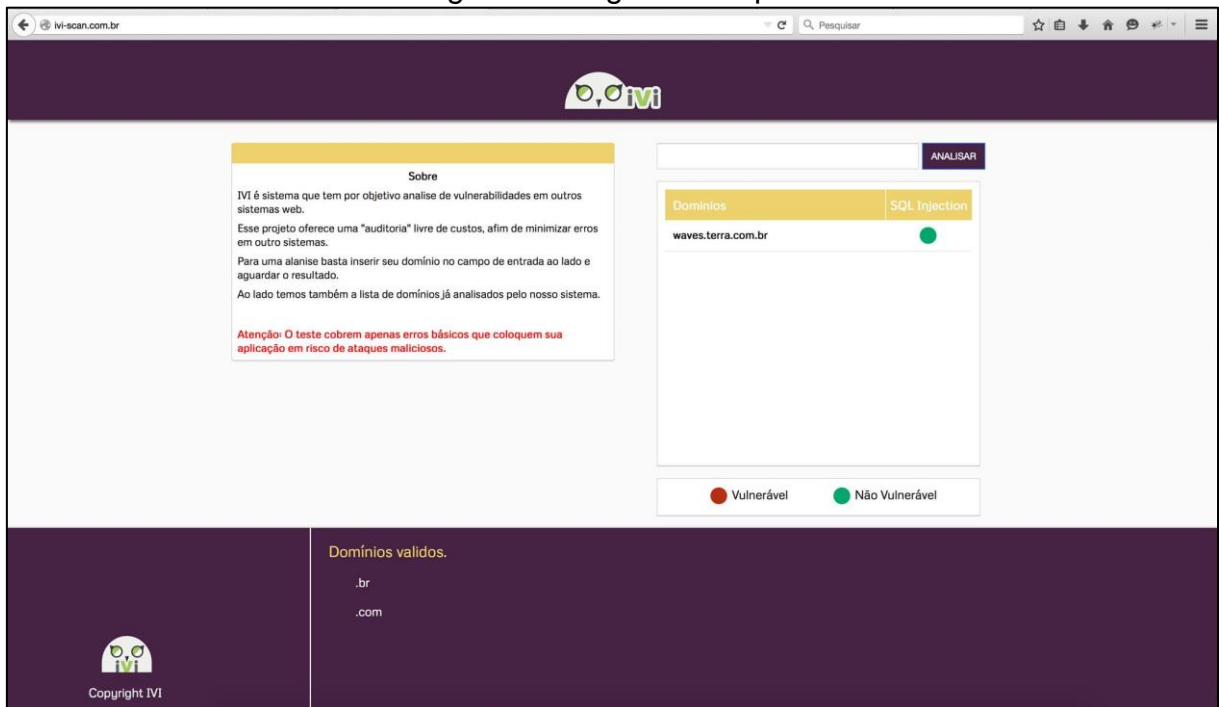
6.2.4 Funcionamento do Sistema

As funcionalidades do sistema serão apresentadas a seguir com ilustrações da captura das telas do próprio sistema.

6.2.4.1 Página principal

A página principal do sistema apresenta informações sobre a aplicação, além do campo de inserção de domínio para análise e uma lista de todos os domínios que já foram analisados por ele.

Figura 7 - Página Principal

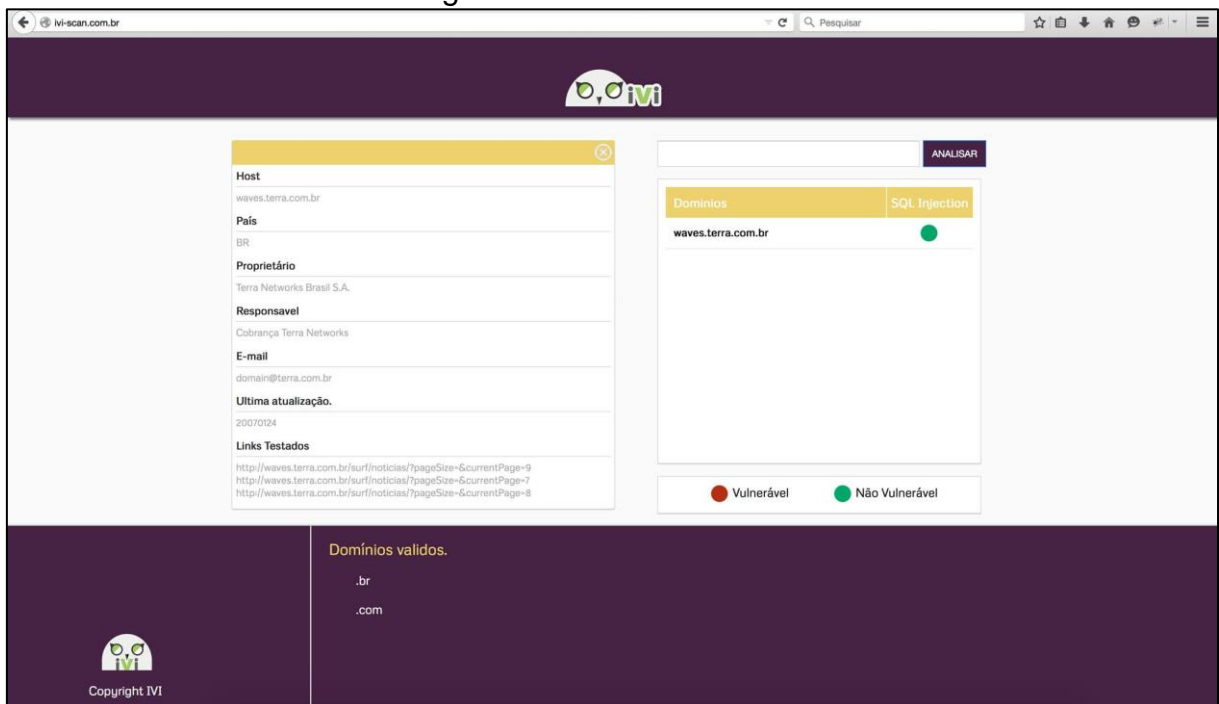


Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.4.2 Página de consulta

A página de consulta apresenta as informações sobre o domínio selecionado na lista da página principal, conforme ilustra a figura a seguir.

Figura 8 - Tela de Consulta

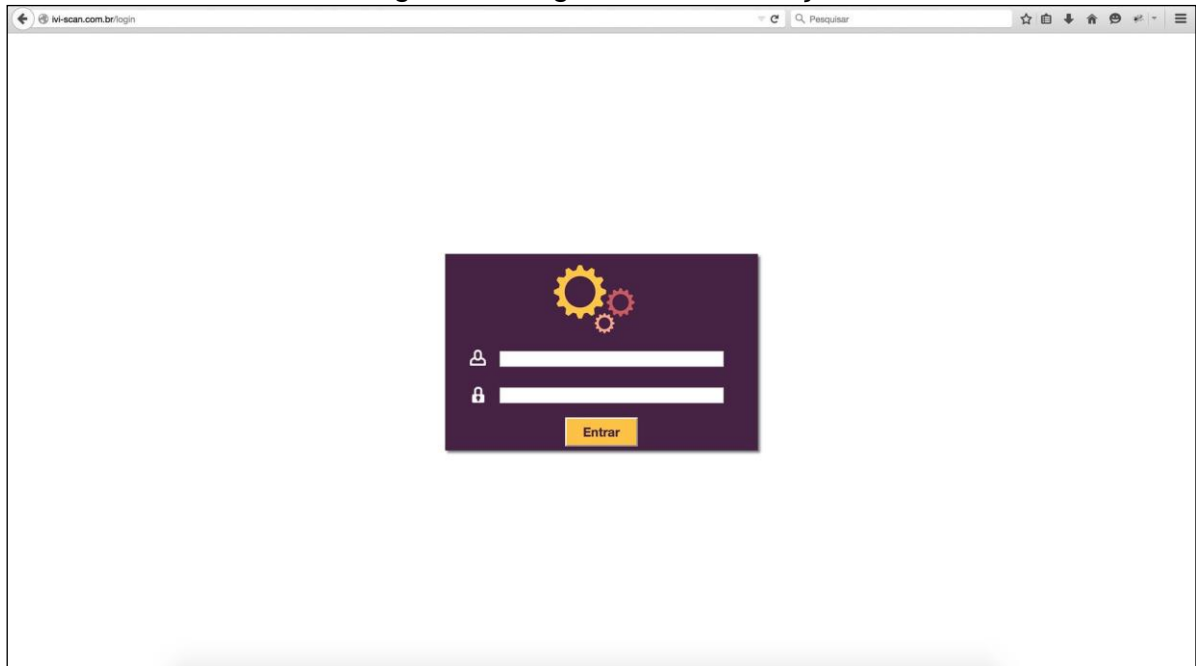


Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.4.3 Página de autenticação do usuário

A página de autenticação permite que o administrador insira os seus dados (nome de usuário e senha), permitindo que, após o envio, a área do administrador seja liberada.

Figura 9 - Página de Autenticação

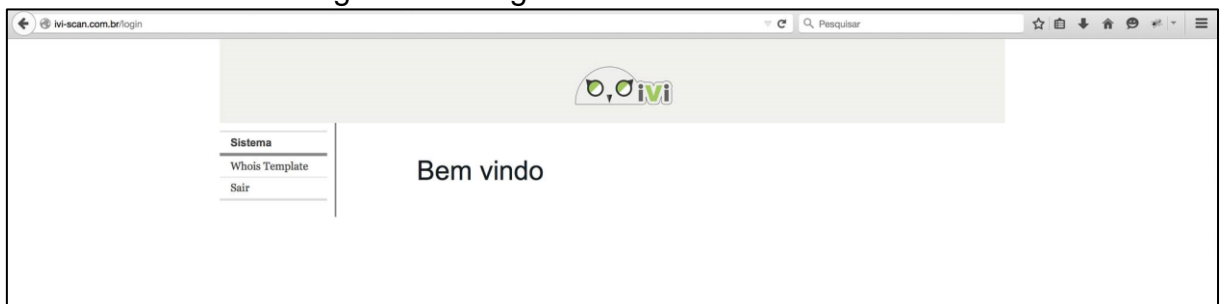


Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.4.4 Página Inicial da Área do Administrador

Após a autenticação do administrador, o sistema apresenta uma página de “Boas Vindas” com um menu na barra lateral, onde se encontram as opções de sair e acessar os *templates*.

Figura 10 - Página Inicial do Administrador

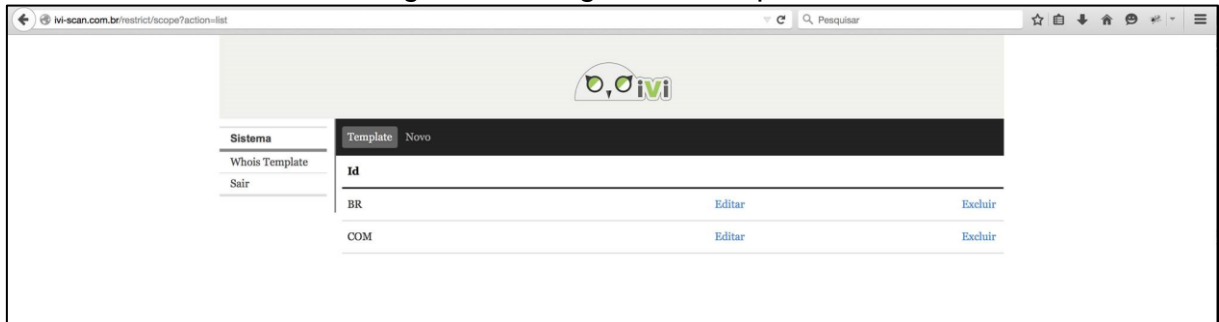


Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.4.5 Página de Templates

Esta página exibe os *templates* cadastrados, apresentando as opções de editá-los ou excluí-los. Além disso, apresenta também a opção de inserir um novo modelo.

Figura 11 - Página de Templates



Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.4.6 Página de criação de novo template

Ao selecionar a opção “novo” na página anterior, o sistema direciona para uma página com os campos de preenchimento de um novo *template*. Esta página permite que o administrador insira os dados que vai caracterizá-lo.

Figura 12 - Página Novo Template

Id:

Owner:

Person:

Country:

Email:

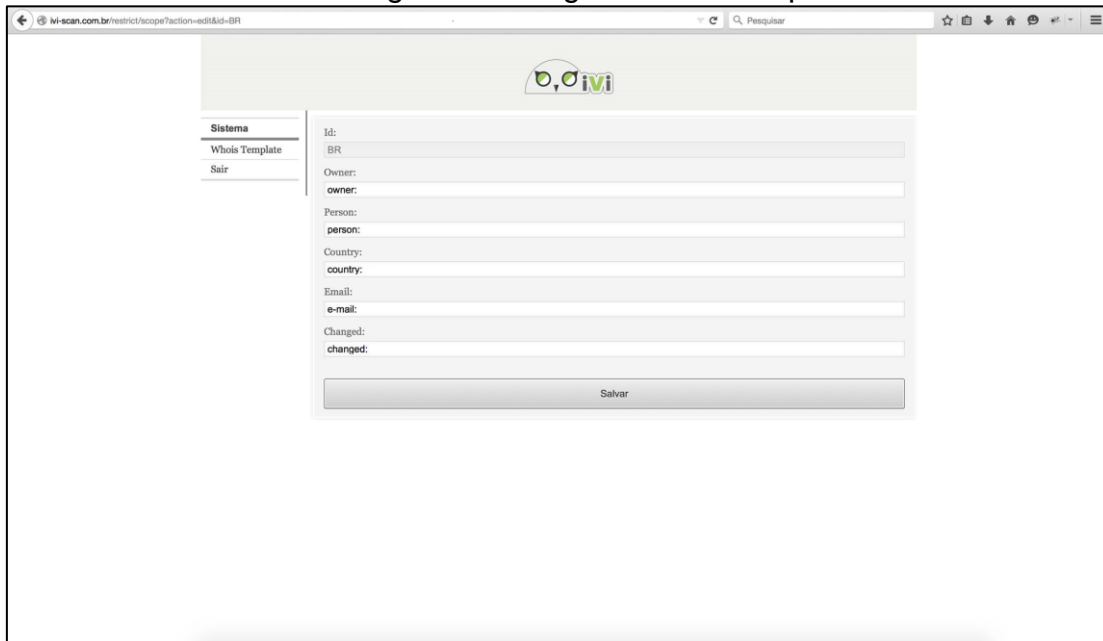
Changed:

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.4.7 Página de edição de template

A página exibe os dados de um *template* cadastrado em modo editável, permitindo a alteração do mesmo.

Figura 13 - Página Editar Template



The screenshot shows a web browser window with the URL `ivi-scan.com.br/restrict/scope?action=edit&id=BR`. The page has a sidebar with a menu containing 'Sistema', 'Whois Template', and 'Sair'. The main content area displays a form for editing a template. The form includes the following fields:

- Id:** A text input field containing 'BR'.
- Owner:** A text input field containing 'owner:'.
- Person:** A text input field containing 'person:'.
- Country:** A text input field containing 'country:'.
- Email:** A text input field containing 'e-mail:'.
- Changed:** A text input field containing 'changed:'.

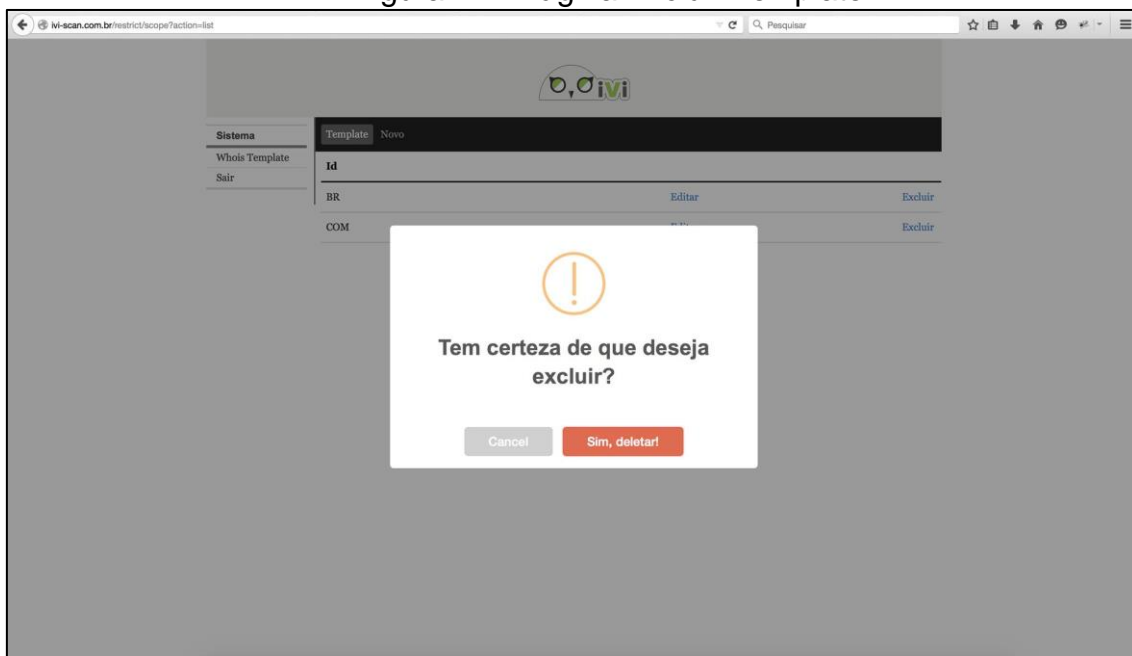
At the bottom of the form is a 'Salvar' button.

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

6.2.4.8 Página exclusão de template

Esta página apresenta ao administrador uma solicitação para confirmar a exclusão de um *template*.

Figura 14 - Página Excluir Template



The screenshot shows a web browser window with the URL `ivi-scan.com.br/restrict/scope?action=list`. The page has a sidebar with a menu containing 'Sistema', 'Whois Template', and 'Sair'. The main content area displays a table of templates. The table has the following columns:

- Id**: A column containing the IDs of the templates.
- Editar**: A column containing links to edit the templates.
- Excluir**: A column containing links to delete the templates.

The table contains two rows of data:

Id	Editar	Excluir
BR	Editar	Excluir
COM		Excluir

A confirmation dialog box is displayed in the foreground, asking 'Tem certeza de que deseja excluir?' (Are you sure you want to delete?). The dialog box has two buttons: 'Cancel' and 'Sim, deletar!' (Yes, delete!).

Fonte: FERREIRA; RODRIGUES FILHO; SANTOS, 2015.

7 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado um sistema de teste de vulnerabilidades de aplicações web. O objetivo era fornecer ao usuário uma análise sobre a existência, ou não, de risco ao ataque *SQL Injection* em sua aplicação.

Considerando os custos que envolvem a prática de medidas para identificação de vulnerabilidades de um sistema, muitas empresas e/ou desenvolvedores podem encontrar dificuldades na adoção das mesmas. Dessa forma, o sistema pode ser útil para os produtores de conteúdo da internet, tendo em vista que os informa a respeito dos riscos de segurança para sua infra e negócio, possibilitando que medidas de segurança necessárias sejam tomadas por eles.

Sendo de caráter gratuito, o serviço é capaz de beneficiar muitos usuários, por oferecer conhecimento acerca de características de sistemas online e alertar não só proprietários, como também usuários em relação ao nível de confiança a ser depositado na página verificada.

Transcorrido todo o desenvolvimento, e com o objetivo do trabalho alcançado, afirma-se que ele contribuiu consideravelmente para a prática do conhecimento adquirido no decorrer do curso, proporcionando a oportunidade de ampliar este conhecimento para áreas correlatas.

Ao considerar que o *SQL Injection* é o risco mais comum em aplicações web, mas não o único, surge, para trabalhos futuros, a possibilidade de extensão do sistema para análises mais aprofundadas, através da inserção de mais dois riscos na análise, o *XSS (Cross-site scripting)* e sequestro de sessão.

REFERÊNCIAS

ALMEIDA, Henrique Cesar de et al. SQL Injection, entenda o que é, aprenda a evitá-lo. 2010. Disponível em: <<http://re.granbery.edu.br/artigos/Mzk2.pdf>>. Acesso em: 02 junho 2015.

ANICETO, Jefferson. Aplicações Web. Apostila ASP.net. Escola Técnica da Univale (ETEIT). 2009.

ASCENCIO, A.F.G, CAMPOS, E.A.V. Fundamentos da programação de computadores. São Paulo: Pearson Prentice Hall, 2007.

BELMIRO, João N. Sistemas de Informação. São Paulo: Pearson Education do Brasil, 2012.

CAPRON, H.L, JOHNSON, J.A. Introdução à informática. São Paulo: Pearson Education do Brasil, 2004.

CHIAVENATO, Idalberto. Iniciação a Sistemas, Organização e Métodos–SO&M. São Paulo: Editora Manole Ltda., 2010.

DANTAS, Marcus Leal. Segurança da Informação: Uma Abordagem Focada em Gestão de Riscos. Olinda: Livro Rápido, 2011.

DEITEL, Paul J. Ajax, Rich Internet Applications e desenvolvimento Web para programadores. São Paulo: Pearson Prentice Hall, 2008.

DOMINGUES, Davi Eduardo R. Testes de Invasão em Ambiente Corporativo. Brasília: 2012. Disponível em: <<http://repositorio.ucb.br/jspui/bitstream/10869/2068/7/Davi%20Eduardo%20Rodrigues%20Domingues.pdf>>. Acesso em: 02 junho 2015.

ELMASRI, Ramez, NAVATHE, Shamkant B. Sistemas de banco de

FONTES, Edison Luiz Gonçalves. Praticando a segurança da informação. Rio de Janeiro: Brasport, 2008.

GRAVES, Mark. Projeto de Banco com XML. São Paulo: Pearson Education do Brasil, 2003.

GUIMARÃES, Thelma de Carvalho. Comunicação e Linguagem. São Paulo: Pearson Education do Brasil, 2012.

JACYNTHO, Mark Douglas de Azevedo. Processo para desenvolvimento de Aplicações Web. Rio de Janeiro, 2008.

JAVA. Disponível em: <http://www.java.com/en/download/faq/whatis_java>. Acesso em: 16 maio 2015.

JORGE, Marcos. Java Passo a Passo Lite. São Paulo: Pearson Education do Brasil, 2004.

KOSUTIC, Dejan. Política de classificação da informação. 2014.

KUROSE, J.F, ROSS, K.W. Rede de Computadores e a Internet: uma abordagem top-down. São Paulo: Pearson Addison Wesley, 2006.

LANNA, Eduardo. Penetration Test, 2010. Disponível em: <<http://www.redesegura.com.br/2010/10/penetration-test/>>. Acesso em: 02 junho 2015.

MCGEE, James V., Gerenciamento Estratégico da Informação. Rio de Janeiro: Editora Elsevier, 1992.

MEDEIROS, Luciano Frontino de. Banco de dados: princípios e prática. Curitiba: InterSaberes, 2013.

MOURA, Hélio A.S, TONIETO, Márcia T., Modelagem e Aplicação de Regras de Negócio para Banco de Dados Relacional. 2003. Disponível em: <<http://www.flf.edu.br/revista-flf.edu/volume03/38.pdf>> Acesso em: 7 de junho de 2015.

MYSQL. Disponível em: <<http://www.mysql.com/about/>>. Acesso em: 16 maio 2015.

OLIVEIRA, Andréa Silva de. Desenvolvimento de um sistema de compras para uma fábrica de alimentos. São Paulo: 2008. Disponível em: <<http://www.feg.unesp.br/ceie/Monografias-Texto/CEIE0801.pdf>>. Acesso em: 8 de junho de 2015.

OLIVEIRA, Djalma de Pinho Rebouças de. Sistemas de Informação gerenciais: estratégias, táticas, operacionais. 8ª ed., São Paulo: Atlas, 1992.

OWASP, The Open Web Application Security Project. OWASP Top 10: Os dez riscos de segurança mais críticos em aplicações web. 2013.

PFLEEGER, Shari Lawrence. Engenharia de Software: Teoria e Prática, 2ª edição. São Paulo: Prentice Hall, 2004.

PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional, 7ª edição. São Paulo: AMGH Editora, 2011.

PUGA, Sandra, RISSETTI, Gerson. Lógica de programação e estrutura de dados, com aplicações em Java. São Paulo: Pearson Prentice Hall, 2009.

ROUSE, Margaret. Web Application (Web app). Disponível em: <<http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>>. Acesso em: 22 abril 2015.

SÊMOLA, Marcos. Gestão da Segurança da Informação: Uma visão executiva. Rio de Janeiro: Editora Campus, 2003.

SHARMA, Vivek, Rajiv. Desenvolvendo sites e-Commerce. São Paulo: MAKRON Books, 2001.

SILVA, Eduardo Leme da. Programação de Computadores. São Paulo: Pearson Education do Brasil, 2015.

SOMMERVILLE, Ian. Engenharia de software. 9ª edição. São Paulo: Pearson Prentice, 2011.

The Apache Software Foundation. Maven: Apache Maven Project. Disponível em: <<https://maven.apache.org/>> . Acesso em: 23 maio 2015.

WEIDMAN, Georgia. Testes de Invasão: Uma introdução prática ao hacking. São Paulo: Novatec, 2014.

WHOIS. Disponível em: <<http://whois.icann.org/en/about-whois#field-section-1>>. Acesso em 22 maio 2015.

YOSHIMA, Rodrigo. Projeto de Software com UML 2.0. Aspercom, 2005.

YOUNG, Paul H. Técnicas de Comunicação Eletrônica. 5ª ed., São Paulo: Pearson Education do Brasil, 2006.

GLOSSÁRIO

AND	E, assim como, também como.
BROWSERS	Navegadores.
CLIPS	Clipes.
DATACENTERS	Ambiente que abriga servidores e componentes de um sistema.
DESIGN	Desenho, Projeto.
E-MAIL	Correio eletrônico.
EXPLOITS	Explorar, executar um código malicioso.
EXTENDED	Estendido.
FRAMEWORK	Estrutura em que algo é construído.
FROM	De.
FRONT-END	Aspecto gráfico, corresponde ao html.
HARDWARE	Conjunto de unidades físicas que compõem um computador ou seus periféricos.
HEAD	Cabeça, referente ao cabeçalho no html.
HOSTS	Hospedeiro, estalajadeiro.
PROTOCOL	Protocolo, registro, minuta.
INTERNET	Inter: internacional. Net: rede. Ou seja, rede mundial de computadores.
LAPTOPS	Computador portátil.
LAYOUT	Esboço do trabalho final.
LINKS	Atalhos.

LOGIN	Log: registro. In: dentro. Neologismo para “Ter acesso”.
ONLINE	Conectado, “na linha”.
PASSWORD	Senha.
REGISTRANT	Registrante.
REGISTRARS	Registradores.
REGISTRIES	Registros.
SCRIPTS	Escritas
SELECT	Selecionar.
SERVER	Servidor.
SHOPPINGS	Centro de compras.
SOFTWARE	Programas que comandam o funcionamento de um computador.
SQL INJECTION	Injeção de Código.
STRINGS	Palavras.
STYLE	Estilo.
SUN MICROSYSTEMS	Fabricante de computador.
TABLET	Computador portátil em forma de prancheta.
TAG	Etiqueta, rótulo.
TEMPLATE	Molde, modelo, padrão.
T-MOBILE	Empresa alemã de telefones móveis.
USER	Usuário.
WEB	Teia. Nome pelo qual a internet se popularizou.

WEB APP	Aplicação da internet.
WEBSITES	Sítios da internet.
WHERE	Onde.
WHO IS	Quem é.
FRONT-END	Parte da apresentação visual de um site.