



Departamento de Engenharia Informática  
Faculdade de Ciências e Tecnologia da  
Universidade de Coimbra

## CodeV - Interacção com Repositórios de Código Dognædis

Orientador DEI: Prof. Marco Paulo Amorim Vieira  
Orientador Dognædis: Eng. Hugo Trovão

Rui Manuel Tomaz Venâncio  
venancio@student.dei.uc.pt

Julho de 2013

## Resumo

O presente documento representa o trabalho realizado por Rui Manuel Tomaz Venâncio, no âmbito da Dissertação de Mestrado em Engenharia Informática, no Departamento de Engenharia Informática, da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. O objetivo do projeto consiste em criar uma plataforma de interação entre o Codev, ferramenta de análise de vulnerabilidades de segurança em código fonte, e diversos repositórios de código externos a este. A dissertação consiste na especificação, planeamento e implementação de um projeto de investigação e desenvolvimento da Dognædis, Lda., designado *Codev - Interação com Repositórios de Código*. Pretende-se nesta dissertação apresentar o estado da arte referente à temática em estudo, arquitetura do sistema que sustenta o projeto e as várias fases de especificação, implementação e validação da solução desenvolvida.



**Agradecimentos:**

Agradeço ao Professor Marco Paulo Amorim Vieira todo o apoio, dedicação e cuidado incansável que me deu enquanto meu orientador.

Agradeço ainda a toda a equipa da Dognaedis, Lda., por toda a ajuda disponibilizada e conhecimentos transmitidos.

Um agradecimento especial, igualmente ao orientador, Engenheiro Hugo Trovão, por todo o apoio, dedicação, paciência e amizade que demonstrou ao longo de todo o percurso desta dissertação.

**Dedicatória:**

Dedico este trabalho à minha família e a toda a equipa da Dognaedis, Lda.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Entidade Acolhedora	1
1.2	Projeto	2
1.3	Objetivos	2
1.4	Estrutura do Documento	3
<b>2</b>	<b>Estado da Arte</b>	<b>4</b>
2.1	Produtos Semelhantes ao CodeV	4
2.1.1	CodeSecure	4
2.1.2	VeraCode	5
2.1.3	RIPS	5
2.1.4	PHP Vulnerability Hunter	5
2.1.5	IBM Security AppScan	5
2.1.6	Comparação de produtos	6
2.2	Frameworks para desenvolvimento baseado em plugins	6
2.2.1	Zend Framework	6
2.2.2	Symfony Framework	7
2.3	Softwares de interação com repositórios	7
2.3.1	Interação via SVN	7
2.3.2	Interação via SCP	7
2.3.3	Interação via CVS	7
2.3.4	Interação via SMB	8
2.4	Cifras	8
2.5	Conclusão	9
<b>3</b>	<b>Método</b>	<b>10</b>
3.1	Processo de desenvolvimento	10
3.2	Controlo de qualidade	11
3.2.1	Cenário de validação	11
3.2.2	Testes de validação	12
3.3	Conclusão	14
<b>4</b>	<b>Planeamento</b>	<b>15</b>
4.1	Metodologia	15
4.2	1º Semestre	15
4.2.1	Diagrama de Gantt	15
4.3	2º Semestre	16
4.3.1	Work breakdown structure	16

4.3.2	Esforço . . . . .	17
4.3.3	Diagrama de Gantt . . . . .	17
4.3.4	Identificação de Riscos . . . . .	18
4.3.5	Alterações ao plano . . . . .	18
4.3.6	Conclusão . . . . .	19
<b>5</b>	<b>Análise de Requisitos</b>	<b>20</b>
5.1	Restrições . . . . .	20
5.2	Requisitos Funcionais . . . . .	20
5.2.1	Casos de Uso . . . . .	21
5.2.2	Outros requisitos . . . . .	33
5.2.3	Sumário de classificação de Requisitos . . . . .	35
5.3	Requisitos Não-Funcionais . . . . .	35
5.3.1	Alterações na classificação de requisitos . . . . .	37
5.4	Conclusão . . . . .	37
<b>6</b>	<b>Arquitetura do Sistema</b>	<b>38</b>
6.1	Visão Geral . . . . .	38
6.2	Estrutura Interna . . . . .	39
6.2.1	Descrição dos módulos: . . . . .	40
6.2.2	Estrutura de Plugins . . . . .	41
6.3	Base de Dados PostgreSQL . . . . .	42
6.4	Especificação das tecnologias usadas . . . . .	43
6.4.1	Sistema Operativo . . . . .	43
6.4.2	PHP . . . . .	43
6.4.3	JavaScript . . . . .	44
6.4.4	jQuery . . . . .	44
6.4.5	Twitter Bootstrap . . . . .	44
6.4.6	PostgreSQL . . . . .	45
6.4.7	Apache HTTP Server . . . . .	45
6.4.8	Apache HTTP Server Benchmarking Tool . . . . .	46
<b>7</b>	<b>Especificação da Implementação</b>	<b>47</b>
7.1	Integração com o CodeV . . . . .	47
7.2	Comunicação entres módulos . . . . .	49
7.2.1	Tipos de comunicação . . . . .	49
7.2.2	CodeV-SMT . . . . .	50
7.2.3	PluginManager . . . . .	50
7.2.4	Schedule Imports . . . . .	52
7.2.5	Crypter . . . . .	52
7.2.6	Database Handler . . . . .	53
7.3	Interface gráfico . . . . .	53
7.4	Processo de importação de código fonte . . . . .	54
7.4.1	Explorador de ficheiros . . . . .	55
7.5	Extensões às Ferramentas e bibliotecas usadas . . . . .	55
7.5.1	PECL SVN Extension . . . . .	56
7.5.2	PECL SSH2 Extension . . . . .	56
7.5.3	smbclient . . . . .	56
7.6	Conclusão . . . . .	56

<b>8</b>	<b>Verificação de Funcionalidades e Atributos de Qualidade</b>	<b>57</b>
8.1	Testes aos Requisitos Funcionais . . . . .	57
8.2	Testes aos Requisitos Não-Funcionais . . . . .	57
8.2.1	Usabilidade . . . . .	57
8.2.2	Robustez . . . . .	58
8.2.3	Segurança . . . . .	58
8.2.4	Modularidade . . . . .	58
8.2.5	Escalabilidade . . . . .	59
8.3	Conclusão . . . . .	60
<b>9</b>	<b>Reflexões Finais</b>	<b>61</b>
9.1	Conclusão da Dissertação . . . . .	61
9.2	Trabalho futuro . . . . .	61
<b>I</b>	<b>Apêndices</b>	<b>65</b>
<b>A</b>	<b>Diagramas da Base de Dados</b>	<b>i</b>
A.1	Base de dados - parte 1 . . . . .	ii
A.2	Base de dados - parte 2 . . . . .	iii
<b>B</b>	<b>Diagramas de Gantt</b>	<b>iv</b>
B.1	Gantt de 1º Semestre . . . . .	v
B.2	Gantt Inicial de 2º Semestre . . . . .	vi
B.3	Gantt Final de 2º Semestre . . . . .	vii
<b>C</b>	<b>Diagrama do WBS</b>	<b>viii</b>
<b>D</b>	<b>Validação de Requisitos Funcionais</b>	<b>x</b>
D.1	Testes aos Requisitos Funcionais . . . . .	xi
<b>E</b>	<b>Questionário ao Utilizador</b>	<b>xxxix</b>
<b>F</b>	<b>Interface gráfica</b>	<b>xli</b>



# Lista de Figuras

3.1	Iteração do processo de desenvolvimento. . . . .	10
5.1	Diagrama de Casos de uso do Administrador. . . . .	21
5.2	Diagrama de Casos de uso do Utilizador. . . . .	21
6.1	Visão geral. . . . .	38
6.2	Diagrama da estrutura interna do SIRC. . . . .	40
6.3	Diagrama da estrutura de plugins do SIRC. . . . .	41
6.4	Utilização de servidores web[40]. . . . .	46
7.1	Exemplo de utilização do PluginManager. . . . .	47
7.2	Interface inicial do SIRC. . . . .	48
7.3	Exemplo de utilização das funcionalidades de Admin do PluginManager. . . . .	48
7.4	Interface do painel de Admin. . . . .	48
7.5	Fluxo de mensagens no SIRC. . . . .	49
7.6	Fluxo de mensagens no <i>PluginManager</i> . . . . .	50
7.7	Diagrama de sequência do processo de importação. . . . .	54
7.8	Explorador de ficheiros. . . . .	55
8.1	Teste de escalabilidade para 70 clientes em simultâneo. . . . .	60
A.1	Base de dados - parte 1. . . . .	ii
A.2	Base de dados - parte 2. . . . .	iii
B.1	Gantt - 1º semestre. . . . .	v
B.2	Gantt Inicial - 2º semestre. . . . .	vi
B.3	Gantt Final - 2º semestre. . . . .	vii
C.1	Diagrama do WBS. . . . .	ix
E.1	Questionário de Usabilidade. . . . .	xl
F.1	Importação via CVS - Login. . . . .	xli
F.2	Importação via CVS - Sistema de credenciais. . . . .	xli
F.3	Importação via CVS - Menu principal. . . . .	xlii
F.4	Importação via CVS - Menu principal com Agendamentos. . . . .	xlii
F.5	Importação via CVS - Tratamento de exceções. . . . .	xliii
F.6	Importação via FileUpload. . . . .	xliii
F.7	Importação via HttpZip. . . . .	xliv

F.8	Importação via SCP - Login. . . . .	xliv
F.9	Importação via SCP - Menu principal. . . . .	xlv
F.10	Importação via SMB - Login. . . . .	xlv
F.11	Importação via SMB - Menu principal. . . . .	xlvi
F.12	Importação via SVN - Login. . . . .	xlvi
F.13	Importação via SVN - Menu principal, pesquisa de revisões. . . . .	xlvii
F.14	Importação via SVN - Menu principal, seleções no explorador. . . . .	xlviii
F.15	Agendamentos ativos. . . . .	xlviii
F.16	Detalhes de agendamentos ativos. . . . .	xlviii
F.17	Importação de novo plugin. . . . .	xlviii
F.18	Importação de novo plugin com erro no nome do ficheiro. . . . .	xlix
F.19	Importação de novo plugin com erro na definição da classe. . . . .	xlix
F.20	Importação de novo plugin sem <i>override</i> a funções de uso obrigatório. . . . .	xlix
F.21	Importação de novo plugin sem <i>override</i> a funções de uso obrigatório. . . . .	l
F.22	Remover plugins. . . . .	l
F.23	Remover plugins - Diálogo de confirmação sem agendamentos dependentes. . . . .	l
F.24	Remover plugins - Diálogo de confirmação com agendamentos dependentes. . . . .	li
F.25	Ativar e desativar plugins. . . . .	li
F.26	Histórico de importações. . . . .	lii
F.27	Ação de exportação em relatório sem vulnerabilidades. . . . .	lii
F.28	Ação de exportação em relatório com vulnerabilidades. . . . .	lii

# Lista de Tabelas

2.1	Comparação Geral de Produtos. . . . .	6
2.2	Comparação de Funcionalidades. . . . .	9
3.1	Modelo para validação dos requisitos funcionais. . . . .	13
4.1	Tabela auxiliar ao Diagrama de Gantt de 1º semestre. . . . .	16
4.2	Esforço das tarefas em horas. . . . .	17
4.3	Tabela auxiliar ao Diagrama de Gantt de 2º semestre. . . . .	18
4.4	Tabela auxiliar ao Diagrama de Gantt Inicial de 2º semestre. . . . .	19
5.1	Tabela de classificação de requisitos . . . . .	35
8.1	Resultados do questionário de usabilidade. . . . .	58
8.2	Testes de escalabilidade. . . . .	59

# Acrónimos

**ACID** Atomicity, Consistency, Isolation, Durability.

**AES** Advanced Encryption Standard.

**AJAX** Asynchronous JavaScript and XML.

**BSD** Berkeley Software Distribution.

**CPU** Central Processing Unit.

**CSV** Comma-separated Values.

**CVS** Concurrent Versioning System.

**GPL** General Public Licence.

**HTML** HyperText Markup Language.

**HTTP** Hypertext Transfer Protocol.

**IPN** Instituto Pedro Nunes.

**ISO** International Organization for Standardization.

**JSON** JavaScript Object Notation.

**NSA** National Security Agency.

**PECL** PHP Extension Community Library.

**PHP** PHP: Hypertext Preprocessor.

**RAM** Random Access Memory.

**RAR** Roshal ARchive.

**SCP** Secure copy protocol.

**SMB** Server Message Block.

**SSH** Secure Shell.

**SVN** Apache Subversion.

**TAR** Tape ARchive.

**URL** Uniform Resource Locator.

**WBS** Work Breakdown Structure.

**XML** eXtensible Markup Language.

# Glossário

**Beta** Versão de um produto (geralmente software) que ainda se encontra em fase de desenvolvimento e testes.

**Biblioteca** Conjunto de recursos utilizados na construção de software.

**Black-box testing** Testes efetuados às funcionalidades da aplicação.

**Bottleneck** Fenómeno em que um elemento limita a capacidade de funcionamento de todo o sistema.

**CodeV** Plataforma para análise automática de vulnerabilidades a código fonte.

**Core** Bloco nuclear de implementação de um sistema.

**Debug** Processo de pesquisa de erros durante a execução de uma determinada aplicação.

**Framework** Conjunto de ferramentas que promove a abstração entre vários projetos de modo a fornecer uma funcionalidade genérica.

**Fuzzing** Técnica de testes a software onde são fornecidos dados de entrada aleatórios, por forma a verificar possíveis erros.

**GET** Tipo de request suportado pelo HTTP.

**Linux** Sistema Operativo baseado em UNIX.

**Open Source** Adjetivo utilizado para caracterizar software que disponibiliza o seu código fonte.

**Plataforma** Conjunto de software e ou hardware que permite que outro software execute.

**Plugin** Ferramenta que estende as funcionalidades de uma determinada aplicação.

**POST** Tipo de request suportado pelo HTTP.

**Repositório** Local onde se encontra código fonte para importar.

**Request** Pedido HTTP.

**Script** Código executado no interior de um programa para estender as suas funcionalidades, ou para o controlar.

**Shell** Software que oferece uma interface para utilizadores de um sistema operativo.

**Tag** Funcionalidade do CVS que permite marcar uma revisão para obtenção posterior do seu respetivo código.

**White-box testing** Testes efetuados ao código e estrutura de uma aplicação.

# Capítulo 1

## Introdução

O presente documento representa a dissertação elaborada no âmbito do estágio de Mestrado de Engenharia Informática do Departamento de Engenharia Informática da Universidade de Coimbra. O estágio consiste no desenvolvimento do projeto *CodeV-Interação com repositórios de código (SIRC)* e tem a duração de dois semestres - no primeiro, foi delineado e debatido o planeamento do projeto, foram identificados os requisitos necessários para a sua implementação e procedeu-se a alguma implementação; no segundo, procedeu-se à continuação do desenvolvimento e documentação adjacente a este. Esta dissertação retrata todo o trabalho realizado ao longo de todo o ano letivo.

### 1.1 Entidade Acolhedora

A Dognædis, Lda. é uma empresa que foi constituída em Junho de 2010, tendo como área de atividade a segurança de informação. Resulta da união de duas equipas: um núcleo de investigadores do CERT-IPN e investigadores da Universidade de Coimbra. O CERT-IPN, criado no seio do Instituto Pedro Nunes, era uma equipa CSIRT (*Computer Security Incident Response Team*), que tinha como missão cultivar e implementar uma atitude de segurança pró-ativa na comunidade envolvente, incentivar o alargamento e a criação de comunidades Portuguesas ligadas à segurança de informação, ser uma entidade de referência internacional, detentora e coletora de informação sobre segurança, e ser referência no mercado nacional de segurança de informação, assumindo como função primordial a execução, coordenação e suporte à resposta a incidentes de segurança existentes no seu âmbito de ação.

Depois de cinco anos de atividade, e devido ao crescente sucesso e ao *feedback* positivo de vários clientes privados e institucionais, a Dognædis iniciou-se como uma empresa privada.

A empresa presta serviços de *Software Assurance*, disponibilizando garantias de segurança e robustez para software, *Business Continuity Support* - serviços de apoio e garantia da continuidade de negócio através de capacidades de deteção, prevenção, mitigação e resolução de eventos de segurança; e *Security Audit and Consultancy* - realização de auditoria e consultoria de segurança.

Ao longo do seu curto tempo de existência, a Dognædis foi distinguida já com diversos prémios, designadamente o “Prémio Coimbra Inovação Parque”, no âmbito do Concurso Arrisca Coimbra 2010, “Melhor Empresa Start-Up Nacional” na Gala do Empreendedorismo em Ciência e Tecnologia promovido pela TecParques e o BES Inovação 2011 na área de Tecnologia[1]. A missão da Dognædis é estar na vanguarda das tecnologias de segurança, trazendo segurança



de informação a organizações e indivíduos, através da excelência em soluções inovadoras.

Sendo o projeto CodeV-SIRC idealizado no seio da Dognædis, seguirá naturalmente a sua filosofia e missão no mundo da informação digital e da segurança da informação.

## 1.2 Projeto

O CodeV é uma plataforma criada pela entidade acolhedora que realiza análises automáticas de código fonte para detetar vulnerabilidades de segurança com foco em aplicações web, tentando aumentar a segurança das mesmas ao máximo. No momento, o CodeV encontra-se em funcionamento, suportando para já a linguagem PHP.

Hoje em dia as empresas utilizam diferentes locais (repositórios) de armazenamento de código fonte durante o desenvolvimento de um determinado software. Tendo isto em conta, e o facto de que o CodeV analisa código fonte com vista a identificação de vulnerabilidades, é importante que este interaja com os diferentes tipos de repositórios existentes na atualidade, de modo a aproximar o mais possível a análise de vulnerabilidades e os ambientes de desenvolvimento.

Atualmente, o CodeV suporta importação de código fonte através de um simples upload de ficheiros e Samba. É neste contexto que surge o presente projeto, pretendendo-se alargar o leque de métodos de importação de código para o CodeV. Para solucionar este problema foi desenvolvido um motor de plugins em que cada plugin irá suportar um tipo diferente de importação de código, como por exemplo: importação via svn, importação via cvs, entre outras retratadas ao longo deste documento.

É importante salientar que este motor de plugins não foi desenvolvido apenas para esta solução de importação de código para o CodeV. É possível utiliza-lo em diferentes contextos, bastando para tal consultar o processo de utilização deste e as regras para o desenvolvimento de novos plugins, encontrando-se todos estes aspetos em mais detalhe no decorrer do presente documento.

## 1.3 Objetivos

O presente projeto (SIRC) consiste num módulo que integra a plataforma CodeV, com o objetivo de a capacitar com diferentes tipos de importação de código. Este módulo foi desenvolvido sob a forma de um motor de plugins, que não só permite solucionar esta temática de importações código, mas também poderá ser utilizado em diferentes contextos, uma vez que o seu desenvolvimento foi pensado para uma utilização genérica e não apenas para este caso em específico. Desta forma permite-se que sejam desenvolvidos outros tipos de plugins, como por exemplo, plugins para exportação de relatórios de vulnerabilidades em diferentes formatos, ou plugins que permitam importar código fonte localizado em sistemas nativos de empresas, dado que nem sempre estas recorrem a repositórios de código *standard* para desenvolvimento dos seus produtos.

Além destes objetivos principais identificam-se outros enumerados abaixo:

- Integração com o CodeV, com lugar a alterações mínimas;
- Módulo SIRC responsável por desenhar toda a interface necessária ao seu funcionamento;
- Desenvolvimento de plugins de importação de código;
- Desenvolvimento de plugins de exportação de relatórios de vulnerabilidades em diferentes formatos;

- Garantir a segurança dos dados processados e guardados pelo SIRC, nomeadamente, encriptação de dados sensíveis, como credenciais de acesso a repositórios.

## 1.4 Estrutura do Documento

Com o intuito de facilitar a leitura e compreensão desta dissertação, optou-se por uma divisão em seis capítulos distintos.

No primeiro capítulo é fornecida uma introdução a todo o projeto, identificando o seu âmbito e descrevendo a entidade acolhedora, fornecendo assim ao leitor uma primeira noção do que se pretende desenvolver e do enquadramento do projeto. No segundo capítulo apresenta-se o estudo de estado da arte nos campos em que o projeto se enquadra, designadamente focando as soluções existentes no mercado e as tecnologias utilizadas para resolver os problemas expostos nesta dissertação. No terceiro capítulo são apresentadas as especificações e a análise de requisitos necessários para o desenvolvimento e implementação do projeto. No quarto capítulo há lugar à especificação do método de desenvolvimento adotado. No quinto capítulo procura-se apresentar o planeamento dos trabalhos realizados ao longo de toda a dissertação. No sexto capítulo, expõe-se a arquitetura do projeto. No sétimo capítulo apresenta-se a especificação da implementação do projeto, onde são abordados diversos aspetos acerca da implementação deste. No oitavo capítulo, expõe-se a verificação de funcionalidades e atributos de qualidade, onde são validados os requisitos funcionais e não-funcionais. No nono e último capítulo, retiram-se e expõem-se as conclusões do projeto, referindo o estado atual e o trabalho futuro.

## Capítulo 2

# Estado da Arte

O presente capítulo descreve os resultados do estudo realizado sobre o estado da arte referente às temáticas, tecnologias e técnicas abordadas, organizando-se desta forma em quatro secções distintas.

Uma primeira secção referente ao levantamento e estudo de produtos semelhantes ao CodeV, com objetivo de perceber a sua integração (ou não) com repositórios de código.

Uma segunda secção em que se procede ao levantamento e igualmente, estudo de frameworks que permitam um desenvolvimento de PHP baseado em sistema de *plugins*.

Uma terceira secção dedicada ao levantamento e estudo de algumas soluções que permitam interação com repositórios de código.

Por último, uma secção onde se procede a um levantamento e estudo sobre mecanismos de cifra de dados.

### 2.1 Produtos Semelhantes ao CodeV

Foi efetuado um levantamento de produtos semelhantes ao CodeV com intuito de estudar a sua integração com repositórios de código. Nas secções abaixo procede-se a uma descrição geral de cada um deles, com especial atenção para aspectos relacionados com a interação com repositórios de código.

Dentro deste aspeto, procurou-se perceber as forças e fraquezas de cada um deles, pelo que os resultados do estudo efetuado são neste presente relatório apresentados sobre a forma de tabelas que ilustram os diferentes tipos de interação destes produtos com repositórios de código, uma classificação de 0 a 5 (*0 - muito mau e 5 - muito bom*) acerca da sua *interface* de utilizador e algumas observações para complementar. É de igual forma efetuada uma breve descrição dos produtos para uma maior contextualização acerca dos mesmos.

#### 2.1.1 CodeSecure

O CodeSecure[2] é um analisador estático de código fonte da empresa Norte Americana Armorize que se dedica exclusivamente à segurança de aplicações web. O CodeSecure tem como objetivo a identificação de vulnerabilidades em aplicações web ao longo do seu desenvolvimento. Fornece uma análise de código automatizado em que mostra os modelos de fluxo de dados vulneráveis dentro da aplicação alvo. Apresenta relatórios onde identifica locais de código vulneráveis oferecendo soluções prioritizadas.

Relativamente acerca da interação deste produto com repositórios de código, este interage com quatro tipos diferentes de repositórios de código, sendo eles: simples upload de zips, subversion, windows share folder e CVS, apresentando um design muito simples e intuitivo.

### 2.1.2 VeraCode

O VeraCode[3] é uma plataforma da empresa Norte Americana Veracode, que se dedica exclusivamente à segurança de aplicações web. Esta é uma plataforma que tanto permite uma análise estática de código como uma análise dinâmica, fornecendo relatórios com as falhas de segurança encontradas e igualmente uma lista priorizada das soluções a implementar.

No que toca a interação com repositórios de código, esta cinge-se unicamente a um simples upload de ficheiros compilados. Contudo, a user interface é robusta e intuitiva.

### 2.1.3 RIPS

O RIPS[4] é um analisador estático de código fonte escrito em PHP e criado por Johannes Dahse. Este analisador destina-se única e exclusivamente à análise de aplicações desenvolvidas em PHP.

O seu funcionamento consiste em partir o código fonte da aplicação alvo em pedaços por forma a analisar função a função e lhe atribuir vulnerabilidades consoante os vários tipos de *input* que esta permita. Esta plataforma, além de um relatório com as vulnerabilidades encontradas, dispõe também de um local onde consultar possíveis soluções para as mesmas.

Em termos de interação com repositórios de código, esta é extremamente limitada, permitindo apenas especificar a localização de uma pasta onde reside a aplicação a analisar ou as aplicações a analisar (caso se marque a opção *subdirs*). Relativamente à interface de utilizador, esta é bastante rudimentar.

### 2.1.4 PHP Vulnerability Hunter

O PHP Vulnerability Hunter[5] é uma plataforma de análise de código PHP criada por John leitch. Esta ferramenta é capaz de efetuar tanto análises estáticas como dinâmicas, mostrando no final de cada execução um relatório com as vulnerabilidades encontradas.

Relativamente a interação com repositórios de código, à semelhança das últimas três plataformas estudadas, esta é fraca. É apenas permitido analisar aplicações PHP que estejam localizadas na *Webroot*. Em termos de user interface, esta é demasiado simples e arcaica.

### 2.1.5 IBM Security AppScan

O IBM Security AppScan[6] é um produto de análise de vulnerabilidades da empresa americana IBM[7]. Este produto da IBM é capaz de efetuar tanto análises estáticas de código como análises dinâmicas. No final de cada análise é mostrada uma lista das vulnerabilidades encontradas, ao explorá-las é possível identificar a respetiva linha ou local de código que as despoletaram, assim como uma listagem de possíveis soluções.

Este produto suporta diversas linguagens de programação, como exemplo, o COBOL, o Java e linguagens .NET (Microsoft Visual C#, VB.NET, ASP.NET). Para além destas características, com este produto é possível efetuar tanto uma análise *white-box* como *black-box*, sendo que em caso de análises *black-box*, é possível verificar a origem das vulnerabilidades ao nível dos pacotes HTTP trocados entre o cliente e o servidor.

É importante salientar que este software não se restringe apenas a análise de projetos Web, possuindo a capacidade de análise de aplicações móveis para *smartphones*.

No que se prende com interação com repositórios de código, este produto apenas permite, no caso de análise *white-box*, upload simples de ficheiros.

### 2.1.6 Comparação de produtos

Com todo o estudo envolvente e inerente à realização do projeto CodeV-Interação com Repositórios de Código, é possível salientar que este constitui uma inovação e uma substancial melhoria relativamente aos concorrentes apresentados, uma vez que a interação destes com repositórios de código é algo fraca.

Na tabela 2.1 que seguidamente se apresenta, procura-se organizar informação pertinente relativa aos concorrentes apresentados.

Produto	Repositórios Suportados	User interface	Observações
<i>CodeSecure</i>	zip upload, svn, samba, cvs	4	User interface simples e intuitivo
<i>VeraCode</i>	zip upload	4	User interface robusto e intuitivo
<i>RIPS</i>	compiled file upload	2	User interface rudimentar
<i>PHP Vulnerability Hunter</i>	pesquisa na webroot	2	User interface demasiado simples
<i>IBM Security AppScan</i>	file upload	4	User interface robusto

Tabela 2.1: Comparação Geral de Produtos.

Analisando a tabela acima e o teor do estudo efetuado acerca destes cinco produtos, o CodeSecure é aquele que se aproxima mais daquilo que se pretende realizar, dado que interage com um bom número de repositórios e o sua user interface é simples e intuitiva, ainda que poderia ser um pouco mais apelativa ao estilo do Veracode.

Contudo, o CodeSecure não suporta importações via HTTP (*download* de ficheiros através de URL) e SCP, funcionalidades estas que estão incluídas no presente projeto.

## 2.2 Frameworks para desenvolvimento baseado em plugins

Procedeu-se a um levantamento de frameworks para PHP que permitissem um desenvolvimento modular de aplicações, mais em concreto, que permitissem o desenvolvimento de plugins pelo utilizador de forma simples e rápida, sendo que o fator comunidade também teve o seu peso neste estudo.

### 2.2.1 Zend Framework

A Zend Framework[8] é uma framework open source para desenvolvimento de aplicações em PHP. Esta framework teve como principal patrocinador a Zend Technologies, mas empresas como a Google, Microsoft e Strikelron[9] também contribuíram para o seu desenvolvimento.

Esta framework entre outras coisas permite um desenvolvimento modular de aplicações, facilitando a criação de plugins pelo utilizador. Um outro fator importante a ter em conta é o facto de possuir uma comunidade grande e ativa, o que a torna numa mais valia para possíveis problemas que se possam encontrar durante o desenvolvimento de uma aplicação.

## 2.2.2 Symfony Framework

A Symfony Framework[10] é uma framework open source para desenvolvimento em PHP. Esta framework foi criada pela empresa Sensio Labs[11], a qual atualmente lhe fornece suporte. Para além do suporte dado pela empresa, a Symfony conta também com uma comunidade bastante ativa.

No contexto do presente projeto, esta framework tem como fator positivo, o facto de possibilitar o desenvolvimento de plugins pelo utilizador e até agilizar este processo. Ainda neste campo, ela detém um repositório suportado pela comunidade, com diversos plugins livres de utilização, podendo qualquer pessoa contribuir para o aumento ou melhoramento destes.

## 2.3 Softwares de interação com repositórios

Esta secção destina-se a apresentar os resultados do levantamento e estudo de diversas tecnologias ou softwares inerentes a interação com os diferentes tipos de repositórios de software. São igualmente discutidos os pontos fortes e fracos de cada software/extensão apresentados.

### 2.3.1 Interação via SVN

- **PECL SVN Extension**

O *PECL SVN Extension* [12] é uma extensão do repositório PECL[13]. Atualmente, esta extensão é suportada por Alan Knowles, Wez Furlong e Scott MacVicar, encontrando-se na sua versão estável 1.0.2. Esta disponibiliza métodos para autenticação que permitem manter uma sessão com determinado repositório e métodos para listagem de ficheiros e pastas em repositórios remotos. Contudo, o método de *export* apenas permite interagir com repositórios locais, sendo este um aspeto negativo na utilização desta extensão no presente projecto. Apesar disto, parece ser uma boa alternativa ao tradicional uso de comandos nativos do sistema operativo através da função *shell.exec*[14] do PHP.

### 2.3.2 Interação via SCP

- **PECL SSH2 Extension**

O *PECL SSH2 Extension*[15] é uma extensão do repositório PECL. Atualmente, esta extensão é suportada por Sara Golemon, Mike Sullivan, Pierre Joye (lead) e Casper Lange-meijer, encontrando-se na sua versão beta 0.12.

Esta extensão vai tendo atualizações para versões mais recentes do PHP, sendo que por norma também tem bom feedback por parte de quem a utiliza ou já a utilizou.

A mais valia desta extensão é que permite manter uma sessão ativa entre a execução de diversas operações e permite fazer *download* de ficheiros do repositório remoto. Contudo, não disponibiliza nenhum método para listagem de ficheiros e pastas do repositório e o *download* de ficheiros não tem opção de *download* recursivo.

### 2.3.3 Interação via CVS

- **phpnetcvs**

O projeto *phpnetcvs*[16], é uma implementação do protocolo CVS em puro PHP, que se encontra na sua versão 1.3.7 e com uma licença BSD.

Este projeto implementa um cliente que interage com repositórios CVS. Contudo, não implementa todas as funções do protocolo CVS e aquela que mais interesse tem, o *export*, é uma delas.

### 2.3.4 Interação via SMB

- **Samba Web Client**

O *Samba Web Client*[17] é uma classe desenvolvida em PHP com licença GPL[18], que permite diversas operações sobre um repositório *Samba*. Esta classe, além das operações no repositório, permite também listar pastas e até imprimir conteúdos remotos. Contudo, esta faz parte de uma solução web completa e seria necessário alterar bastante o código fonte por forma a ser usado no âmbito do presente projeto.

- **smbclient**

O *smbclient*[19] é uma classe desenvolvida em PHP que faz uso de chamadas ao *smbclient* do sistema operativo. Esta classe é semelhante à anterior, mas muito mais simples em termos de funcionalidades. O desenvolvimento do plugin para interação com repositórios SMB, poderá adotar esta filosofia de chamadas ao cliente do sistema operativo, sendo esta classe um bom exemplo e igualmente um bom ponto de partida.

## 2.4 Cifras

Numa visão geral, um algoritmo de cifra consiste numa função matemática que, aplicada num texto ou objeto binário, gera um novo objeto ou texto final, que se diz cifrado. A segurança do mecanismo de cifra está diretamente relacionada com a chave utilizada, quer por cifras simétricas ou assimétricas.

As cifras podem ser divididas em duas categorias: as cifras simétricas e as assimétricas. Uma cifra simétrica pressupõe a existência de partilha de uma chave previamente acordada entre o emissor e o recetor da mensagem, e mantida fora do âmbito público - ambas as partes em causa são conhecedoras daquela chave, o que permite o processo de cifra e decifragem nos dois sentidos. Já no caso da cifra assimétrica, são geradas duas chaves - uma pública e outra privada. A chave privada é utilizada no processo de decifragem e de assinatura (garantia de não repúdio), já a chave pública é utilizada para cifra. Este mecanismo permite cifrar mensagens para um determinado grupo, desde que se tenha prévio conhecimento das suas chaves públicas. Por exemplo, pretendendo duas entidades A e B passar uma mensagem entre elas, e sendo do conhecimento de A a chave pública de B e reciprocamente do conhecimento de B a chave pública de A, A cifra a sua mensagem com a chave pública de B, e B recebe essa mesma mensagem executando a decifragem da mesma recorrendo à sua chave privada; de forma análoga, B envia uma mensagem cifrada com a chave pública de A e este decifra com a sua chave privada.

No âmbito da presente dissertação, o tipo de cifra utilizado foi a simétrica. A razão desta escolha deve-se ao facto de que a informação a cifrar se encontra dentro de um ambiente considerado seguro, ou seja, encontra-se dentro das infraestruturas da entidade acolhedora, e igualmente pelo facto do processo de encriptação e desencriptação de dados ser desempenhado sempre pela mesma entidade/módulo. É importante referir, que este tipo de cifra apenas é aplicável a esta situação porque a base de dados e a entidade responsável pela encriptação e desencriptação da informação se encontram em máquinas distintas, pelo que se alguém tiver acesso à base de dados, não terá acesso a chave que permite a desencriptação dos dados nela contidos.

Relativamente ao algoritmo de encriptação simétrica, o escolhido foi o AES [20] 256-bit. Esta escolha deve-se ao facto de que este algoritmo é o escolhido pela NSA [21], como sendo um algoritmo *standard* para troca de informação secreta em ambientes considerados como desconhecidos.

## 2.5 Conclusão

Com todo o estudo envolvente e inerente à realização do projeto CodeV-Interação com repositórios de código (SIRC), é possível salientar que este constitui uma inovação e uma substancial melhoria relativamente aos concorrentes apresentados. Na tabela 2.2 que seguidamente se apresenta, procura-se organizar a informação sobre os concorrentes ao CodeV, no âmbito de interação com repositórios de código.

Produto	File Upload	SVN	SCP	HttpZip	SMB	CVS	HTTP
<i>CodeV</i>	✓	✓	✓	✓	✓	✓	✓
<i>CodeSecure</i>	✓	✓	✗	✗	✓	✓	✗
<i>VeraCode</i>	✓	✗	✗	✗	✗	✗	✗
<i>RIPS</i>	✓	✗	✗	✗	✗	✗	✗
<i>PHP Vulnerability Hunter</i>	✓	✗	✗	✗	✗	✗	✗
<i>IBM Security AppScan</i>	✓	✗	✗	✗	✗	✗	✗

Tabela 2.2: Comparação de Funcionalidades.

Analisando a tabela 2.2, pode-se constatar que este projeto vem colmatar de certa forma uma lacuna existente em relação as funcionalidades disponibilizadas pelos seus concorrentes no âmbito de importações de código. Inicialmente o CodeV já suportava importação via upload simples de ficheiros e Samba, contudo no decorrer do presente projeto, estes foram excluídos e desenvolvidos na integra com o objetivo de serem incluídos sob a forma de plugins. Para além disto, a interação com repositórios SMB, encontrava-se desenvolvida em Python [22], tendo sido eliminada esta dependência.

As principais vantagens inovadoras do SIRC consistem portanto, na agregação de diversas funcionalidades, as quais já existem no mercado, mas não integradas numa solução única, como é possível verificar na tabela 2.2 acima.

Relativamente ao uso de uma framework para o desenvolvimento do projeto, apesar da análise efetuada apontar para alguns aspetos positivos, optou-se por não usar nenhuma delas. A justificação para tal, prende-se com o facto de existir uma curva de aprendizagem e de no final se acabar por obter um resultado que esta dependente de terceiros.

Em relação ao uso de extensões ou bibliotecas que permitam interagir com repositórios de código, foram escolhidos o *PECL SVN Extension* para interação com repositórios SVN e *PECL SSH2 Extension* para interação via SCP, essencialmente por facilitarem o processo de autenticação. No caso da interação via Samba, escolheu-se o *smbclient* em detrimento do *Samba Web Client* visto a licença deste último obrigar a publicação das alterações efetuadas ao código fonte. A utilização do *phpnetcv* foi excluída devido a este não possuir a opção de *export* e apresentar alguns problemas de conexão com os repositórios.



# Capítulo 3

## Método

O presente capítulo descreve o método e abordagem seguidos na implementação do presente projeto. De seguida será documentado o processo de desenvolvimento e o controlo de qualidade.

### 3.1 Processo de desenvolvimento

Para o presente projeto, foi seguida uma abordagem baseada no Método Ágil de Desenvolvimento ou **Desenvolvimento Ágil de Software**, sendo que este se caracteriza pelo desenvolvimento em períodos relativamente curtos denominados de iterações, nunca superiores a uma ou duas semanas.

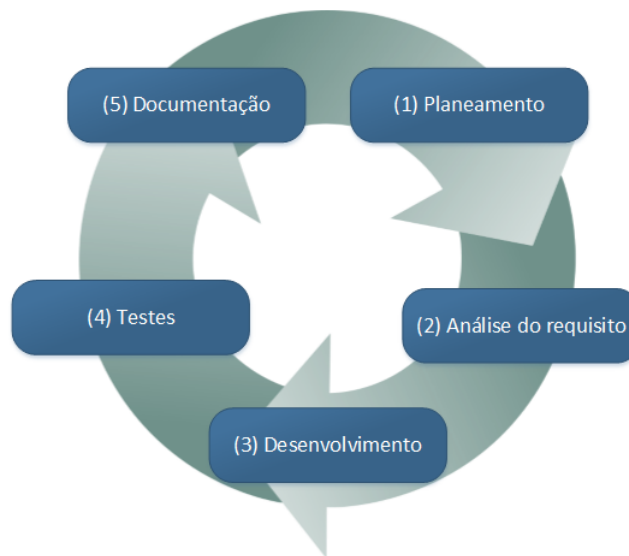


Figura 3.1: Iteração do processo de desenvolvimento.

Cada iteração, como se pode observar na figura 3.1, pode ser comparada a um mini-projeto, pois contém associadas as tarefas necessárias para implementar uma nova funcionalidade. Assim cada iteração/ciclo de desenvolvimento divide-se em:

- **Planeamento** Fase correspondente ao planeamento da funcionalidade a implementar em determinada iteração;
- **Análise do requisito** Analisar o requisito no sentido da funcionalidade em que se enquadra;
- **Desenvolvimento** Representa o desenvolvimento de código da funcionalidade a implementar;
- **Testes** Execução de testes (Unitários, Integração, Sistema e Aceitação);
- **Documentação** Documentar aspetos relevantes sobre a funcionalidade implementada durante a iteração.

A aplicação e o sucesso de um desenvolvimento ágil de software divide-se por três áreas distintas, a organização, a cultura e o pessoal. Os fatores que levam ao sucesso no desenvolvimento de software segundo esta metodologia, traduzem-se na confiança das pessoas envolvidas no projeto, no número de pessoas ser reduzido mas competente, no comportamento da organização que deve promover as decisões dos programadores, assim como na facilidade e rapidez e na comunicação entre os membros. Todos estes fatores de sucesso enquadram-se no comportamento da organização *Dognaedis*, logo sobrevaloriza a escolha de um método ágil de desenvolvimento. Mais se afirma que existem vantagens na escolha do método acima referido pois este prevê situações que se encaixam no presente estágio, como a possibilidade de a qualquer momento ser necessário reestruturar algum requisito, ou no custo da adaptação a novas tecnologias. Conseguindo tirar partido das características que lhe são impostas, transformando por isso imposições em vantagens.

Optou-se por não se seguir linearmente nenhuma metodologia de desenvolvimento ágil como Scrum [23], ou a Programação Extrema [24], dado que a dimensão do projeto e a quantidade de trabalhadores associados não se enquadram perfeitamente em nenhuma metodologia e iria logicamente falhar alguma das duas premissas fundamentais. Em exemplo significativo temos o *Daily Scrum*<sup>1</sup> que iria provocar uma sobrecarga desnecessária, não trazendo neste caso benefícios ao presente projeto.

Resumindo, com a escolha de uma metodologia ágil procura-se obter um método em que seja possível medir o progresso do projeto, responder a mudanças que possam ocorrer durante o desenvolvimento, diminuir os riscos e acima de tudo garantir a colaboração e aprovação do consumidor/orientador da empresa, entregando-lhe rápida e continuamente software funcional.

## 3.2 Controlo de qualidade

No decorrer desta secção encontram-se discriminados os processos que têm como objetivo garantir a qualidade de todo o trabalho desenvolvido no âmbito do presente projeto. É apresentado o cenário utilizado para a validação e seguidamente os testes de validação.

### 3.2.1 Cenário de validação

Todo o controlo de qualidade tem por base um cenário de utilização. No caso do presente projeto, o cenário de utilização é constituído pelos seguintes elementos.

---

<sup>1</sup>Na metodologia Scrum, a Daily Scrum corresponde a uma reunião diária, em que alguns dos participantes no projeto falam sobre o progresso conseguido, o trabalho a ser realizado e/ou o que o impede de seguir o planeamento.

- **Servidor:** 2Gb RAM, Intel(R) Xenon(R) CPU E5606 @ 2.13Ghz, com 100% de disponibilidade.
- **Cliente:** Intel Core i5, 4Gb RAM.
- **Velocidade da Ligação:** 100Mb/s.

### 3.2.2 Testes de validação

A presente secção de testes tem o objetivo de descrever a qualidade de um software, sendo que para tal será realizado e exposto um plano de testes sobre o software a desenvolver. O termo *teste* é usado para avaliar o software através da execução do código, assim como da observação do *output* gerado por essa execução.

Com base na norma ISO 9126 [25], os atributos qualitativos na avaliação com recurso a testes, são: a funcionalidade, a eficiência, usabilidade e portabilidade. Dado estes atributos serem importantes para a obtenção de um bom software, é necessário comparar o seu comportamento com base em resultados esperados, especificações, versões anteriores do mesmo software e expectativas do cliente.

A realização de testes tem o objetivo de testar o sistema, garantindo que as funcionalidades requeridas estão implementadas e a funcionar corretamente de acordo com os requisitos pré-estabelecidos. Além destes, pretende-se ainda garantir que o software cumpre os requisitos de performance, requisitos de usabilidade, garantir que o software é seguro e acima de tudo que no final do desenvolvimento é entregue atempadamente um software funcional e qualidade.

#### Testes aos requisitos funcionais

Com base no livro *Introduction to Software*, de Amman e Offutt [26], em que são mencionados diversos testes, com significados e modelos de aplicações distintos, considera-se vantajoso a execução de alguns desses testes enumerados, após cada funcionalidade desenvolvida. Desta forma foram executados os seguintes testes de forma ascendente:

**Testes de unidade** Analisa, se a unidade mais pequena do projeto (ex. funções) está de acordo com a especificação;

**Testes de integração** Nos testes de integração, o objetivo é verificar se os módulos estão em conformidade com o detalhado, nomeadamente a comunicação/interação entres estes;

**Testes de sistema** O teste de sistema caracteriza-se pela verificação do software já completamente integrado, quanto ao cumprimento dos seus requisitos já num ambiente de produção;

**Testes de aceitação** O teste de aceitação é indicado para verificar se o software se encontra de acordo com os requisitos, este teste exige a validação do cliente, neste caso orientador da empresa.

Com base na estrutura adotada para escrever os requisitos, ID-Requisito, Descrição, Ação, Assunções e Requisitos Especiais, desenhou-se um modelo que facilita na execução de testes ao software desenvolvido. De seguida ilustra-se o modelo em questão para validação dos requisitos funcionais.

<b>ID-Teste:</b> 01			
<b>ID-Requisito:</b> 01-Importação de código			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Importar código para o CodeV			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Autenticar-se perante o sistema inserindo corretamente o <i>username</i> e palavra-chave.	O utilizador acede ao conteúdo do <i>SIRC</i> .	O utilizador acede ao conteúdo do <i>SIRC</i> .	✓
<b>Problemas:</b> ...			

Tabela 3.1: Modelo para validação dos requisitos funcionais.

### Testes aos requisitos não funcionais

Seguidamente apresentam-se os mecanismos usados para validação dos requisitos não-funcionais que foram anteriormente especificados na secção 5.3.

Os resultados provenientes da execução destes mecanismos, encontram-se devidamente apresentados no capítulo 8.

#### Validação da Usabilidade

Existem dois métodos para realizar a validação de usabilidade de um software, ou se segue as *guidelines* respetivas, ou se realiza um conjunto de inquéritos a pessoas que não estiveram no desenvolvimento do software, questionando-as sobre a usabilidade do produto.

Optou-se pelo segundo método descrito, em que houve lugar a elaboração de inquéritos sobre a usabilidade do software resultante do presente projeto.

#### Validação da Robustez

A robustez foi avaliada com recurso a testes de Fuzzing, com o objetivo de verificar se foram tratadas todas as exceções provenientes de *user inputs*. Contudo, estes testes não foram efetuados de uma forma automatizada, mas sim manualmente durante o desenvolvimento. Para além destes testes, foram cortadas ligações a determinados repositórios para verificar se o SIRC identifica corretamente estes casos. Passando nestes dois testes, o requisito robustez é considerado como validado.

#### Validação da Segurança

O SIRC-CodeV foi sujeito primeiramente a uma análise de vulnerabilidades por parte do motor de análise de vulnerabilidades da entidade acolhedora (CodeV) e a uma auditoria por parte de uma equipa especializada

Para validação da segurança do SIRC-CodeV no futuro, aquando a importação de um novo plugin, este é sujeito a uma análise de vulnerabilidades por parte do CodeV.

#### Validação da Modularidade

A validação da modularidade nunca é uma tarefa fácil, pois depende do projeto e do produto a desenvolver. No caso do SIRC - CodeV, tratando-se de um motor de plugins, para que o requisito de modularidade seja cumprido será avaliado se existe alguma dependência entre entre

os plugins, onde se espera que se possa remover ou adicionar novos plugins sem afetar o bom funcionamento dos restantes.

#### **Validação da Escalabilidade**

A validação da escalabilidade passou por variar a carga de clientes com o objetivo de perceber qual o número máximo de clientes suportados em simultâneo. O número máximo será aquele a partir do qual o tempo de resposta da interface do *SIRC-CodeV* passe a ser superior em um segundo, até um máximo de dois segundos.

Os valores acima mencionados, têm por base o livro *Response Time in Man-computer Conversational Transactions* de Miller, R. B (1968)[27], em que um segundo é o tempo limite para que no decorrer de interações Humano Computador, o utilizador sintá que está a interagir livremente com as informações. No mesmo estudo Miller realça ainda que o tempo de resposta pode ir até dois segundos sem que o utilizador perca a atenção sobre o sistema.

### **3.3 Conclusão**

No presente capítulo explorou-se o método e o processo de desenvolvimento do software a desenvolver. Este foi executado sobre uma abordagem de Desenvolvimento Ágil de Software, constituída por iterações de curta duração com várias tarefas associadas. Foi ainda exposto o levantamento e validação da escrita de requisitos, assim como os testes a implementar sobre os requisitos funcionais e não funcionais.

# Capítulo 4

## Planeamento

### 4.1 Metodologia

Como metodologia de trabalho foram adotados, sempre que possível, ciclos de trabalho de duas semanas. A razão do espaço temporal atribuído a estes, deve-se ao facto de que em caso de ciclos demasiado grandes, a gestão de tarefas seria mais complexa, uma vez que seria mais complicado para o autor situar-se no tempo. Em caso de ciclos menores que duas semanas, haveria lugar a tarefas mais pequenas, o que faria com que o número destas aumentasse, tornando desta forma a gestão de trabalhos mais complexa.

Estes ciclos de trabalho iniciaram-se com dois dias reservados para revisão e documentação do trabalho realizado anteriormente, bem como para reuniões com o orientador da empresa no sentido de validar tanto o trabalho efetuado como o trabalho a realizar, procurando sempre despistar previamente possíveis desvios ao plano de trabalhos. No final de cada ciclo, houve lugar a alguns testes, como se encontra descrito adiante na secção 4.3.1, relativa ao WBS. De notar que ao longo de cada ciclo, foram efetuados testes mínimos (testes unitários e de integração) para despistar possíveis erros que pudessem ocorrer.

É de salientar, que estes ciclos de desenvolvimento foram validados pelo orientador da empresa, uma vez que como foi dito atrás, houve lugar a reuniões no início de cada um dos ciclos e onde o tempo atribuído a estes teve em conta uma pequena margem de erro para colmatar possíveis derrapagens decorrentes dos riscos identificados na realização do projeto.

### 4.2 1º Semestre

#### 4.2.1 Diagrama de Gantt

Para facilitar a visualização do planeamento do primeiro semestre, foi construído um gráfico de Gantt com as principais fases e respetivas metas do projeto. Na tabela 4.1 estão expressas as principais fases e metas, encontrando-se o gráfico de Gantt completo disponível no anexo B.1.

	Início	Fim	Duração(dias)
<b>Ambientação à Empresa</b>			
Configuração da estação de trabalho	17/09/12	17/09/12	1
Familiarização com ferramentas internas	18/09/12	18/09/12	1
<b>Estado da Arte</b>			
Estudo de produtos semelhantes	19/09/12	25/09/12	5
Estudo de Frameworks Plugins Based	26/09/12	02/10/12	5
Estudo de software de interação com repositórios	03/10/12	09/10/12	5
Documentação	10/10/12	12/10/12	3
<b>Levantamento e especificação de requisitos</b>			
Arquitetura do Sistema	05/11/12	09/11/12	5
Work Breakdown Structure	12/11/12	23/11/12	10
<b>Desenvolvimento do motor de plugins</b>			
Mini planeamento e análise do requisito	26/11/12	27/11/12	2
Implementação da estrutura base do motor	28/11/12	07/12/12	8
Mini planeamento e análise do requisito	10/12/12	11/12/12	2
Implementação da Ativação / desativação de Plugins	12/12/12	21/12/12	8
<b>Elaboração do Relatório Intermédio</b>			
	24/12/12	28/01/13	26

Tabela 4.1: Tabela auxiliar ao Diagrama de Gantt de 1º semestre.

Por forma a esclarecer alguns tópicos da tabela 4.1, fornece-se de seguida uma breve descrição de alguns destes.

A configuração da estação de trabalho, consistiu na montagem e instalação de todo o software no computador fornecido para trabalho. A familiarização com ferramentas de trabalho deveu-se a aprendizagem com normas e regras da instituição acolhedora. A tarefa *Documentação* incluída no final da tarefa *Estado da arte*, diz respeito a uma documentação preliminar desse capítulo no relatório intermédio. Os restantes tópicos são auto explicativos através do seu nome.

## 4.3 2º Semestre

Esta secção encontra-se subdividida em cinco partes, sendo uma relativa ao *Work Breakdown Estruture*, seguida das estimativas de tempo atribuídas às tarefas descritas na subsecção anterior, uma relativa ao diagrama de Gantt e ainda uma secção destinada a alterações que ocorreram no plano. A última subsecção destina-se aos riscos associados à realização do presente projeto.

### 4.3.1 Work breakdown structure

Como já foi introduzido anteriormente no presente capítulo, houve lugar a ciclos de desenvolvimento de duas semanas, pelo que as tarefas pensadas no *WBS* tiveram em conta esta margem temporal para a sua realização. Observando o gráfico do *WBS* que se encontra no anexo C.1, todas as tarefas dos penúltimos níveis comportam este espaço temporal de duas semanas para a sua realização, havendo apenas quatro exceções relativamente às tarefas *Integração Preliminar com o CodeV*, *Integração Final com o CodeV*, *Testes Finais* e *Deploy da Versão final*.

Falando um pouco de tarefas que possam ser menos óbvias, a *Integração com o CodeV* representa a integração do módulo resultante do presente projeto, com o módulo *Security*

*management tool* já retratado no capítulo anterior. Após a integração dos dois módulos houve lugar a *Testes Finais*, onde se aferiu se o funcionamento era o esperado. Depois de aferido o o funcionamento, houve lugar ao *Deploy da versão final*.

### 4.3.2 Esforço

Na tabela 4.2 encontra-se descrito o esforço em número de horas atribuídas a cada tarefa de penúltimo nível do WBS. Como se pode constatar, a duração destas tarefas segue o padrão de tempo definido(ciclos de duas semanas). É de salientar que o tempo despendido em tarefas que foram efetuadas no segundo semestre é maior que no primeiro, uma vez que no segundo o número de horas semanais equivale a 40 horas e no primeiro apenas a 16 horas.

Tarefas	Esforço(h)
Implementação da estrutura base do motor de plugins	32
Implementação da Ativação/desativação de plugins	32
Implementação da importação/exclusão de Plugins	80
Implementação do sistema de gestão de credenciais e do explorador de ficheiros	80
Implementação da importação via SVN	80
Implementação da importação via SCP e SMB	80
Implementação da importação via CVS e HTTP	80
Implementação da exportação de relatórios e da importação via FileUpload	80
Implementação da repetição de importações e sistema de agendamentos	80
Integração com o CodeV e testes finais	40
Deploy da versão final	8

Tabela 4.2: Esforço das tarefas em horas.

### 4.3.3 Diagrama de Gantt

Para facilitar a visualização do planeamento do segundo semestre, foi construído um gráfico de Gantt com as principais fases e respetivas metas do projeto. Na tabela 4.3 estão expressas as principais fases e metas, encontrando-se o gráfico de Gantt completo disponível no anexo B.3.



Tarefas	Início	Fim	Duração(dias)
<b>1) Ciclos de desenvolvimento</b>			
1.1) Implementação importação/exclusão de Plugins	11/02/13	22/02/13	10
1.2) Implementação do sistema de gestão de Credenciais e explorador de ficheiros	25/02/13	8/03/13	10
1.3) Implementação da importação via SVN	11/03/13	22/03/13	10
1.4) Implementação da exportação de relatórios e importação via FileUpload	25/03/13	5/04/13	10
<b>1.5) Integração preliminar com o CodeV</b>			
1.5.1) Integração e execução de testes preliminares	8/04/13	8/04/13	10
1.5.2) Deploy de uma versão intermédia	9/04/13	9/04/13	10
1.6) Implementação da importação via SCP e SMB	10/04/13	23/04/13	10
1.7) Implementação da importação via HTTP e CVS	24/04/13	7/05/13	10
1.8) Implementação da repetição de importações e sistema de agendamentos	8/05/13	21/05/13	10
2) Integração final com o CodeV e testes finais	22/05/13	28/05/13	5
3) Deploy da versão final	29/05/13	29/05/13	1
4) Documentação do relatório final	30/05/13	28/06/13	22

Tabela 4.3: Tabela auxiliar ao Diagrama de Gantt de 2º semestre.

#### 4.3.4 Identificação de Riscos

Na implementação do projeto SIRC, identificaram-se possíveis riscos em duas vertentes. Uma das vertentes foi a familiarização com tecnologias desconhecidas, neste caso em concreto, a linguagem PHP e o JavaScript. A outra vertente foi a dependência de trabalho de terceiros.

Na primeira vertente, foi necessária a familiarização com a linguagem PHP e JavaScript uma vez que estas não foram linguagens abordadas no plano de estudos da Licenciatura e Mestrado em Eng. Informática.

Na segunda vertente, existiu a dependência de terceiros, que neste caso foi a dependência do módulo *Security management tool*, que se encontrava a ser desenvolvido por um colega de trabalho, igualmente enquadrado na sua dissertação de mestrado. Esta dependência foi gerada, porque este módulo SIRC por si só não funciona como um todo, necessitando sempre de informação externa.

#### 4.3.5 Alterações ao plano

Como já foi introduzido anteriormente, devido a interesses do ponto de vista do negócio por parte da entidade acolhedora, a classificação do requisito *Exportação de relatórios de vulnerabilidades* passou de *Nice to Have* para *Must Have*. Esta alteração motivou alterações no plano de desenvolvimento relativo ao 2º semestre. A tabela 4.4 abaixo, ilustra o resumo do plano que inicialmente fora traçado e no apêndice B.2 encontra-se o respetivo diagrama de Gantt.

Contrastando o planeamento inicial 4.4, com o planeamento final 4.3 é possível verificar que a maior das alterações foi a inclusão do requisito *Exportação de relatórios de vulnerabilidades*

Tarefas	Início	Fim	Duração(dias)
<b>Desenvolvimento do motor de plugins</b>			
Implementação importação/exclusão de Plugins	11/02/13	22/02/13	10
Implementação do sistema de gestão de Credenciais	25/02/13	8/03/13	10
<b>Desenvolvimento de plugins</b>			
Implementação da importação via SVN e Fileupload	11/03/13	22/03/13	10
Implementação da importação via SCP e HTTP	25/03/13	05/04/13	10
Implementação da importação via SMB	8/04/13	19/04/13	10
Implementação da importação via CVS	22/04/13	3/05/13	10
<b>Desenvolvimento de outras funcionalidades</b>			
Implementação de reimportações e programação de importações	6/05/13	17/05/13	10
<b>Integração com o CodeV e testes finais</b>	20/05/13	24/05/13	5
<b>Deploy da versão final</b>	27/05/13	27/05/13	1
<b>Documentação do relatório final</b>	28/05/13	28/06/13	24

Tabela 4.4: Tabela auxiliar ao Diagrama de Gantt Inicial de 2º semestre.

e da *Integração preliminar com o CodeV*. Houve também lugar a alterações na ordem de desenvolvimento das tarefas, mas devido a metodologia de desenvolvimento escolhida no capítulo 3, estas alterações não implicaram quaisquer consequências. Além destas alterações, houve necessidade agrupar o desenvolvimento de dois plugins de importação de código em apenas um ciclo de desenvolvimento, por forma a libertar o máximo espaço temporal possível. Esta alteração motivou que o trabalho nesse ciclo de desenvolvimento ocorresse a um ritmo bastante elevado, havendo a necessidade de diminuir o tempo para análise de requisitos, tempo para testes e documentação.

O impacto causado pelas alterações efetuadas ao plano afetaram o tempo total reservado para escrita do relatório final, tendo este sido encurtado em dois dias.

### 4.3.6 Conclusão

Com o planeamento efetuado, chegou-se à conclusão que o projeto tinha tudo para ser concretizado com sucesso. Existiram algumas dependências identificadas claramente nos riscos, embora não sendo muito críticas. Contudo, a segunda dependência foi algo que foi acompanhado durante o decorrer do projeto, para que no final não existissem quaisquer entraves.

O requisitos *Nice to have* não foram incluídos no plano, visto não ter existido margem temporal para o seu desenvolvimento.

Para finalizar é importante salientar que as alterações no plano de trabalhos do 2º semestre tiveram apenas como consequência uma sobrecarga de trabalho num dos ciclos de desenvolvimento e o encurtamento em dois dias na escrita da dissertação final, contudo foi possível respeitar as datas delineadas.

## Capítulo 5

# Análise de Requisitos

Foi efetuado um levantamento das restrições do projeto e posteriormente efetuada uma análise de requisitos que o produto final deveria comportar. Os resultados desse estudo são apresentados no decorrer deste capítulo, em que uma secção se encontra atribuída aos requisitos funcionais e outra aos requisitos não funcionais. Cada requisito encontra-se designado por um ID, por forma a facilitar na validação destes.

Relativamente aos requisitos funcionais, estes apresentam-se sob a forma de um diagrama de casos de uso que é apresentado de uma forma geral e seguidamente acompanhado de descrições mais detalhadas relativas a cada um dos casos de uso apresentados.

### 5.1 Restrições

Para a realização do presente projeto, foram desde logo impostas duas restrições. Uma delas prende-se com o uso da linguagem PHP [28] no desenvolvimento e a outra refere-se à obrigatoriedade de utilização de uma base de dados PostgreSQL[29]. A utilização do PHP deve-se essencialmente ao facto da interface atual e futura do CodeV ter sido desenvolvida nesta linguagem. Em relação a segunda restrição, esta deve-se simplesmente ao facto de ser o motor de base de dados mais usada pela entidade acolhedora, contudo a justificação que leva a escolha deste encontra-se na subsecção 6.4.6.

### 5.2 Requisitos Funcionais

O estudo efetuado relativo aos requisitos funcionais teve como ponto de partida a elaboração de um diagrama de casos de uso em que foram identificados desde logo três *atores*, o *Gestor de projetos*, o *Utilizador* e o *Administrador do sistema*. Contudo, no âmbito deste projeto, dado que o *Gestor de Projetos* e o *Utilizador* têm acesso às mesmas funcionalidades, são ambos representados pelo ator *Utilizador*.

Depois de elaborada a lista de atores do sistema, foram definidos os casos de uso.

No decorrer da elaboração do diagrama de casos de uso, foram emergindo alguns requisitos funcionais que apesar de não implicarem uma interação com nenhum dos atores do sistema, não deixam por si só de ser requisitos funcionais, pelo que se encontram igualmente enumerados e descritos nesta secção.

## 5.2.1 Casos de Uso

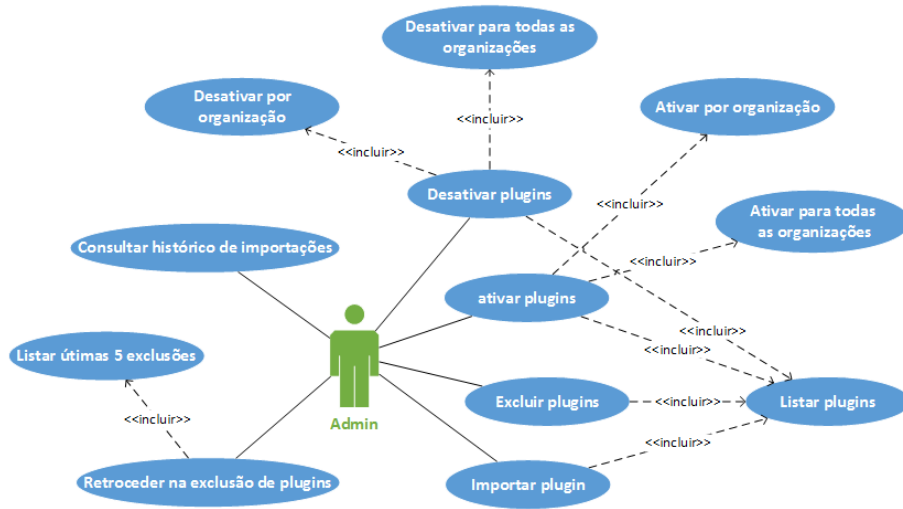


Figura 5.1: Diagrama de Casos de uso do Administrador.

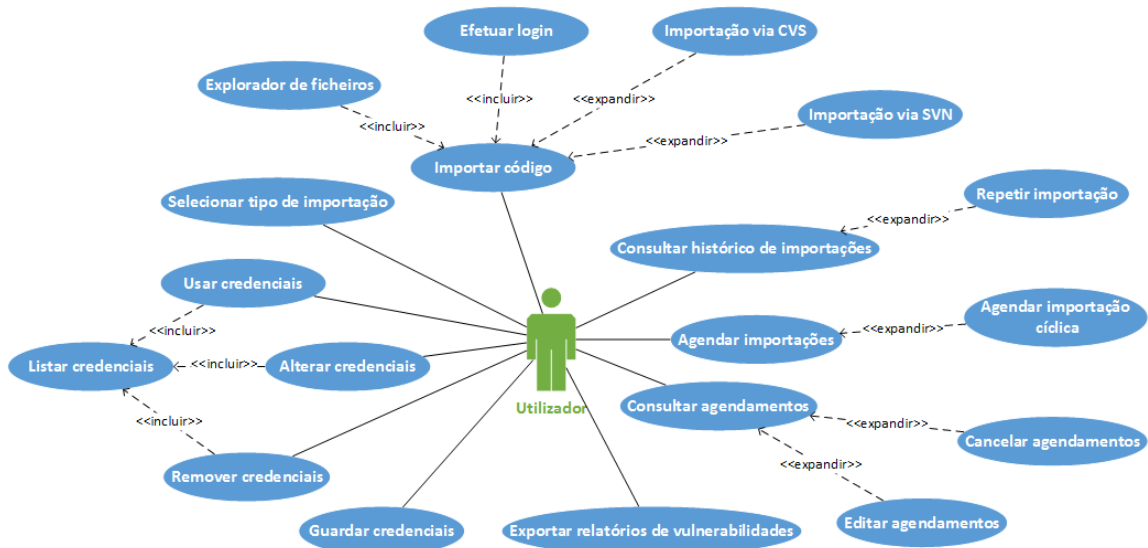


Figura 5.2: Diagrama de Casos de uso do Utilizador.

## Descrições em detalhe:

### Atores

*Utilizador:* O utilizador é responsável pelo desenvolvimento de determinado software e pretende aferir se este contém ou não vulnerabilidades.

*Administrador:* O administrador é responsável pela configuração e gestão do motor de plugins resultante da implementação do presente projeto.

### ID-01: Selecionar tipo de importação

**Descrição:** O utilizador tem a possibilidade de selecionar diferentes tipo de importação de código. Entre estes encontram-se, importação através de SVN, CVS, upload simples de ficheiros, Samba, HTTP, SCP.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende importar código fonte para o CodeV analisar.

**Assunções:** Assume-se que o utilizador já se encontra autenticado no sistema.

**Requisitos Especiais:** Nenhum.

**Justificação:** No âmbito deste projeto, este requisito é aquele que mais importância assume, dado que um dos objetivos principais é capacitar o *CodeV* com diferentes tipos de importação.

### ID-02: Efetuar login

**Descrição:** Este caso uso permite ao utilizador autenticar-se perante um repositório.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende efetuar login num repositório.

**Assunções:** Assume-se que o utilizador já se encontra autenticado no sistema.

**Requisitos Especiais:** A palavra chave não deve ser exibida em texto simples.

**Justificação:** Existência de repositórios que requerem autenticação.

### ID-03: Importação de código

**Descrição:** A importação de código é um caso de uso referente ao utilizador. Como o próprio nome indica, permite ao utilizador importar código para análise de vulnerabilidades.

O utilizador tem a possibilidade de escolher diversos tipos de importação de código dos que estejam disponíveis, conforme o administrador assim o permitir, isto porque o administrador pode ativar ou desativar plugins, por exemplo. Para além disto, o utilizador pode configurar uma importação a ser efetuada no futuro, sendo igualmente possível configurar uma importação de forma a que esta ocorra ciclicamente, ou seja, todos os dias a uma determinada hora, por exemplo. Neste caso de uso é ainda possível guardar credenciais de acesso e usar credenciais de acesso previamente guardadas para auto-completar os campos de acesso a repositórios.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende importar código fonte para o CodeV analisar.

**Assunções:** Assume-se que o utilizador já se encontra autenticado no repositório, caso o repositório necessite de credenciais de acesso.

**Requisitos Especiais:** Consoante o tipo de importação que o utilizador desejar efetuar, poderá existir a necessidade de inserir credenciais de acesso. Nesses casos, a palavra-chave não deverá ser exibida em texto simples.

**Justificação:** No âmbito deste projeto, este requisito é aquele que mais importância assume, dado que um dos objetivos principais, é a importação de código para o *CodeV*.

#### **ID-04: Importação de código via SVN**

**Descrição:** Este caso de uso é uma extensão do anterior porque este tipo de repositórios permite adicionar mais funcionalidades a importação de código, nomeadamente, é possível escolher a revisão que se deseja importar. Para tal, é mostrada uma lista das revisões disponíveis no repositório assim como a pessoa responsável por cada uma delas, podendo o utilizador pesquisar uma revisão pelo seu número ou pelo seu autor. Existe ainda um explorador de ficheiros que permite visualizar a estrutura de ficheiros e pastas correspondente a determinada revisão, podendo o utilizador selecionar os ficheiros ou pastas que deseja importar para análise.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende importar código fonte para o CodeV analisar.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Caso o acesso ao repositório necessite de credenciais, a palavra-chave não deverá ser exibida em texto simples.

**Justificação:** O utilizador pode desejar importar uma versão mais antiga do repositório em que tenha sido ele a efetuar *commit*. Em casos de repositórios com uma vasta lista de revisões, o sistema pesquisa de revisões é uma mais valia.

#### **ID-05: Importação de código via CVS**

**Descrição:** Este caso de uso é uma extensão do caso de uso *Importação de código* porque este tipo de repositórios permite adicionar mais funcionalidades a importação de código, nomeadamente, é possível escolher a *tag* que se deseja importar. Para tal, é mostrada uma lista das *tags* disponíveis no repositório, podendo o utilizador efetuar pesquisas e selecionar qualquer destas. Existe igualmente um explorador de ficheiros que ilustra a estrutura de ficheiros e pastas correspondente a determinada *tag*.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende importar código fonte para o CodeV analisar.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Caso o acesso ao repositório necessite de credenciais, a palavra-chave não deverá ser exibida em texto simples.

**Justificação:** O utilizador pode desejar importar uma versão mais antiga do repositório. Em casos de repositórios com uma vasta lista de *tags*, o sistema pesquisa de é uma mais valia.

#### **ID-06: Explorador de ficheiros**

**Descrição:** Este caso de uso pretende dar mais flexibilidade ao utilizador no momento da importação de um código fonte. Este oferece ao utilizador a possibilidade visualizar a estrutura de ficheiros e pastas que este deseja importar. Existe a possibilidade de selecionar apenas o conteúdo que se pretende importar.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende importar código fonte para o CodeV analisar.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Caso o acesso ao repositório necessite de credenciais, a palavra-chave não deverá ser exibida em texto simples.

**Justificação:** Evitar que se tenha de importar todo o conteúdo de um repositório. Permitir a visualização do conteúdo de um repositório dentro do próprio *CodeV*.

#### **ID-07: Consultar histórico de importações**

**Descrição:** Como é possível verificar pelos diagramas da figura 5.2.1, este é o único caso de uso que tanto pode ser acedido pelo utilizador como pelo Administrador. O objetivo é que no caso do utilizador, este tenha a possibilidade de consultar o seu histórico de importações e para além disto, que possa também repetir submissões (caso de uso retratado mais adiante). No caso do Administrador, este tem a possibilidade de consultar o histórico de importações relativo a qualquer utilizador, este será um caso de uso com uma finalidade de *debug* ao sistema, não podendo o Administrador ter acesso a credenciais de login nos repositórios nem aos *paths* destes.

O histórico de importações, encontra-se apresentado sob a forma de uma tabela, em que cada linha representa uma importação e as colunas as respetivas informações desta, sendo este o esquema adotado para os dois tipos de atores no sistema. Além disto, este caso de uso disponibiliza diversos tipos de ordenação, como por exemplo, ordenação por nome, descrição, data de importação, número de vulnerabilidades, número de total de ficheiros e estado atual (analisada, cancelada, agendada).

**Prioridade:** *Must Have*

**Ação:** O utilizador ou o administrador pretende consultar o histórico de importações.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** O administrador não pode visualizar informações relativas a credenciais de acesso utilizadas nas importações.

**Justificação:** Caso o utilizador deseje repetir uma importação, é importante que se faça uma listagem das importações efetuadas por este. No caso do administrador, este requisito é importante porque permite aferir se tem havido importações de código ou não. Desta forma é uma ferramenta que permite aferir sobre a frequência de utilização do *CodeV*.

#### **ID-08: Repetir importação**

**Descrição:** Este é um caso de uso referente ao utilizador e que fornece a possibilidade de repetir uma importação efetuada no passado. Este caso de uso faz *include* ao caso de uso anterior, uma vez que para um utilizador repetir uma importação, ele terá de visualizar o seu histórico de importações.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende repetir uma importação passada.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Caso sejam detetados falsos positivos ou falsos negativos em determinada importação e depois dos ajustes necessários para colmatar esses aspetos, com este caso de uso é possível que se repita novamente a importação em causa, com o objetivo de aferir se a situação ficou corrigida.

#### **ID-09: Guardar credenciais de acesso**

**Descrição:** Um utilizador tem a possibilidade de guardar as credenciais de acesso a um determinado repositório. Para tal, dispõe de uma opção, que quando marcada, surge um campo para atribuir um nome às credenciais a guardar. Uma vez, efetuada a importação de código com sucesso, aquele grupo de credenciais fica imediatamente disponível para utilizações futuras.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende guardar credenciais de acesso.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Caso um utilizador se encontre a trabalhar frequentemente com o mesmo repositório, para evitar que haja sempre a necessidade de inserção das credenciais de acesso, este caso de uso permite que estas sejam guardadas para utilizações futuras.



#### **ID-10: Listar credenciais de acesso**

**Descrição:** Este caso de uso consiste, como o nome indica, na listagem de credenciais de acesso de um utilizador a um repositório. O objetivo deste caso de uso é evitar que o utilizador ao estar a usar frequentemente o mesmo repositório, seja obrigado a introduzir sempre os dados de *login*.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende listar credenciais de acesso.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** O utilizador não tem de se lembrar de todas as credenciais de acesso que guardou. Com este requisito, ele tem a possibilidade de verificar as credenciais que se encontram guardadas.

#### **ID-11: Remover credenciais de acesso**

**Descrição:** Assim como existe a possibilidade de guardar credenciais de acesso, o utilizador tem também a possibilidade de as remover, para que credenciais que deixem de ser usadas com mais frequência, deixem de constar na lista.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende remover credenciais de acesso.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** No caso de algumas das credencias guardadas deixar de ser útil, o utilizador tem a possibilidade de as eliminar.

#### **ID-12: Usar credenciais de acesso**

**Descrição:** O utilizador ao aceder ao menu de importação de código, tem à disposição uma lista com as credenciais previamente guardadas. Ao selecionar uma delas, haverá lugar a um *autocomplete* dos campos relativos às credenciais de acesso ao repositório em questão.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende usar credenciais de acesso previamente guardadas.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Este caso de uso é importante dado que permite usar as credencias de acesso guardadas previamente por um utilizador.

#### **ID-13: Alterar credenciais de acesso**

**Descrição:** O utilizador tem a possibilidade de editar credenciais de acesso previamente guardadas.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende editar credenciais de acesso.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** No caso de ter havido um engano no preenchimento de credencias de acesso previamente guardadas, ou estas tenham sofrido alterações entretanto, o utilizador tem com este requisito a possibilidade de as corrigir.

#### **ID-14: Agendar importações**

**Descrição:** Quando o utilizador estiver perante o menu de importações de código, este tem a possibilidade de especificar uma data e hora em que a importação deverá ser efetuada.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende agendar uma importação.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** O utilizador pode pretender que a importação ocorra mais tarde, por exemplo, a uma hora em que a plataforma esteja a ser menos utilizada.

#### **ID-15: Agendar importações cíclicas**

**Descrição:** O presente caso de uso permite ao utilizador agendar importações cíclicas, em mais detalhe, é possível ao utilizador agendar importações diárias, semanais e mensais.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende agendar uma importação cíclica.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Este caso de uso permite dar mais flexibilidade ao agendamento de importações.

Desta forma é possível a um gestor de projetos proceder a agendamentos em projetos à sua responsabilidade, possibilitando uma análise de vulnerabilidades mais consistente e facilitada ao longo de todo o desenvolvimento, dado que uma vez programado, todo o processo decorre de uma forma automática.

#### **ID-16: Consultar agendamentos**

**Descrição:** Este caso de uso permite ao utilizador consultar os agendamentos que se encontram ativos na plataforma. Esta consulta é efetuada por projeto, podendo o utilizador visualizar os seus agendamentos em cada um dos projetos em que se encontra alocado.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende verificar se tem agendamentos em vigor.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Este caso de uso retira ao utilizador a obrigação de saber se detém ou não agendamentos ativos, podendo este consultar esta informação a qualquer momento.

#### **ID-17: Cancelar agendamentos**

**Descrição:** Este caso de uso permite ao utilizador proceder ao cancelamento de agendamentos.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende cancelar agendamentos que se encontrem em vigor.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** O presente caso de uso pretende responder à necessidade de eliminação de agendamentos cíclicos caso estes percam importância para o projeto, contudo é igualmente importante dar ao utilizador a possibilidade de cancelar qualquer tipo de agendamento a qualquer momento.

#### **ID-18: Editar agendamentos**

**Descrição:** Este caso de uso permite ao utilizador proceder à edição de agendamentos que se encontrem em vigor.

**Prioridade:** *Must Have*

**Ação:** O utilizador pretende editar agendamentos em vigor.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** As credenciais de acesso a repositórios ao serem alteradas, invalidariam agendamentos que necessitassem de aceder a estes. Este caso de uso pretende dar ao utilizador a possibilidade de colmatar esta problemática, evitando que este tenha de cancelar e criar novamente os agendamentos afetados.

#### **ID-19: Listar plugins**

**Descrição:** Este caso de uso é utilizado por outros casos de uso que lhe fazem *include*. A sua finalidade é simplesmente, como o seu próprio nome indica, mostrar os plugins existentes na plataforma.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende importar, excluir, desativar ou ativar plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** O administrador pode não ter conhecimento dos plugins de importação de código existentes no SIRC. Este requisito permite que o administrador tenha a possibilidade de verificar os plugins existentes, independentemente se estes se encontram ativados ou desativados.

#### **ID-20: Importar plugins**

**Descrição:** Com este caso de uso é dado ao administrador, a possibilidade de importar plugins desenvolvidos para a plataforma. O administrador terá disponíveis dois meios de importação, sendo eles: a importação através do browser e a colocação dos plugins numa determinada pasta.

A importação de um novo plugin apenas ocorre com sucesso se este cumprir todo um conjunto de regras que se encontram definidas em detalhe na secção 7.2.3.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende importar um novo plugin.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Caso exista a necessidade de importar código a partir de uma localização que não se encontre contemplada na solução final deste projeto, existe a possibilidade de desenvolver um novo plugin para tal. Este caso de uso pretende dar a possibilidade de importação de uma forma simplificada de novos plugins que sejam desenvolvidos.

#### **ID-21: Excluir plugins**

**Descrição:** Assim como é possível importar novos plugins que possam vir a ser desenvolvidos, existe também a possibilidade de excluir aqueles que passem a ter menos utilidade. Para evitar que sejam eliminados plugins por lapso, é necessário confirmar que se deseja proceder à eliminação de determinados plugins, sendo esta funcionalidade levada a cabo por uma mensagem que resume os plugins que irão ser excluídos e pede para se confirmar exclusão. É importante salientar que caso existam agendamentos de importações no sistema que usem alguns dos plugins a ser excluídos, o administrador será alertado na mensagem referida, ficando a cargo deste a decisão de os excluir. Caso o administrador opte pela exclusão, todos os agendamentos serão igualmente excluídos.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende excluir um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** É importante manter o estado interno do módulo, ou seja, tratar todas as dependências, nomeadamente, agendamentos dependentes dos plugins a excluir.

**Justificação:** Caso se desenvolva um novo plugin para substituir outro por motivos de mau funcionamento ou apenas por uma mera atualização de funcionalidades, este caso de uso permite a remoção do antigo.

#### **ID-22: Ativar plugins**

**Descrição:** Este caso de uso permite ao administrador ativar plugins. Esta ativação de plugins é efetuada para cada uma das organizações presentes no sistema ou para todas.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende ativar um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** A entidade acolhedora tem possibilidade de ativar e desativar plugins para os clientes que pretender. Seja por razões técnicas em que o uso de determinado plugin não faça sentido, ou por razões de estratégia de negócio.

#### **ID-23: Ativar plugins por organização**

**Descrição:** Este caso de uso permite que se ativem plugins por organização.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende ativar um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** A entidade acolhedora tem a possibilidade ativar plugins por organização, permitindo que esta possa disponibilizar mais ou menos plugins a determinada organização, consoante o modelo de negócio.

#### **ID-24: Ativar plugins para todas as organizações**

**Descrição:** Este caso de uso permite que se ativem plugins para todas as organizações.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende desativar um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Este caso de uso é útil no sentido em que se detete alguma anomalia no funcionamento de um plugin e se pretenda desativa-lo de imediato em todas as organizações.

#### **ID-25: Desativar plugins**

**Descrição:** À semelhança do caso de uso anterior, este permite a desativação de plugins. Apesar de já existir um caso de uso para exclusão de plugins, é importante que exista apenas a desativação. Isto, porque é interessante que haja a possibilidade de limitar o acesso a determinados plugins a algumas organizações. Por exemplo, uma organização pode ter interesse em ter apenas alguns plugins ao seu dispor.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende desativar um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** A entidade acolhedora tem possibilidade de ativar e desativar plugins para os clientes que pretender. Seja por razões técnicas em que o uso de determinado plugin não faça sentido, ou por razões de estratégia de negócio.

#### **ID-26: Desativar plugins por organização**

**Descrição:** Este caso de uso permite que se desativem plugins por organização.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende desativar um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** A entidade acolhedora tem a possibilidade desativar plugins por organização, permitindo que esta possa disponibilizar mais ou menos plugins a determinada organização, consoante o modelo de negócio.

#### **ID-27: Desativar plugins para todas as organizações**

**Descrição:** Este caso de uso permite que se desativem plugins para todas as organizações.

**Prioridade:** *Must Have*

**Ação:** O administrador pretende desativar um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** Este caso de uso é útil no sentido em que se detete alguma anomalia no funcionamento de um plugin e se pretenda desativa-lo de imediato em todas as organizações.

#### **ID-28: Listar últimas 5 exclusões de Plugins**

**Descrição:** Este caso de uso permite, como o próprio nome indica, listar as últimas 5 exclusões de plugins. Ele tem relação de inclusão como caso de uso abaixo *Retroceder na exclusão de Plugins* e apenas pode ser acedido através deste.

**Prioridade:** *Nice to Have*

**Ação:** O administrador pretende retroceder na eliminação de um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** O administrador pode por lapso remover determinado plugin e para tal pode retroceder nessa ação.

#### **ID-29: Retroceder na exclusão de Plugins**

**Descrição:** Este é um caso de uso referente ao Administrador do sistema. Ele permite que se possa voltar atrás na exclusão de um determinado plugin caso este se encontre entre as últimas cinco exclusões.

**Prioridade:** *Nice to Have*

**Ação:** O administrador pretende retroceder na eliminação de um ou mais plugins.

**Assunções:** Assume-se que o administrador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** O administrador pode por lapso remover determinado plugin e para tal pode retroceder nessa ação.

### **ID-30: Exportação de relatórios de vulnerabilidades**

**Descrição:** Este é um caso de uso referente ao utilizador. Com esta funcionalidade permite-se que exista a possibilidade de exportar um relatório de análise de vulnerabilidades em diferentes formatos, entre eles: *xml*, *json* e *csv*.

**Prioridade:** *Nice to Have*

**Ação:** O utilizador pretende exportar relatórios de vulnerabilidades em diferentes formatos.

**Assunções:** Assume-se que o utilizador se encontra autenticado.

**Requisitos Especiais:** Nenhum.

**Justificação:** O utilizador pode desejar incorporar os relatórios de vulnerabilidade gerados pelo *CodeV* em sistemas nativos da empresa em que enquadra. Por este facto, este caso de uso permite exportar relatórios em formatos padrão amplamente usados.

## **5.2.2 Outros requisitos**

Os requisitos apresentados nesta secção são igualmente requisitos funcionais, contudo, pelo facto de não terem uma interação direta com os atores, não foram incluídos na secção anterior relativa aos casos de uso.

### **ID-31: Módulo auto desenhável**

**Descrição:** O principal objetivo deste requisito é de que o módulo é capacidade de autodesenho da interface do SIRC na interface do *CodeV*, sendo necessário o mínimo de ajustes possível. Mais concretamente, este módulo é ser capaz de desenhar toda a interface necessária ao seu funcionamento.

**Prioridade:** *Must Have*

**Ação:** Nenhuma.

**Assunções:** Nenhuma.

**Requisitos Especiais:** Nenhum.

**Justificação:** Este requisito permite que se reduza a dependência do SIRC, relativamente a outros módulos constituintes do *CodeV*, aumentando desta forma o grau de modularidade.



### **ID-32: Detecção automática de plugins**

**Descrição:** A importação de plugins é um dos casos de uso já discutidos acima. Contudo, este requisito é como que uma alternativa a essa importação via interface web. Este requisito permite que no futuro a pessoa responsável por desenvolver novos plugins para a plataforma, possa adicioná-los diretamente na pasta desta segundo determinadas regras, como por exemplo, o nome do ficheiro de código de um plugin terá de seguir o formato *nomeExemplo.plugin.tipo.php*, terá de existir uma classe dentro do plugin com o mesmo nome atribuído a este (nomeExemplo) e terá de se encontrar dentro de uma pasta igualmente com o nome que lhe foi atribuído. Para além da verificação destas situações, existe ainda a verificação de funções de presença obrigatória e que não se encontrem implementadas, assim como uma análise de vulnerabilidades ao próprio plugin. Caso alguma destas situações não passe nos testes efetuados quando se iniciar a aplicação, esta não irá carregar o plugin em questão. Na secção 7.2.3 relativa a implementação de plugins, encontram-se definidas as regras de desenvolvimentos destes em mais detalhe.

**Prioridade:** *Must Have*

**Ação:** Nenhuma.

**Assunções:** Nenhuma.

**Requisitos Especiais:** Nenhum.

**Justificação:** Caso o módulo SIRC não se encontre em funcionamento, o administrador pode desde logo adicionar o novo plugin desenvolvido na pasta respetiva (consoante as regras mencionadas na descrição). Posteriormente, o novo plugin será detetado e adicionado ao SIRC, caso cumpra as regras de implementação.

### **ID-33: Emissão de erros de comunicação**

**Descrição:** Este requisito relaciona-se com o tratamento de erros da aplicação. Cada vez que surja um erro de comunicação a um determinado repositório, seja um erro de queda de ligação, ou até mesmo logins incorretos, é importante informar sempre o utilizador para que este possa perceber a motivação do erro e possivelmente corrigir algo que esteja ao seu alcance.

**Prioridade:** *Must Have*

**Ação:** Nenhuma.

**Assunções:** Nenhuma.

**Requisitos Especiais:** Nenhum.

**Justificação:** Caso algo alguma operação não ocorra como esperado devido a condições anormais de funcionamento, é importante alertar os atores do sistema para o que se possa estar a passar.

### 5.2.3 Sumário de classificação de Requisitos

ID	Requisito	Classificação
1	Selecionar tipo de importação	Must Have
2	Efetuar login	Must Have
3	Importação de código	Must Have
4	Importação de código via SVN	Must Have
5	Importação de código via CVS	Must Have
6	Explorador de ficheiros	Must Have
7	Consultar histórico de importações	Must Have
8	Repetir importação	Must Have
9	Guardar credenciais de acesso	Must Have
10	Listar credenciais de acesso	Must Have
11	Remover credenciais de acesso	Must Have
12	Usar credenciais de acesso	Must Have
13	Alterar credenciais de acesso	Must Have
14	Agendar importações	Must Have
15	Agendar importações cíclicas	Must Have
16	Consultar agendamentos	Must Have
17	Cancelar agendamentos	Must Have
18	Editar agendamentos	Must Have
19	Listar plugins	Must Have
20	Importar plugins	Must Have
21	Excluir plugins	Must Have
22	Ativar plugins	Must Have
23	Ativar plugins por organização	Must Have
24	Ativar plugins para todas as organizações	Must Have
25	Desativar plugins	Must Have
26	Desativar plugins por organização	Must Have
27	Desativar plugins para todas as organizações	Must Have
31	Módulo auto desenhável	Must Have
32	Deteção automática de plugins	Must Have
33	Emissão de erros ao utilizador	Must Have
34	Listar últimas 5 exclusões de Plugins	Must Have
29	Retroceder na exclusão de Plugins	Nice to Have
30	Exportação de relatórios de vulnerabilidades	Nice to Have

Tabela 5.1: Tabela de classificação de requisitos

### 5.3 Requisitos Não-Funcionais

Esta secção, como o próprio nome indica, é relativa aos requisitos não funcionais. Estes encontram-se enumerados no decorrer da secção e acompanhados de uma justificação para o seu enquadramento no presente projeto. À semelhança dos requisitos funcionais, estes encontram-se com um ID atribuído por forma a facilitar a validação destes no capítulo 8.

É importante salientar que todos os requisitos não funcionais descritos na presente secção

pressupõem o cenário de validação descrito na secção Controlo de Qualidade 3.2.

**ID-26: Modularidade [MUST]** - A aplicação deve comportar uma arquitetura modular. O principal objetivo deste requisito é de que no futuro, a integração de novos plugins seja conseguida da forma mais transparente possível para a pessoa responsável pelo seu desenvolvimento, ou seja, o utilizador apenas terá de se preocupar em implementar as funções em que é obrigatória a sua presença no código do novo plugin, sendo este alertado caso exista alguma anomalia na sua implementação, como por exemplo, uma função obrigatória que por lapso não foi implementada, entre outros erros que se encontram definidos em detalhe na secção 7.2.3 relativa a implementação de plugins.

Outros dois aspetos igualmente importantes são: o facto deste motor de plugins, ser desenvolvido não a pensar nesta solução em específico, mas sim, numa solução mais abstrata que permite que este seja utilizado tanto no âmbito do presente projeto, como noutros projetos de âmbitos totalmente diferentes. Relativamente ao último aspeto justificativo para a utilização deste requisito, este prende-se com o facto de não existir a necessidade de desligar o módulo SIRC para efetuar alterações em determinado tipo de plugin, dado que é possível retirar de funcionamento qualquer um sem comprometer o bom funcionamento da plataforma.

**ID-27: Segurança [MUST]** - Por questões de segurança, todos os *inputs* provenientes do utilizador são validados. Esta validação inclui desde tratamento do input, à utilização de *white-lists* de funções. Este é um requisito de enorme importância, uma vez que a entidade acolhedora, é uma empresa dedicada ao desenvolvimento de produtos nesta área, como é exemplo o CodeV. Para além destes aspetos os plugins submetidos são alvo de uma análise de vulnerabilidades pelo CodeV aquando a sua importação para a plataforma.

**ID-28: Escalabilidade [MUST]** - O tempo de resposta com vários clientes em simultâneo, não pode ser superior em 2 segundos. Na secção de Controlo de Qualidade 3.2 do presente documento, pode-se verificar o cenário detalhado para validação deste requisito.

**ID-29: Robustez [MUST]** - O módulo deve ser capaz de manter o seu funcionamento normal em condições anormais. Para tal há lugar a tratamento de exceções, como por exemplo, dados de input inválidos e ligações a repositórios que estejam em baixo, são reportados como tal.

**ID-30: Usabilidade [MUST]** - A usabilidade é um atributo de qualidade dos produtos que permite aferir se uma interface com o utilizador é intuitiva. O termo é ainda empregue para aferir o conjunto de métodos destinados a melhorar a interface com determinado produto, procurando dar resposta aos seguintes pontos:

- Aprendizagem: quão fácil é para os utilizadores realizarem as suas tarefas no primeiro contacto com a interface;
- Eficiência: depois dos utilizadores se tornarem experientes na manipulação da aplicação, quão rápido conseguem realizar as suas tarefas;
- Memorização: depois de um período de ausência, quão fácil é para os utilizadores restabelecer o seu nível de produtividade;

- Satisfação: quão agradável é a utilização do sistema.

Por forma a mitigar a questão da usabilidade foi usada para o desenvolvimento deste projeto uma framework utilizada à escala mundial para o desenvolvimento de projetos de carácter web, nomeadamente o Twitter bootstrap que se encontra analisado na secção 6.4.5. A utilização desta framework permite e força de certa forma a resposta aos aspetos mencionados anteriormente.

### 5.3.1 Alterações na classificação de requisitos

Devido a interesses do ponto de vista do negócio por parte da entidade acolhedora, a classificação do requisito **Exportação de relatórios de vulnerabilidades** passou de *Nice to Have* para *Must Have*. Consequentemente, houve lugar a alterações no plano de desenvolvimento e as quais se encontram especificadas no capítulo 4 relativo ao Planeamento.

## 5.4 Conclusão

Os requisitos funcionais descritos nesta secção foram pensados e classificados em duas vertentes, requisitos *Must Have* e requisitos *Nice to Have*. Os requisitos relativos ao primeiro tipo, são prioritários na sua implementação enquanto que os do segundo tipo são descartáveis quando se concluiu não existir margem temporal para a sua implementação. No grupo dos *Nice to Have*, houve lugar uma priorização na sua implementação, onde o requisito *Exportação de relatórios de vulnerabilidades* foi considerado o de maior importância, seguido do *Listar últimas 5 exclusões de Plugins* e *Retroceder na exclusão de Plugins*. Contudo, houve necessidade de alterações conforme explanado no decorrer desta secção, passando o requisito *Exportação de relatórios de vulnerabilidades* de *Nice to Have* para *Must Have*.

## Capítulo 6

# Arquitetura do Sistema

No presente capítulo é abordada a arquitetura do SIRC. A organização deste consiste numa primeira secção onde se ilustra a visão geral da arquitetura, uma segunda secção que aborda a estrutura interna do SIRC, uma terceira respetiva a base de dados e uma quarta onde se abordam as tecnologias usadas.

### 6.1 Visão Geral

Como foi referido, o SIRC é um módulo que integra o CodeV, enriquecendo-o com a possibilidade de importar código fonte para análise de vulnerabilidades a partir de diversos tipos de repositórios, assim como a exportação de relatórios de vulnerabilidades.

A imagem da figura 6.1 pretende ilustrar a visão geral do módulo enquanto parte integrante do CodeV e a interação que este mantém com outros módulos, tanto dentro como fora do sistema.

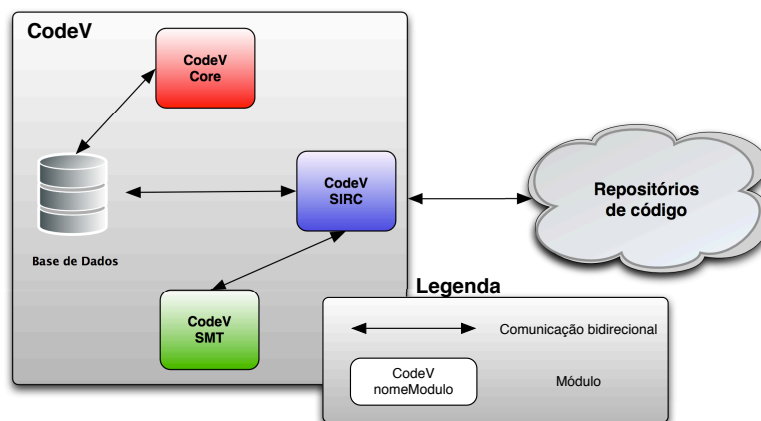


Figura 6.1: Visão geral.

## Descrição dos módulos:

- **Base de dados**

A base de dados é responsável por manter toda a informação guardada de uma forma persistente. Esta encontra-se assente num motor de base de dados *PostgreSQL* 6.4.6 e o esquema idealizado encontra-se detalhado na secção 6.3. É importante salientar que a base de dados é o meio de comunicação entre o SIRC e o *CodeV Core*, dado que estes não comunicam diretamente. Mais concretamente, o *CodeV Core* encontra-se monitorizar a base de dados por forma a identificar novas análises de código que devam ocorrer, sendo o SIRC responsável pela gestão destas análises na base de dados.

- **CodeV Core**

Este módulo é responsável pela análise de vulnerabilidades aos códigos fonte submetidos. Em termos de funcionamento, este módulo vai verificando na *Base de dados PostgreSQL* se existem novas importações de código que devam ser analisadas. No caso em que seja detetada alguma, este módulo procede à sua análise, emitindo no final um relatório que é inserido na base de dados.

- **Security Management Tool**

Este módulo representa toda a interface gráfica do CodeV. Esta é uma interface web desenvolvida PHP, que gere toda a lógica de interações que um utilizador tem à disposição ao trabalhar com o CodeV. É com este módulo que o SIRC se integra e auto desenha, como já foi referido no capítulo anterior referente à *Análise de Requisitos*.

- **SIRC - Sistema de interação com repositórios de código**

O SIRC é o módulo que foi desenvolvido no âmbito da presente dissertação. Como se pode verificar pela figura 6.1, este módulo interage com diversos repositórios de código e com a base da dados para diversas operações. Apesar deste módulo não ter uma interação direta com o *CodeV Core*, esta existe, tendo a base de dados com intermediária.

Por cada vez que um novo código é importado com sucesso, o SIRC adiciona uma nova entrada na tabela de submissões da base de dados, que posteriormente o *CodeV Core* deteta, procedendo à análise do código em questão. É importante referir que a base da dados não é apenas usada pelo SIRC para gestão de importações de código, mas também para guardar informação pertinente ao seu funcionamento, nomeadamente informação relativa a gestão de *plugins*, gestão de credenciais de acesso e gestão de agendamentos.

Analisando ainda a figura 6.1, é visível que existe interação entre o SIRC e o *Security Management Tool* que ainda não foi abordada. Esta interação existe, porque um dos principais requisitos é que o SIRC seja auto desenhável, ou seja, como já foi referido no capítulo anterior é necessário que este se integre com lugar a alterações mínimas de ambas as partes e proceda ao desenho de toda a interface necessária ao seu funcionamento de um forma autónoma. Apesar dos módulos se integrarem mutuamente, na prática nunca deixam de ser dois módulos distintos que apenas comunicam entre si para partilhar informação relevante ao seu funcionamento.

## 6.2 Estrutura Interna

Nesta secção são abordados aspetos acerca da estrutura interna. A imagem da figura 6.2 ilustra os diferentes módulos constituintes do SIRC assim como as interações existentes entre estes.

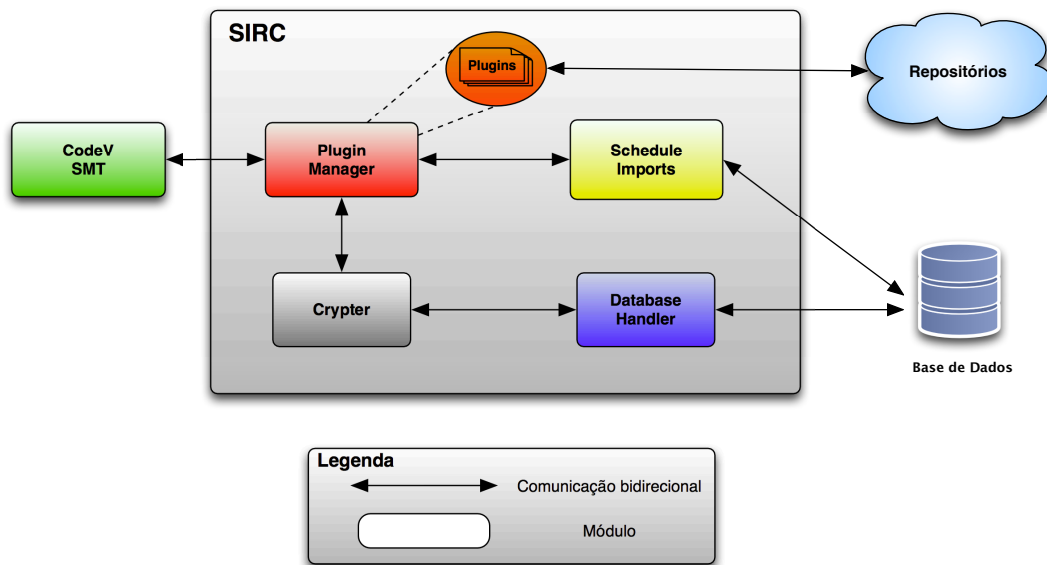


Figura 6.2: Diagrama da estrutura interna do SIRC.

### 6.2.1 Descrição dos módulos:

- **Plugin Manager**

Toda a lógica do módulo fica à responsabilidade desta entidade. É nela que se encontram implementadas funcionalidades fulcrais como é o exemplo do auto-desenho de plugins bem como a sua auto-deteção, gestão de plugins, gestão de credenciais e gestão de agendamentos.

É importante salientar que esta é a única entidade que fica exposta ao exterior, estando esta encarregue de receber pedidos (via *http post*), processá-los e enviar o output gerado de volta para a interface web do módulo *Security Management Tool*.

- **Schedule Imports**

O módulo *Schedule Imports* é responsável por monitorizar a base de dados por forma a identificar agendamentos de importações que devam ocorrer. Esta monitorização é efetuada de minuto a minuto e quando são detetados agendamentos de importações que devam ocorrer, é enviado um pedido de importação de código ao *PluginManager* com toda a informação necessária para que esta ocorra.

- **Crypter**

Este módulo é responsável pela encriptação de desencriptação de dados considerados sensíveis, nomeadamente, credenciais de acesso a repositórios. Quando um utilizador pretender guardar credenciais de acesso para utilizações futuras, estas são encriptadas e guardadas na base de dados. Desta forma pretende-se que caso exista algum acesso não autorizado a base de dados, os autores desse acesso não consigam obter os dados sensíveis na sua forma "legível". No capítulo 7 relativo a Implementação encontra-se uma descrição mais pormenorizada acerca do funcionamento deste módulo.

- **Database Handler**

O *Database Handler* serve de intermediário na comunicação com a base de dados. Este contém todas as funções que permitem efetuar operações sobre esta, bem como toda a validação de *inputs* provenientes do utilizador, por forma a minimizar o risco de aparecimento de vulnerabilidades.

## 6.2.2 Estrutura de Plugins

Nesta secção são abordados aspetos acerca da estrutura de plugins que é gerida pelo *Plugin-Manager*. A imagem da figura 6.3 ilustra a hierarquia presente na estrutura de plugins.

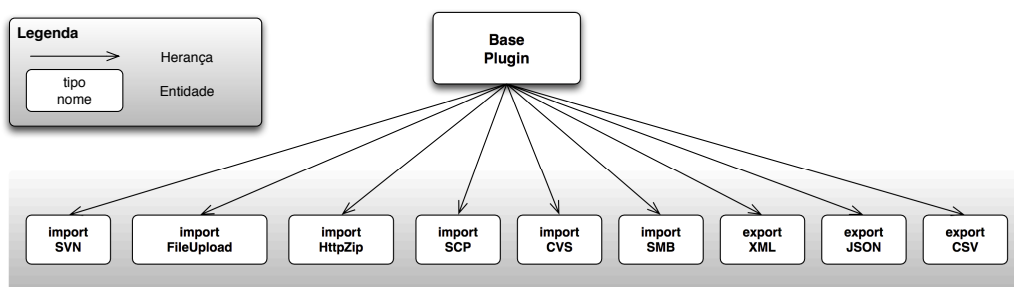


Figura 6.3: Diagrama da estrutura de plugins do SIRC.

### Descrição das entidades:

- **BasePlugin**

Esta é a entidade base de um plugin onde se definem as funcionalidades básicas que estes devem conter.

Esta entidade encontra-se definida como uma classe PHP e contém funcionalidades que *a priori* se sabe que têm de estar presentes nos plugins, como por exemplo, as funções necessárias ao auto desenho de interfaces, funções para a obtenção de código fonte dos repositórios, funções para inserção de novas submissões de código e funções para a criação de novos agendamentos. A pessoa responsável pelo desenvolvimento de novos plugins para a plataforma, ao *estender* esta classe, tem desde logo à disposição todas estas funcionalidades. É importante salientar que para o desenvolvimento de novos plugins é mandatário que estes *estendam* esta classe, caso contrário não podem ser utilizados na plataforma. Esta é uma das regras básicas para a importação de novos plugins, sendo que no capítulo 7 relativo a implementação se encontram todas as regras a respeitar no desenvolvimento de novos plugins.

- **import SVN, import SMB, import SCP, import HttpZip, import FileUpload, import CVS, export CSV, export XML, export JSON**

Estas entidades são classes PHP e representam vários os vários plugins da plataforma. Como se pode verificar, existem plugins de importação, os quais comunicam com repositórios de código e plugin de exportação, os quais permitem exportar relatórios de vulnerabilidades em diferentes formatos. Como já foi introduzido, estas classes devem *estender*



a classe *Plugin* com o objetivo de herdar tanto funções de presença obrigatória bem como outras funcionalidades que possam ser pertinentes.

Ao se desenvolver um *plugin* deve-se não só estender a classe *BasePlugin*, mas também fazer *override* às funções de presença obrigatória, caso contrário a importação do plugin em desenvolvimento falhará devido às verificações que são efetuadas aquando a sua importação.

## 6.3 Base de Dados PostgreSQL

A base de dados que se encontra representada na figura 6.1 é utilizada por toda a plataforma integrante do CodeV. Contudo, num universo de vinte e uma tabelas, o módulo SIRC faz uso de quatro, as quais passam a ser descritas a baixo. O modelo físico da base de dados completa pode ser consultada no capítulo B dos apêndices.

- **tbl\_plugin**

Esta tabela contém os plugins existentes no módulo, independentemente se estes estão ativados ou desativados. Cada plugin é identificado por um id auto-incremental aquando da importação de um novo, dispondo ainda de uma coluna que representa o nome que lhe foi atribuído. Existe um campo que especifica o tipo de plugin e que no contexto do presente projeto resume-se a plugins do tipo *importação* (plugins para importação de código fonte) e *exportação* (plugins de exportação de relatórios de vulnerabilidades). O campo *native* pretende distinguir o conjunto de plugins "base" desenvolvidos no âmbito do presente projeto, de plugins que possam ser desenvolvidos no futuro a pedido de clientes da entidade acolhedora.

- **tbl\_organization\_has\_plugins**

O conteúdo desta tabela representa todos os plugins ativados e desativados de uma empresa. Cada entrada é representada por id auto-incremental, um booleano que indica o estado do plugin (*false para desativado e a true para ativado*), um id relativo ao plugin e um id relativo à organização.

- **tbl\_user\_plugin\_credentials**

Esta tabela permite armazenar informação relativa a credenciais de acesso a um repositório que o utilizador decida guardar para usar futuramente. Cada entrada da tabela é identificada por id auto-incremental. Existe ainda uma chave estrangeira relativa a entradas na tabela *tbl\_plugin*, sendo que determinado conjunto de credenciais é respetivo a um plugin (tipo de importação), existe um id de utilizador porque as credenciais pertencem a um utilizador. A coluna *user\_plugin\_credentials\_configs* conterá uma *string* encriptada com as informações necessárias para se efetuar a autenticação perante um repositório, sendo que apenas o próprio utilizador terá acesso a estas.

O conteúdo desta tabela é inteiramente controlado pelos utilizadores, mais concretamente, um utilizador é livre de guardar, apagar ou editar as suas credenciais.

O principal objetivo desta tabela foi proporcionar um sistema de *autocomplete* nos campos de autenticação, para evitar que um utilizador o faça repetidamente quando se encontra a aceder várias vezes ao mesmo repositório.

- **tbl\_submission**

De cada vez que um utilizador efetua uma importação de código ou um agendamento, é criada uma nova entrada nesta tabela. É importante salientar que o campo *submission\_schedule\_configs* é responsável por albergar a informação necessária para que um agendamento ocorra, como por exemplo, credenciais de acesso ao repositório onde se encontra o código fonte a importar. Esta informação encontra-se sob a forma de uma *string* encriptada. Além destes aspetos, esta tabela suporta funcionalidades como, agendamentos cíclicos e consulta do histórico de importações.

## 6.4 Especificação das tecnologias usadas

Nesta secção são apresentadas as diversas tecnologias que integram o projeto SIRC, fornecendo-se uma descrição do sistema operativo Linux, das linguagens PHP e javascript, da biblioteca JQuery e da *front-end toolkit* Bootstrap.

### 6.4.1 Sistema Operativo

O sistema operativo que irá suportar o CodeV, será o OpenBSD. Este é um sistema operativo multi-plataforma, com licença BSD[30], baseado no sistema BSD UNIX e exclusivamente pensado e desenhado tendo em vista garantir a segurança de um sistema. As métricas estudadas e que levaram a concluir que este sistema garantia um bom suporte ao CodeV, foram a sua estabilidade, implementação e técnica de segurança.

Baseado num Unix BSD 4.4, uma das grandes vantagens que apresenta é a sua estabilidade, tanto a nível operacional como ao nível da sua arquitetura e desenho. Ao nível operacional esta estabilidade é garantida através de constantes auditorias de código e apenas o hardware selecionado pela equipa de trabalho, é suportado com drivers desenvolvidos pelos próprios colaboradores. Ao nível de desenho e arquitetura, a estabilidade verifica-se no facto de que o sistema, ao contrário do que acontece com outros não está constantemente a ser redesenhado, tendo uma política conservadora na implementação de novas funcionalidades e adição de novos pacotes de software.[31]

O código desenvolvido para este sistema operativo, é sempre sujeito a rigorosas regras e condutas de desenvolvimento, como a definição de utilização de código seguro, a realização de auditorias de código regulares, efetuadas por equipas especializadas e dedicadas a esse trabalho, com políticas de segurança muito bem definidas e testadas.[32]

Durante o processo de revisão de código por parte dos programadores do OpenBSD, é feito um esforço de inovação do mesmo, através de novas técnicas que visam oferecer mecanismos de proteção contra vulnerabilidades de sistema operativo.[31]

### 6.4.2 PHP

O PHP sucede de um produto mais antigo, chamado PHP/FI. PHP/FI foi criado por Rasmus Lerdorf em 1995, inicialmente como simples scripts Perl como estatísticas de acesso para o seu currículo online. Ele nomeou esta série de script de *Personal Home Page Tools*. Como mais funcionalidades foram requeridas, Rasmus escreveu uma implementação C muito maior, que era capaz de comunicar com base de dados, e possibilitava aos utilizadores desenvolver simples aplicativos dinâmicos para Web. Rasmus resolveu disponibilizar o código fonte do PHP/FI para que todos pudessem ver, e também usá-lo, bem como fixar bugs e melhorar o código. O PHP/FI, que significa *Personal Home Page/Forms Interpreter*, incluía algumas funcionalidades básicas do PHP que nós conhecemos hoje. Ele usava variáveis no estilo Perl, interpretação

automática de variáveis vindas de formulário e sintaxe embutida no HTML. A sua própria sintaxe era similar a do Perl, porém muito mais limitada, simples, e um pouco inconsistente. Em 1997, o PHP/FI 2.0, a segunda versão da implementação C, obteve milhares de usuários ao redor do mundo (estimado), com aproximadamente 50,000 domínios reportando que tinha PHP/FI 2.0 instalado, angariando 1% dos domínios da Internet. Enquanto isto milhares de pessoas contribuíam com pequenos códigos para o projeto, e ainda assim o PHP/FI 2.0 foi oficialmente lançado somente em Novembro de 1997, após perder a maior parte de sua vida em versões *beta*. Ele foi rapidamente substituído pelos alphas do PHP 3.0. Ao logo do tempo foram lançadas novas versões do PHP, até que em Julho de 2004 surgiu o PHP 5, sendo que atualmente se encontrava na sua versão 5.4 que é a utilizada no âmbito deste projeto.

### 6.4.3 JavaScript

O JavaScript é uma linguagem de programação. Foi desenvolvido pela *Sun Microsystems* com a intenção de criar um método de manipulação de páginas web no próprio computador do utilizador, sem que houvesse a necessidade de um *refresh* à página. O primeiro browser a suportar o JavaScript foi o *Netscape 2.0B3*. O *Internet Explorer* nunca suportou o JavaScript oficial, ao invés disso, suporta a sua própria versão chamada de *Jscript*. O *Jscript* tem uma compatibilidade acima de 99% com o JavaScript, tanto que a maioria das pessoas não se apercebe de que não se trata realmente do JavaScript. Por vezes vão surgindo alguns problemas de compatibilidade, principalmente devido ao facto da *Microsoft* não seguir certos *standards*. Em termos de segurança, o JavaScript tem sofrido, uma vez que assenta sobre um modelo em que o código é executado do lado do cliente, falhas de segurança são muitas vezes encontradas e exploradas. Por isto, ao longo do tempo muitos serviços online foram bloqueando o uso desta linguagem, contudo é uma linguagem muito popularizada e utilizada em larga escala hoje em dia.

### 6.4.4 jQuery

O jQuery[33] é uma biblioteca JavaScript *multi-browser* desenvolvida para simplificar o *scripting* de HTML. Foi desenvolvida em Janeiro de 2006 em *BarCamp NYC* por John Resig. Atualmente, o seu desenvolvimento encontra-se a cargo de uma equipa de programadores liderada por Dave Methvin, sendo usada por 55% dos 10,000 sites mais visitados do mundo, assumindo o título da biblioteca JavaScript mais popular no mundo. O jQuery é *open-source* e licenciado segundo a MIT License [34]. A sua sintaxe é desenhada com intuito de facilitar a navegação de um documento, seleccionar elementos *DOM*, criar animações, *handle* de eventos e desenvolvimento de aplicações que usem *Ajax*. O jQuery permite também a criação de plugins JavaScript, o que proporciona e facilita o desenvolvimento de páginas web dinâmicas.

### 6.4.5 Twitter Bootstrap

O Twitter Bootstrap[35] é uma *front-end toolkit* que permite um rápido desenvolvimento de aplicações web. É constituído por uma coleção de convenções CSS e HTML e usa as mais recentes técnicas presentes atualmente nos browsers para permitir uma elegante tipografia, *forms*, *buttons*, *tables*, *grids* e *navigations*. O Bootstrap nasceu no primórdios do Twitter, através da necessidade de resolver os diversos problemas de compatibilidade entre diversas bibliotecas utilizadas naquela altura. A sua primeira versão oficial teve início e, Agosto de 2011 como um projeto open-source, sendo que a Fevereiro de 2012 já era o projeto de desenvolvimento mais popular do Git-Hub.

## 6.4.6 PostgreSQL

As métricas que levaram à adoção do sistema de gestão de base de dados no presente projeto foram as seguintes:

- Licença de utilização,
- Fiabilidade e integridade de dados,
- Velocidade de utilização,
- Facilidade de gestão,
- Segurança de dados.

Assim, tendo estas métricas em consideração, foi feita uma primeira tiragem daqueles que se ofereciam como os mais adequados ao problema, MySQL[36] e PostgreSQL[29].

O primeiro, trata-se do mais popular SGBD Open source, que advém principalmente da sua portabilidade. Licenciado por uma licença GPL, disponibiliza também uma versão comercial licenciada sobre um end-user license agreement (EULA). Estes sistemas de licenciamento podem tornar-se redutores para o SIRC, visto que a licença GPL obriga a que o produto final se mantenha sobre a mesma licença, e o EULA não permite sequer a modificação ou alteração do software em si.

Surge assim a alternativa PostgreSQL, licenciado sobre a licença BSD, que permite, entre outros, que o software seja modificado e vendido sem problemas.

Apesar do MySQL ser amplamente reconhecido como mais rápido que o postgresSQL, estes estudos utilizam muitas vezes o motor MyISAM (indexed sequential access method) no MySQL, que não respeita as regras ACID<sup>1</sup>, enquanto que o PostgreSQL, satisfaz totalmente estas regras, vantagem evidente para um sistema no qual a integridade dos dados é um dos seus requisitos fundamentais[37].

Observando um estudo comparativo entre estas duas tecnologias[52], que utiliza as versões 5.1.30 do MySQL recorrendo ao motor InnoDB e à versão 8.3.7 do PostgreSQL e no qual são carregados 1024MB de dados e em que foram efetuados testes single e multi-utilizadores com 100 utilizadores em simultâneo, concluímos que os resultados ao nível de velocidade são até bastante próximos.

No que diz respeito à segurança de dados, ambos os SGBDs oferecem um vasto leque de características que vão desde a criptografia a vários métodos de autenticação[38].

## 6.4.7 Apache HTTP Server

O servidor web Apache[39], criado em 1995 por Rob McCool, representa o mais bem sucedido servidor Open Source, sendo o mais usado, como se pode verificar na figura 6.4.7.

Contudo, esta característica não foi essencial para a sua escolha. Esta foi impedida pela utilização do OpenBSD, que vem com este servidor web instalado por defeito.

A versão do Apache que vem instalada neste sistema operativo trata-se de um servidor web que foi modificado pelas equipas de desenvolvimento do OpenBSD com o intuito de lhe dar capacidades que permitam a segurança dos serviços disponibilizados.

---

<sup>1</sup>Atomicity, Consistency, Isolation, Durability

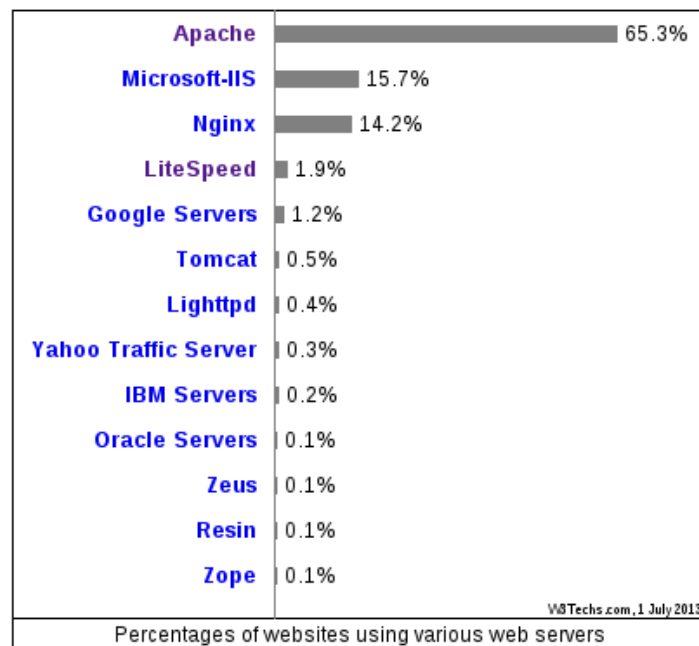


Figura 6.4: Utilização de servidores web[40].

#### 6.4.8 Apache HTTP Server Benchmarking Tool

O Apache HTTP server benchmarking tool[41], é uma ferramenta que permite medir a performance de servidores HTTP. A utilização desta é efetuada através de linha de comandos e é disponibilizada por omissão no pacote de instalação do Apache HTTP Server.

## Capítulo 7

# Especificação da Implementação

Neste capítulo encontram-se apresentadas as várias funcionalidades implementadas, assim como descrições acerca do seu desenvolvimento. Foi seguida uma abordagem *top-down*, onde primeiro se descreve os aspetos relevantes da integração do SIRC com o CodeV, seguidamente apresenta-se os módulos que o constituem assim como a comunicação existente entre estes, passando depois a uma descrição mais pormenorizada de cada um dos módulos. Posteriormente, apresentam-se detalhes acerca do interface gráfico, processo de importação de código e extensões que foram realizadas as ferramentas e bibliotecas usadas.

### 7.1 Integração com o CodeV

Como foi mencionado anteriormente, é importante que o SIRC se integre com a plataforma CodeV com lugar ao mínimo de alterações de ambos os lados. Como tal, para utilizar o motor de plugins desenvolvido no âmbito do presente projeto, é apenas necessário incluir o PluginManager e proceder a chamada de uma função que é responsável pelo desenho de toda a *user interface* inicial necessária ao funcionamento do SIRC. Abaixo encontra-se um excerto de código que o pretende ilustrar.

```
<?php
//incluir o PluginManager para que possa ser usado
include_once("../PluginManager.php")

//desenhar a interface de escolha de repositório
drawInitialInterface()
?>
```

Figura 7.1: Exemplo de utilização do PluginManager.

A interface de utilizador inicial do SIRC, resultante da inclusão do excerto de código acima no CodeV encontra-se demonstrada na imagem 7.2 abaixo, onde se pode verificar os diferentes tipos de importação de código fonte listados através de um *dropdown menu*, sendo este o ponto de partida para a maior parte das funcionalidades relativas ao ator utilizador, que já foram especificadas no capítulo 5.

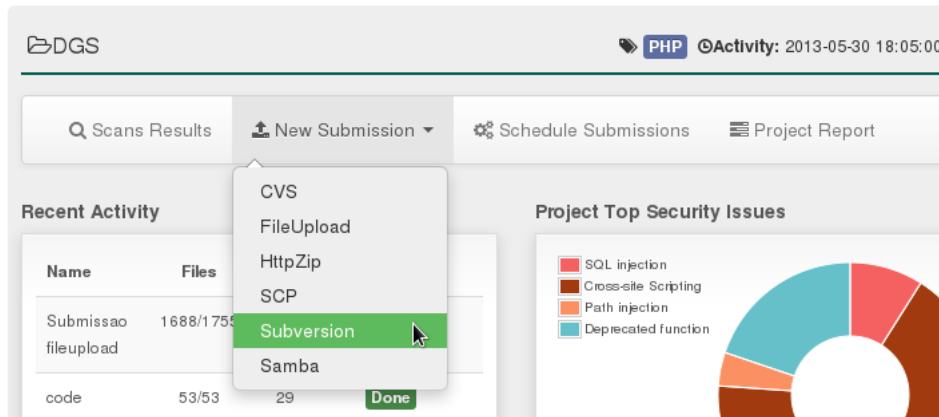


Figura 7.2: Interface inicial do SIRC.

As funcionalidades relativas ao ator administrador, são desenhadas na interface de utilizador do CodeV através de outra função do PluginManager. Esta separação deve-se apenas por motivos de organização e legibilidade do código fonte. Abaixo ilustra-se um excerto de código que exemplifica como utilizar as funcionalidades de administração especificadas no capítulo 5.

```

<?php
//incluir o PluginManager para que possa ser usado
include_once("../PluginManager.php")

//desenhar as interfaces de Admin
drawAdminInterfaces()
?>

```

Figura 7.3: Exemplo de utilização das funcionalidades de Admin do PluginManager.

Como resultado da utilização do excerto de código de fonte acima, obtém-se o ponto de partida para a maior parte das funcionalidades de Administração, como se pode verificar pela imagem 7.4 abaixo.

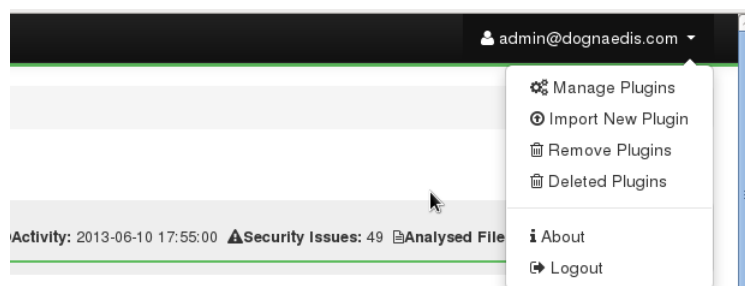


Figura 7.4: Interface do painel de Admin.

## 7.2 Comunicação entres módulos

O SIRC é constituído por cinco grandes módulos , sendo eles o *CodeV-SMT(interface web)*, o *PluginManager*, o *Schedule Imports*, o *Crypter* e o *Database Handler*. A comunicação entre os diferentes módulos encontra-se ilustrada na imagem abaixo, onde é possível observar o fluxo de mensagens existente.

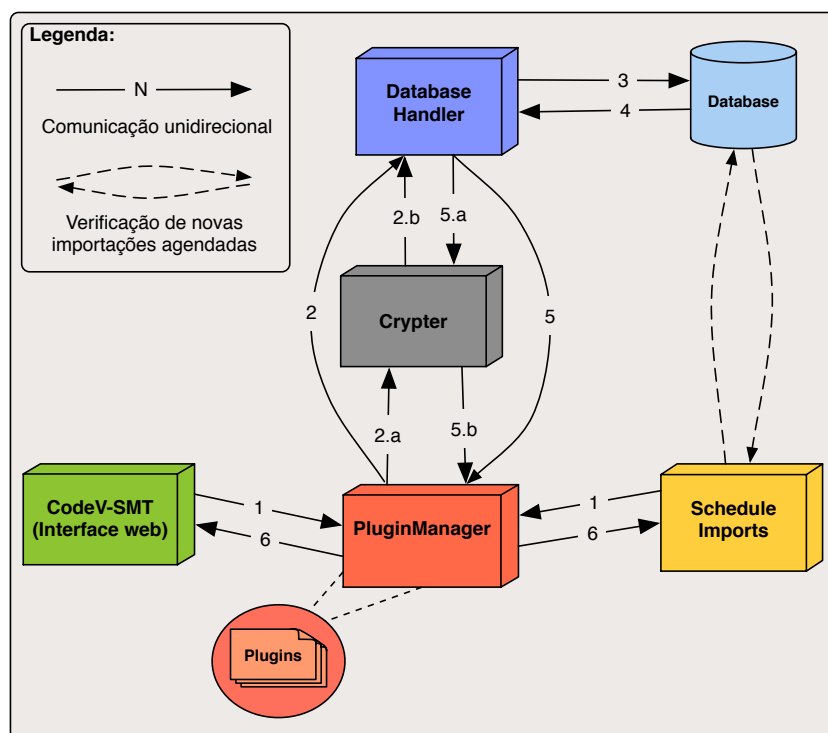


Figura 7.5: Fluxo de mensagens no SIRC.

Como se pode verificar pela figura 7.5 acima, os módulos *CodeV-SMT* e *Schedule Imports* são responsáveis pelo início de dois fluxos de mensagens com destino ao *PluginManager*. As mensagens enviadas pelo *CodeV-SMT* são resultantes de operações efetuadas na interface web através das funcionalidades que foram colocadas a disposição dos atores do sistema e que se encontram discriminadas no capítulo 5. As mensagens provenientes do módulo *Schedule Imports* correspondem a pedidos de importação de código que se encontram agendados. Este módulo encontra-se a monitorizar a base de dados e cada vez que encontra um agendamento, envia uma mensagem ao *PluginManager* com a informação necessária ao processamento da importação de código em causa. A comunicação entre o *PluginManager* e o *Database Handler*, consoante o tipo de operação, pode ou não passar pelo módulo *Crypter*.

### 7.2.1 Tipos de comunicação

Relativamente aos tipos de comunicação existentes. Existe comunicação HTTP entre os módulos *CodeV-SMT*, *Schedule Imports* e o *PluginManager*. As restantes comunicações são efetuadas com recurso a chamadas de funções.



Uma vez incluído o *PluginManager* no CodeV, este fica acessível e disponível para receber pedidos através do endereço <https://codev-smt.dognaedis.com/CodeV-2.0/PluginManager.php>. Por motivos de segurança, o tipo de pedidos adotado para a comunicação HTTP entre a interface e o *PluginManager* foi o *POST*, dado que pode haver lugar a troca de dados sensíveis e não se pretende, por exemplo, que estes fiquem guardados no histórico ou cache do browser. No caso de desenvolvimento de novos plugins, a comunicação será forçosamente efetuada por *POST*, uma vez que o módulo *Request filter* apenas valida e processa este tipo de pedidos, ignorando qualquer outro.

## 7.2.2 CodeV-SMT

Como já foi mencionado no presente capítulo, o *CodeV-SMT* representa a interface web. É aqui que se procede a inclusão do *PluginManager* para que possa ser usado de acordo com o que foi dito na secção 7.1.

## 7.2.3 PluginManager

Este é o módulo central e que assume maior relevância no presente projeto. Este é responsável por todos os aspetos pertencentes aos plugins, desde listagens de plugins disponíveis, importação dinâmica de plugins, encaminhamento de mensagens para os plugins corretos, entre outras funcionalidades já descritas no capítulo 5 de análise de requisitos. A figura 7.6 ilustra de uma forma geral o funcionamento deste módulo.

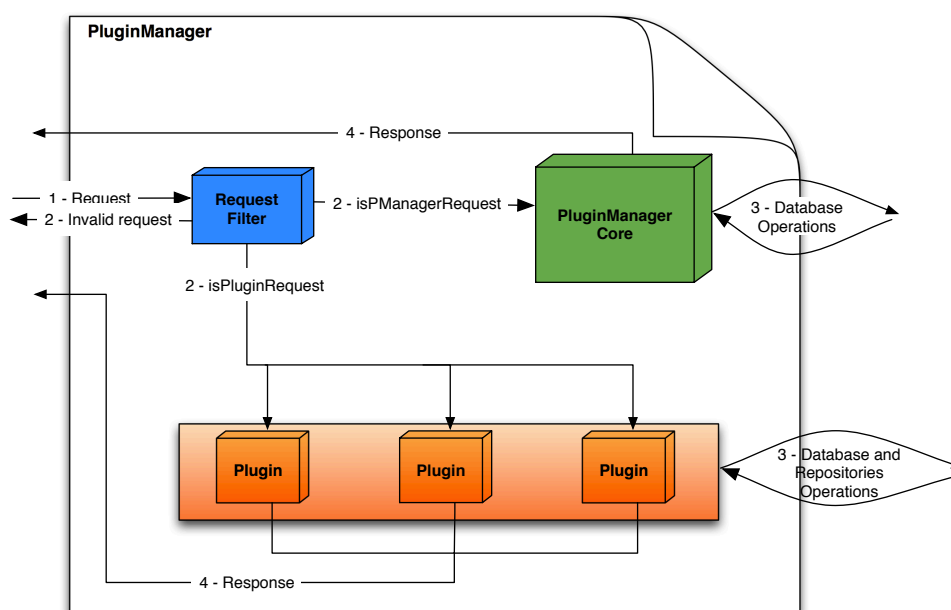


Figura 7.6: Fluxo de mensagens no *PluginManager*.

Como se pode observar pela figura 7.6 acima, o *PluginManager* encontra-se subdividido em três módulos distintos, o *Request Filter*, o *PluginManager Core* e os *Plugins*. Seguidamente apresentam-se as descrições destes módulos em mais detalhe.

## Request Filter

Este módulo é responsável por receber, validar e encaminhar todos os pedidos com destino ao *PluginManager*. Quando chega um novo pedido, verifica-se primeiramente se é destinado ao *PluginManager Core*, a um dos plugins ou se é um pedido inválido. Caso o este tenha um destino conhecidos, passa-se ao processo de validação. Este processo de validação consiste em dois fatores, a verificação de chamadas a funções válidas e a verificação de permissões do utilizador para aceder às funcionalidades pretendidas. Caso o pedido seja validado com sucesso, este prossegue para o seu destino, caso contrário o presente módulo responde ao remetente da mensagem a informar que o pedido foi considerado como inválido.

## PluginManager Core

O presente módulo é responsável pelas operações de ativação e desativação de plugins, gestão de credenciais de acesso, gestão de agendamentos e gestão de plugins. Este recebe os pedidos já validados pelo módulo *Request filter* e procede ao seu processamento. Durante o processamento há lugar a diversas operações na base de dados e as quais se encontram ilustradas na figura 7.6 como *Database Operations*. Uma vez concluído o processamento do pedidos, é enviada uma resposta ao remetente deste a informar se o foi ou não processado com sucesso.

## Plugins

Foi desenvolvida uma classe base (classe *Plugin*) para os plugins, contendo esta funções necessárias para comunicação com o *PluginManager* e que serão iguais para todos os plugins. Por uma questão de abstração e facilidade no desenvolvimento de novos plugins, estas funções ficam desde logo a disposição de quem tiver de desenvolver novos plugins, bastando para tal estender esta classe *Plugin*. Os plugins desenvolvidos terão de respeitar um conjunto de regras de completção para que o sistema os aceitar como válidos.

1. Terão de estar numa pasta dentro da diretoria *Plugins*.
2. O nome dessa pasta terá de ser igual ao nome da classe do novo plugin.
3. O nome do ficheiro do plugin terá de ser na forma *Nome.plugin.import.php*, no caso de plugins de importação, ou *Nome.plugin.export.php* no caso de plugins de exportação.
4. O novo plugin deverá estender a classe *BasePlugin*.
5. O novo plugin não deve conter vulnerabilidades.

Estas são as regras básicas para a construção de um plugin, seguindo estes passos será possível importar um plugin para o SIRC.

Devido a forma como o sistema de plugins foi criado existe uma grande liberdade na criação dos mesmos, é possível criar plugins de diversos tipos, ou seja, é possível adicionar qualquer tipo de funcionalidade que seja uma mais valia para o administrador. No contexto do presente projeto, foram desenvolvidos plugins para importação de código e plugins para exportação de relatórios de vulnerabilidades em formatos *standard*. Em mais detalhe, os plugins desenvolvidos encontram listados abaixo.

### Plugins de importação:

1. *Plugin Subversion*, que permite interagir e importar código de repositórios SVN.
2. *Plugin FileUpload*, que permite importar código através de um upload simples de ficheiros.
3. *Plugin HttpZip*, que permite importar código contido num *Zip*, *Rar* ou *Tar.gz* que esteja alojado algures na internet e acessível através de um endereço *URL*.
4. *Plugin CVS*, que permite interagir e importar código de repositórios CVS.
5. *Plugin SCP*, que permite interagir e importar código através SCP.
6. *Plugin Samba*, que permite interagir e importar código através de Samba.

### Plugins de exportação:

1. *Plugin XML*, que permite exportar um relatório no formato XML.
2. *Plugin Json*, que permite exportar um relatório no formato Json.
3. *Plugin CSV*, que permite exportar um relatório no formato CSV.

## 7.2.4 Schedule Imports

O módulo *Schedule Imports*, como foi já introduzido, encontra-se a monitorizar a base de dados por forma a identificar agendamentos de importações que devam ocorrer em determinado momento. Esta monitorização é efetuada de minuto a minuto e conseguida através de um *cron job* configurado para este efeito. Quando são detetados agendamentos de importações que devam ocorrer, é enviado um pedido de importação de código ao *PluginManager* com toda a informação necessária para que esta ocorra.

## 7.2.5 Crypter

Como se pode verificar pela figura 7.5, o fluxo de mensagens tanto passa por este módulo como passa diretamente do *PluginManager* para o *Database Handler* e vice versa, consoante o tipo de operações envolvidas. Este é responsável por cifrar e decifrar dados considerados sensíveis e que por esta razão não devem ficar armazenados em *Plain Text* na base de dados. Para que o *Crypter* proceda à encriptação ou desencriptação de dados é necessário que o *PluginManager* lhe indique qual das ações deve executar, para tal recorre-se a um parâmetro denominado *action* e o qual é usado da seguinte forma:

- *action* = 1 para **cifrar** os dados.
- *action* = 2, para **decifrar** dados.

Devido a forma como este módulo foi criado, existe ainda a possibilidade se especificar qual o mecanismo de cifra a usar, possibilitando o uso desde módulo em contextos diferentes do presente projeto. Contudo, devido ao estudo efetuado sobre cifras no capítulo 2 foi escolhida a *AES-256* e a qual é usada por defeito pelo *Crypter*.

### 7.2.6 Database Handler

O *Database Handler* estabelece a comunicação com a base de dados através da disponibilização de um conjunto de funções que permite, guardar, editar, consultar e apagar dados desta. Para minimizar o risco de aparecimento de vulnerabilidades no sistema, este módulo valida todos os dados provenientes do utilizador. Desta forma, no futuro quem desenvolver novos plugins para o SIRC não terá de se preocupar com esta questão, dado que as funções disponibilizadas pelo *Database Handler* já realizam todo o trabalho de validação.

## 7.3 Interface gráfico

Foi criado um modelo HTML para interface gráfica. No apêndice F encontram-se imagens da interface criada acompanhadas com explicações sobre os vários componentes existentes nesta. Existiram alguns detalhes que estiveram sempre presentes ao criar toda a interface gráfica, nomeadamente, deixa-la o mais simples possível, seguir os exemplos base fornecidos pelos criadores do *Twitter Bootstrap*, nunca ter páginas sobre-carregadas de informação quer visual quer textual e tentar mostrar os dados sempre com suporte visual, como cores, imagens, balões para que se torne mais fácil a sua interpretação.

## 7.4 Processo de importação de código fonte

Devido à complexidade do processo de importação de código fonte, foi criado um diagrama de sequência onde se ilustra a troca de mensagens entre os vários intervenientes no processo.

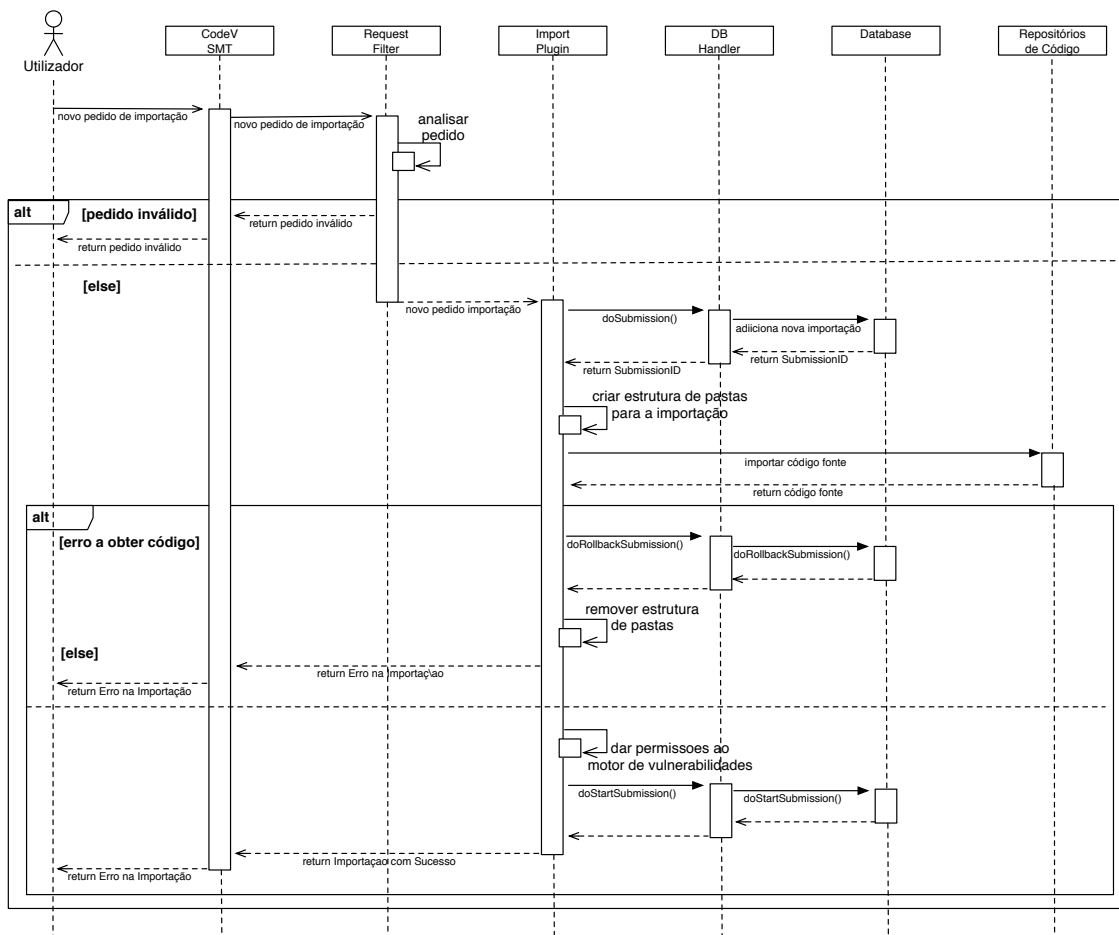


Figura 7.7: Diagrama de sequência do processo de importação.

Existem alguns tratamentos de erros que não se encontram descritos no diagrama 7.7 acima devido ao facto de tornar este demasiado complexo, como por exemplo, os acessos a base de dados. Nesta vertente foram apenas ilustrados os tratamentos de erro mais importantes, como os pedidos inválidos efetuados e erros na obtenção de código fonte dos repositórios.

Um aspeto que pode não estar claro no diagrama é a ação *dar permissões ao motor de vulnerabilidades*. Em mais detalhe, esta pretende-se com o facto de ser necessário dar permissões de leitura e escrita ao motor de análise de vulnerabilidades, na diretoria onde se encontra o código fonte que fora importado com sucesso.

### 7.4.1 Explorador de ficheiros

Para o desenvolvimento do explorador de ficheiros foi tido em conta que a listagem de conteúdo de um repositório deve ser efetuada antes de haver lugar a qualquer importação de código fonte. Desta forma o utilizador tem possibilidade de visualizar e escolher qual o conteúdo que deseja realmente importar para análise, tornando-se o processo de importação mais flexível e principalmente mais eficiente. No desenvolvimento do explorador de ficheiros foi usada uma biblioteca *javascript*, a *DynaTree* [42], que é baseada em *jQuery* 6.4.4 e permite renderizar a estrutura de uma diretoria através de uma *string json* proveniente de um servidor. Contudo, houve necessidade de alterações nesta biblioteca devido ao facto de ter sido desenvolvida para uma versão mais antiga do *jQuery*. A imagem da figura 7.8 abaixo ilustra o aspeto gráfico do explorador de ficheiros.

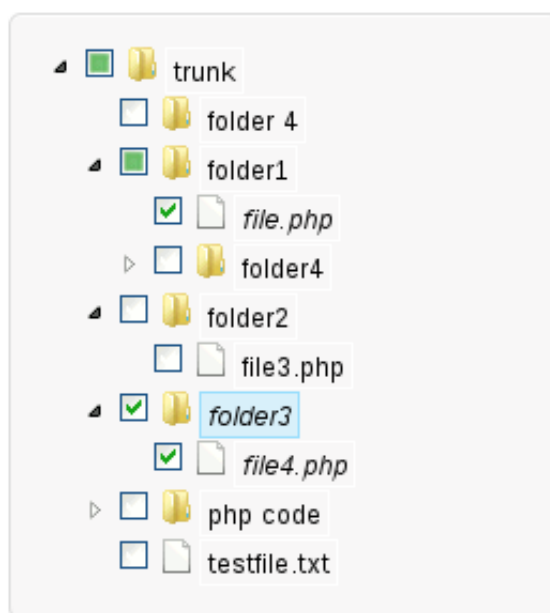


Figura 7.8: Explorador de ficheiros.

Dado que o explorador é renderizado através de uma *string json*, foram desenvolvidas funções que pretendem uniformizar a conversão de endereços de ficheiros no formato Linux/Unix (ex. */projeto/folder/file.php*) para o formato json pretendido. Estas funções encontram-se incluídas na classe base *Plugin* para que possam ser usadas no desenvolvimento de novos plugins, facilitando desta forma o uso do explorador de ficheiros.

## 7.5 Extensões às Ferramentas e bibliotecas usadas

Nesta secção apresentam-se ferramentas e bibliotecas usadas assim como as alterações necessárias nestas para que pudessem desempenhar as suas funções de acordo com o pretendido no presente projeto.

### 7.5.1 PECL SVN Extension

Para a interação com repositórios SVN optou-se pelo uso desta extensão. Apesar desta ter simplificado o processo de autenticação perante os repositórios e listagem de repositórios remotos, acabou por complicar na importação de código propriamente dita. Isto porque a funcionalidade de *export* disponibilizada apenas permite realizar um *export* de um repositório e não era isso que se pretendia. Devido a isto, houve a necessidade de contornar este problema com recurso a chamadas ao binário de sistema do SVN, tendo sido desenvolvida uma classe PHP para este mesmo efeito.

### 7.5.2 PECL SSH2 Extension

Esta extensão simplificou o processo de autenticação devido a possibilidade de manter uma sessão ativa durante a execução de diversas operações. Contudo, houve a necessidade de desenvolver um mecanismo para a listagem da estrutura de ficheiros e pastas do repositório remoto assim como o desenvolvimento de um método que permitisse um *download* recursivo de ficheiros e pastas, dado que esta extensão apenas permite *downloads* individuais.

### 7.5.3 smbclient

O processo de autenticação foi igualmente simplificado pelo uso desta biblioteca, contudo houve a necessidade de dotar esta biblioteca de funcionalidades como, listagem recursiva de ficheiros na *share* e *download* recursivo de ficheiros, uma vez que esta só permite *downloads* individuais.

## 7.6 Conclusão

No decorrer do desenvolvimento existiu sempre grande preocupação em deixar o SIRC o mais modular possível, criando várias camadas de abstração entre módulos. Isto contribuiu para que o SIRC se tornasse extensível, ou seja, é possível adicionar novos plugins com outras funcionalidades e comportamentos. Com isto, torna-se exequível a utilização do SIRC em contextos diferentes daquele que é o presente projeto.

Uma outra vantagem deste projeto prende-se com o seu sistema de plugins, que possibilita a gestão dinâmica destes, onde se pode proceder a importação, remoção, ativação e desativação de plugins sem nunca ser necessário parar o SIRC, o que é uma mais valia em termos de administração da plataforma.

## Capítulo 8

# Verificação de Funcionalidades e Atributos de Qualidade

Após o desenvolvimento da solução foi necessário avaliar as funcionalidades implementadas e qual o cumprimento dos requisitos não-funcionais, descritos no capítulo 5. Para atingir este objetivo, é apresentada de seguida uma análise de cada um dos requisitos refletindo sobre se atingiu ou não o pretendido para cada um deles.

### 8.1 Testes aos Requisitos Funcionais

Os requisitos funcionais foram validados segundo o modelo de validação especificado na secção 3.2 de controlo de qualidade. No apêndice D.1, encontram-se descritos os vários testes efetuados ao sistema durante a implementação. Os testes unitários e de integração foram omitidos devido a sua natureza, direcionando-se essencialmente para o programador (confirmação do funcionamento de pequenos pedaços de código e comunicação entre diferentes módulos), sendo demasiado extensos e desadequados a um documento de dissertação. Tendo em conta estes aspetos, os testes que se encontram documentados são testes de integração e de aceitação. É importante referir ainda que todos os requisitos funcionais foram testados pelo menos uma vez e que no final nenhum ficou a apresentar problemas, podendo-se concluir que os objetivos destes foram concluídos com sucesso.

### 8.2 Testes aos Requisitos Não-Funcionais

Os requisitos não-funcionais foram validados conforme o que foi explanado na secção 5.3. O requisitos que foram validados foram: a usabilidade, a modularidade, a robustez, a segurança e a escalabilidade. Seguidamente apresentam-se individualmente os resultados das validações efetuadas a estes cinco requisitos.

#### 8.2.1 Usabilidade

Na validação da usabilidade optou-se pela realização de inquéritos, como foi inicialmente introduzido na secção 5.3. O questionário entregue encontra-se no apêndice E e visa recolher dados sobre a primeira experiência do utilizador com a plataforma. Dado que o SIRC é considerado



módulo do *CodeV*, os resultados do inquérito refletem a avaliação da plataforma como um todo, ou seja, foi avaliado o *CodeV* já com o SIRC completamente integrado e funcional.

Name	Role	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
Client1	Pentester	3	1	1	1	1	1	5	1	1	1	65
Client2	Developer	5	2	4	1	5	1	4	1	4	2	87,5
Client3	Developer	4	2	4	2	4	4	4	2	4	3	67,5
Client4	Security Auditor	5	4	5	2	5	1	4	1	5	1	87,5
Client5	Security Consultor	1	2	1	1	5	1	5	1	1	2	65
Client6	System Administrator	5	1	5	3	5	1	5	1	5	1	95

Tabela 8.1: Resultados do questionário de usabilidade.

Com base no *System Usability Scale* criado por *John Brooke* [43], um resultado médio acima de 68 é considerado bastante positivo. Tendo em conta que o resultado médio obtido foi de 77.91, pode-se concluir que o requisito foi validado com sucesso.

É importante referir que estes resultados contam com questionários respondidos por clientes reais a nível nacional e internacional da entidade acolhedora, o que permite consolidar mais firmemente a validação do requisito.

### 8.2.2 Robustez

Os testes que levam à avaliação da robustez de um programa, podem ser comuns aqueles que são utilizados para analisar a sua escalabilidade. Para que um sistema seja robusto deve possuir mecanismos de controlo, que lhe permitam quando sob stress, reagir corretamente a quebras na rede ou outro tipo de condições consideradas como adversas no ambiente em que se encontra.

A robustez foi testada de acordo com o que foi especificado na secção 5.3, tendo-se verificado uma taxa de sucesso total no controlo de erros através dos testes de *fuzzing*. Relativamente à deteção de quebras de ligações, verificou-se um pequeno problema no acesso a repositórios CVS, que quando identificado, não voltou a dar problemas. Por tudo isto, pode-se concluir que o presente requisito foi validado com sucesso.

### 8.2.3 Segurança

Relativamente à segurança do SIRC, esta foi avaliada conforme descrito na secção 5.3. Após a análise de vulnerabilidades por parte do *CodeV* e de uma equipa especializada para o efeito, foram detetados alguns problemas, mas que atempadamente ficaram resolvidos. Devido a isto, o requisito é considerado como validado com sucesso.

### 8.2.4 Modularidade

A modularidade foi testada segundo o método de validação definido na secção 5.3. Foram importados e removidos diversos plugins de teste, onde se verificou que o funcionamento do SIRC não apresentou qualquer tipo de anomalia. Foram igualmente testados enumeras vezes os mecanismos de gestão de plugins, como a ativação e desativação, não sendo detetados nenhuns problemas no funcionamento da plataforma. Além disto, é importante referir que o facto de ser possível o desenvolvimento de plugins de diferentes tipos (não só plugins para importação de código), reforça o grau de modularidade que se atingiu. Concluí-se que o presente requisito foi validado com sucesso.

## 8.2.5 Escalabilidade

A avaliação da escalabilidade do sistema pressupõe o cenário descrito na secção 5.3. Como ferramenta para os testes, foi utilizado o Apache Benchmark [41]. Esta é uma ferramenta que é utilizada via linha de comandos e onde foram utilizados e variados os seguintes parâmetros.

- **Comando de exemplo:** `ab -C "Parâmetros do Cookie" -p "Parâmetros do Post" -n "Total de requests" -c "Requests em simultâneo"`

Como o objetivo é perceber o número máximo de clientes em simultâneo que a plataforma suporta tendo em conta as métricas especificadas, tornou-se claro que o parâmetro a variar é o `-c` que representa os requests em simultâneo. Já o número total de requests por teste foi fixado nos 10000, isto porque é um número suficientemente grande para permitir que estes sejam suficientemente longos. Os testes realizados envolveram pedidos diretamente ao *PluginManager*, onde foi especificado um *Cookie* para a autenticação e um *Post* que simula a seleção de um tipo de repositório.

Foram efetuados seis tipos de teste, onde o número de clientes em simultâneo foi, respetivamente: 1, 30, 40, 50, 60 e 70. Cada tipo de teste foi efetuado cinco vezes com o objetivo de eliminar possíveis comportamentos anormais e os resultados apresentados na tabela abaixo representam a média dos resultados obtidos.

Nº de Clientes	Tempo médio por request (ms)	Desvio Padrão (ms)	Failed requests (%)
1	43	4	0
30	359	63	0
40	479	88	0
50	599	93	0
60	721	97	0.0001
70	832	121	0.0005

Tabela 8.2: Testes de escalabilidade.

Como se pode verificar pela tabela 8.2, com 70 clientes em simultâneo obteve-se um tempo médio de resposta de 832ms e começa a haver lugar a falhas de comunicação, ainda que numa percentagem ínfima. O gráfico da figura 8.1 abaixo, demonstra o tempo médio de resposta por request relativo ao teste com 70 clientes.

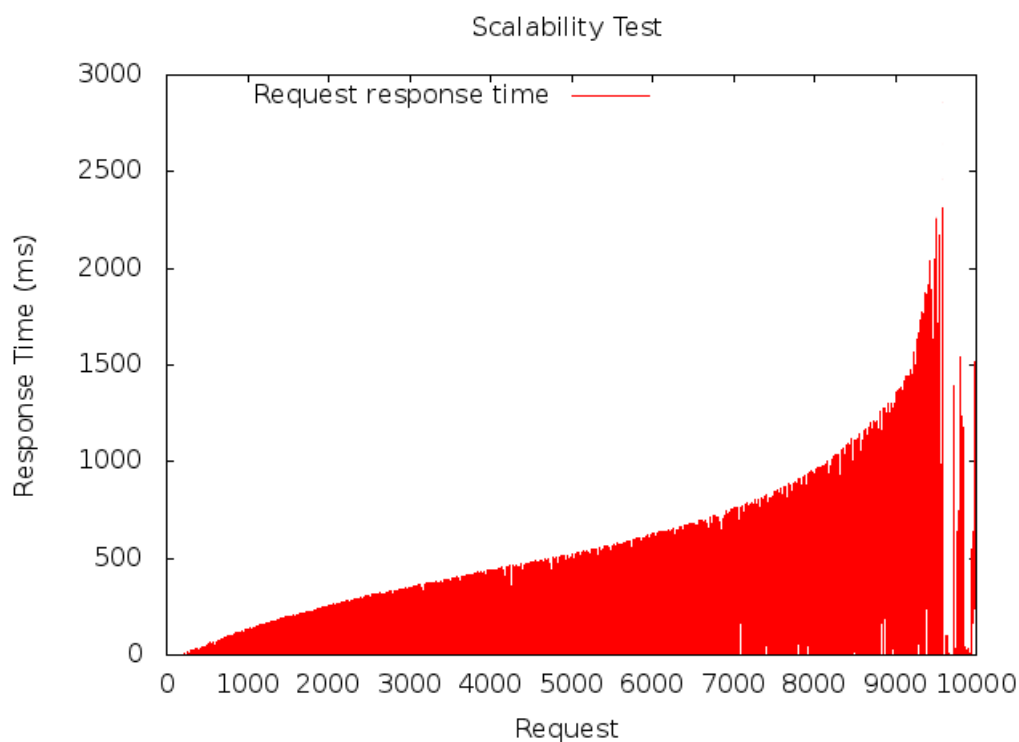


Figura 8.1: Teste de escalabilidade para 70 clientes em simultâneo.

Observando a figura 8.1, verifica-se que aproximadamente 85% dos requests foram efetuados com tempos abaixo de 1 segundo. Os restantes tiveram no máximo tempos de resposta até 2 segundos, o que ainda se encontra dentro dos limites estabelecidos para validação do requisito. Com a realização desta bateria de testes, verificou-se que o SIRC é escalável até um máximo de 70 clientes em simultâneo tendo em conta o cenário de validação apresentado.

### 8.3 Conclusão

Após o estudo de validação dos requisitos não-funcionais do sistema, foi possível chegar a conclusão que o sistema cumpre os requisitos propostos. É importante salientar ainda que os testes de escalabilidade apesar de terem sido efetuados segundo o cenário proposto, estes permitem à entidade acolhedora ter valores de referência no caso de utilização SIRC num cenário diferente do testado. Em suma, concluí-se que a fase de controlo de qualidade ocorreu com sucesso.

## Capítulo 9

# Reflexões Finais

Seguidamente são expostas as notas finais da presente dissertação, apresentando-se as principais conclusões relativas ao projeto, dando conta dos objetivos que foram atingidos assim como uma descrição do trabalho futuro a desenvolver.

### 9.1 Conclusão da Dissertação

O presente documento representa o esforço, empenho e dedicação do autor ao longo um ano letivo. Neste documento é visível de forma clara o processo de construção da solução. Este iniciou-se com a pesquisa por ferramentas que pudessem tanto auxiliar no desenvolvimento da solução como acrescentar valor a esta. De seguida definiu-se quais as metas a atingir com a mesma. Após o desenho e implementação da solução, esta foi validada segundo os requisitos idealizados. Esta foi uma tarefa que assumiu grande importância, não só porque é a forma que permite comprovar que o autor atingiu os objetivos a que se propôs, mas também porque existe uma terceira entidade (para além do autor e da Universidade de Coimbra), a *Dognædis*, que também validou esses requisitos. Esta entidade forneceu as condições necessárias para que o autor pudesse desenvolver a solução, sendo também importante referir que a solução idealizada em conjunto, vai de encontro aquilo que eram os desejos da entidade acolhedora. Dados os resultados das validações aos requisitos, o autor conclui que atingiu os objetivos a que se propôs: integração com o *CodeV* com lugar a alterações mínimas, módulo SIRC responsável por desenhar toda a interface necessária ao seu funcionamento, desenvolvimento de plugins de importação de código, desenvolvimento de plugins de exportação de vulnerabilidades, garantia da segurança na manipulação de dados sensíveis e a construção de um motor de plugins modular com grande capacidade de expansão devido à facilidade de desenvolvimento de novos plugins com funcionalidades distintas.

### 9.2 Trabalho futuro

Apesar do presente projeto preencher todos os requisitos funcionais e não-funcionais, existe margem para o seu melhoramento. Este inicialmente pode passar pelo desenvolvimento dos requisitos *Nice to have* que não foram implementados, como o *Retroceder na exclusão de plugins* e *Listagem da últimas cinco exclusões de plugins*. Numa segunda instância, devido à modularidade do SIRC será possível dotar este com novos plugins que acrescentem mais valor

a plataforma e tudo isto de uma forma facilitada como foi explanado ao longo da presente dissertação.

# Bibliografia

- [1] "Dogædis Press." <https://www.dognaedis.com/press/>.
- [2] "CodeSecure." <http://www.armorize.com/codesecure/>.
- [3] "VeraCode." <http://www.veracode.com/>.
- [4] "RIPS." <http://rips-scanner.sourceforge.net/>.
- [5] "PHP Vulnerability Hunter." <http://www.autosectools.com/>.
- [6] "IBM Security AppScan family." <http://www-01.ibm.com/software/awdtools/appscan/>.
- [7] "IBM." <http://www.ibm.com>.
- [8] "Zend Framework." <http://framework.zend.com/>.
- [9] "Strikelron." <http://www.strikeiron.com/>.
- [10] "Symfony Framework." <http://symfony.com/legacy>.
- [11] "SensioLabs." <http://sensiolabs.com/>.
- [12] "Pecl, svn extension." <http://pecl.php.net/package/svns>.
- [13] "PECL." <http://pecl.php.net/>.
- [14] "PHP, shell exec." <http://php.net/manual/en/function.shell-exec.php>.
- [15] "Pecl, ssh2 extension." <http://pecl.php.net/package/ssh2>.
- [16] "phpnetcvs." <http://code.google.com/p/phpnetcvs/>.
- [17] "Samba Web Client." <http://sourceforge.net/projects/smbwebclient/>.
- [18] "GPL License." <http://www.opensource.org/licenses/GPL-2.0>.
- [19] "smbclient." <http://smorgasbork.com/component/content/article/34-web/66-accessing-smb-servers-with-php>.
- [20] "Advanced Encryption Standard (AES)." <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
- [21] "National Security Agency." <http://www.nsa.gov/index.shtml>.
- [22] "Python." <http://www.python.org/>.
- [23] "Scrumalliance." <http://scrumalliance.org/>.

- [24] "Extreme programming." <http://www.extremeprogramming.org/>.
- [25] "Iso 9126 software quality." <http://www.sqa.net/iso9126.html>.
- [26] P. J. Ammann, *Introduction to Software Testing*. tech. rep, 2008.
- [27] R. B. Miller, *Response Time in Man-Computer Conversational Transactions*. tech. rep, 1968.
- [28] "PHP." <http://php.net/>.
- [29] "Postgresql." <http://www.postgresql.org/>.
- [30] "Open source initiative osi - the bsd license." <http://www.opensource.org/licenses/bsd-license.php>.
- [31] "Openbsd security." <http://www.openbsd.org/security.html>.
- [32] "Openbsd, manual pages." <http://www.openbsd.org/cgi-bin/man.cgi>.
- [33] "jQuery." <http://jquery.com/>.
- [34] "MIT License." <http://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt>.
- [35] "Bootstrap, from Twitter.." <http://twitter.github.com/bootstrap/>.
- [36] "Mysql." <http://www.mysql.com/>.
- [37] "A. S. P. Ltd, Mysql vs. postgresql comparison." [http://www.anchor.com.au/hosting/dedicated/mysql\\_vs\\_postgres/](http://www.anchor.com.au/hosting/dedicated/mysql_vs_postgres/).
- [38] "WikiVs, Mysql vs postgresql." [http://www.wikivs.com/wiki/MySQL\\_vs\\_PostgreSQL](http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL).
- [39] "Apache HTTP Server." <https://httpd.apache.org/>.
- [40] "Usage of web servers for websites." [http://w3techs.com/technologies/overview/web\\_server/all](http://w3techs.com/technologies/overview/web_server/all).
- [41] "Apache Benchmark." <https://httpd.apache.org/docs/2.2/programs/ab.html>.
- [42] "DynaTree." <https://code.google.com/p/dynatree/>.
- [43] J.Brooke, "SUS, a quick and dirty usability scale," tech. rep., Redhatch Consulting Ltd, 1996.
- [44] "Apache License." <http://www.opensource.org/licenses/Apache-2.0>.
- [45] "Openbsd - goals." <http://openbsd.org/goals.html>.
- [46] "Pecl, rsync extension." <http://pecl.php.net/package/rsync>.
- [47] "PHP copy." <http://php.net/manual/en/function.copy.php>.

**Parte I**

**Apêndices**



**Apêndice A**

# **Diagramas da Base de Dados**

## A.1 Base de dados - parte 1

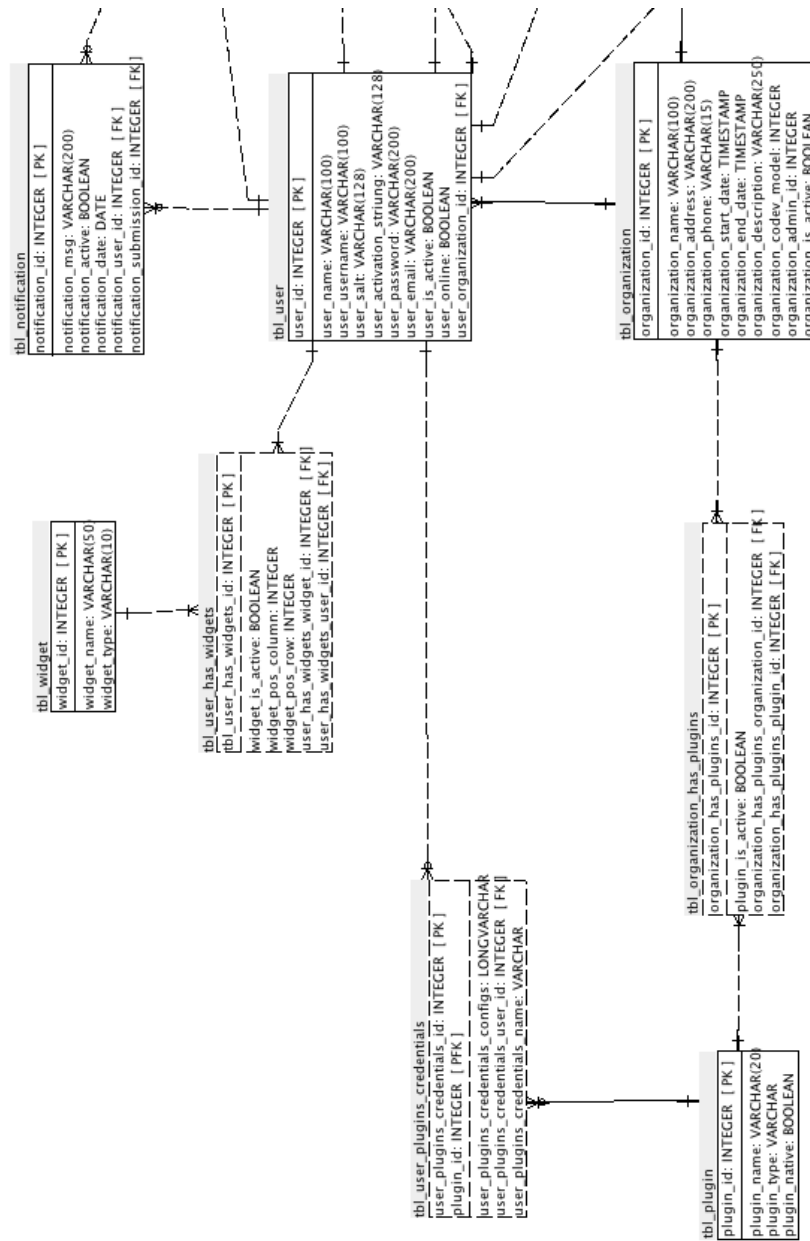


Figura A.1: Base de dados - parte 1.

## A.2 Base de datos - parte 2

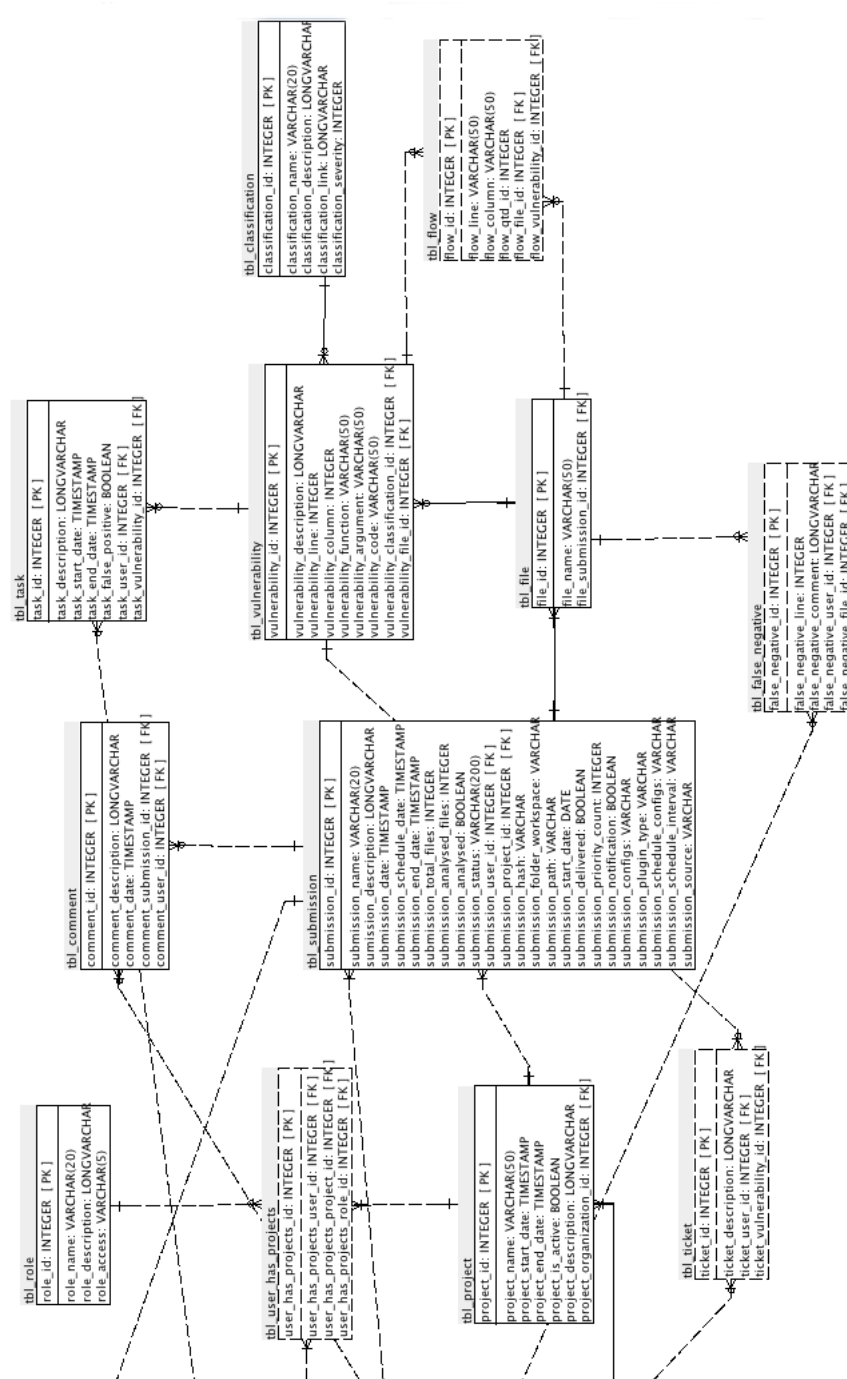


Figura A.2: Base de datos - parte 2.

## Apêndice B

# Diagramas de Gantt

## B.1 Gantt de 1º Semestre

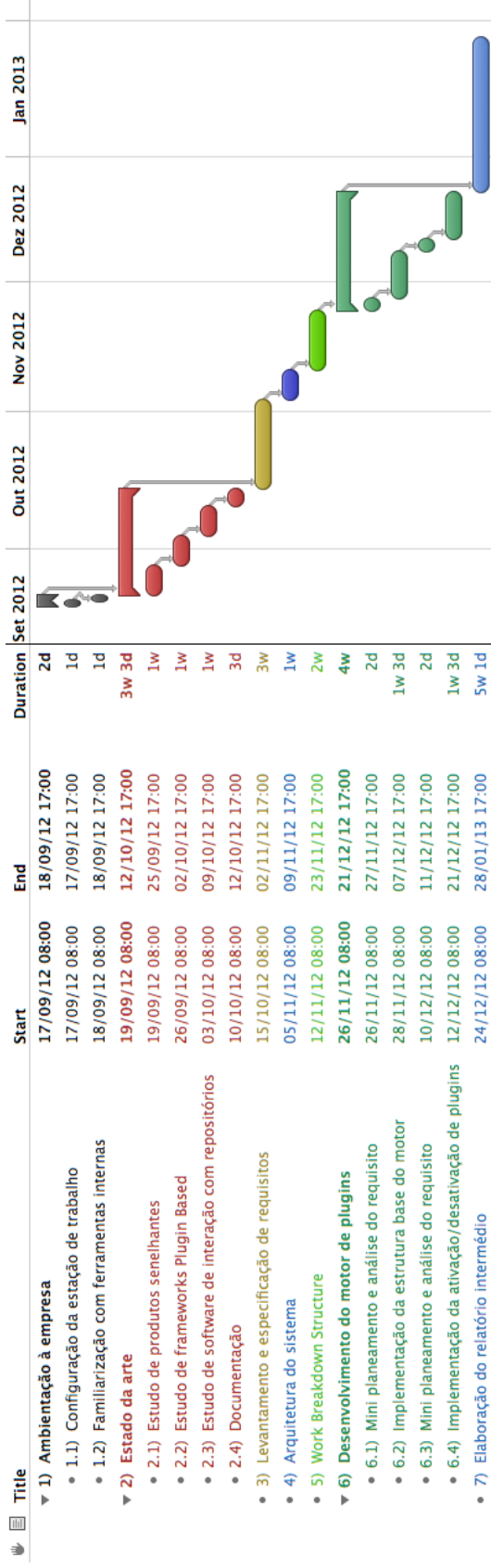


Figura B.1: Gantt – 1º semestre.

## B.2 Gantt Inicial de 2º Semestre

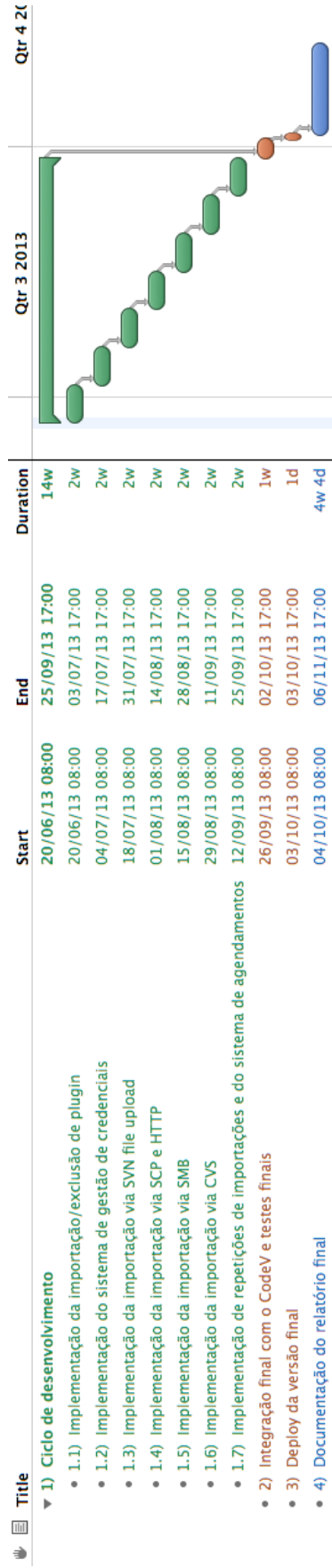


Figura B.2: Gantt Inicial - 2º semestre.

## B.3 Gantt Final de 2º Semestre



Figura B.3: Gantt Final - 2º semestre.

## Apêndice C

# Diagrama do WBS



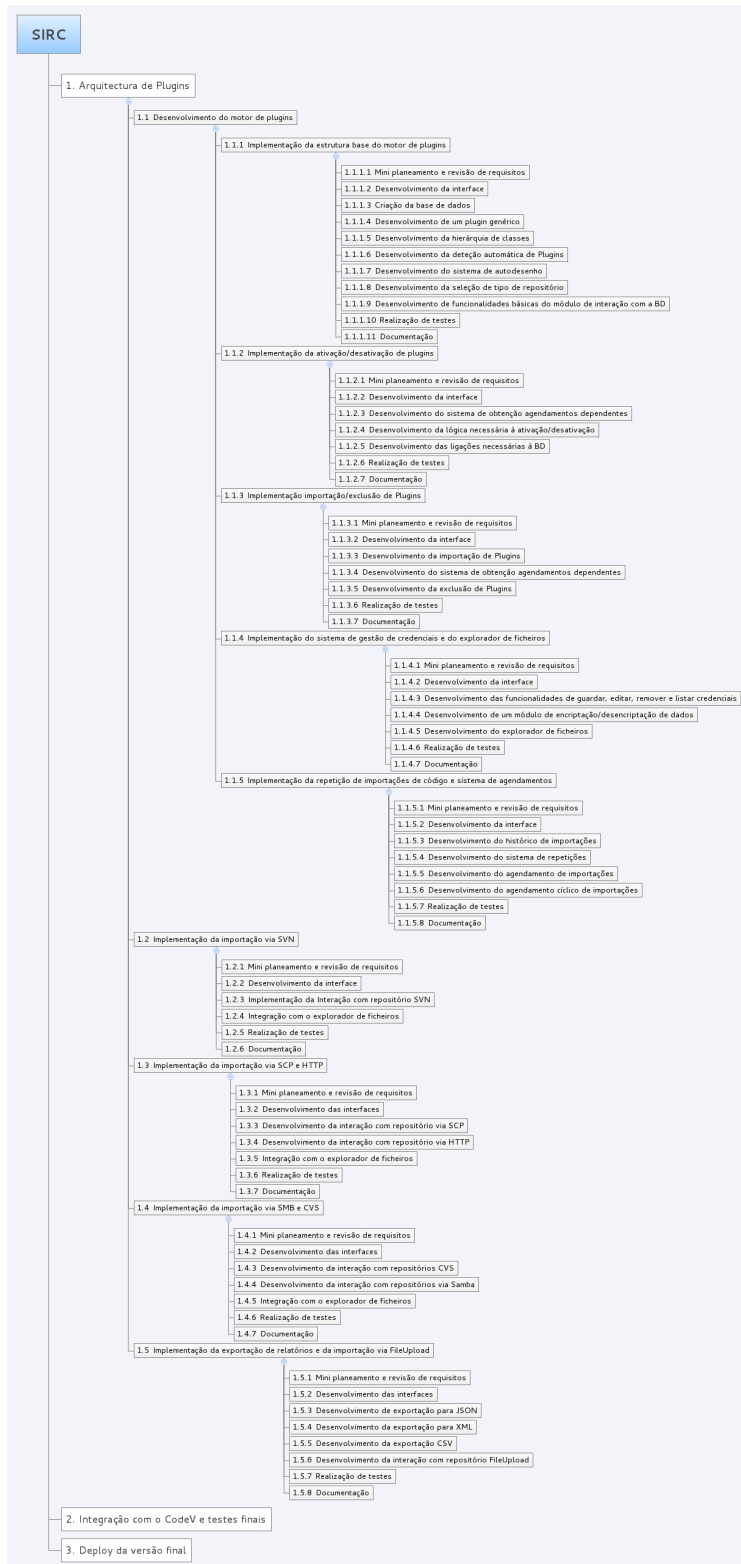


Figura C.1: Diagrama do WBS.  
ix

## Apêndice D

# Validação de Requisitos Funcionais

Os testes que se seguidamente se apresentam refletem o estado final de cada requisito, sendo esta a razão pela qual nenhum destes apresenta problemas. Contudo, durante o desenvolvimento, a ocorrência de problemas foi uma realidade e os testes em que estes foram identificados não são aqui retratados devido à sua extensão, sendo apenas representado o teste a partir do qual determinado requisito deixou de apresentar problemas.

## D.1 Testes aos Requisitos Funcionais

<b>ID-Teste:</b> 01			
<b>ID-Requisito:</b> 01-Selecionar tipo de importação			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se todos os plugins ativos são listados.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de projetos, selecionar um projeto e tentar efetuar uma submissão de código.	Surgir uma listagem dos plugins de importação que se encontram activos.	Surgiu uma listagem dos plugins de importação que se encontram activos.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 02			
<b>ID-Requisito:</b> 01-Selecionar tipo de importação			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Selecionar cada um dos plugins ativos afim de verificar se surge a respectiva interface de utilizador.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar cada um dos plugins de importação ativos.	Surgir a interface de login, relativa ao plugin de importação selecionado, numa <i>modal</i> .	Surgiu a interface de login, numa <i>modal</i> , relativa a cada plugins de importação selecionado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 03</b>			
<b>ID-Requisito: 01-Selecionar tipo de importação</b>			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se é possível selecionar qualquer um dos tipos de importação disponíveis.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar cada um dos plugins de importação ativos.	Surgir a interface de login, relativa ao plugin de importação selecionado, numa <i>modal</i> .	Surgiu a interface de login, numa <i>modal</i> , relativa a cada plugins de importação selecionado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 04</b>			
<b>ID-Requisito: 02-Efetuar login</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se o login é efetuado com sucesso em cada um dos tipos de importação.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar login em cada um dos tipos de importação com credenciais válidas.	Surgir a interface de importação de código, com ao explorador de ficheiros e campos para o programador preencher.	Surgiu a interface de login, numa <i>modal</i> , relativa a cada plugins de importação selecionado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 05</b>			
<b>ID-Requisito: 02-Efetuar login</b>			
<b>Tipo de Teste: Teste de sistema.</b>			
<b>Descrição:</b> Verificar se o ator do sistema é alertado em caso de inserção de credenciais inválidas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar login em cada um dos tipos de importação com credenciais inválidas.	Surgir um alerta a informar sobre o uso de credenciais inválidas.	Surgiu um alerta a informar sobre o uso de credenciais inválidas.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 06</b>			
<b>ID-Requisito: 02-Efetuar login</b>			
<b>Tipo de Teste: Teste de sistema.</b>			
<b>Descrição:</b> Verificar o comportamento do sistema em caso de credenciais em falta.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Fornecer credenciais de login incompletas. Ex: nome de utilizador sem palavra-chave e vice-versa.	Surgir um alerta a informar para a falta de credenciais, em cada um dos campos que seja obrigatória a sua presença.	Surgiram alertas nos campos em que se encontravam credenciais em falta.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 07</b>			
<b>ID-Requisito: 02-Efetuar login</b>			
<b>Tipo de Teste: Teste de aceitação.</b>			
<b>Descrição:</b> Verificar o correto funcionamento de todo os sistema de logins.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar logins com credenciais válidas, inválidas e com credenciais em falta.	Logins efetuados com sucesso, insucesso nos seus respetivos casos.	Logins foram efetuados com sucesso, insucesso nos seus respetivos casos.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 08</b>			
<b>ID-Requisito: 03-Importação de código</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar o comportamento do sistema em caso de não preenchimentos dos campos obrigatórios.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar uma importação de código através de cada um dos tipos disponíveis, sem preencher os campos obrigatórios.	A importação de código não deve prosseguir, devendo surgir alertas nos campos em falta.	As importações não prosseguiram, surgindo os respetivos alertas nos campos em falta.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 09</b>			
<b>ID-Requisito: 03-Importação de código</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar o comportamento do sistema em caso de importação de ficheiros inválidos.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar uma importação com tipos de ficheiros inválidos, através de HTTP e simples upload de ficheiros.	O ator deve ser informado de que o tipo de ficheiros que tenta submeter é inválido e a importação não deve prosseguir.	Surgiu o alerta esperado e as importações não prosseguiram.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 10			
<b>ID-Requisito:</b> 03-Importação de código			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se as importações através de qualquer tipo de importação funcionam correctamente e se as posteriores análises de vulnerabilidades começam após cada uma das importações.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar importações de código válidas através de cada um dos tipos de importação disponíveis.	Surgir um alerta a informar que as importações ocorreram com sucesso e a análise de vulnerabilidades ter início logo após cada uma das importações.	Surgiram os alertas a informar que as importações ocorreram com sucesso e as análises de vulnerabilidades tiveram início como o esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 11			
<b>ID-Requisito:</b> 04-Importação de código através de SVN			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a listagem de revisões e respectivos autores.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar login num repositório e percorrer a lista de revisões disponíveis.	Surgirem todas as revisões listadas com os respectivos autores.	Surgiram as revisões disponíveis e respectivos autores.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 12</b>			
<b>ID-Requisito: 04-Importação de código através de SVN</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a importação parcial de determinada revisão à escolha.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar uma revisão das disponíveis e selecionar alguns ficheiros e pastas desta.	Surgir um alerta a informar que a importação ocorreu com sucesso.	Surgiu o alerta esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 13</b>			
<b>ID-Requisito: 04-Importação de código através de SVN</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a importação completa de determinada revisão à escolha.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar uma revisão das disponíveis e selecionar a pasta de raiz desta no explorador de ficheiros.	Surgir um alerta a informar que a importação ocorreu com sucesso.	Surgiu o alerta esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 14</b>			
<b>ID-Requisito: 04-Importação de código através de SVN</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar o correto funcionamento dos filtros de revisões e autores destas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Pesquisar por revisões e autores.	Surgir as revisões consoante o tipo de filtro aplicado.	Surgiram as revisões como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			



<b>ID-Teste:</b> 15			
<b>ID-Requisito:</b> 04-Importação de código através de SVN			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar o correto funcionamento de todas as funcionalidades.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar importações parciais e completas. Usar os filtros de revisões e tags disponíveis.	As importações devem ocorrer com sucesso e os filtros devem filtrar a informação corretamente.	Decorreu tudo como o esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 16			
<b>ID-Requisito:</b> 05-Importação de código através de CVS			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a importação parcial de determinada Tag.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar uma tag e selecionar algumas pasta e ficheiros a importar.	Surgir um alerta a informar que a importação ocorreu com sucesso.	Surgiu o alerta esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 17			
<b>ID-Requisito:</b> 05-Importação de código através de CVS			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a importação completa de uma tag CVS.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar uma tag e selecionar a pasta de raiz do repositório.	Surgir um alerta a informar que a importação ocorreu com sucesso.	Surgiu o alerta esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 18</b>			
<b>ID-Requisito: 05-Importação de código através de CVS</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar funcionamento do filtro de tags.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Utilizar o filtro de tags.	Surgirem as tags corretas consoante a procura efetuada.	Surgiram as tags corretas como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 19</b>			
<b>ID-Requisito: 05-Importação de código através de CVS</b>			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar o correto funcionamento de todas as funcionalidades do requisito.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar importações parciais, completas e usar os filtros de tags.	As importações devem ocorrer com sucesso e o filtro de tags deve mostrar as tags procuradas.	Os comportamentos esperados foram observados.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 20</b>			
<b>ID-Requisito: 06- Explorador de ficheiros</b>			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a listagem de estrutura de ficheiros e pastas em qualquer um dos tipos de importação disponíveis.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Navegar pela estrutura de ficheiros e pasta apresentada no explorador.	As pastas devem abrir mostrando o seu conteúdo, devendo ser possível selecionar tanto ficheiros como pastas completas para importar.	As pasta abrem corretamente e mostram o seu conteúdo. É possível selecionar ficheiros e pastas.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 21</b>			
<b>ID-Requisito: 06-</b> Explorador de ficheiros			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar comportamento do explorador de ficheiros em caso de repositório vazio.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder a um repositório vazio.	Surgir um alerta a informar que o repositório se encontra vazio.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 22</b>			
<b>ID-Requisito: 06-</b> Explorador de ficheiros			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar o comportamento do explorador de ficheiros ao testar as suas funcionalidades.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Navegar através da estrutura de ficheiros e pastas e aceder a repositórios vazios.	As pastas abrirem corretamente e no caso de um repositório vazio deve surgir um alerta.	Os comportamentos esperados observaram-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 23</b>			
<b>ID-Requisito: 07-</b> Consultar histórico de importações			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a listagem de importações.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de <i>Scans</i> .	Surgir uma lista com a importações efetuadas até ao momento.	Surgiu a lista como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 24</b>			
<b>ID-Requisito: 07-</b> Consultar histórico de importações			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar os diversos tipos de ordenamento disponíveis.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Ordenar a lista de importações por cada um dos campos disponíveis.	A lista de importações deve ser ordenada com sucesso consoante a coluna selecionada.	As ordenações ocorreram com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 25</b>			
<b>ID-Requisito: 07-</b> Consultar histórico de importações			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar a listagem de importações e os diversos tipos de ordenamento disponíveis.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de <i>Scans</i> e efetuar ordenamentos.	Surgir a listagem das importações e as ordenações devem ser efetuadas com sucesso.	Os comportamentos esperados observaram-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 26</b>			
<b>ID-Requisito: 08-</b> Repetir importação			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se a repetição de importação ocorre sem problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de <i>Scans</i> e efetuar uma reimportação.	Surgir um alerta a informar que a reimportação ocorreu com sucesso.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 27</b>			
<b>ID-Requisito: 08-</b> Repetir importação			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar as reimportações não apresentam problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de <i>Scans</i> e efetuar várias reimportações.	Surgir alertas a informar que as reimportações ocorreram com sucesso.	Os alertas surgiram como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 28</b>			
<b>ID-Requisito: 09-</b> Guardar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se as credenciais são guardadas em todos os tipos de importação.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar cada um dos tipos de importação e guardar credenciais de acesso.	Em caso de <i>login</i> com sucesso as credenciais de acesso ficam disponíveis para futuros acessos.	As credenciais ficaram disponíveis conforme o esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 29</b>			
<b>ID-Requisito: 09-</b> Guardar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se as credenciais são guardadas em todos os tipos de importação.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar cada um dos tipos de importação e guardar credenciais de acesso.	Em caso de <i>login</i> com sucesso as credenciais de acesso ficam disponíveis para futuros acessos.	As credenciais ficaram disponíveis conforme o esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 30			
<b>ID-Requisito:</b> 10- Listar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se as credenciais previamente guardadas são listadas com sucesso.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder a cada um dos tipos de importação disponíveis.	Surgir as credenciais previamente guardadas sob a forma de um <i>dropdown menu</i> .	As credenciais ficaram disponíveis conforme o esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 31			
<b>ID-Requisito:</b> 10- Listar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se as credenciais previamente guardadas são listadas com sucesso.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder a cada um dos tipos de importação disponíveis.	Surgir as credenciais previamente guardadas sob a forma de um <i>dropdown menu</i> .	As credenciais ficaram disponíveis conforme o esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 32			
<b>ID-Requisito:</b> 11- Remover credenciais de acesso			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se as credenciais são removidas com sucesso.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder ao painel de gestão do utilizador e seleccionar <i>remove credenciais de acesso</i> .	As credenciais seleccionadas para remoção não devem constar mais na listagem.	As credenciais foram removidas com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 33			
<b>ID-Requisito:</b> 11- Remover credenciais de acesso			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se as credenciais são removidas com sucesso.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder ao painel de gestão do utilizador e seleccionar <i>remover credenciais de acesso</i> .	As credenciais seleccionadas para remoção não devem constar mais na listagem.	As credenciais foram removidas com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 34			
<b>ID-Requisito:</b> 12- Usar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar sistema de <i>autocomplete</i> ao seleccionar credenciais de acesso previamente guardadas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Seleccionar credenciais de acesso previamente guardadas.	Os diversos campos de autenticação num repositório, devem ser correctamente preenchidos com as credenciais seleccionadas.	Os campos foram devidamente preenchidos.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 35			
<b>ID-Requisito:</b> 12- Usar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar sistema de <i>autocomplete</i> ao selecionar credenciais de acesso previamente guardadas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar credenciais de acesso previamente guardadas.	Os diversos campo de autenticação num repositório, devem ser corretamente preenchidos com as credenciais selecionadas.	Os campos foram devidamente preenchidos.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 36			
<b>ID-Requisito:</b> 13- Editar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se as credenciais são devidamente alteradas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar credenciais de acesso a editar e proceder a edições de qualquer um dos campos.	As credenciais são alteradas, surgindo um alerta a informar que alteração fora efetuada com sucesso.	O comportamento esperado observou-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			



<b>ID-Teste:</b> 37			
<b>ID-Requisito:</b> 13- Editar credenciais de acesso			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se as credenciais são devidamente alteradas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Selecionar credenciais de acesso a editar e proceder a edições de qualquer um dos campos.	As credenciais são alteradas, surgindo um alerta a informar que alteração fora efetuada com sucesso.	O comportamento esperado observou-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 38			
<b>ID-Requisito:</b> 14- Agendar importações			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se as importações ocorrem na data e hora definidas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Agendar importações para momentos específicos.	As importações devem ocorrer com sucesso na data especificada.	As importações ocorreram com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 39			
<b>ID-Requisito:</b> 14- Agendar importações			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se as importações ocorrem na data especificada.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Agendar importações para momentos específicos.	As importações devem ocorrer com sucesso na data especificada.	As importações ocorreram com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 40			
<b>ID-Requisito:</b> 15- Agendar importações cíclicas			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se as importações ocorrem na data especificada e se o período de repetição é respeitado.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Agendar importações diárias, semanais e mensais.	As importações devem ocorrer com sucesso na data e hora definidas.	As importações ocorreram com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 41			
<b>ID-Requisito:</b> 15- Agendar importações cíclicas			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se as importações ocorrem na data especificada e se o período de repetição é respeitado.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Agendar importações diárias, semanais e mensais.	As importações devem ocorrer com sucesso na data e hora definidas.	As importações ocorreram com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 42			
<b>ID-Requisito:</b> 16- Consultar agendamentos			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se os agendamentos são listados sem problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de projeto, selecionar um projeto e consultar os agendamentos em vigor.	Deve surgir uma lista de agendamentos caso existam agendamentos em vigor.	A lista surgiu com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 43			
<b>ID-Requisito:</b> 16- Consultar agendamentos			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se os agendamentos são listados sem problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder à página de projeto, selecionar um projeto e consultar os agendamentos em vigor.	Deve surgir uma lista de agendamentos caso existam agendamentos em vigor.	A lista surgiu com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 44			
<b>ID-Requisito:</b> 17- Cancelar agendamentos			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se o cancelamento de agendamentos funciona sem problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder à página de projetos, selecionar um projeto, consultar os agendamentos em vigor e proceder ao cancelamento de vários.	Deve surgir um alerta por cada cancelamento a informar que a operação ocorreu com sucesso e a lista de agendamentos deve refletir estas alterações.	O esperado observou-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 45			
<b>ID-Requisito:</b> 17- Cancelar agendamentos			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se o cancelamento de agendamentos funciona sem problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de projetos, selecionar um projeto, consultar os agendamentos em vigor e proceder ao cancelamento de vários.	Deve surgir um alerta por cada cancelamento a informar que a operação ocorreu com sucesso e a lista de agendamentos deve refletir estas alterações.	O esperado observou-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 46			
<b>ID-Requisito:</b> 18- Editar agendamentos			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se a edição de agendamentos funciona sem problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à página de projetos, selecionar um projeto, consultar os agendamentos em vigor e proceder à edição de vários.	Deve surgir um alerta por cada cancelamento a informar que a operação ocorreu com sucesso e a lista de agendamentos deve refletir estas alterações.	O esperado observou-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 47			
<b>ID-Requisito:</b> 18- Editar agendamentos			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se a edição de agendamentos funciona sem problemas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder à página de projetos, selecionar um projeto, consultar os agendamentos em vigor e proceder à edição de vários.	Deve surgir um alerta por cada cancelamento a informar que a operação ocorreu com sucesso e a lista de agendamentos deve refletir estas alterações.	O esperado observou-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 48			
<b>ID-Requisito:</b> 19- Listar plugins			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se os plugins são todos listados.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder à interface de administração relativa a gestão de plugins.	Surgir uma lista com todos os plugins de importação presentes no sistema.	A lista surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 49			
<b>ID-Requisito:</b> 19- Listar plugins			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar se os plugins são todos listados.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à interface de administração relativa a gestão de plugins.	Surgir uma lista com todos os plugins de importação presentes no sistema.	A lista surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 50			
<b>ID-Requisito:</b> 20- Importar plugins			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se o sistema de deteção de erros de desenvolvimento em plugins funciona com sucesso.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Importar vários plugins com diversos erros.	Surgir um alerta a informar que os plugins contêm erros e não podem ser incluídos na plataforma.	O comportamento esperado observou-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 51			
<b>ID-Requisito:</b> 20- Importar plugins			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se o sistema de deteção de vulnerabilidades em plugins funciona como esperado.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Importar vários plugins com vulnerabilidades.	As importações devem canceladas e deve surgir um alerta com as vulnerabilidades encontradas.	O alerta surgiu como esperado e as importações foram canceladas.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 52			
<b>ID-Requisito:</b> 20- Importar plugins			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar o funcionamento do sistema de detecção de erros e do sistema de detecção de vulnerabilidades.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Importar plugins com erros de desenvolvimento e com vulnerabilidades.	Surgir alertas a informar que os plugins contêm erros de desenvolvimento ou vulnerabilidades e o respectivo cancelamento da importação.	Os comportamentos esperados observaram-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 53			
<b>ID-Requisito:</b> 21- Excluir plugins			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar o funcionamento do sistema de detecção de importações agendadas que dependam do plugin a excluir.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Excluir plugins com importações agendadas.	Surgir alertas a informar que os plugins contêm erros de desenvolvimento ou vulnerabilidades e o respectivo cancelamento da importação.	Os comportamentos esperados observaram-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 54			
<b>ID-Requisito:</b> 21- Excluir plugins			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar se os plugins são removidos com sucesso.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Proceder a exclusão de plugins.	Surgir um alerta a informar que os plugins foram excluídos com sucesso.	O alerta esperado surgiu.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 55			
<b>ID-Requisito:</b> 21- Excluir plugins			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar o sistema de alertas de importações com dependências e sistema de remoção de plugins.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar exclusões de plugins com e sem importações de código agendadas.	Surgir um alerta a informar que os plugins a excluir contém importações de código agendadas e surgir um alerta a informar que os plugins sem importações dependentes foram excluídos com sucesso.	Os comportamentos esperados verificaram-se.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 56			
<b>ID-Requisito:</b> 22- Ativar plugins; 24- Ativar para todas as organizações			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a ativação de plugins para todas as organizações.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Ativar vários plugins para todas as organizações.	Deve surgir um alerta a informar que os plugins foram ativados com sucesso.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			



<b>ID-Teste:</b> 57			
<b>ID-Requisito:</b> 22- Ativar plugins; 23- Ativar por organização			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a ativação de plugins para organizações em específico.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar ativações de plugins para empresas em específico.	Deve surgir um alerta a informar que os plugins foram ativados com sucesso.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 58			
<b>ID-Requisito:</b> 22- Ativar plugins; 23- Ativar por organização; 24- Ativar para todas as organizações			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar a ativação de plugins para todas as organizações e apenas para algumas organizações em específico.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar ativações de plugins para organizações em específico e para todas as organizações.	Deve surgir um alerta a informar que os plugins foram ativados com sucesso.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 59			
<b>ID-Requisito:</b> 25- Desativar plugins; 27- Desativar para todas as organizações			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a desativação de plugins para todas as organizações.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Desativar vários plugins para todas as organizações.	Deve surgir um alerta a informar que os plugins foram desativados com sucesso.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 60			
<b>ID-Requisito:</b> 25- Desativar plugins;26- Desativar por organização			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar a desativação de plugins para organizações em específico.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar desativações de plugins para empresas em específico.	Deve surgir um alerta a informar que os plugins foram desativados com sucesso.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 61			
<b>ID-Requisito:</b> 25- Desativar plugins;26- Desativar por organização;27- Desativar para todas as organizações			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar a desativação de plugins para todas as organizações e apenas para algumas organizações em específico.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar desativações de plugins para organizações em específico e para todas as organizações.	Deve surgir um alerta a informar que os plugins foram desativados com sucesso.	O alerta surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 62			
<b>ID-Requisito:</b> 28- Listar últimas 5 exclusões de Plugins			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar listagem das últimas 5 exclusões.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Aceder à interface de retrocesso de exclusões.	Os últimos plugins removidos, devem surgir listados até um máximo de 5.	Os plugins foram listados com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 63			
<b>ID-Requisito:</b> 28- Listar últimas 5 exclusões de Plugins			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar listagem das últimas 5 exclusões.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Acéder à interface de retrocesso de exclusões.	Os últimos plugins removidos, devem surgir listados até um máximo de 5.	Os plugins foram listados com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 64			
<b>ID-Requisito:</b> 29- Retroceder na exclusão de Plugins			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar sistema de retrocesso na exclusão de plugins.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar retrocessos em exclusões de plugins.	Surgir um alerta a informar que a operação ocorreu com sucesso.	As operações ocorreram com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 65			
<b>ID-Requisito:</b> 29- Retroceder na exclusão de Plugins			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar sistema de retrocesso na exclusão de plugins.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar retrocessos em exclusões de plugins.	Surgir um alerta a informar que a operação ocorreu com sucesso.	As operações ocorreram com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 66			
<b>ID-Requisito:</b> 30- Exportação de relatórios de vulnerabilidades.			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Exportar relatórios de vulnerabilidades nos três formatos disponíveis.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar exportações nos três formatos disponíveis.	Espera-se o download do ficheiro de relatório no formato selecionado.	Foram efetuados os <i>downloads</i> com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 67			
<b>ID-Requisito:</b> 30- Exportação de relatórios de vulnerabilidades.			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Exportar relatórios de vulnerabilidades nos três formatos disponíveis.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar exportações nos três formatos disponíveis.	Espera-se o download do ficheiro de relatório no formato selecionado.	Foram efetuados os <i>downloads</i> com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 68			
<b>ID-Requisito:</b> 31- Módulo auto desenhável.			
<b>Tipo de Teste:</b> Teste de integração.			
<b>Descrição:</b> Verificar se o módulo SIRC desenha a interface necessária ao seu funcionamento na interface do <i>CodeV</i> .			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar exportações nos três formatos disponíveis.	Espera-se o download do ficheiro de relatório no formato selecionado.	Foram efetuados os <i>downloads</i> com sucesso.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 69			
<b>ID-Requisito:</b> 31- Módulo auto desenhável.			
<b>Tipo de Teste:</b> Teste de integração.			
<b>Descrição:</b> Verificar se o desenho de todas as funcionalidades do SIRC ocorre sem problemas na interface do <i>CodeV</i> .			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Testar todas as interfaces relativas a funcionalidades do SIRC.	Devem ser desenhadas todas as interfaces sem problemas aparentes.	As interfaces desenharam-se sem problemas no <i>CodeV</i> .	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 70			
<b>ID-Requisito:</b> 32- Detecção automática de plugins.			
<b>Tipo de Teste:</b> Teste de integração.			
<b>Descrição:</b> Verificar o sistema de detecção automática de plugins.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Colocar um novo plugin sem problemas de desenvolvimento nem vulnerabilidades na pasta de plugins.	O novo plugin deve surgir como desativado no painel de gestão de plugins do Administrador.	O plugin surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste:</b> 71			
<b>ID-Requisito:</b> 32- Detecção automática de plugins.			
<b>Tipo de Teste:</b> Teste de integração.			
<b>Descrição:</b> Verificar o sistema de detecção automática de plugins.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Colocar um novo plugin com problemas de desenvolvimento nem vulnerabilidades na pasta de plugins.	O novo plugin não deve surgir no painel de gestão de plugins do Administrador.	O plugin não surgiu como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 72</b>			
<b>ID-Requisito: 33-</b> Emissão de erros de comunicação.			
<b>Tipo de Teste:</b> Teste de sistema.			
<b>Descrição:</b> Verificar todas as funcionalidades do SIRC passíveis de ocorrência de erros, provocando erros de comunicação voluntariamente, como desligar o acesso a repositórios e providenciar credencias de acesso erradas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar operações, como logins e importações.	Emissão de erros ao utilizador caso ocorra algum problema.	As emissões de erros ocorreram como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

<b>ID-Teste: 73</b>			
<b>ID-Requisito: 33-</b> Emissão de erros de comunicação.			
<b>Tipo de Teste:</b> Teste de aceitação.			
<b>Descrição:</b> Verificar todas as funcionalidades do SIRC passíveis de ocorrência de erros, provocando erros de comunicação voluntariamente, como desligar o acesso a repositórios e providenciar credencias de acesso erradas.			
<b>Ação</b>	<b>Esperado</b>	<b>Observado</b>	<b>Pass/Fail</b>
Efetuar operações, como logins e importações.	Emissão de erros ao utilizador caso ocorra algum problema.	As emissões de erros ocorreram como esperado.	✓
<b>Problemas:</b> Sem problemas relevantes.			

## Apêndice E

# Questionário ao Utilizador

No presente apêndice é apresentado o questionário que foi dado ao utilizador para aferir a usabilidade da plataforma, tendo sido distribuído aos utilizadores aquando o seu primeiro contacto com esta.

O questionário é baseado na *System Usability Scale* criada por John Brooke [43] e o seu objetivo é fazer uma avaliação global da usabilidade de um sistema. É constituída por 10 perguntas apresentadas na figura E.1 e para a sua avaliação dever-se-ão seguir as regras do *sus*, obtendo-se uma pontuação de 0 a 100 para o sistema (quanto mais usável for, maior a sua pontuação).

## Usability Questionnaire

Company:

Role:

Date:



Please respond to the next questions were 1 is Strongly Disagree and 5 is Strongly Agree. Please note that this set of questions is only related with CodeV usability.

	1	2	3	4	5
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technical person to be able to use this system.					
5. I found the various functions in this system Were well integrated.					
6. I thought there was too much inconsistency In this system.					
7. I would imagine that most people would learn to Use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. 0. I needed to learn a lot of things before I could get going with this system.					

Thank you for your time.

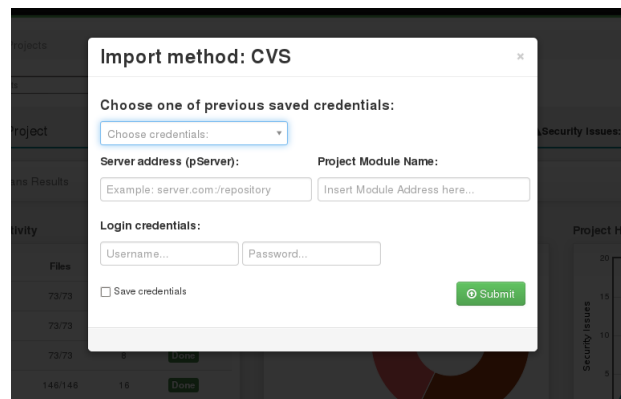
Figura E.1: Questionário de Usabilidade.



## Apêndice F

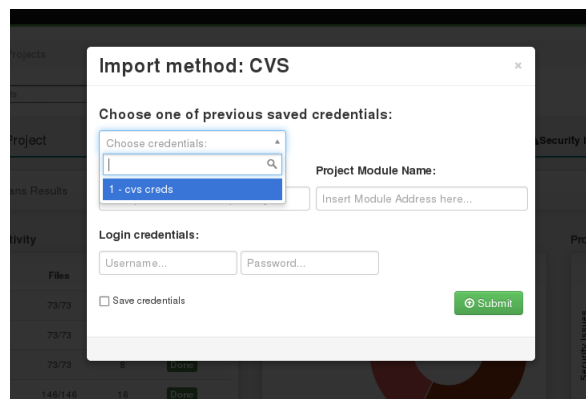
# Interface gráfica

Nesta secção encontraram-se imagens retiradas da interface gráfica do SIRC com vista a demonstrar o resultado da implementação dos principais requisitos funcionais.



The screenshot shows a dialog box titled "Import method: CVS". It contains a dropdown menu for "Choose one of previous saved credentials:" with the text "Choose credentials:". Below this are two input fields: "Server address (pServer):" with a placeholder "Example: server.com/repository" and "Project Module Name:" with a placeholder "Insert Module Address here...". Underneath are "Login credentials:" fields for "Username..." and "Password...". At the bottom left is a checkbox "Save credentials" and at the bottom right is a green "Submit" button.

Figura F.1: Importação via CVS - Login.



The screenshot shows the same "Import method: CVS" dialog box. The "Choose one of previous saved credentials:" dropdown menu is open, displaying a search bar and a list item "1 - cvs creds". The rest of the form, including the server address, project module name, login credentials, and submit button, remains visible in the background.

Figura F.2: Importação via CVS - Sistema de credenciais.

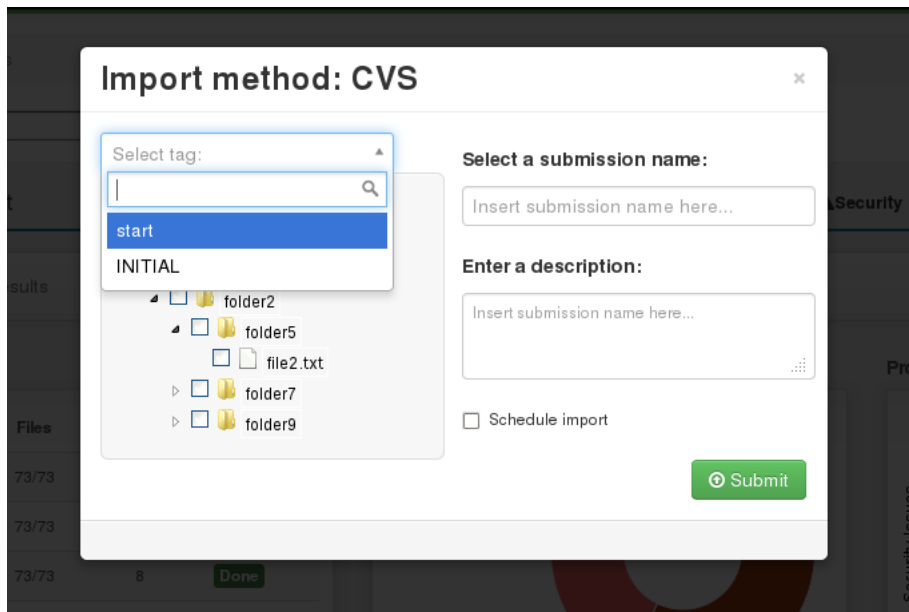


Figura F.3: Importação via CVS - Menu principal.

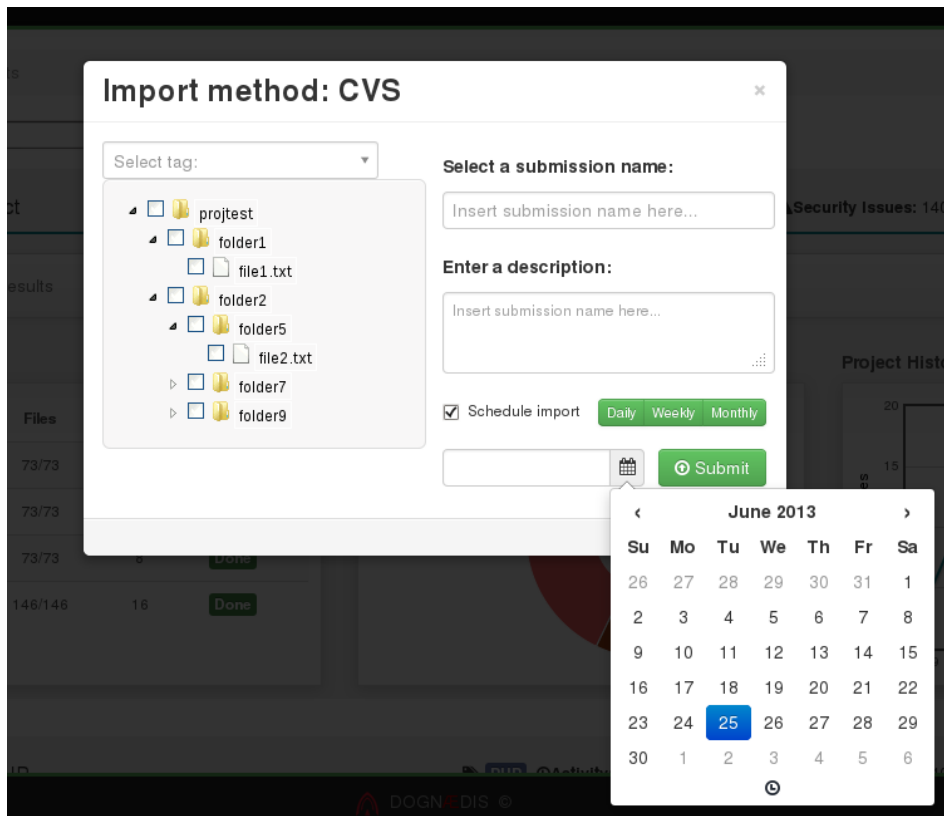


Figura F.4: Importação via CVS - Menu principal com Agendamentos.

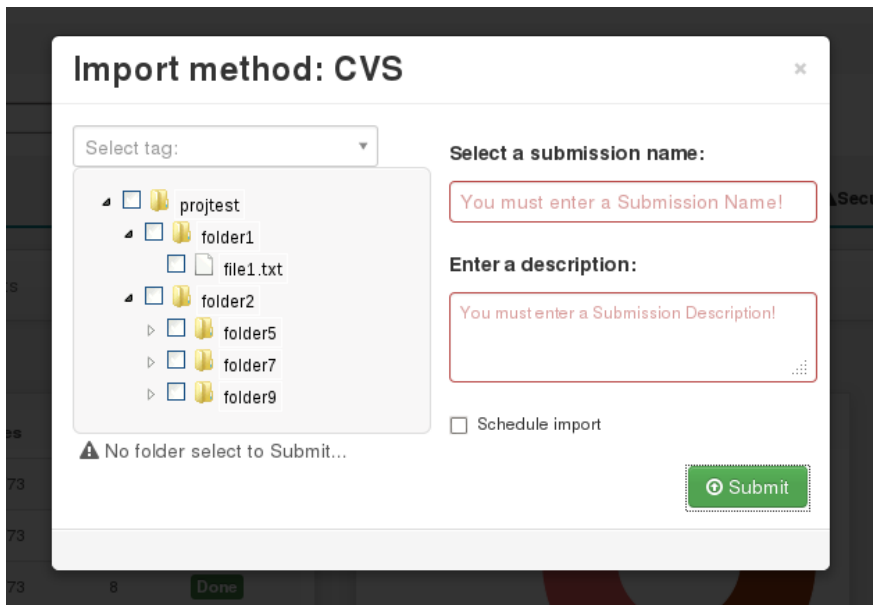


Figura F.5: Importação via CVS - Tratamento de exceções.

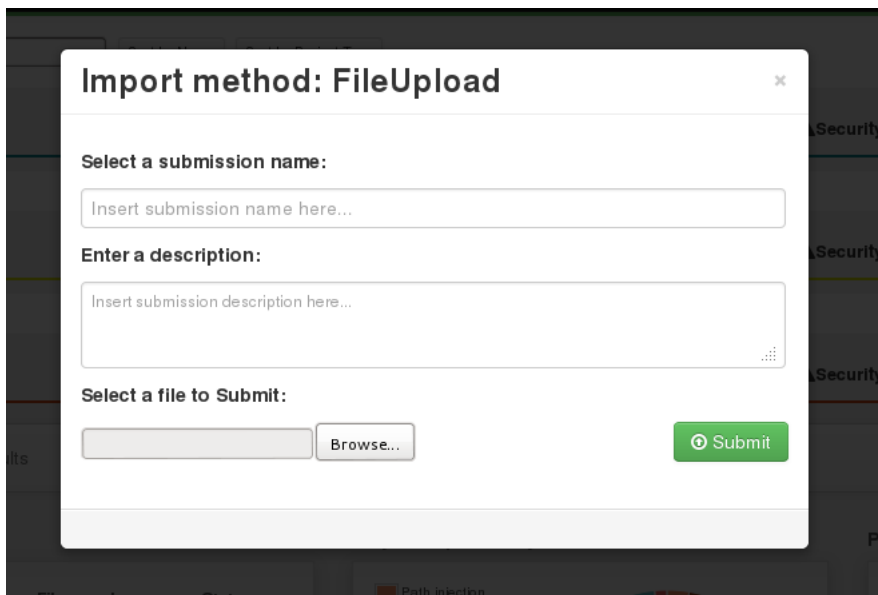


Figura F.6: Importação via FileUpload.

**Import method: HttpZip**

**Enter a submission name:**  
Insert submission name here...

**Enter a description:**  
Insert submission name here...

**Enter URL of Zip File:**  
Insert url here...

Schedule import    Daily  Weekly  Monthly

Figura F.7: Importação via HttpZip.

**Import method: SCP**

**Choose one of previous saved credentials:**  
Choose credentials:  
1 - scp creds

**Remote folder path:**  
Example: Desktop/project

**Login credentials:**  
Username... Password...

Save credentials

Figura F.8: Importação via SCP - Login.

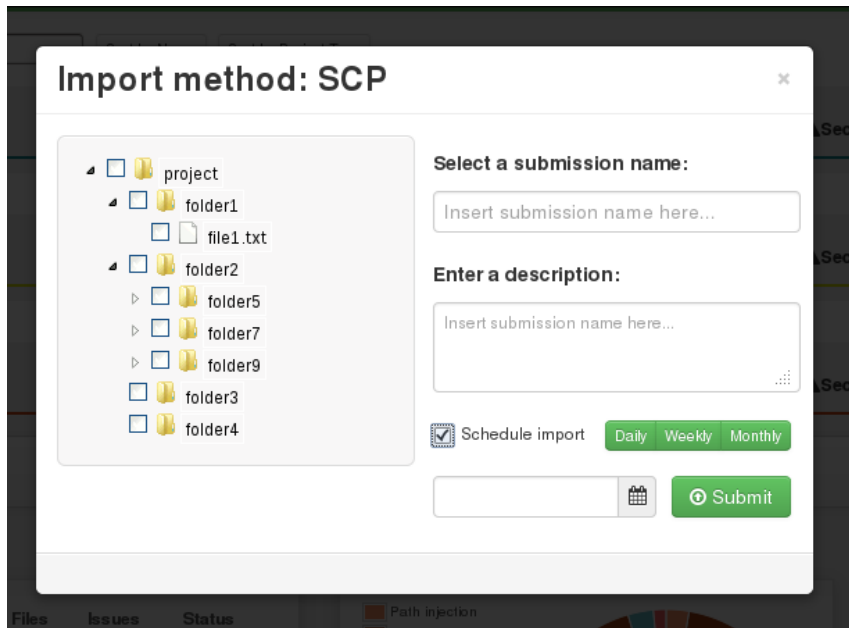


Figura F.9: Importação via SCP - Menu principal.

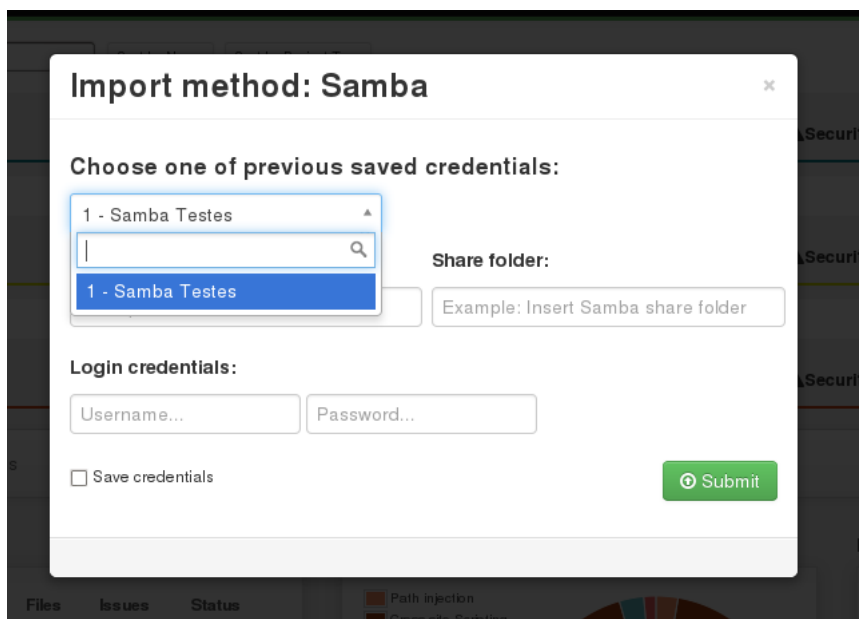


Figura F.10: Importação via SMB - Login.

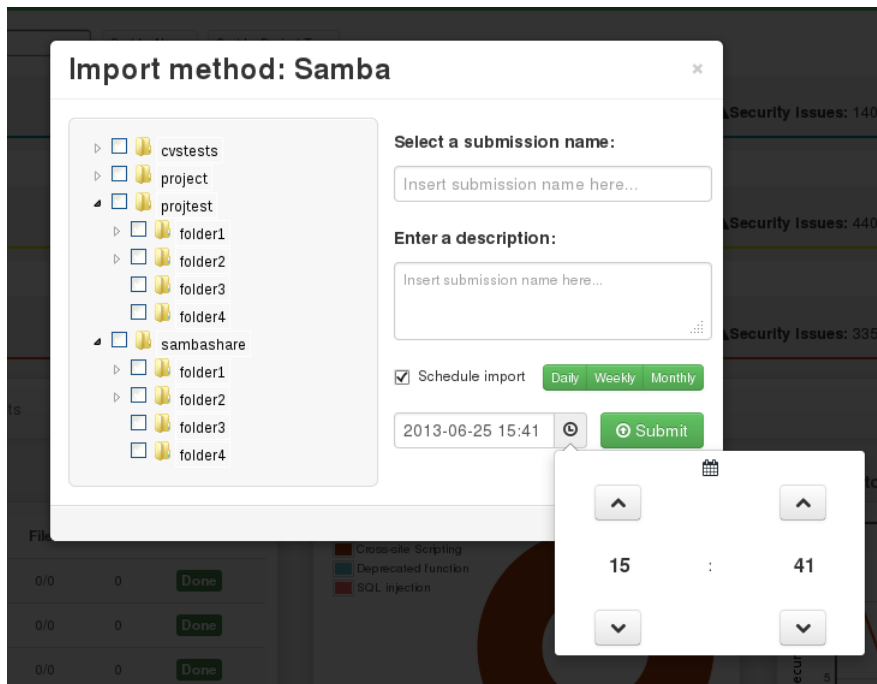


Figura F.11: Importação via SMB - Menu principal.

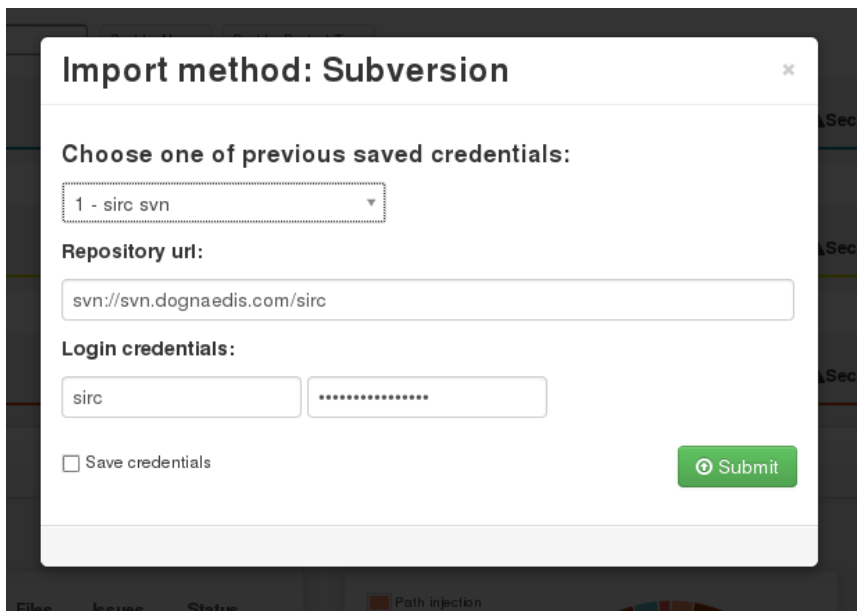


Figura F.12: Importação via SVN - Login.

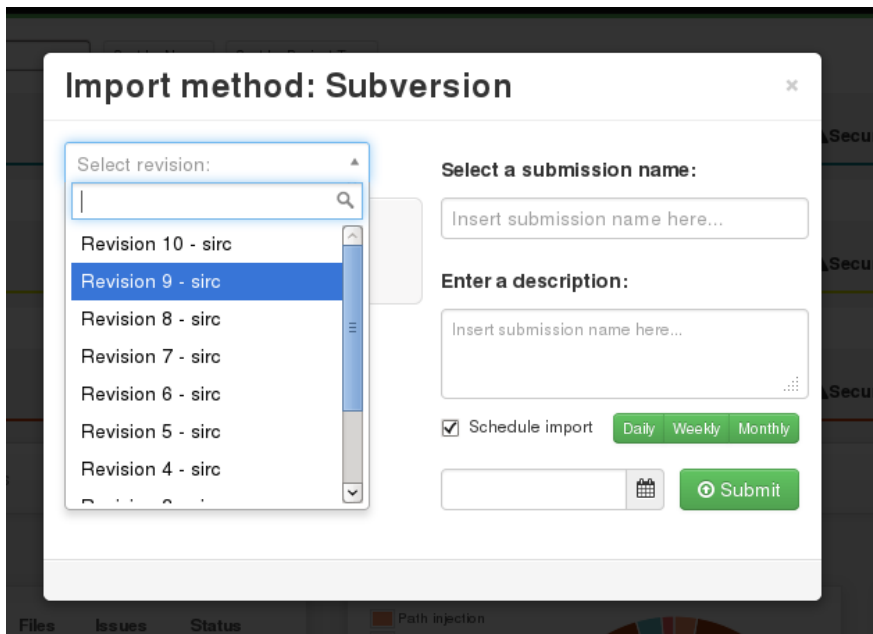


Figura F.13: Importação via SVN - Menu principal, pesquisa de revisões.

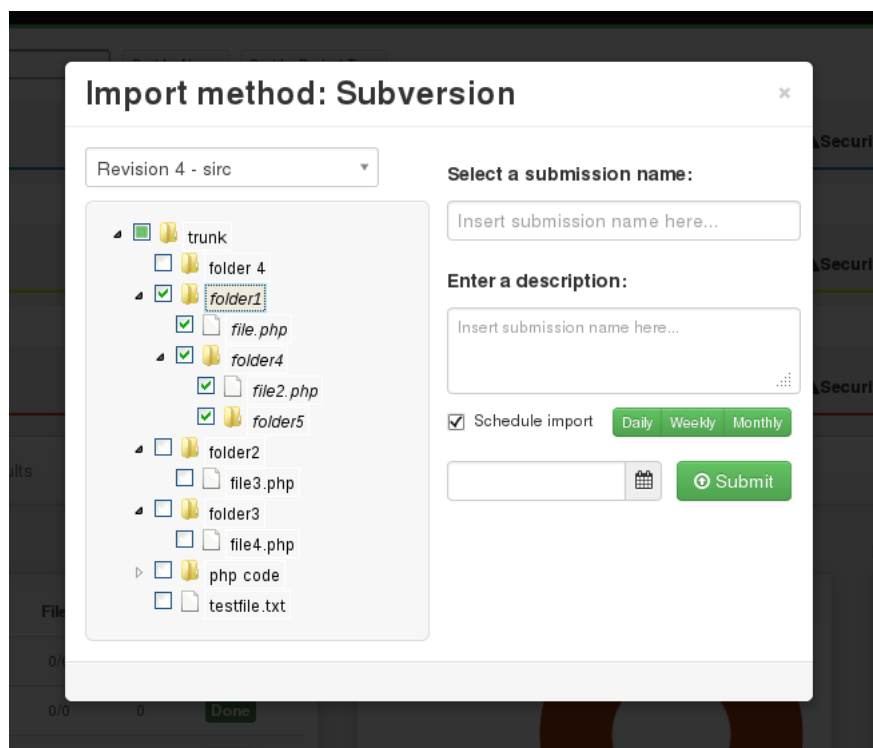


Figura F.14: Importação via SVN - Menu principal, seleções no explorador.

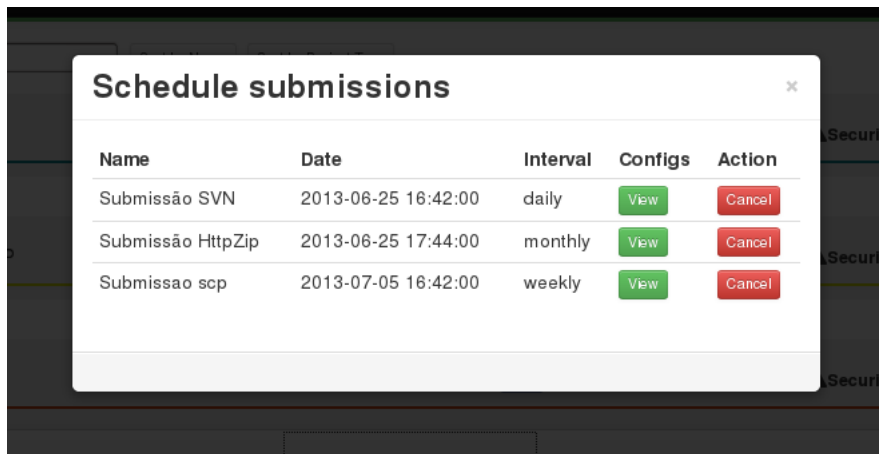


Figura F.15: Agendamentos ativos.

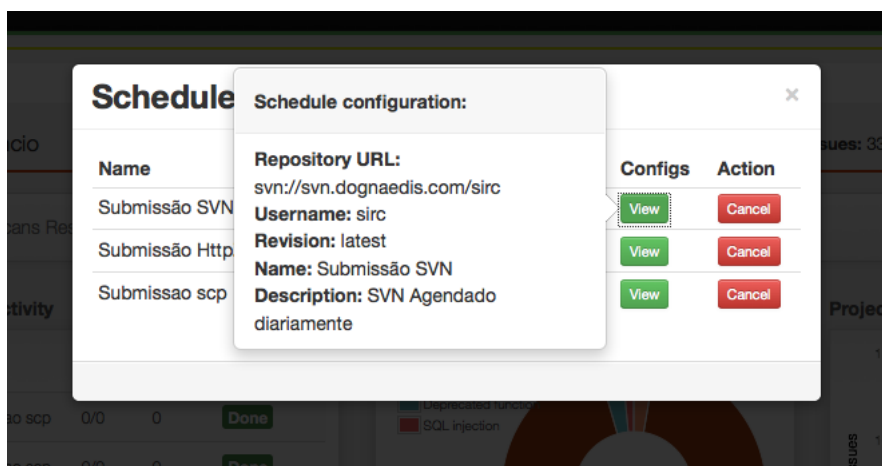


Figura F.16: Detalhes de agendamentos ativos.

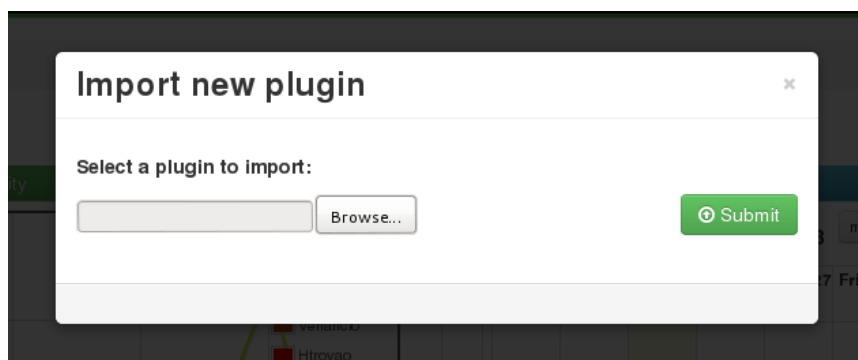


Figura F.17: Importação de novo plugin.



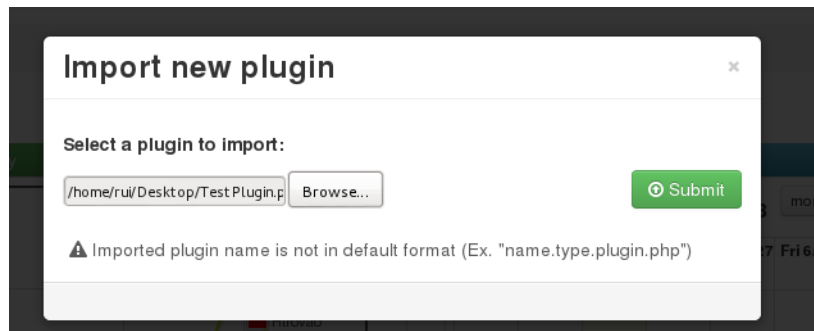


Figura F.18: Importação de novo plugin com erro no nome do ficheiro.

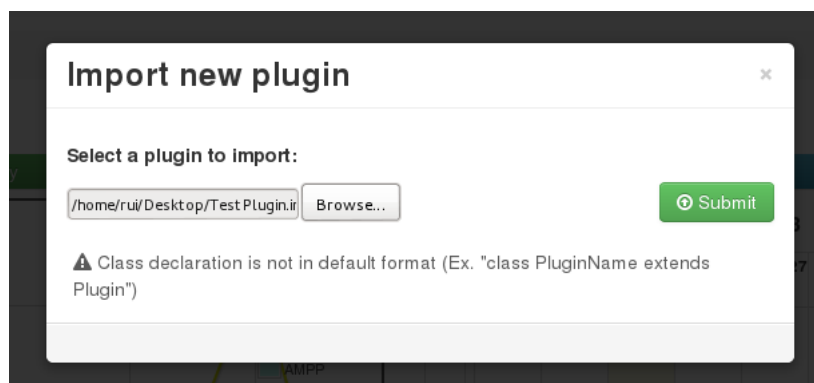


Figura F.19: Importação de novo plugin com erro na definição da classe.

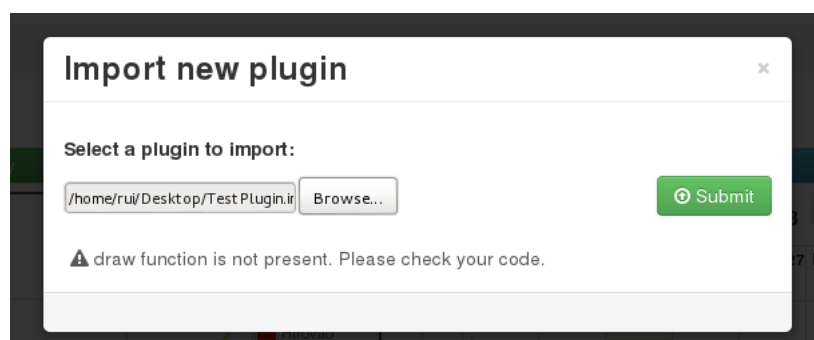


Figura F.20: Importação de novo plugin sem *override* a funções de uso obrigatório.

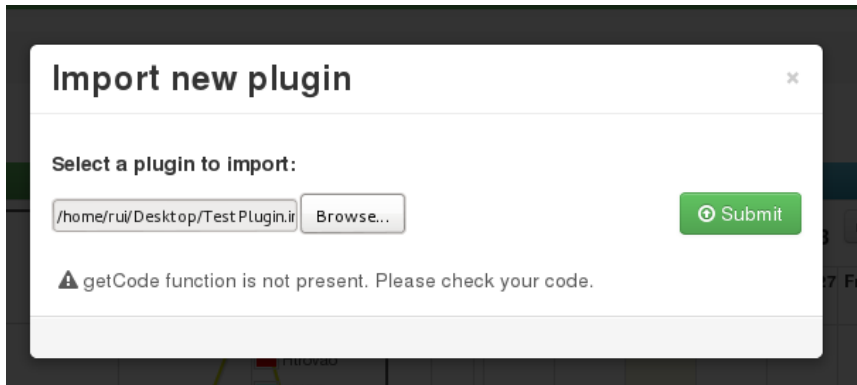


Figura F.21: Importação de novo plugin sem *override* a funções de uso obrigatório.

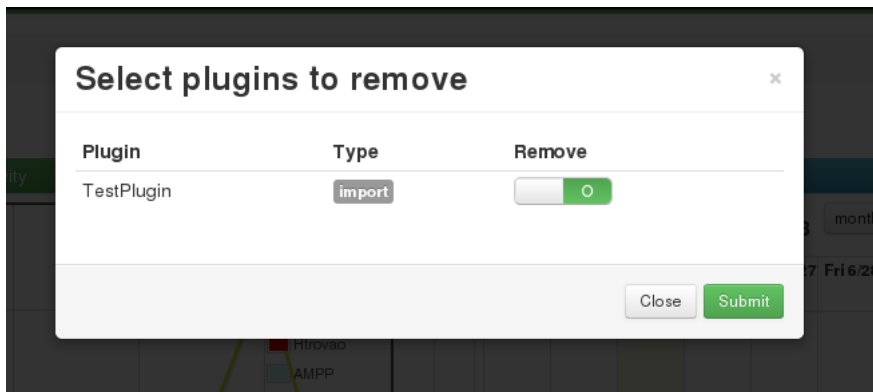


Figura F.22: Remover plugins.

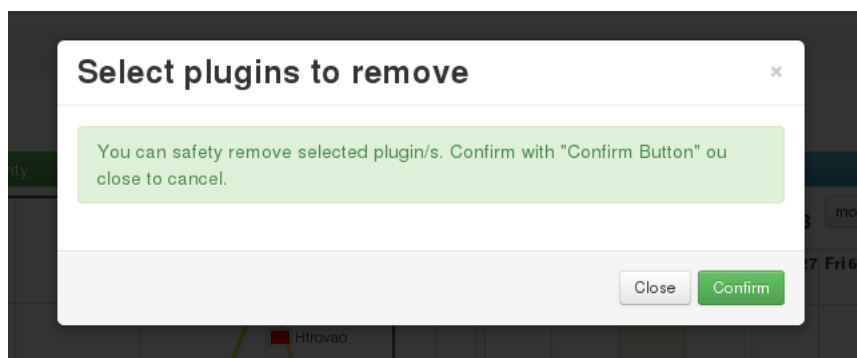


Figura F.23: Remover plugins - Diálogo de confirmação sem agendamentos dependentes.

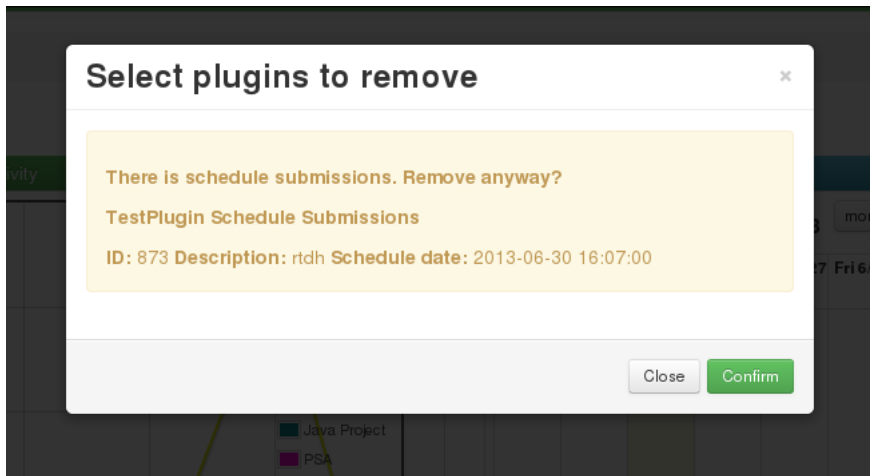


Figura F.24: Remover plugins - Diálogo de confirmação com agendamentos dependentes.

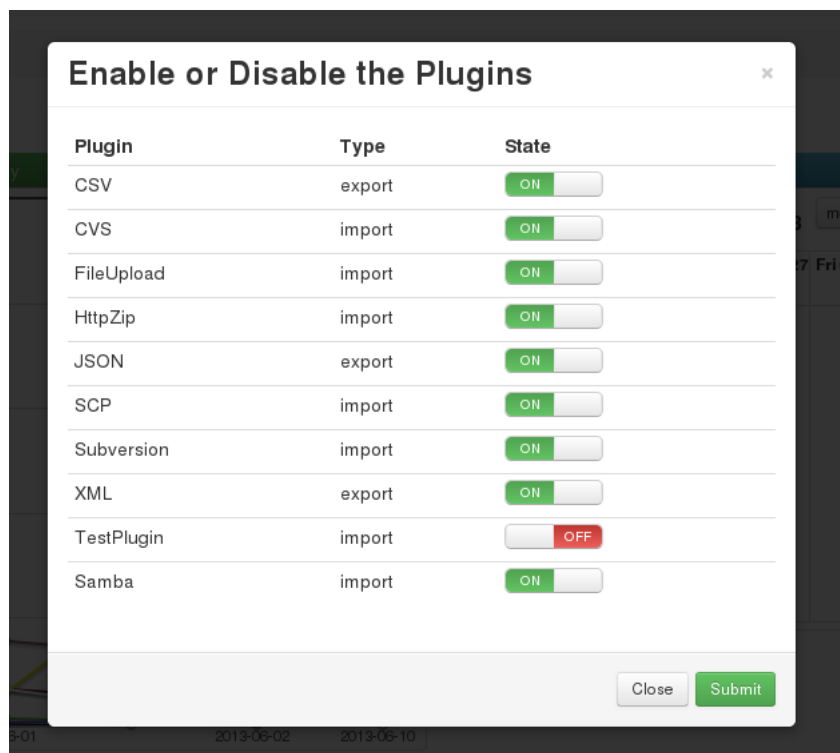


Figura F.25: Ativar e desativar plugins.

10 records per page Search:

Name	Date	Analysed / Total Files	Security Issues	Status	Options
Submissao scp	2013-05-29 16:42:00	0 / 0	0	Done	Export
Submissao scp	2013-05-29 16:27:00	-	-	Scheduled	Export
Submissao SCP	2013-05-29 12:01:00	0 / 0	0	Done	Export
Submissao teste	2013-05-29 11:55:00	0 / 0	0	Done	Export
Submissao svn daily	2013-05-29 00:02:00	1598 / 1755	150	Time Exceeded	Export
Submissao weekly	2013-05-29 00:01:00	0 / 0	0	Done	Export
Submissao weekly	2013-05-28 22:22:00	0 / 0	0	Done	Export
Submissao daily	2013-05-28 22:20:00	0 / 0	0	Done	Export
dfhgrth	2013-05-28 17:57:00	0 / 0	0	Done	Export
eee	2013-05-28 17:57:00	0 / 0	0	Done	Export

Showing 11 to 20 of 52 entries

– Previous 1 2 3 4 5 Next –

Figura F.26: Histórico de importações.

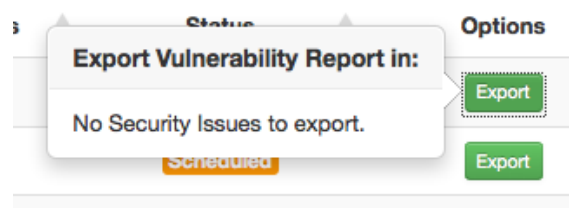


Figura F.27: Ação de exportação em relatório sem vulnerabilidades.

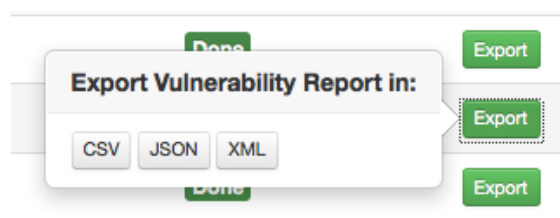


Figura F.28: Ação de exportação em relatório com vulnerabilidades.