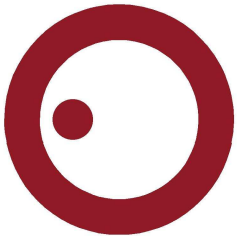


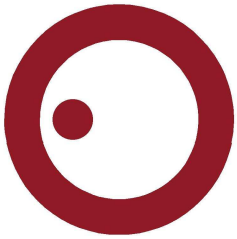
escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

# VULNERABILITY MANAGEMENT GAMIFICATION

ESTUDANTE MIGUEL DE ALMEIDA MARTINS LIBÂNIO

Leiria, Março de 2022



escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

# VULNERABILITY MANAGEMENT GAMIFICATION

ESTUDANTE MIGUEL DE ALMEIDA MARTINS LIBÂNIO
Número: 2190378

Projeto realizada sob orientação do Professor Doutor Paulo Manuel Almeida Costa
paulo.costa@ipleiria.pt

Leiria, Março de 2022

# AGRADECIMENTOS

O desenvolvimento deste projeto não seria possível sem o apoio de algumas pessoas e entidades que, direta ou indiretamente, o possibilitaram.

Antes de mais, gostaria de agradecer a Rui Gonçalo Amaro e ao meu orientador Paulo Costa. O primeiro, pela ajuda que sem qualquer hesitação prestou ao longo da realização deste projeto, e ao professor Paulo Costa, pela sua disponibilidade e orientação ao longo do desenvolvimento deste projeto.

Em especial, também quero agradecer à minha família pelo seu apoio ao longo deste percurso académico. Sem o vosso apoio isto não teria sido possível.

Gostaria de agradecer ainda à equipa Cipher, por ter sempre demonstrado a sua disponibilidade e vontade para me apoiar com o projeto.

Finalmente, gostaria de agradecer a todos os amigos que pude conhecer ao longo destes últimos dois anos. Sempre me apoiaram e fizeram-me sentir em boa companhia, enriquecendo, assim, toda esta experiência.

# RESUMO

Os episódios de deteção de vulnerabilidades, sendo essenciais para a segurança das infraestruturas de informação das organizações, expõem as vulnerabilidades existentes nos sistemas, possibilitando assim a sua resolução ou mitigação.

Contudo, devido a não serem problemas com impacto imediato, a dimensão de correções a implementar em grandes redes e à falta de consciência do impacto real destas vulnerabilidades, a sua mitigação de forma priorizada e continua, é um desafio a que as ferramentas atuais de deteção não são capazes de responder.

Assim, este projeto foi desenvolvido de modo a proporcionar uma plataforma, implementada através da *framework* Django, capaz de agregar os resultados dos vários episódios de deteção e de priorizar a resolução de vulnerabilidades tendo em consideração as suas severidades, *assets*, soluções e outros indicadores com o uso de projetos de remediação, conjuntos de uma ou mais vulnerabilidades a serem resolvidas até uma data limite por um grupo de utilizadores.

O foco deste projeto nos utilizadores responsáveis pelas remediações criadas manifesta-se também através de vários sistema que procuram automatizar ou reduzir os processos necessários para as remediações, como a simplificação de ações como pedidos de verificação de vulnerabilidades ou de maquinas, um sistema de preferências para melhor atribuição de projetos ou novas verificações do estado atual de vulnerabilidades que diminuem o numero de falsos positivos detetados pelos sistemas de deteção.

Simultaneamente a plataforma desenvolvida consegue também incentivar o seu uso e consequentemente que a remediação de vulnerabilidades seja mantida através de um conjunto de sistemas de gamificação.

Os sistemas implementados incluem o uso de recompensas atribuindo aos utilizadores pontuação por determinadas ações e removendo pela sua ausência, ou atribuído *badges* por metas atingidas, ou ainda o fomento à competição entre os utilizadores através do uso torneios, *leaderboards* ou mesmo certos gráficos nos relatórios de atividade e pagina principal.

# ABSTRACT

The episodes of vulnerability detection, being essential for the security of the information infrastructures of an organisation, expose the existing vulnerabilities in the systems, thus enabling their resolution or mitigation.

However, due to the fact that they are not problems with immediate impact, the size of the corrections to be implemented in large networks and the lack of awareness of the real impact of these vulnerabilities, their mitigation in a prioritised and continuous manner is a challenge which the current detection tools are unable to meet.

Thus, this project was developed in order to provide a platform, implemented using the *framework* Django, capable of aggregating the results of the various detection episodes and to prioritize the resolution of vulnerabilities taking into account their severities, assets, solutions and other indicators with the use of remediation projects, sets of one or more vulnerabilities to be resolved by a deadline by a group of users.

The focus of this project on the users responsible for the remediations is also manifested through various systems that seek to automate or reduce the processes required for remediation, such as simplifying actions like requests for verification of vulnerabilities or machines, a system of preferences for better allocation of projects or new checks of the current state of vulnerabilities that reduce the number of false positives detected by detection systems.

Simultaneously, the developed platform is also able to encourage its use and consequently that the remediation of vulnerabilities is maintained through a set of gamification systems.

The systems implemented include the use of rewards by assigning users points for certain actions and removing them for their absence, or assigning *badges* for goals achieved, or encouraging competition between users through the use of tournaments, *leaderboards* or even certain graphics in the activity reports and main page.

# ÍNDICE

# LISTA DE FIGURAS

# LISTA DE TABELAS

# LISTA DE ABREVIATURAS

| | |
|---|---|
| CVSS | Common Vulnerability Scoring System. |
| NIST | National Institute of Standards and Technology. |
| OT | Operational Technology. |
| REST | Representional State Transfer. |
| SGBD | Sistema de Gerenciamento de Bancos de Dados. |
| SLA | Service Level Agreement. |
| SUS | System Usability Scale. |
| UC | Universidade de Coimbra. |
| VMG | Vulnerability Management Gamification. |
| XSS | Cross Site Scripting. |

# 1

# INTRODUÇÃO

## 1.1 ENQUADRAMENTO

A análise e detção de vulnerabilidades permitem conhecer falhas existentes nos sistemas e infraestuturas de informação de organizações. Através de relatórios detalhados das vulnerabilidades descobertas pode-se então priorizar-las de acordo como os seus impactos e processo de resolução.

Contudo, em redes de grande dimensão, a mitigação destas vulnerabilidades, de forma priorizada e continua, é um desafio a que as ferramentas atuais não conseguem responder ou possuem grande dificuldade em responder.

Assim, é necessário uma ferramenta que permita às organizações acompanhar a evolução da mitigação destas vulnerabilidades continuamente, atribuindo as suas resoluções a diferentes grupos, priorizando-as e medindo o cumprimento, ou não, dos tempos de resolução dos nível de risco correspondentes.

Deste modo, este projeto visa desenvolver uma plataforma web que seja capaz de auxiliar as organizações na atribuição da resolução de vulnerabilidades, agrupando vulnerabilidades em diferentes projetos de resolução, e que através de um sistema de gamificaçao seja capaz ainda de fomentar um espírito competitivo saudável entre as equipas responsáveis pelas resoluções através da classificação e recompensa face ao trabalho realizado.

Este documento irá descrever o estudo efetuado a plataformas semelhantes à proposta, o desenvolvimento de uma aplicação e também os vários tipos de teste realizados para avaliar e verificar o sistema bem como se foi possível alcançar os objetivos propostos para este projeto.

## 1.2 ENTIDADE ACOLHEDORA

A Dognaedis é uma empresa centrada na segurança de informação, criada em 2010 por uma equipa de investigadores do CERT-IPN e da Universidade de Coimbra (UC) [1].

A empresa criou o software CodeV [2], que funciona como um perito de segurança informática, identificando de forma automática vulnerabilidades em programas. Este projeto foi distinguido em múltiplas ocasiões com o prémio BES Inovação em 2011, sendo reconhecido pela Garter em 2013 e 2015, e em 2018 foi uma das empresas distinguidas com o DIT Business Awards.

A Dognaedis faz parte do grupo Prosegur desde do final de 2018. Em 2019, todas as empresas da divisão cibersegurança da Prosegur começaram uma convergir numa única marca global, a Cipher.

## 1.3 DECLARAÇÃO DE PROBLEMA

Esta plataforma web foi desenvolvida de modo a satisfazer uma necessidade identificada no tratamento e mitigação de vulnerabilidades em redes de grande dimensão. Esta necessidade será mais detalhada nesta secção, juntamente com os objetivos deste projeto.

### 1.3.1 *O Problema*

Sendo cruciais para a segurança das infraestruturas de informação de uma organização, as analises de deteção de vulnerabilidades expõem vulnerabilidades existentes nos sistemas destas, que através da devida gestão podem ser priorizadas para remediação.

Contudo, devido a não serem problemas com impacto imediato, a dimensão de correções a implementar em grandes redes e à falta de noção do impacto real, causa com que a mitigação destas vulnerabilidades, de forma priorizada e continua, seja um desafio a que as ferramentas atuais de deteção não conseguem responder ou têm dificuldade a responder.

Assim, é necessário uma ferramenta que permita às organizações acompanhar a evolução da mitigação destas vulnerabilidades continuamente.

### 1.3.2 *Objetivos*

O principal objetivo deste projeto é pesquisar, planear e implementar uma plataforma web capaz de processar as vulnerabilidades, agrupando e atribuindo as suas resoluções a grupos diferentes de utilizadores medindo o cumprimento, ou não, cumprimento dos tempos de resolução estipulados.

A plataforma deve ainda possuir a capacidade de classificar e recompensar o colaborador responsável pelas correções face ao trabalho realizado, fomentando assim um espírito competitivo saudável entre os grupos. A plataforma web, deve ser capaz de:

- Exportar de uma plataforma de deteção de vulnerabilidades os seus resultados de forma regular;
- Processar resultados, agrupando-os sob diversos atributos consoante preferência do utilizador;
- De acordo com os critérios de agrupamento, atribuir vulnerabilidades a utilizadores, responsáveis pelas mitigações correspondentes;
- Fazer operações com o nível de risco resultante de tais agrupamentos, apresentando aos analistas quais as vulnerabilidades que devem mitigar primeiro, dado o risco que acarretam para a organização;
- Classificar vulnerabilidades como resolvidas, aceites ou falsos-positivos, comunicando tais designações à plataforma de deteção de vulnerabilidades facilitando a classificação de resultados de análises futuras;
- Classificar equipas e analistas de acordo com o nível de risco que foram capazes de mitigar, criando um *ranking* comparativo entre estes, favorecendo o modelo de gamificação, cujo resultado esperado será a recompensa de utilizadores e grupos que mais se destaquem em tais processos de redução de risco;
- Criação de *dashboards* que permitam mostrar à empresa resultados estatísticos de todas as funcionalidades previstas e descritas nos requisitos funcionais representando assim também indicadores de performance das equipas no processo de resolução de vulnerabilidades;
- Permitir que, após corrigida determinada vulnerabilidade, o utilizador possa lançar um teste sobre a aplicação, que lhe irá retornar se esta continua ou não a existir.

## 1.4 ESBOÇO DO DOCUMENTO

O presente relatório foi regido no âmbito do projeto do curso de Mestrado em Cibersegurança e Informática Forense. O presente trabalho foi realizado em três fases, sendo que a primeira o planeamento e a identificação dos requisitos necessários para a implementação da aplicação, na segunda se procedesse ao desenvolvimento e implementação do projeto e por fim à verificação e validação do sistema. Os capítulos estão organizados na seguinte forma:

1. **Introdução**: Este primeiro capítulo serve para apresentar uma breve descrição do trabalho que foi realizado e uma breve introdução ao contexto do projeto. Em seguida, apresenta a declaração do problema e os objetivos a serem alcançados.
2. **Planeamento**: O segundo capítulo detalha as fases do projeto e a metodologia escolhida para o processo de desenvolvimento e as tarefas a ela associadas.
3. **Estado da arte**: O terceiro capítulo apresenta um estudo das várias ferramentas e as suas abordagens e métodos face ao problema que esta plataforma propõem responder, procurando ainda apresentar as suas vantagens e desvantagens em detalhe.
4. **Requisitos**: O quarto capítulo detalha a análise de requisitos realizada e também os requisitos funcionais e não funcionais acordados para o projeto.
5. **Arquitetura**: O quinto capítulo é utilizado para detalhar a arquitetura do sistema resultante, o modo de tratamento de erros pela plataforma e enumera as tecnologias adotadas para o seu uso na aplicação web.
6. **Desenvolvimento**: O sexto capítulo apresenta o processo de implementação.
7. **Verificação, Validação e Avaliação**: O sétimo capítulo apresenta uma visão geral do processo utilizado para verificar que os componentes funcionam como esperado e validar a conformidade do produto com os requisitos propostos.
8. **Conclusão**: O oitavo e último capítulo serve para encerrar o documento com observações sobre o sistema resultante, olhando para trás para o que foi alcançado e detalhando trabalho futuro que possa ser construído sobre esta plataforma atual.

# 2

# PLANEAMENTO

Num projeto destas dimensões, um bom orçamento do tempo e escolha de metodologia são essenciais para gerir o tempo de desenvolvimento e atingir as metas planeadas. Como tal este capítulo será dividido em duas diferentes secções, a metodologia escolhida e o orçamento de tempo onde é apresentado o tempo necessário para o desenvolvimento do projeto e as fases de desenvolvimento da plataforma.

## 2.1 METODOLOGIA

No desenvolvimento de software é crucial o uso de uma metodologia de desenvolvimento, um conjunto bem estruturado de passos ou actividades que permitem aumentar a produtividade e a comunicação da equipa, bem como a qualidade do produto final. Neste projeto, foi decidido adotar uma metodologia ágil.

Uma abordagem incremental não só é mais tolerante em relação a pequenos erros como também permite a liberdade de aprender e explorar os conceitos e tecnologias necessários, ao mesmo tempo que acomoda a possível necessidade de mudanças rápidas. À medida que o conhecimento é recolhido e o aluno experimenta diferentes abordagens, o projeto pode ser gradualmente montado e avaliado passo a passo pelo supervisor da empresa. Para atingir esses objetivos, foi escolhida uma metodologia ágil inspirada em Scrum.

### 2.1.1 *Scrum*

Scrum[3] é uma *framework* incremental de desenvolvimento de software ágil[4] que pretende cortar a complexidade e tempo necessário para criar um produto através de uma forma incremental com ciclos de desenvolvimento curtos. Embora a definição da *framework* seja orientada para uma equipa, optou-se por adotar os conceitos-chave utilizados pela Scrum como parte da metodologia deste projeto, com alterações sempre que necessário.

#### 2.1.1.1 *Proprietário do produto*

O proprietário do produto, sendo neste projeto o orientador da empresa, é responsável por maximizar o valor do produto, dando prioridade aos itens no *Backlog*, assegurando a compreensão dos itens, assegurando que a visão do produto é mantida e fornecendo *feedback* durante cada Revisão do *sprint*.

#### 2.1.1.2 *Backlog do Produto*

O *Backlog* do Produto é uma lista ordenada de tudo o que possa ser necessário no produto e é a única fonte de requisitos para quaisquer alterações a serem feitas ao produto. É da responsabilidade do proprietário do produto.

#### 2.1.1.3 *Sprint*

O Sprint representa a unidade básica de trabalho na metodologia, uma caixa de tempo, neste projeto de duas semanas durante as quais foi implementada uma ou mais funcionalidades. Os Sprints têm uma duração consistente durante todo o esforço de desenvolvimento, tendo cada um deles um objetivo específico. Um novo Sprint começa imediatamente após a conclusão do anterior.

Os Sprints são divididos nas fases seguintes:

- Planeamento do Sprint: O trabalho a ser realizado durante o Sprint é planeado nesta fase. No contexto deste projeto, o objetivo do Sprint foram decididos pelo aluno e o orientador da empresa de acordo com o *backlog* do produto na altura. Após a definição do objetivo do Sprint, deve ser escolhido o método para a sua implementação.
- Revisão do Sprint: No final de cada Sprint deve ser realizada uma Revisão para assegurar que o objetivo do sprint foi alcançado, refletir sobre o estado atual e possíveis alterações ao *backlog* do produto e obter *feedback*.

#### 2.1.1.4 *Incremento*

O Incremento é a soma de todos os itens do backlog do produto completados durante um sprint e todos os incrementos anteriores, cuidadosamente verificado e testado que todos os incrementos funcionam em conjunto. Depois de cada incremento o produto deve também estar potencialmente entregável.

#### 2.1.1.5 *Processo Resultante*

O processo resultante é uma versão simplificada do Scrum dirigida a uma equipa de desenvolvimento composta por um único membro. Durante o processo de desenvolvimento foi utilizado um modelo de entregas semanais ou quinzenais onde era relatado ao orientador atribuído pela empresa quais as tarefas concluídas ou em desenvolvimento, visando garantir que a calendarização inicial se mantinha sob controlo.

Tratou-se de uma metodologia orientada às funcionalidades, para a qual foram definidas quais correspondem a cada requisito do projeto. Onde cada um destes requisitos constituía uma meta de avaliação sobre o trabalho desenvolvido.

Para que o projeto pudesse ser desenvolvido segundo esta metodologia, foi necessário ainda elaborar uma taxonomia de requisitos, classificando quais os mais e menos prioritários para a solução final, de forma a priorizar as suas implementações e definir objetivamente que funcionalidades pertencem a cada um deles.

Quanto à avaliação de cada um dos incrementos, foi feita pelo aluno e com o orientador da empresa, no final de cada sprint, confirmando que as funcionalidades planeadas para o requisito se encontravam implementadas e recolhendo feedback e para acomodar quaisquer alterações necessárias.

## 2.2 ORÇAMENTO DO TEMPO

Uma vez que o prazo de entrega inicial foi adiado de Junho para Novembro, o projeto possuía um orçamento de tempo maior ao inicialmente previsto, isto possibilitou a implementação de novas funcionalidades e uma fase de validação do projeto maior.

No final, foi ainda necessário um tempo extra devido a um atraso no desenvolvimento do relatório do projeto. Esta questão é detalhada na seguinte tabela, que apresenta o orçamento de tempo final para o desenvolvimento do projeto:

| **Período** | **Tempo** |
|---|---|
| Data de Inicio | 5/10/2020 |
| Data de Fim | 26/10/2021 |
| Data do Relatório | 31/03/22 |
| Esforço Semanal | 16 horas |
| Semanas | 55 |
| Esforço Total | 880 horas |

Tabela 1: Orçamento do Tempo

Para auxiliar no planeamento das fases do projeto, recorreu-se ao auxilio de um diagrama de Gantt com as respectivas fases e tarefas apresentadas em detalhe. O diagrama de Gantt poderá ser encontrados no apêndice A. A seguinte tabela apresenta uma versão resumida do diagrama através do orçamento do tempo despendido em cada fase:

| **Fase** | **Inicio** | **Fim** | **Duração** |
|---|---|---|---|
| Planeamento | 5/10/20 | 26/10/20 | 21 dias |
| Estado de Arte | 5/10/20 | 26/10/20 | 21 dias |
| Arquitectura | 5/10/20 | 26/10/20 | 21 dias |
| Desenvolvimento | 26/10/20 | 10/08/21 | 288 dias |
| Testing | 10/8/21 | 26/10/21 | 77 dias |

Tabela 2: Fases do projeto

Cada uma destas fases serão exploradas em detalhe ao longo dos capítulos deste documento com excepção da fase de design, onde foi planeada e desenhada a base de dados e interface da plataforma, que será descrita no capítulo 6.

## 2.3 SÍNTESE

Ao longo deste capítulo foi descrita a metodologia escolhida e como esta foi adaptada. De seguida foi apresentado o orçamento de tempo onde foram identificadas as varias fases de desenvolvimento da plataforma e o tempo necessário para o seu desenvolvimento.

# 3

# ESTADO DA ARTE

No presente capítulo serão apresentados os resultados do estudo do estado da arte referente às várias aplicações, ferramentas e técnicas similares à plataforma desenvolvida para este projeto, de modo a procurar os métodos mais adequados para facilitar e incentivar aos utilizadores a resolução de vulnerabilidades.

Nestes termos, optou-se por dividir o estudo do estado da arte, entre ferramentas de gestão de vulnerabilidades e aplicações que utilizam a gamificação na sua utilização.

## 3.1 VULNERABILIDADES

Em segurança informática, uma vulnerabilidade é uma falha ou fraqueza de segurança que permite a um atacante reduzir a firmeza da informação de um sistema [5] [6].

As vulnerabilidades podem ser causadas por muitos fatores, alguns dos mais comuns são a má configuração de um sistema, a falta da encriptação de dados ou erros no código de um software utilizado. Perante alguma destas falhas, um atacante é capaz de explorar as vulnerabilidades de uma organização de modo aceder indevidamente aos dados de uma organização, comprometendo assim a confidencialidade, integridade e disponibilidade da informação.

As vulnerabilidades podem ser divididas, com base no seu tipo, em cinco categorias[7]:

1. **Vulnerabilidades do Sistema Operativo**: Estas são vulnerabilidades dentro de um sistema operativo particular que os atacantes podem explorar para obter acessos ou privilégios que não era suposto. Alguns exemplos comuns incluem contas *default* de *superusers* ou *backdoors* ocultos.
2. **Vulnerabilidades de Rede**: São problemas com o hardware ou software de uma rede que a expõem a uma possível intrusão. Os exemplos desta vulnerabilidade incluem pontos de acesso Wi-Fi inseguros e *firewalls* mal configuradas.
3. **Vulnerabilidades Humanas**: Os utilizadores podem facilmente expor dados sensíveis ao criar pontos de acesso exploráveis para atacantes, ou perturbar os sistemas. Um exemplo deste tipo de vulnerabilidades é *Social Engineering*, uma técnica que utiliza a psicologia humana para ganhar acesso indevido a dados e sistemas.
4. **Vulnerabilidades de Processos**: Algumas vulnerabilidades podem ser criadas por falta de controlos e processos. Um exemplo, seria a utilização de palavras-passe fracas.

Novas vulnerabilidades são descobertas todos os dias levando a que sua gestão e remediação seja necessária caso uma organização não queira ficar exposta a novos riscos e potenciais ataques.

## 3.2 GESTÃO DE VULNERABILIDADES

A gestão de vulnerabilidades é uma prática de segurança especificamente concebida para identificar, classificar, remediar ou mitigar a potencial exploração de vulnerabilidades dos sistemas informáticos de uma organização. Esta prática cíclica é demonstrada através do ciclo de vida da gestão de vulnerabilidades.

O ciclo de vida da gestão de vulnerabilidades, ver figura 1, permite às organizações identificar as vulnerabilidades de segurança dos seus sistemas informáticos, priorizar os seus *assets*, avaliar o risco, informar-se e corrigir as vulnerabilidades identificadas e por fim verificar se foram, ou não, eliminadas.

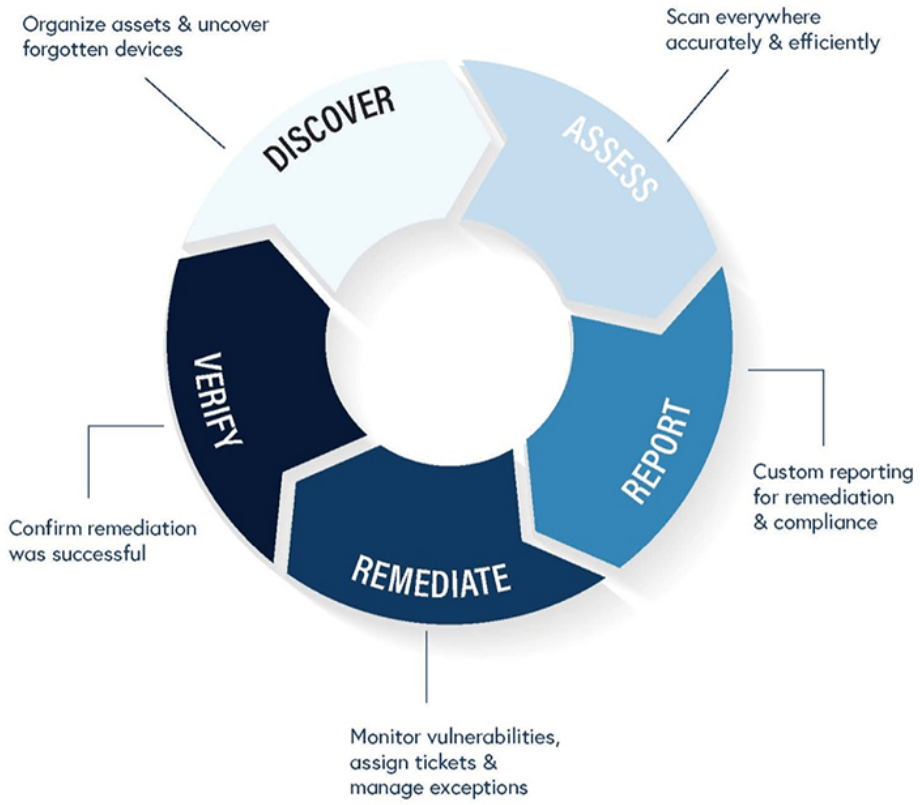



Figura 1: Vulnerability Management Life Cycle [8].

As etapas do Ciclo de Vida da Gestão da Vulnerabilidade são, as seguintes:

**Identificar**: construir o inventário de todos os *assets* e identificar os detalhes de cada máquina, incluindo o sistema operativo e os serviços expostos para identificar possíveis vulnerabilidades. Esta identificação de vulnerabilidades de segurança deve ser realizada de forma automatizada e regular.

**Avaliar**: determinar as vulnerabilidades dos *assets* no inventario e certificar que cada *asset* é analisado, tanto com precisão como com eficiência.

**Reportar**: medir o nível de risco associado aos ativos de acordo com as políticas de segurança. Documentar um plano de segurança, monitorizar atividades suspeitas e indicar as vulnerabilidades conhecidas.

**Remediar**: definir prioridades e corrigir vulnerabilidades de acordo com o seu risco. Estabelecer controlos e demonstrar progresso.

**Verificar**: verificar que as ameaças foram eliminadas através de auditorias de acompanhamento.

A gestão de vulnerabilidades é uma tarefa cuja dificuldade advém do grande numero de vulnerabilidades existentes, sendo que cada uma delas possui ainda várias formas de serem remediadas. Mesmo as vulnerabilidades de baixo risco podem muitas vezes transformar-se em vulnerabilidades de alto risco dependendo do *asset* em que se encontra e o ambiente da organização. Além disso, normalmente, organizações com uma grande infraestrutura de rede com muitos dispositivos onde todos eles poderão ser vulneráveis e em que uma única vulnerabilidade poderá comprometer a segurança de toda a rede, a gestão de vulnerabilidades torna-se uma tarefa prioritária.

Atualmente, a gestão da vulnerabilidade já não é apenas uma necessidade técnica, pois tornou-se um requisito institucional para muitas organizações que procuram manter-se a par dos modernos requisitos de conformidade e conduzir os seus negócios internacionalmente, minimizando os riscos associados a cibersegurança.

As seguintes normas são as mais comuns em matéria de gestão da vulnerabilidade:

1. **CVE**[9] (Common Vulnerabilities and Exposure): esta norma fornece um identificador único a vulnerabilidades publicadas.
2. **CCE**[10](Common Configuration Enumeration): fornece um identificador único para questões de configuração de sistemas relacionadas com a segurança, de modo a facilitar uma associação de configurações através de múltiplas fontes e ferramentas.
3. **CPE**[11] (Common Platform Enumeration): é uma norma de nomenclatura para sistemas de informação que facilita a ligação entre vulnerabilidades e plataformas.
4. **CWE**[12] (Common Weakness Enumeration): este sistema permite a categorização de vulnerabilidades de software, através dele é possível uma melhor descrição, avaliação e seleção de ferramentas que podem encontrar estas fraquezas no código fonte de software.
5. **CVSS**[13] (Common Vulnerability Scoring System): fornece uma unidade comum para a avaliação de risco de uma vulnerabilidade que exista na base de dados de vulnerabilidade da NIST (National Institute of Standards and Technology). A unidade está relacionada com a facilidade de exposição de cada vulnerabilidade combinada com o risco da sua exposição. O CVSS de uma vulnerabilidade tem um intervalo entre 0 e 10, sendo o 0 de baixo risco e o 10 de alto risco.

Seguidamente serão apresentadas as ferramentas estudadas e as suas respetivas abordagens à gestão de vulnerabilidades, concluindo-se com uma breve síntese e comparação das várias ferramentas.

### 3.2.1 *Vulnerability Control by Skybox Security*

A solução da Skybox Security[14] para a gestão de vulnerabilidades engloba a descoberta, avaliação, priorização de vulnerabilidades e também o acompanhamento das remediações. Esta solução abrange redes locais, redes na cloud e OT (Operational technology).

Utilizando o contexto das redes, *assets* e do negócio, esta solução é capaz de determinar os riscos e ajudar a focar as remediações nos *assets* mais prioritários[15]. Através da sua complexa interface gráfica a aplicação possibilita a um utilizador consultar detalhadamente o estado da infraestrutura e a avaliar com precisão os *assets* e vulnerabilidades, algo normalmente complexo em ambientes de rede dinâmicos.

Isto ajuda também a construir um modelo de rede abrangente e ao utilizador, compreender o estado de segurança da sua organização, vendo as vulnerabilidades dos seus *assets*, onde residem os dados e operações críticas, bem como os potenciais caminhos de ataque.

A solução é capaz de importar informação de múltiplas fontes de dados, possibilitando às organizações com mais de um tipo de *scanners*, sistemas ou produtos de deteção de vulnerabilidades, a gestão de vulnerabilidades através de uma ferramenta central [16].

Na página do centro de vulnerabilidade é apresentado através de um conjunto de gráficos o índice de vulnerabilidade. Este índice consiste num valor que dá uma indicação da escala e a severidade das vulnerabilidades que afetam a organização.

O índice de vulnerabilidade da plataforma é calculado diariamente a partir de uma soma de fatores atribuídos a cada vulnerabilidade reportada num espaço de tempo de 90 dias, por defeito, embora possa ser alterado para permitir ás organizações ver o impacto dos seus diferentes ciclos de resolução.

O centro de priorização, apresentado na figura 3, mostra potenciais ameaças ou ameaças iminentes através de um gráfico de cone. O risco baseia-se na exposição do *asset* e na existência de um vector de ataque até ao *asset*, ou seja, se um *asset* está

exposto directamente ou indirectamente é considerado na avaliação de risco, bem como o numero de vulnerabilidades do *asset*.

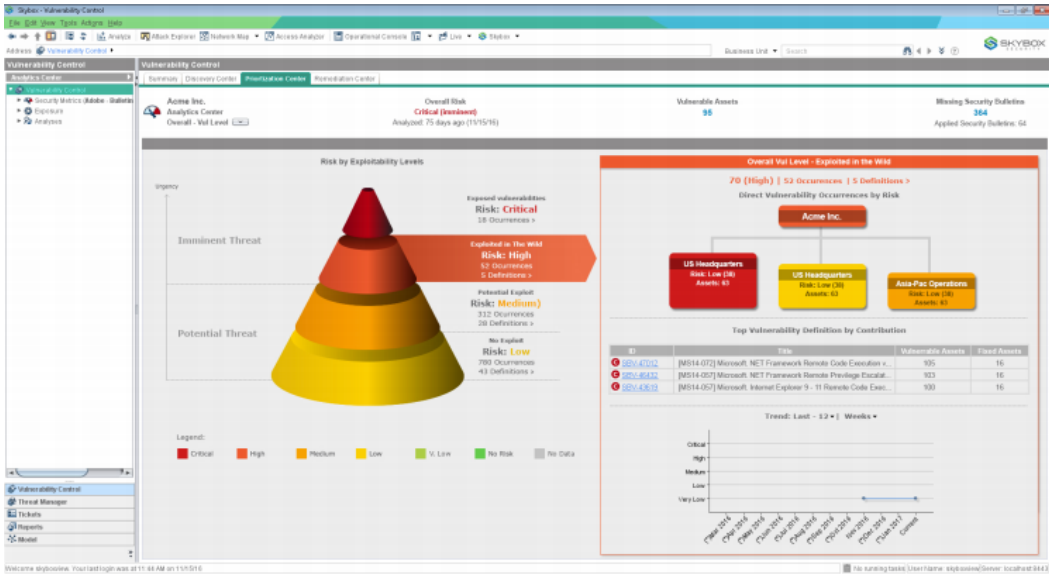



Figura 2: Centro de Prioritização da Skybox [17].

Por fim no centro de ataques é demonstrado visualmente os possíveis ataques á organização mostrando as possíveis vulnerabilidades exploradas, como elas poderão ser exploradas, e diferentes possíveis ataques. Os dados para as soluções não são só retirados das análises de vulnerabilidades mas também da *threat intelligence* realizada pela equipa da Skybox.

### 3.2.2 *SaltStack SecOps*

O SaltStack SecOps é o componente de *policy compliance* e de gestão de vulnerabilidades do vRealize Automation. Através deste componente é possível definir políticas de segurança para a organização, analisar os seus sistemas por vulnerabilidades e avaliar a conformidade das politicas de segurança definidas [18].

Como na solução anterior o SaltStack SecOps[19] permite a remediação de vulnerabilidades através de episódios de deteção de vulnerabilidades, dando priorização ás vulnerabilidades, utilizando os dados do *asset* envolvido e a informação da vulnerabilidade detetada e por fim fornecendo soluções para a sua remediação.

Quanto ao aspeto visual, como apresentado na figura 3, esta solução optou por uma abordagem de simplificação onde disponibiliza a informação essencial utilizando um pequeno número de gráficos e tabelas que dispõem a informação de forma clara e rápida ao utilizador, infelizmente, não são capazes de proporcionar ao utilizador uma visão completa dos seus sistemas e o seu estado atual.


Figura 3: Página do SaltStack Config [20]

Para além da sua capacidade de deteção de vulnerabilidades nativa, o SaltStack SecOps também é capaz de ingerir os análises de vulnerabilidades de outras soluções como a Tenable, Rapid7, Qualys, e Kenna [21].

### 3.2.3 *Rapid7 InsightVM*

A solução de remediação de projetos do Rapid7, InsightVM avalia os riscos e vulnerabilidades de cada sistema através de um *scan* ou de avaliações baseadas em agentes [22]. Os dados das vulnerabilidades recolhidas são tratados em cada anfitrião individual incluído na plataforma do InsightVM.

O InsightVM estima a criticidade de uma vulnerabilidade não só utilizando o CVSS, mas também através de uma equação proprietária para calcular adicionalmente o "risco real". A equação leva em conta o CVSS Score e várias métricas temporais, além disso, incorpora a exposição, a existência de kits de *malware* e possibilidade de exploração [23]. Classifica as vulnerabilidades com base nestes resultados num número de risco num intervalo de 1 a 1000.

Um projeto de remediação possui configurações de modo a definir qual o âmbito das vulnerabilidades que precisam de ser remediadas, por quem e quando deve estar remediado, mas a sua criação é realizada através de um conjunto de filtros simples que embora funcionais tornam a sua utilização um processo mais laborioso que o necessário.

Quando configurado com Jira, uma ferramenta que permite a monitorização e acompanhamento de tarefas, os projetos são criados automaticamente e poderão ser atribuídos aos proprietários dos sistemas para serem tratados por eles próprios.

Uma vez tratada a remediação das vulnerabilidades, o projeto de remediação em InsightVM mostra que as vulnerabilidades são marcadas como resolvidas automaticamente, evitando o pedido de um novo ao HIAB (hacker-in-a-box) do Outpost24.

### 3.2.4 *Outpost24 HIAB*

O HIAB é uma ferramenta para a gestão de vulnerabilidades que inclui um *scanner* de rede de vulnerabilidades e também um *scanner* para aplicações web.

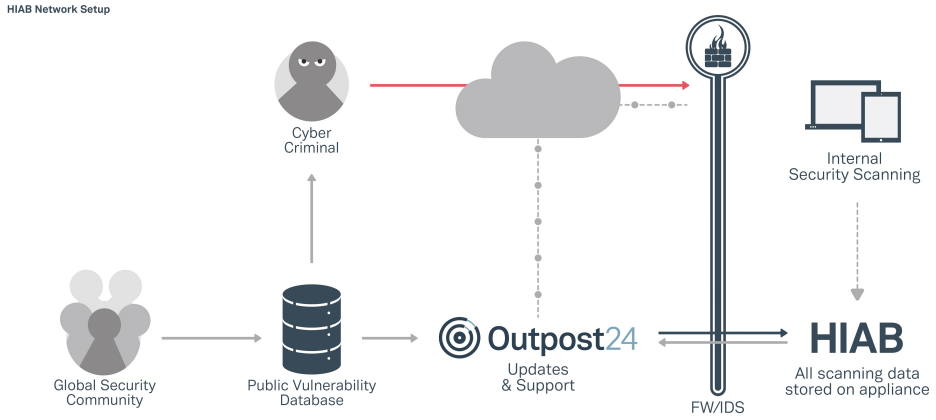



Figura 4: HIAB Diagram.

A plataforma é acessível através de um *browser* onde é apresentado, ao utilizador uma página principal onde um conjunto de gráficos e tabelas dispõem a informação sobre o estado dos sistemas motorizados. A primeira categoria consiste numa visão geral onde através de um gráfico de linhas é apresentado o numero de vulnerabilidades e remediações ao longo do tempo e uma tabela das 10 melhores soluções com o numero de vulnerabilidades e *assets* que estas corrigem.

A segunda categoria, *targets*, pretende transmitir ao utilizador o estado dos dispositivos monitorizados através da percentagem de *assets* com vulnerabilidades altas, os *assets* mais vulneráveis, os *assets* com vulnerabilidades exploráveis e outros dados, tais como máquinas com problemas de certificados.

Por fim temos a categoria das *findings* ou vulnerabilidades onde o utilizador é apresentado como as vulnerabilidades mais antigas ou as mais críticas.

Os alvos para a avaliação de vulnerabilidades podem ser especificados através de uma lista de IPs, domínios ou então indicando um intervalo de uma rede onde será realizado uma deteção de vulnerabilidades de modo identificar os alvos disponíveis para que o processo de avaliação seja realizado.

Para definir o conjunto de vulnerabilidades que devem ser procurados é disponibilizado pela aplicação uma série de diferentes políticas pré-definidas para diferentes cenários e máquinas alvo. Em particular, existem as seguintes políticas :

**Normal**: Esta é a politica padrão que inclui a verificação de todas as vulnerabilidades, que são não intrusivas.

**Normal with WebApp**: É aplicada a política "Normal"com verificações adicionais para Cross Site Scripting (XSS) e ataques de SQL Injection.

**Port Scan**: É efetuada a verificação para portos TCP e UDP específicos.

**Unsafe**: A política "Normal"é aplicada, exceto que os *scripts* perigosos são também executados, o que pode afetar o funcionamento normal dos sistemas alvo.

**New checks**: A política padrão é aplicada apenas para novas vulnerabilidades que apareceram desde da última análise.

**New checks and most recent findings**: Esta política verifica apenas as novas vulnerabilidades que foram identificadas na última análise e para vulnerabilidades que foram recentemente identificadas.

**PCI**: A politica PCI (Payment Card Industry) é utilizada em transações de cartões de crédito, onde ambas as partes da transação são verificadas para vulnerabilidades nos seus sistemas.

Depois da avaliação de vulnerabilidades, os resultados são armazenados localmente na aplicação onde é possível exportar relatórios dos resultados obtidos em vários tipos de ficheiros. Com base nestes relatórios, os administradores podem então avaliar as vulnerabilidades, o seu grau de exposição ao risco e por fim implementar as soluções que a aplicação atribuiu para cada vulnerabilidade.

### 3.2.5 *Comparação*

| **Ferramenta** | Interface Informativa | Interface Intuitiva | Projetos de Remediação | Projetos Úteis |
|---|---|---|---|---|
| **Vulnerability Control** | Sim | Não | Não | N/A |
| **SaltStack SecOps** | Não | Sim | Não | N/A |
| **InsightVM** | Sim | Não | Sim | Sim |
| **HIAB** | Sim | Não | Sim | Não |

Tabela 3: Comparação de ferramentas de gestão de vulnerabilidades

Como descrito ao longo deste capítulo e novamente reforçado na tabela 3, as soluções já existentes, embora funcionais e providenciem uma interface informativa, elas não são capazes de tratar e apresentar a informação de maneira intuitiva.

Não utilizam projetos de remediação, com exceção do HIAB que permite a criação de um ticket por cada projeto para seguir a sua resolução, uma solução pouco pratica e do InsightVM que realmente utiliza projetos de resolução para a gestão e controlo de remediações.

Como indicado anteriormente, através deste projeto pretende-se o desenvolvimento, com a *framework* Django, de uma plataforma web que permita aos utilizadores a gestão de vulnerabilidades num formato mais amigável e simplificado. Através de uma interface que formate e agregue os resultados das análises de deteção de vulnerabilidades, como faz o InsightVM, mas também que disponibilize todos os detalhes das vulnerabilidades sem perder a sua facilidade de uso, como se verificou com o HIAB.

O desenvolvimento de episódios de deteção de vulnerabilidades automáticos também será essencial para evitar que os utilizadores necessitem de pedir novos episódios para confirmar as suas resoluções, adicionando um passo desnecessário ao processo. Algo que uma solução com agentes como InsightVM da Rapid7 já efetua.

Felizmente, o uso de filtragem na criação de projetos já implementado no InsightVM fornece um ótimo molde para seguir e melhorar a utilização para este projeto, e que assinala novamente as vantagens da sua utilização.

Os seus problemas com a dificuldade de uso devem-se principalmente ao uso de poucos filtros mais abstratos e ao facto do utilizador necessitar de conhecer os comandos dos filtros para os aplicar.

Esta situação é facilmente remediada com o uso de um maior número de filtros mais especializados na sua disponibilização como opções e não comandos.

A capacidade da plataforma Vulnerability Control de processar os resultados de episódios de vários tipos de *scanners* também apresenta uma oportunidade, que embora fora do âmbito deste projeto, deva ser planeada e preparada de modo a facilitar ainda mais o uso da plataforma desenvolvida e estender a sua utilidade para o futuro.

Por fim, um dos pontos fracos de todas as soluções discutidas foi a falta de mecanismos que incentivem o seu uso, motivo pelo qual, foi estudado e implementado o uso de vários mecanismos de gamificação.

## 3.3 GAMIFICAÇÃO

A Gamificação, no contexto de desenvolvimento de software, é a utilização de mecânicas, estratégias e características de jogos de modo a motivar e facilitar o uso da aplicação, aumentando a retenção de utilizadores [24].

Através de elementos de gamificação, é possível transformar rotinas laboriosas em pequenos jogos, desencadear uma sensação de realização e deste modo motivar os utilizadores a utilizar ainda mais a aplicação e a dedicar-se às tarefas e desafios por ela propostos.

Existem diferentes elementos [25] que podem influenciar a forma como as pessoas interagem com uma aplicação, tais como, recompensas, questionários, badges outros bens virtuais, *leaderboards* e outras exibições de progresso. Todos estes elementos ajudam a criar uma experiência única para os utilizadores e encoraja-os a voltar e utilizar novamente a aplicação. De seguida serão descritos os elementos de gamificação que foram identificados como os que melhor se enquadram com os objetivos da plataforma e mais sucesso obtiveram:

- **Pontos**: pontos são recompensas básicas que o utilizador recebe pelas suas ações à medida que elas são realizadas, de modo que estas sejam compensadas e incentivadas.
- **Badges**: estas recompensas são uma representação visual dos objetivos atingidos pelos utilizadores, que indicam o seu desempenho no âmbito da aplicação. Por exemplo, algumas aplicações de fitness introduzem badges com base no número de passos que o utilizador percorreu ao longo da utilização da aplicação.

- **Níveis**: níveis, em conjunto com pontos, servem como um mecanismo de medir e recompensar a interação que o utilizador tem com a aplicação. A cada nível, a complexidade e dificuldade das metas aumenta, desafiando o utilizador a ir mais longe.
- **Gráficos de Desempenho**: estes gráficos mostram o desempenho do utilizador em relação aos seus resultados e mantém um registo do seu histórico.
- **Leaderboards**: listas com classificações de utilizadores ajudam a definir quem tem o melhor desempenho da aplicação ou numa determinada atividade/evento. Ao contrário dos gráficos de desempenho que mostram o desempenho de um utilizador num determinado período de tempo, um painel de avaliação mostra o seu desempenho em relação ao desempenho dos outros.
- **Torneios**: torneios são concursos onde os utilizadores competem para ver quem consegue o maior número objetivos ou atingir mais pontos num determinado período de tempo. Quando estes eventos terminam, os utilizadores no topo são depois recompensados.

A gamificação está presente em aplicações de muitos sectores industriais, como o comércio eletrónico, *e-learning*, *fitness*, finanças, vendas, e muito mais.

De seguida serão apresentados vários exemplos de aplicações que implementaram com sucesso gamificação [26]:

### 3.3.1 *Nike+ Run Club*

A Nike, uma líder em todas as áreas de *fitness*, desenvolveu esta aplicação para ajudar a motivar e envolver os utilizadores nos seus treinos. Usando *leaderboards*, desafios e *badges*, a Nike+ permite aos utilizadores competir contra atletas de todo o mundo. Aproveitando os impulsos competitivos dos utilizadores, bem como fornecendo planos de treino, dicas de *fitness*, e a capacidade de seguir as suas melhorias pessoais.

### 3.3.2 *Duolingo*

Duolingo é o exemplo mais conhecido da utilização de gamificação para tornar a aprendizagem mais divertida e envolvente.

A aprendizagem de línguas pode ser uma atividade laboriosa e leva muito tempo a ver resultados. Utilizando metas, objetivos diários, e um número finito de vidas, pode-se motivar os utilizadores a usar a aplicação todos os dias e a continuar a aprender.

### 3.3.3 *Microsoft*

A Microsoft utilizou com sucesso um jogo simples e *leaderboards* para ajudar a reduzir o numero de *bugs* nas suas soluções.

Eles produziram um jogo onde o utilizador verificava a exatidão do código dos programas. Ao introduzir um *leaderboard* entre escritórios de todo o mundo, os seus funcionários ficavam mais envolvidos no seu trabalho e motivados a desafiar-se a um melhor desempenho.

### 3.3.4 *ChoreWars*

Esta aplicação ajuda os utilizadores a completar tarefas mundanas através da sua gamificação. As tarefas são transformadas em metas que devem ser completadas para que as personagens dos utilizadores continuem a sua viagem.

Os utilizadores são recompensados por completarem as suas tarefas, e um pouco de competição saudável pode ser introduzido através das tabelas de liderança.

## 3.4 SÍNTESE

Este capítulo começou por apresentar as ferramentas de gestão de vulnerabilidades, concluindo com uma comparação onde são retiradas as técnicas e processos mais adequadas. Por fim, foram apresentadas as várias técnicas de gamificação que foram identificadas como úteis para a aplicação e as aplicações que utilizam estas técnicas.

# 4

# ANALISE DE REQUISITOS

Para o projeto ser desenvolvido foi primeiro necessário elaborar uma análise de requisitos, classificando quais os mais e menos prioritários para a solução final, de forma a priorizar as suas implementações e definir objetivamente que funcionalidades pertencem a cada um deles. Nesta secção, serão explorados todos os conceitos fundamentais necessários para entender o funcionamento e desenvolvimento da aplicação web.

## 4.1 O PROBLEMA

À medida que a tecnologia e os softwares evoluem, a complexidade das redes cresce e a quantidade de vulnerabilidades conhecidas vai aumentando. Ferramentas de segurança que permitem a gestão e deteção de vulnerabilidades são cada vez mais significativas para que uma organização consiga acompanhar e permanecer protegida da exploração destas novas vulnerabilidades.

Os episódios de deteção de vulnerabilidades expõem vulnerabilidades existentes nos seus sistemas, possibilitando assim a sua gestão de modo a que as vulnerabilidades identificadas sejam remediadas.

Contudo, para que estes episódios de deteção sejam um sucesso a mitigação destas vulnerabilidades tem que ser efetuada de forma priorizada e continua, algo que em muitas organizações é pouco prioritário e como tal é comum existir muitos atrasos nas suas remediações.

Como se pode verificar no capítulo 3, esta situação é um desafio a que as soluções existentes de deteção, embora funcionais, não conseguem responder à falta de interesse por parte das organizações e os seus responsáveis para com a remediação de vulnerabilidades. Assim, é necessária uma ferramenta que permita a estas organizações acompanhar a evolução da mitigação destas vulnerabilidades continuamente e que ao mesmo tempo consiga incentivar os responsáveis pelas mitigações a realizar-las.

## 4.2 REQUISITOS FUNCIONAIS

Ao longo desta secção serão enumerados e detalhados os requisitos funcionais através da utilização de *user stories*, sendo cada requisito organizado numa tabela e priorizado de acordo com o método MoSCoW [27].

Procura-se com cada tabela ainda identificar cada uma das histórias para posterior referencias ao longo deste documento e identificar que permissão um utilizador necessita de possuir para poder aceder á função.

O método MoSCoW é uma técnica de priorização popular para a gestão de requisitos, normalmente utilizado com metodologias ágeis, que envolve a utilização de 4 prioridades para classificar os requisitos conforme a sua importância para o produto final .

### 4.2.1 *Must Have*

Estes requisitos funcionais são considerados essenciais e identificam as funcionalidades mínimas requeridas para o que projeto cumpra os objetivos mínimos definidos para que o projeto seja um funcional.

Tabela 4: Requisitos Must Have

| | |
|---|---|
| **Nome** | Importar dados do Sistema de Deteção de Vulnerabilidades |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de importar dados do sistema de deteção de vulnerabilidades. |
| **Nome** | Iniciar scan no Sistema de Deteção de Vulnerabilidades |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de iniciar scan no sistema de deteção de vulnerabilidades para validar remediações. |
| **Nome** | Resolver Vulnerabilidades no Sistema de Deteção de Vulnerabilidades |
| **Role** | Regular |
| **Titulo** | Como utilizador, quero ser capaz de resolver vulnerabilidades e que seja actualizado o no sistema de deteção de vulnerabilidades. |
| **Nome** | Criação de Grupos de Utilizadores |
| **Role** | Admin |
| **User Story** | Como utilizador, quero ser capaz de criar grupos de utilizadores. |
| **Nome** | Criação de Grupos de Vulnerabilidades |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de criar grupos de vulnerabilidades. |
| **Nome** | Atribuição de Vulnerabilidades a Utilizadores |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de atribuir vulnerabilidades a utilizadores. |
| **Nome** | Criação de Torneios |
| **Role** | Admin |
| **User Story** | Como utilizador, quero ser capaz de criar torneios. |
| **Nome** | Desbloquear Badges |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de desbloquear badges. |
| **Nome** | Ganhar Pontuação |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de receber pontuação por ações tomadas. |
| **Nome** | Perder Pontuação |
| **Role** | Regular |
| **User Story** | Como utilizador quero ser capaz de perder pontuação por acções tomadas. |

### 4.2.2 *Should Have*

Estes requisitos são características não críticas, mas ainda assim são importantes e acrescentam um valor elevado ao produto final.

Tabela 5: Requisitos Should Have

| | |
|---|---|
| **Nome** | Histórico de atividade |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ter acesso ao histórico de atividade. |
| **Nome** | Lançamento de Re-Scanning |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de fazer um lançamento de *scan* a uma máquina monitorizada. |
| **Nome** | Implementação de Sistemas de Prioridades |
| **Role** | N/A |
| **User Story** | Deve ser implementado um sistema de prioridades para as remediações. |
| **Nome** | Implementação de um Sistema de Autenticação |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser ter acesso a um sistema de autenticação. |
| **Nome** | Organizar, Pesquisar, Ordenar e Filtrar Vulnerabilidades |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de organizar, pesquisar, ordenar e filtrar vulnerabilidades. |

### 4.2.3 *Could Have*

Estes requisitos não acrescentam valor suficiente para serem consideradas importantes, mas podem ainda ser incluídos se os requisitos com maior prioridade forem implementados. Estes serão os primeiros a ser removidos da aplicação se necessário.

Tabela 6: Requisitos Could Have

| | |
|---|---|
| **Nome** | Escalação Automática de Vulnerabilidades |
| **Role** | N/A |
| **User Story** | Como utilizador quer ver as remediações a escalar de prioridade automaticamente quando a sua data limite é atingida. |
| **Nome** | Análise Estatística |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de visualizar uma análise estatística das remediações e vulnerabilidades. |
| **Nome** | Relatórios Técnicos e Administrativos |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de obter relatórios técnicos e administrativos. |
| **Nome** | Implementação de um Sistema de Follow-up |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser notificado com o seguimento da minha atividade. |
| **Nome** | Adicionar comentário no sistema de deteção de vulnerabilidades |
| **Role** | Regular |
| **User Story** | Como utilizador, quero ser capaz de adicionar comentários no sistema de deteção de vulnerabilidades. |

### 4.2.4 *Won't have*

Estes são requisitos que foram solicitados, mas que estão excluídos do âmbito de aplicação durante a duração prevista. Podem ser incluídos em fases futuras de desenvolvimento.

Como foi considerado nenhum dos requisitos funcionais não acrescentava valor suficiente para pelo menos ingressar na classificação "Could Have", nenhum foi considerado como "Won't have".

## 4.3 REQUISITOS NÃO-FUNCIONAIS

Os requisitos não-funcionais são requisitos que descrevem o produto numa forma qualitativa e não funcional.

### 4.3.1 *Atributos de Qualidade*

Os atributos de qualidade foram definidos com base na norma ISO-9126 [28]. Esta norma define um conjunto de parâmetros e objetivos para a avaliação da qualidade de um software.

- **Funcionalidade**: De forma a garantir a funcionalidade da aplicação, todas as funcionalidades apontadas como importantes, com classificação "Should Have"ou superior, deverão ser implementados.
- **Confiança**: De forma a garantir a tolerância a erros nos dados importados, tem-se como objetivo desenvolver um conjunto de testes que os validem e possivelmente os corrijam.
- **Usabilidade**: De maneira a facilitar a aprendizagem da aplicação por parte dos utilizadores, a plataforma deverá ter um funcionamento intuitivo. De forma a aumentar a produtividade de quem utiliza a aplicação, a informação relevante será toda agregada. Ter-se à também como objetivo ir de acordo com as preferências dos potenciais utilizadores.
- **Eficiência**: De forma a possibilitar a eficiência e aumentar a produtividade de quem utiliza a aplicação, a plataforma não deverá ter uma latência de não mais de 1 minuto entre um pedido e a sua resposta.
- **Portabilidade**: De modo a evitar a dependência sobre uma única fonte de dados, a aplicação irá suportar um sistema capaz de gerir vários sistemas de deteção de vulnerabilidades.

## 4.4 SÍNTESE

Ao longo deste capitulo foi apresentados resultados da análise de requisitos desenvolvida. Depois de apresentado o problema que o projeto procura responder foram descritos os vários requisitos funcionais por ordem de prioridade de acordo com o método MoSCoW. Por fim foram apresentados os vários atributos de qualidade que a plataforma tem de cumprir.

# 5

# ARQUITETURA

Neste capítulo será apresentado desenho e especificação da arquitetura da plataforma VMG, bem como, os seus componentes, o modo como interagem entre si e finalmente as ferramentas e tecnologias adotadas.

Em primeiro lugar, será feita uma apresentação da visão geral de todo o sistema, mostrando como se interligam as várias tecnologias e módulos que a compõem e como estes interagem entre si. Seguidamente, serão descritos mais detalhadamente os vários componentes da plataforma VMG.

Por fim, serão também apresentadas e identificadas as várias tecnologias e ferramentas adotadas para o funcionamento da plataforma e os motivos que levaram á sua adoção.

## 5.1 VISÃO GERAL

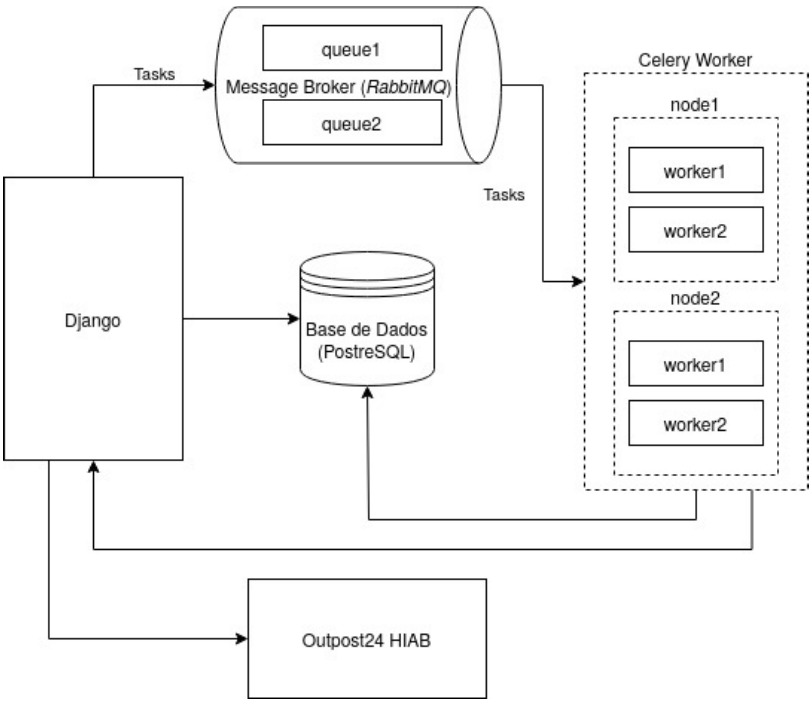



Figura 5: Arquitetura da plataforma VMG

Como representada na figura 5, a arquitetura da plataforma está dividia em quatro principais componentes com um outro componente externo, a plataforma HIAB do Outpost24 com a qual a aplicação web comunica de modo proporciona á aplicação os dados sobre as vulnerabilidades detectadas e os as várias operações relacionadas com o *scanner* de vulnerabilidades.

Visto que o Celery necessita de acesso à base de dados, modelos e lógica, foram definidas as tarefas do Celery Worker dentro da aplicação Django. No entanto, estas tarefas não são executadas na aplicação Django. Em vez disso, o Celery irá gerir servidores separados que podem executar as tarefas em simultâneo em segundo plano.

Uma vez que essa fila necessita de estar acessível tanto ao servidor web do Django, para adicionar novas tarefas, como aos servidores de *workers* para capturar as tarefas em fila, foi utilizado um outro servidor que serve como um *message broker*. Este servidor de corretor de mensagens utiliza Rabbitmq para manter as filas de tarefas.

A plataforma web Django cria uma tarefa, ou *task*, e informa o Celery para a adicionar à *queue* de tarefas. O Celery coloca essa tarefa no Message Broker, libertando a aplicação Django para continuar a executar outros processos. Num servidor separado, o Celery gere *workers* que vão executar tarefas que recolhem do *broker*. Estes escutam o servidor Rabbitmq à espera de recolher uma tarefa para processar.

Quando uma tarefa nova chega, um *workers* recolhe-a e depois processa-a, registando o resultado de volta na base de dados.

## 5.2 COMPONENTES

Nesta secção serão explorados em mais detalhe os principais componentes da plataforma.

### 5.2.1 *Django*

A plataforma Django é composta por diferentes módulos, ou Django Apps, cada um deles com a sua responsabilidade e conjunto de modelos, *views* e *templates*. A seguir serão listados as várias *apps* que constituem a plataforma:

**Manager**: Esta *app* é a responsável pela grande maioria de operações, nela é feita a gestão das vulnerabilidades e os seus respetivos projetos de remediação, alem disso é também nela feita a gestão de preferências, comentários, torneios, perfis e equipas. Como tal a grande maioria de modelos e *views* são do seu âmbito.

**Outpost24**: Esta é a aplicação é responsável pela comunicação entre a plataforma e o HIAB do Outpost24. Através dela são efetuadas as importações, pedidos de análise e outras operações que envolvam o HIAB. Esta separação permite a gestão de vulnerabilidades de várias ferramentas, tendo cada uma delas a sua *app* correspondente que faça a integração pelo Manager e o tratamento dos dados recebidos para o formato utilizado pela plataforma.

**Pinax**: Este conjunto de três *apps* da plataforma web Pinax são utilizados para a gestão de pontuação, a atribuição de *badges* e a gestão de notificações a utilizadores.

**Reports**: Esta aplicação é responsável pela criação de relatórios e pelo envio de emails com o relatório mensal aos utilizadores da plataforma.

**Auth**: Esta *app* foi utilizada para implementar o sistema de autenticação disponibilizado pelo Django. Ela gere as contas de utilizadores, grupos, permissões e sessões de utilizadores baseadas em *cookies*.

### 5.2.2 *Base de dados*

Nesta sub-secção será explicada a estrutura da base de dados utilizada pela plataforma, bem como a relação entre os dados da mesma.

Ao longo do projeto algumas funcionalidades e implementações imprevistas foram descobertas, pelo que, foi necessário planear e implementar estas apps externas como as Pinax na plataforma. Como tal a seguinte secção irá focar-se nos modelos que foram implementados durante o desenvolvimento, o diagrama de dados encontra-se disponível no apêndice B.

**Asset**: Este modelo, baseado no modelo de dados do outpost24, identifica os assets envolvidos na gestão de dados, em destaque estão os campos "exposed"que indica se a máquina está exposta, algo essencial para a gestão de vulnerabilidades e o campo "last_seen"que indica a ultima data a que o *asset* foi descoberto, utilizado para diferenciar um vulnerabilidade remediada de uma em falta.

**Tags**: Através deste modelo é possível a utilização de várias etiquetas para a escolha de preferências e a criação de filtros durante a criação de projetos de remediação. Uma *tag* pode ser asset, um produto ou plataforma, ambos campos do modelo Finding.

**Finding**: O maior modelo da plataforma é responsável pelos dados das vulnerabilidades, o seu tamanho deve-se à riqueza de dados que o *scanner* do outpost24 disponibiliza. De todos os seus atributos os mais relevantes para o funcionamento da aplicação são a "Likelyhood", um valor numérico de 0 a 40 que apresenta a probabilidade desta vulnerabilidade ser explorada, o "last_seen"pelo mesmo motivo que o dos *assets* e o campo "fixed"que indica se a vulnerabilidade foi considerada como resolvida.

**Remediation**: Este modelo representa os projetos de remediação da plataforma. Ele possui uma relação de *many-to-many* com o modelo Finding.

**Customer**: Utilizado para distinguir as várias organizações geridas pela plataforma e gerir os seus tempos limites de resposta. Cada customer possui uma relação de *one-to-many* com as restantes modelos, com a exceção dos utilizadores que podem pertencer a vários customers.

**Comment**: Este modelo é utilizado para possibilitar os comentários de utilizadores em projetos de remediação e Findings, possuindo uma relação de *many-to-many* com ambos.

**User e Profile**: Os utilizadores são representados com este modelo. Visto que este modelo foi implementado através da *framework* Django foi necessário implementar o modelo Profile de modo a associar uma imagem aos utilizadores com o campo "avatar".

**Team**: Este modelo é utilizado para associar vários utilizadores numa equipa. Cada equipa possui o seu nome e descrição.

**Preference**: Para implementar o sistema de preferências foi necessário a implementação deste modelo de modo a associar uma tag ou asset a um utilizador e a atribuir um nível a essa preferência.

**Source**: Este modelo é utilizado para registar as várias fontes de dados a que a plataforma deve aceder para que se possa efectuar a gestão de vulnerabilidades.

**Tournament**: O modelo dos torneios é utilizado para o registo e a sua associação aos utilizadores, através de uma relação *many-to-many*.

**Outpost24**: Este conjunto de 5 modelos são idênticos aos disponibilizados pelo HIAB. Para que não seja necessário um pedido às APIs do *scanner* sempre que um dos gráficos da página principal ou relatórios da aplicação são consultados foram implementados estes modelos. Eles são actualizados todos os dias atraves de um *cron job* ou sempre que existe uma alteração que justifique uma alteração destas estatísticas.

### 5.2.3 *Message Broker*

O Message Broker é solução para enviar e receber mensagens entre um cliente e os Celery Workers. Neste projeto o *broker* utilizado é o rabbitmq.

### 5.2.4 *Celery Worker*

O Celery Worker é um sub-processo que executa tarefas de forma assíncrona distribuídas através da uma *message queue*.

### 5.2.5 *Outpost24 HIAB*

O HIAB do Outpost24 é a ferramenta de gestão e *scanning* de vulnerabilidades que a plataforma comunica de modo a obter os dados e as estáticas sobre as vulnerabilidades encontradas.

## 5.3 FERRAMENTAS E TECNOLOGIAS ADOTADAS

Nas secções seguintes, serão apresentadas as ferramentas e tecnologias adotadas para o desenvolvimento deste projeto e o modo como tais ferramentas possibilitaram a implementação da plataforma que respondem-se aos requisitos levantados no capítulo 4.

### 5.3.1 *Django*

Django [29] é uma *framework* web e *open-source* desenvolvida em Python utilizada para o desenvolvimento de aplicações web. Esta *framework* é popular pela sua robustez, simplicidade de uso e também as diferentes opções relativas à escalabilidade, tais como executar a base de dados num servidor separado ou até utilizar *clustering* ou *load-balancing* para distribuir a aplicação por múltiplos servidores.

O Django utiliza uma arquitetura MVT (Model-View-Template) para construir as suas aplicações web[30]. Esta arquitetura é uma variação Django da famosa estrutura MVC (Model-View-Controller) onde quando o servidor Django recebe um pedido, o pedido é mapeado para a View apropriada. A View vai então buscar os dados através dos Models, preenche o Template e retorna-lo de volta para o utilizador.

- **Model**: funciona como uma camada ORM (Object-Relational-Mapping) que faz o mapeamento dos modelos de dados definidos através de classes Python em tabelas e entradas na base de dados.
- **View**: Numa forma semelhante aos *controllers* do modelo MVC, uma View é responsável pelo processamento dos pedidos HTTP e pelo envio de uma resposta válida. A View acede aos dados do modelo e depois constrói uma reposta transmitindo a um Template acesso aos dados, realizando antes qualquer processamento necessário nos dados obtidos.
- **Template**: A camada Template é utilizada para definir a estrutura de uma pagina com placeholders para representar dados que mais tarde serão populados por uma View. A camada de modelo é semelhante à camada View do MVC.

Uma aplicação desenvolvida com a *framework* Django pode ser expandida através de Django Apps, que são módulos em Python independentes que adicionam novas funcionalidades à plataforma a ser desenvolvida. Algumas Apps estão disponibili-

zadas com a versão base Django e fornecem soluções para casos de uso geral, tais como a gestão de autenticações de utilizadores, gestão de sessões, um painel de administração entre outras.

Apesar da falta de experiência em desenvolvimento de software através desta *framework*, após o estudo das suas capacidades e a sua prévia utilização pela empresa em outros projetos foi decidido que esta seria a utilizada na implementação da plataforma VMG.

### 5.3.2 *Pinax*

Pinax [31] é uma plataforma web open-source desenvolvida com o intuito de disponibilizar um grande conjunto de recursos para o projetos em Django. Alguns desses recursos são Apps para novas funcionalidades ou componentes, são ainda disponibilizados temas visuais, modelos já predefinidos, ou projetos iniciais que podem ser utilizados como base para um novo website.

#### 5.3.2.1 *Pinax Badges*

Uma das Apps da plataforma Pinax utilizadas neste foi a Pinax Badges [32] que forneceu ao projeto uma aplicação de badjes bem testada, documentada e conhecida que permite a atribuição de badjes a utilizadores.

Esta aplicação fornece simples abstrações sobre a atribuição de badges a utilizadores para completar tarefas, incluindo badges de vários níveis, e repetíeis.

#### 5.3.2.2 *Pinax Points*

A aplicação Pinax Points [33] fornece um componente à plataforma com um sistema de pontuação, posições e níveis.

Ela confere a uma plataforma a capacidade de registar a atribuição de pontos a objetos, neste caso utilizadores, da plataforma. Além disso, também é possível controlar e verificar as posições dos utilizadores relativamente aos outros possibilitando a implementação de um *leadersboard*.

#### 5.3.2.3 *Pinax Notifications*

Por fim, foi utilizada a aplicação Pinax Notifications [34] para a gestão de notificações a utilizadores. A plataforma necessita de um sistema de notificações quando certos eventos tenham ocorrido e que através também de um conjunto de opções sobre a forma como estas notificações são recebidas.

### 5.3.3 *Numpy*

NumPy[35] é uma biblioteca para a linguagem de programação Python, que suporta o processamento de grandes matrizes, juntamente com uma grande colecção de funções matemáticas de alto nível para operar sobre estas matrizes.

### 5.3.4 *Matplotlib*

Matplotlib[35] é uma biblioteca *open-source* para criação de gráficos e visualizar dados em geral, feita para e da linguagem de programação Python. Ela proporcionou à plataforma a capacidade de desenhar gráficos e diagramas, para que possa compreender e apresentar melhor os dados visualmente nos relatórios.

### 5.3.5 *Pandas*

Pandas[36] é uma biblioteca de Python para manipulação e análise de dados. Em particular, oferece estruturas e operações para manipular tabelas numéricas e séries temporais. Também possibilita a criação de tabelas, e editar ou adicionar colunas ou linhas às tabelas. Pandas foi utilizado em conjunto com o Numpy e o Matplotlib para a criação de relatórios pela plataforma.

### 5.3.6 *PostreSQL*

Por predefinição, aplicações Django são configuradas para armazenar dados numa de base de dados SQLite.

Embora funcione bem pequenas cargas de dados, para a quantidade de dados que a plataforma iria necessitar de gerir não era possível atingir um desempenho aceitável. Como tal depois de um estudo das opções disponíveis foi-se decidido implementar uma base de dados em PostreSQL.

PostreSQL [37] é SGBD (Sistema de Gestão de Bases de Dados), open-source, com uma forte reputação pela sua fiabilidade e flexibilidade.

Este SGBD oferece tipos de dados mais complexos que outras bases de dados como o MySQL, e permite aos objectos herdar propriedades, que embora vantajoso no aspeto de controlo que o utilizador possui, isto também leva a que a utilização do PostgreSQL seja mais complexo. Os seus principais benefícios [38] que levaram á sua adoção para este projeto são:

- **Tipos de dados suportados**[39]: O elevado numero de tipos de dados suportados relativamente a outros SGBDs. Por exemplo: numérico, data/tempo, carácter, *boolean*, enumerado, JSON, XML, *arrays*, *ranges*, etc.
- **Desempenho e escalabilidade**: Através da utilização de *multiversion concurrency control* é possível a ocorrência simultânea de operações de escrita e leitura. Algo essencial para uma aplicação com um volume elevado de dados como esta.
- **Suporte de linguagens de programação**: A sua compatibilidade e apoio a múltiplas linguagens de programação, incluindo Python, Java, JavaScript, C/C++ e Ruby.
- **Maior suporte e economicamente viável**: Graças ao seu aspeto *open-source* e grande comunidade ativa.

### 5.3.7 *Celery*

O Celery[40] é uma *asynchronous task queue*, *open-source*, baseada na passagem distribuída de mensagens que implementa filas de tarefas como um mecanismo para distribuir trabalho entre *threads* ou máquinas, também conhecidos como *workers* para que estas sejam executadas.

As unidades de execução, chamadas *tasks*, são executadas simultaneamente num único ou mais "*workers*", que monitorizam as *message queues* à espera de *tasks* para as executar. As *tasks* podem ser executadas de forma assíncrona (em segundo plano) ou síncrona, onde a plataforma aguarda até estarem concluídas.

#### 5.3.7.1 *Rabbitmq*

O RabbitMQ[41] é um software *open-source* que através do protocolo "Advanced Message Queuing Protocol"(AMQP) efectua a transferência de mensagens entre aplicações[42].

Uma mensagem pode incluir qualquer tipo de informação. Neste sendo, informação sobre um processo ou tarefa que devem ser executados por um *worker* do Celery. O *message broker* armazena as mensagens de um *producer*, a aplicação Django, até que um Celery Worker, conhecido no protocolo AMQP como *consumer*, se conecte e retire uma mensagem da fila.

Este protocolo permite aos servidores web responderem rapidamente aos pedidos, em vez de serem forçados a executar procedimentos com recursos locais, o que pode atrasar o tempo de resposta. O *message broker* também possibilita distribuir uma mensagem por múltiplos *consumers* para equilibrar cargas entre *workers*.

### 5.3.8 *Redis*

Redis [43] é uma estrutura para o armazenamento de dados em memória utilizada como uma base de dados, cache e message broker. Graças a natureza *in-memory* o Redis é capaz de um alto desempenho em comparação com os sistemas de bases de dados que gravam cada entrada no disco. Deste modo foi implementado no projeto de modo a possibilitar a transmissão de dados para a *progress bar* durante a importação de dados do *scanner* de vulnerabilidades.

### 5.3.9 *Bootstrap*

Para o design da interface da plataforma foi decidido utilizar o Bootstrap[44], uma *framework* de front-end desenvolvida de modo a oferecer um conjunto de componentes personalizados para a implementação de paginas Web através de código HTML, CSS e JavaScript.

#### 5.3.9.1 *NiceAdmin*

NiceAdmin[45] é uma *template* modelo de administração e painel de instrumentos baseado na última versão da estrutura Bootstrap (v5.1.3). Forneceu á aplicação um design limpo e intuitivo de modo a melhorar a experiência do utilizador. NiceAdmin vem com muitos elementos e componentes como tabelas, gráficos, um formulário de login, formulário de registo, página de perfil e muitos mais.

## 5.4 SÍNTESE

Neste capítulo foi apresentada uma visão geral da arquitetura de todo o sistema VMG mostrando como se interligam os módulos que o compõem, de seguida, foram apresentadas descrições detalhadas dos vários módulos e o modo como interagem entre si. Finalmente foram também identificadas e detalhadas as ferramentas e tecnologias adotadas para o funcionamento da plataforma

# 6

# DESENVOLVIMENTO

Este capítulo descreve em pormenor a implementação dos sistemas da plataforma web e na forma como os componentes e sistemas interagem entre si. Será ainda analisado o modo como as ferramentas e arquitetura apresentada no capítulo 5, foram colocados em pratica e utilizados.

## 6.1 COMUNICAÇÃO COM OUTPOST24

Conforme especificado no capítulos 4 e 5, a integração e comunicação com o HIAB constitui uma parte essencial da plataforma, possibilitando a importação de vulnerabilidades e os seus detalhes, estatísticas, máquinas na infraestrutura da organização e operações essenciais tais como, o lançamento de um *scan*, o comentário de uma vulnerabilidade ou a sua resolução, etc.

Após o estudo da documentação da aplicação do Outpost24, percebeu-se que esta integração poderia ser feita através das duas APIs disponibilizadas pelo HIAB:

- A **API legacy**[46] em XML, apresenta um grande conjunto de funções para interagir com a plataforma, embora lhe faltem operações cruciais para atingir os objetivos propostos, como por exemplo, um pedido que devolva todas as vulnerabilidades completas com os seus detalhes.
- A nova **API REST**[47] disponibilizada de modo a responder à falta de operações que a anterior não era capaz de disponibilizar e ainda melhorar a solução anterior que tinha ficado desorganizada e incompleta com as variadas adições de funções que foi sofrendo.

Ambas as APIs permitem personalizar as saídas e solicitar diferentes tipos de informação do sistema. Como se pode ver na ilustração abaixo, figura 8, a interface gráfica do Outpost24 também executa as suas operações a partir da API XML.

Neste projeto foi necessário proceder a uma combinação de operações com as duas APIs de modo a conseguir atingir os objetivos propostos de modo eficaz e completo.

Um exemplo desta combinação seria na importação de vulnerabilidades encontradas nos sistemas monitorizados, que necessitou de um pedido à API REST de forma a obter uma lista dos IDs de todas as vulnerabilidades seguido de pedidos à API Legacy para os detalhes de cada vulnerabilidade.

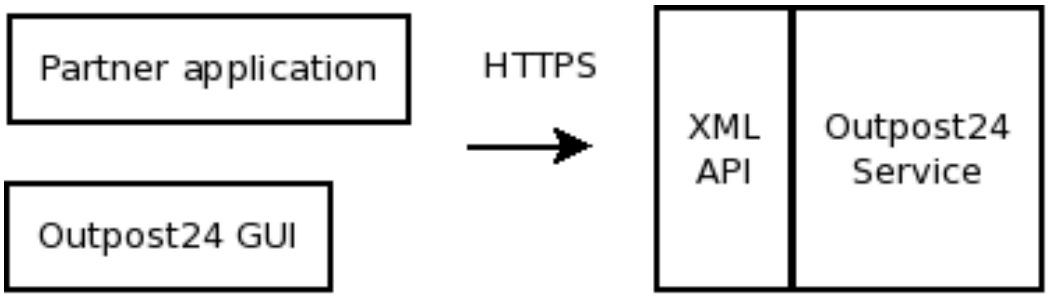



Figura 6: Pedidos API ao Outpost24

### 6.1.1 *Importação*

A importação de dados é realizada de duas diferentes formas, automaticamente, através de um *cron job* agendado para todos os dias e manualmente, através da página "Sources", apresentada na figura 7.

Nesta página podem-se ver as *log sources* configuradas a que a aplicação VMG acede para descarregar todos os dados necessários, bem como comunicar.

Nesta versão inicial da aplicação, com o HIAB do Outpost24 para por exemplo: solicitar novos episódios de deteção de vulnerabilidades, declarar vulnerabilidades como falsos positivos ou aceitá-las, comentar as vulnerabilidades, etc...

Sob a perspetiva técnica, a importação de dados do HIAB é realizada através da *app* "outpost24"da plataforma, que inicia a importação com um pedido à API REST, do domínio estipulado na *log source*, de modo a obter um *bearer token* que mais tarde será utilizado na importação das vulnerabilidades e assets.

Depois deste pedido inicial procede-se à importação dos assets.

Este processo é o mais simples de realizar visto que se resume à realização de um pedido à nova API onde é devolvido uma lista dos assets que de seguida é iterada e as correspondentes entradas na base de dados são criadas ou caso já existam e tenham sido modificadas, actualizadas.

A importação das Findings, ou vulnerabilidades descobertas pelo Outpost24, encontra-se dividida em dois pedidos devido à inabilidade da API REST em devolver uma lista de todas as vulnerabilidades com todos os campos necessários.

Deste modo, é realizado o primeiro pedido á nova API que devolve uma lista de todas as vulnerabilidades, que de seguida é percorrida e por cada vulnerabilidade um pedido à API XML é realizado obtendo-se assim todos os detalhes necessários. Com os resultados obtidos é criada uma entrada que é a seguir associada a um asset importado anteriormente e criadas, caso não existam, novas *tags* para o produto e plataforma da vulnerabilidades. Estas entradas na tabela "tag"são depois utilizadas no sistema de preferências e na criação de projetos de remediação.

Para terminar este processo é verificada a entrada de forma que, caso já exista uma entrada com o mesmo identificador e uma data de ultima visualização diferente, ela seja actualizada, caso contrario, se não for descartada por já existir, esta é adicionada para uma lista de registo na base de dados em *bulk*.

Depois deste pedido faz-se a importação dos comentários dos utilizadores nas vulnerabilidades, denominados como *comments* tanto pela plataforma como pelo Outpost24. Infelizmente, em vez de realizar um único pedido para todas as vulnerabilidades, o método mais eficaz de as importar, passa por percorrer todas as vulnerabilidades encontradas e para cada vulnerabilidade realizar um pedido à API Legacy pelos comentários associados a esta. Caso existam, é criada ou actualizada a entrada na base de dados.

Por fim, são importadas as várias estatísticas e contagens que o outpost disponibiliza para serem posteriormente utilizadas na geração de tabelas e gráficos de *dashboards* e relatórios. Estes dados incluem uma variedade de listas e números como: *top findings*, *top solutions*, remediações e vulnerabilidades.

Para iniciar a importação de dados pode selecionar-se o botão "Importar tudo"o que dará inicio à importação de todos os dados de cada *log source* ou selecionar-se a opção de importação de uma determinada fonte. O progresso da importação é demonstrado através de uma barras de progresso implementada através do Celery Progress[48], uma biblioteca para a implementação de barras de progresso em aplicações Django/Celery.

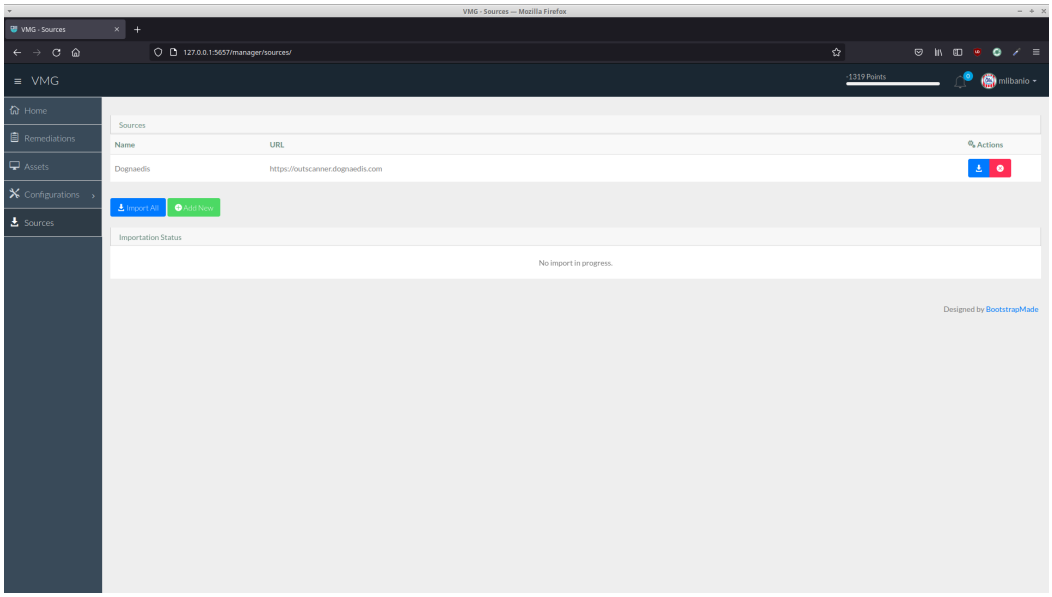



Figura 7: Página das log sources.

Para configurar uma nova fonte de dados, é necessário navegar para a página de criação de *log sources*. Nesta página, será pedido o preenchimento da informação sobre a nova fonte, nomeadamente:

- Name - O nome da *log source*.
- URL - O URL da API do HIAB.
- Customer - O cliente a que pertence esta fonte de registo.
- User - O nome do utilizador que irá aparecer quando o VMG publicar os comentários no HIAB do Outpost24.
- API Key - A chave da API.

## 6.2 CLIENTES

De forma a possibilitar o uso da aplicação para a gestão de vulnerabilidades de várias organizações foi necessário estandardizar o conjunto de vulnerabilidades, *assets* e projetos de remediação de uma organização num único grupo, um cliente.

Cada cliente é constituído pelas *log sources* correspondentes possibilitando à plataforma realizar a gestão de vários *scanners* de vulnerabilidades ao mesmo tempo. Possui ainda um atributo de nome. Os utilizadores que têm acesso aos dados associados às fontes indicadas e por fim, cada cliente possui as *Service level*

*agreement* (SLA), as datas de expiração dos projetos de remediação, e os riscos mínimos correspondentes aos SLAs.

Para aceder aos dados relacionados com os clientes na plataforma é necessário navegar até a página "Clientes"demonstrado na figura 8. Esta página só é acessível a utilizadores autorizados.

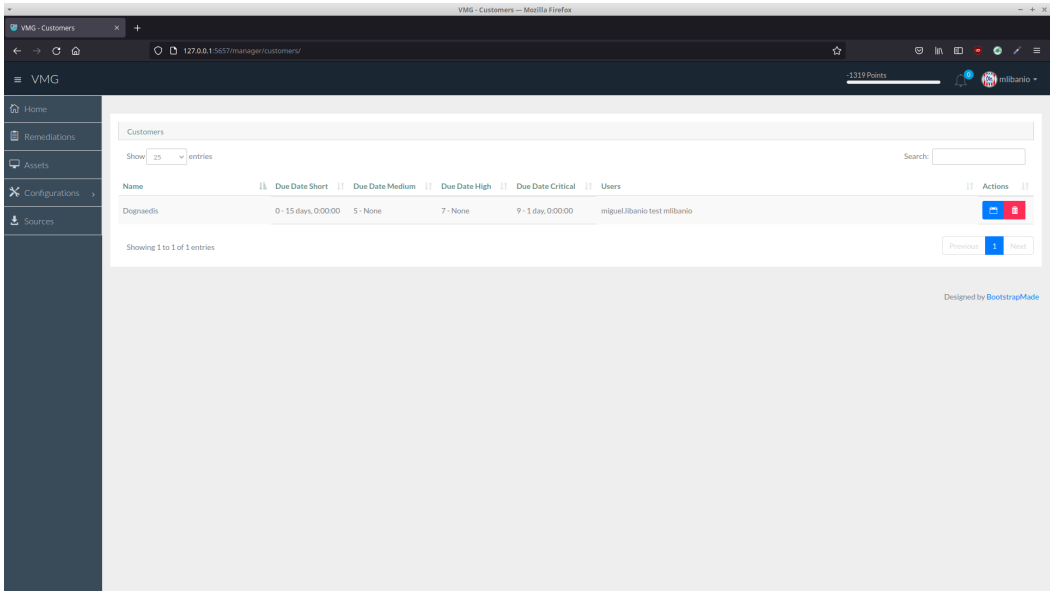



Figura 8: Página dos clientes.

Nesta página são apresentados os clientes geridos pela aplicação e os SLAs com as prioridades correspondentes de cada cliente. Estes prazos podem ser alterados na página "Due Dates"de cada cliente, onde, como podemos ver na figura 9 são pedidas as configurações das SLAs criticas, altas, medias ou baixas. Na hipótese de um campo ter sido deixado vazio, o atributo correspondente será considerado como desativado e nenhuma data será atribuída quando um projeto com a criticidade equivalente é criado.

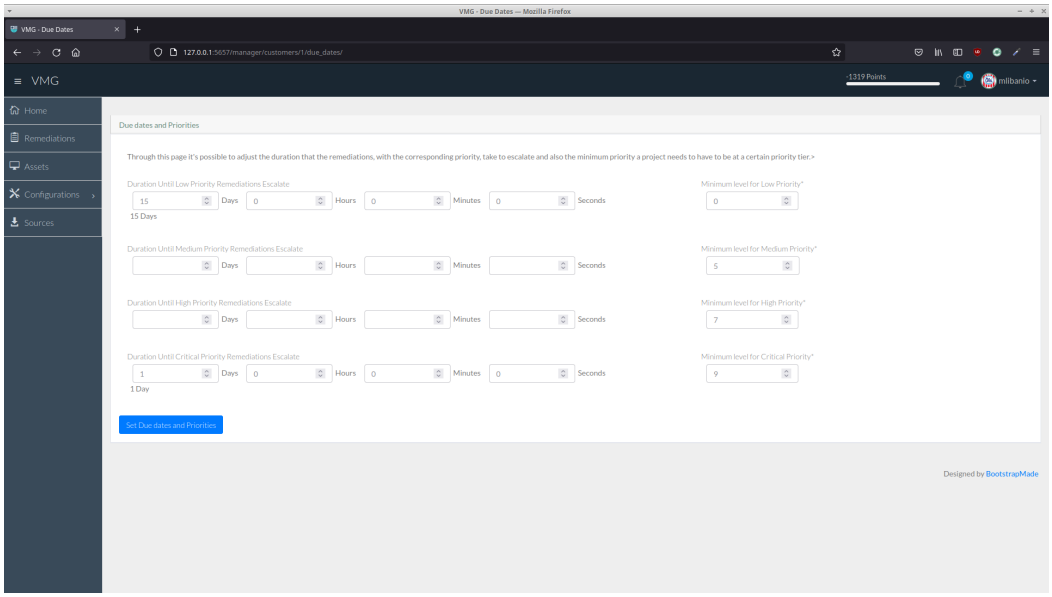



Figura 9: Página das configurações de SLAs.

## 6.3 PREFERÊNCIAS

O sistema de preferências proporciona não só aos utilizadores um método de informar os analistas responsáveis pela atribuição de projetos das suas preferências relativamente ás plataformas, produtos e *assets*, mas também, um mecanismo para personalizar o algoritmo de criação automática de projetos de remediação que, ao fazer a atribuição de equipas vai procurar atribuir a equipa que mais preferências têm em comum com as vulnerabilidades envolvidas.

Caso não consiga atribuir uma equipa, o algoritmo, vai percorrer todos os utilizadores do cliente e atribuir o projeto aos utilizadores com mais preferências em comum.

Na página de preferências, demonstrada na figura 10, o utilizador pode indicar as tecnologias e *assets* que prefere criando uma nova preferência através da página de criação ou arrastando uma das já existentes para primeiro ou ultimo lugar na tabela das preferências.

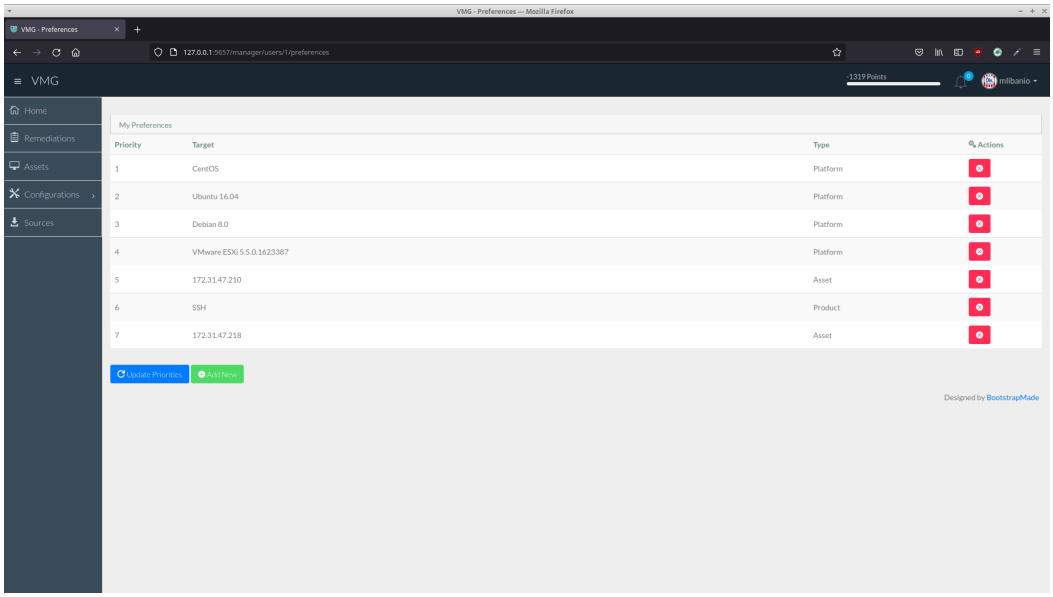



Figura 10: Página das preferências.

## 6.4 UTILIZADORES E EQUIPAS

A página relacionada com a gestão de utilizadores, tal como na página dos clientes, é só acessível a utilizadores autorizados. De forma a controlar os seus acessos os utilizadores foram divididos em três grupos através do sistema de permissões incorporado no Django, nomeadamente:

- Os utilizadores no grupo "Regular"são utilizadores com menos privilégios, embora, os privilégios que possuem lhes permitam o acesso a maior parte das funções disponibilizadas pela aplicação, tais como, o lançamento de novos episódios de deteção, a criação de projetos, comentar vulnerabilidades e também a visualização de quase todos os dados do seu cliente.
- Os utilizadores com privilégios superiores aos anteriores pertencem ao grupo "Priveleged". Neste patamar, para alem dos privilégios que os utilizadores regulares já possuem, o grupo "Priveleged"também pode gerir os torneios, aceder e modificar as configurações do seu cliente como as SLAs e as equipas de utilizadores.
- Por fim existem os utilizardes "Admin". Este grupo é composto pelos analistas responsáveis pela gestão das vulnerabilidades nos vários clientes e o que diferencia os seus privilégios do grupo anterior é a escala do seu acesso, nomeadamente, tem acesso a todos os clientes, à gestão da importação de

dados e por fim à criação e gestão de contas que, entre outros, engloba também a alteração dos clientes associados a um determinado utilizador.

Nesta página, apresentada na figura 11, é possível criar novos utilizadores ou aceder à página de perfil dos mesmos e ainda à criação de equipas. Equipas estas que, como anteriormente referido, servem como forma de indicar ao sistema de atribuição automática de projetos de remediação, quais os utilizadores que deve dar prioridade, ao agrupar ou atribuir novos projetos.

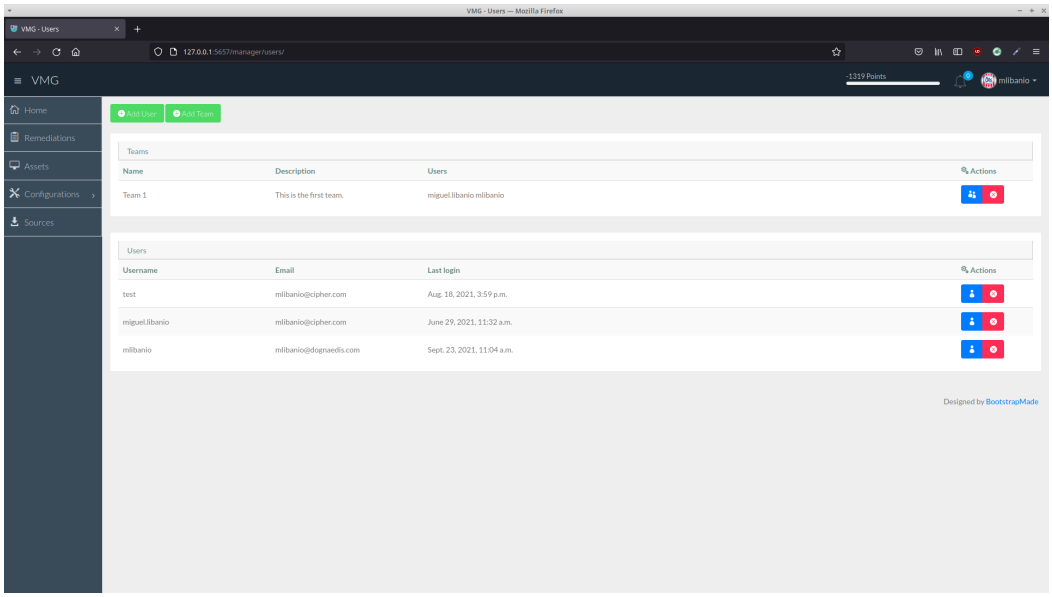



Figura 11: Página dos utilizadores e equipas.

## 6.5 PROJETOS DE RESOLUÇÃO

Os projetos de remediação permitem às equipas coordenar o progresso das remediações de vulnerabilidades. Estes dão visibilidade às tarefas necessárias aos utilizadores responsáveis e permite que possam facilmente acompanhar e medir o progresso do trabalho de remediação.

Os projetos simplificam a definição de prioridades, a remediação e o acompanhamento do progresso da remediação, mostrando ao utilizador o estado da remediação e os seus detalhes.

As métricas do projeto são automaticamente atualizadas à medida que as vulnerabilidades são alteradas ou deixam de existir através de pedidos de resolução, realizados pelo utilizador ou pelo sistema, de sincronização da plataforma que realiza e importa os dados do HIAB todos os dias por meio de um *cron job*, ou sempre

que existe uma atualização na vulnerabilidade de modo que possa visualizar e acompanhar as ações das equipas de remediação.

Na página de remediações são apresentados todos os projetos de remediação dos clientes a que o utilizador pertence.

Como demonstrado na figura 12, para além das informações habituais, tais como, o nome do projeto, a data de expiração, o estado atual e os utilizadores a si alocados, cada projeto é também classificado com uma prioridade calculada com base no *CVSS Score* mais elevado das suas vulnerabilidades e caso alguma dessas estejam expostas a sua severidade é ainda aumentada com mais um valor.

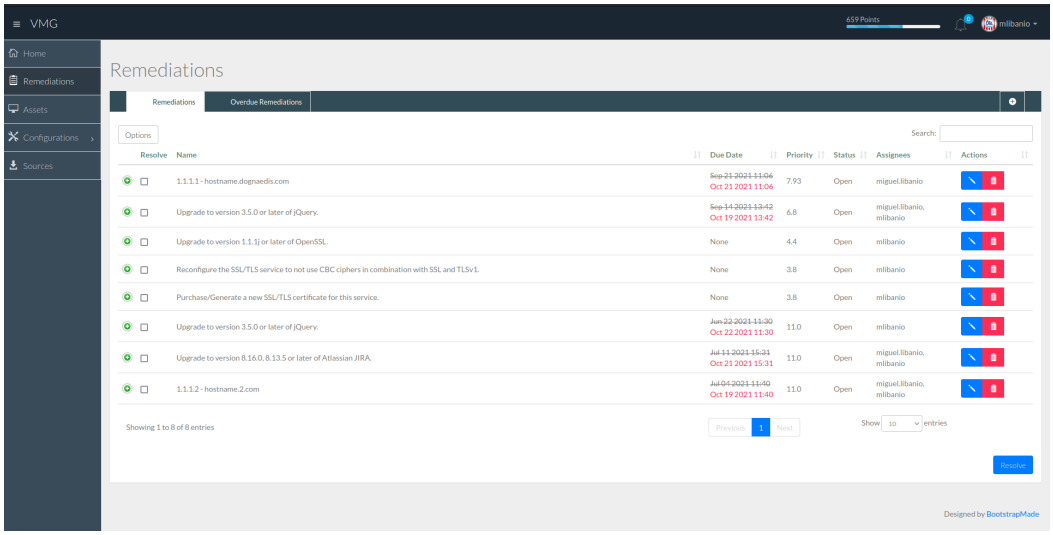



Figura 12: Página das projetos de remediação.

Cada projeto, caso o SLA correspondente à sua prioridade tenha sido configurado, terá um prazo de resolução correspondente. Se data for alcançada e o projeto de remediação possuir ainda vulnerabilidades por resolver, a plataforma penalizará os utilizadores do projeto, removendo uma quantidade substancial de pontos, esta quantidade é calculada com base na prioridade. Se um projeto tiver passado a sua data de expiração uma vez e tiver sido criado à mais de 3 meses, ele aparecerá na secção "Remediações em atraso", demonstrado na figura 13.

Nesta secção os utilizadores administrativos serão informados de projetos em atraso de forma que medidas para corrigir a situação possam ser tomadas, caso essas medidas se mostrem ineficazes os utilizadores poderão então encerrar estes projetos, debitando uma grande quantidade de pontos dos utilizadores responsáveis pela remediação.

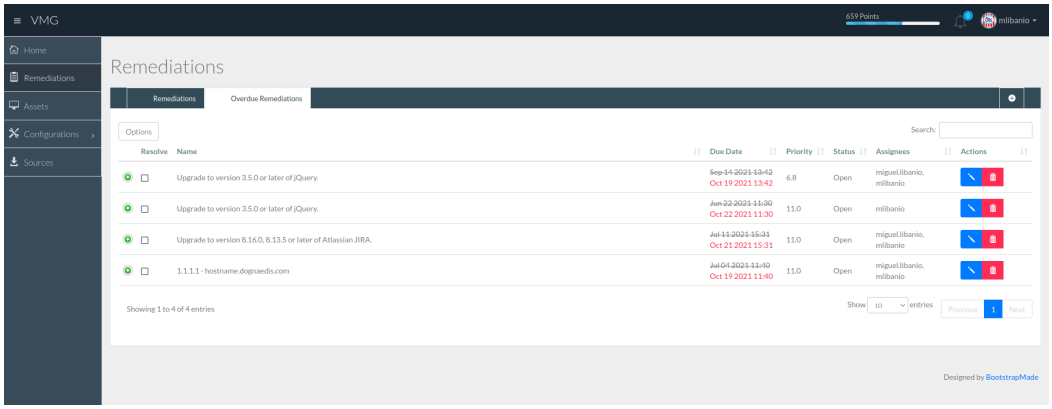



Figura 13: Separador dos projetos em atraso.

Cada projeto de remediação pode ser expandido de modo a ver os comentários a si associados e as vulnerabilidades que o constituem, que por sua vez, podem ainda ser expandidas de modo visualizar os seus detalhes. Nesta janela, demonstrada na figura 14, são apresentados os vários detalhes importados pela aplicação e também os comentários associados á vulnerabilidade criados ou removidos. A cada comentário de uma vulnerabilidade criado ou removido por um dos utilizadores, a plataforma contacta o HIAB do Outpost24 de modo a atualizar a plataforma das novas alterações, enviando um pedido à API REST para criar um comentário no nome do utilizador configurado na *log source* ou para remover o comentário do Outpost24.

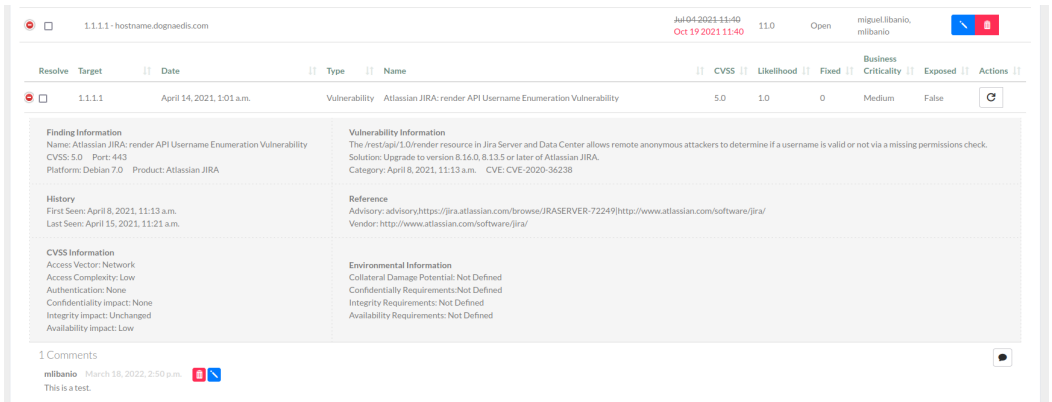



Figura 14: Detalhes de uma vulnerabilidade.

### 6.5.1 *Criação de projetos*

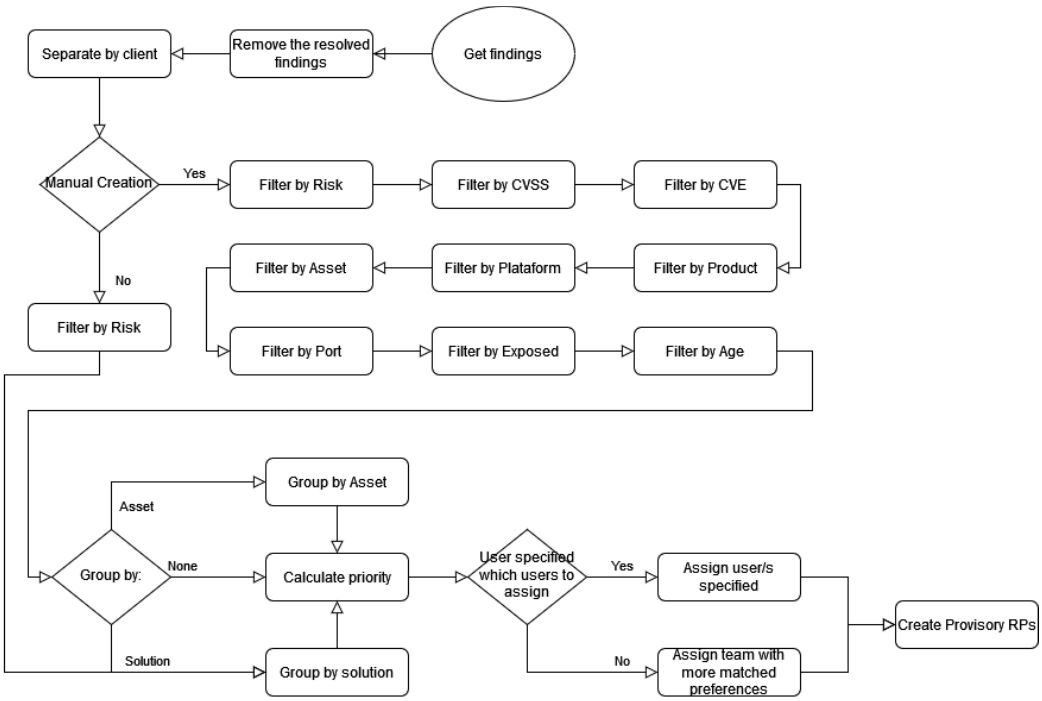



Figura 15: Algoritmo de Criação de projetos de Remediação.

Como apresentado na figura 15, a criação de projetos de remediação pode ser efetuada através de duas opções.

A primeira, consiste na criação e atribuição automática de projetos de remediação e depois apresenta ao utilizador uma página com pré-visualizações dos projetos, apresentada na figura 16, de modo a que o utilizador possa selecionar os que deseja criar. Os restantes serão eliminados.

Nesta opção, os projetos serão gerados pela aplicação através de um conjunto de filtros predefinidos, ou seja todas as vulnerabilidades com o atributo risco maior ou igual a 3 e agrupando as vulnerabilidades pelas suas soluções e atribuído os projetos a equipas, não maiores do que 5 membros, ou no caso de não existirem equipas disponíveis os 3 primeiros utilizadores com as preferências mais correspondentes.

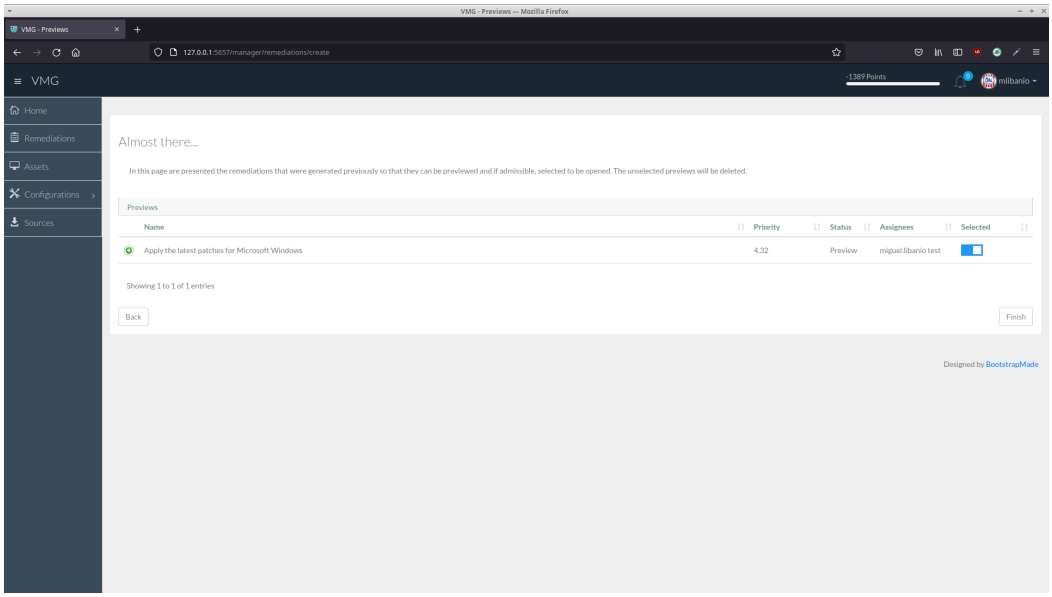



Figura 16: Página de pré-visualização.

A segunda opção permite a criação manual de projetos através de um conjunto de filtros opcionais onde o utilizador pode selecionar as vulnerabilidades com os atributos especificados e ainda, indicar como pretende agrupar os resultados: por solução, *asset* ou mesmo não agrupar as vulnerabilidades criando um projeto por cada vulnerabilidade. O último separador permite a seleção do utilizador a quem deseja atribuir estes projetos, se o deixar vazio será atribuído automaticamente.

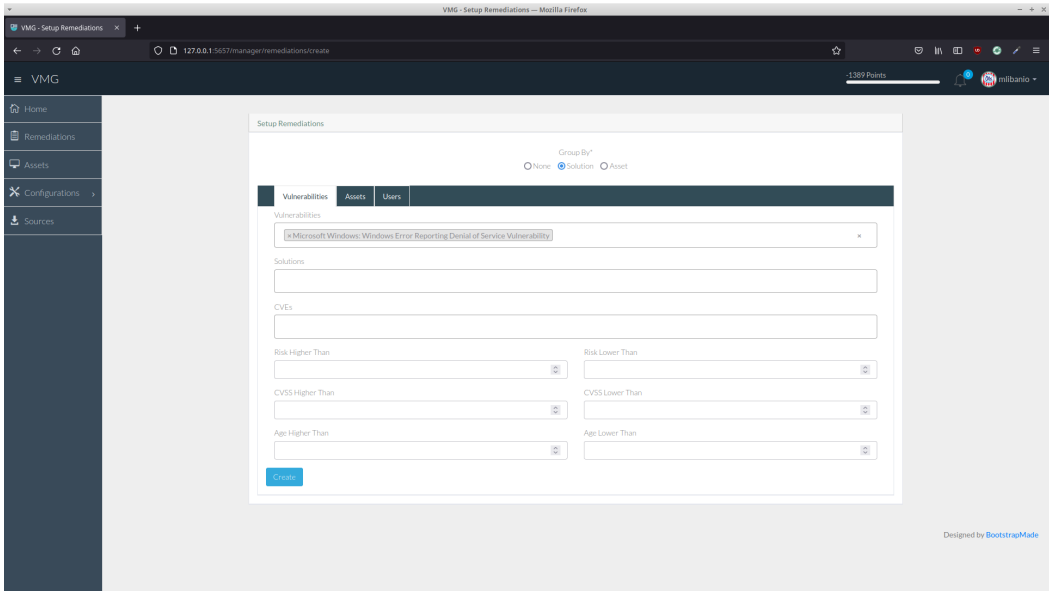



Figura 17: Página de para a criação manual de projetos de remediação.

Tal como na criação automática, depois do utilizador selecionar os filtros que deseja aplicar, ver figura 17, o utilizador será redireccionado para a página de pré-visualizações onde terá de selecionar os projetos que serão criados.

### 6.5.2 *Resolução de projetos*

A resolução de projetos de remediação é efetuada solucionando as vulnerabilidades associadas a um determinado projeto. Esta resolução envolve a seleção e classificação de uma ou mais vulnerabilidades em três classificações, a saber:

**Correcção** onde será lançado uma análise de deteção de vulnerabilidades a fim de verificar se a vulnerabilidade se encontra resolvida e se assim for, a plataforma irá atualizar a informação do projeto e das vulnerabilidades envolvidas atribuindo os pontos aos utilizadores responsáveis pelo projeto. Para uma vulnerabilidade ser considerada como corrigida pela plataforma necessita, de não ser encontrada na última análise, mas a máquina que pertence sim ou então deve ter sido declarada como corrigida pelo Outpost24. Isto deve-se ao facto que os *scanners* do Outpost24 têm dificuldades a diferenciar uma vulnerabilidade desaparecida com uma resolvida e como tal não é marcada devidamente com resolvida.

**Falso Positivo** a plataforma irá declarar a vulnerabilidade como um falso positivo, nesta opção, à semelhança da anterior, a plataforma comunicará com o HIAB do Outpost24 através da sua API XML de modo a atualizar a entrada e relativamente à outra opção, atribui menos pontos.

**Aceitação** declara a vulnerabilidade como aceite pelo número de dias estipulado pelo utilizador ou caso selecionado para sempre. Esta opção atribui também menos pontos aos utilizadores responsáveis.

## 6.6 ASSETS

Como verificado no capítulo 3, o *assets* são um componente essencial para a gestão de vulnerabilidades, como tal, na página de *assets*, apresentado na figura 18, são apresentados todos os *assets* dos clientes a que o utilizador pertence e se, como na página de remediações, o utilizador selecionar a opção de expansão de uma entrada da tabela dos *assets*, as vulnerabilidades pertencentes ao dispositivo selecionado serão exibidos.

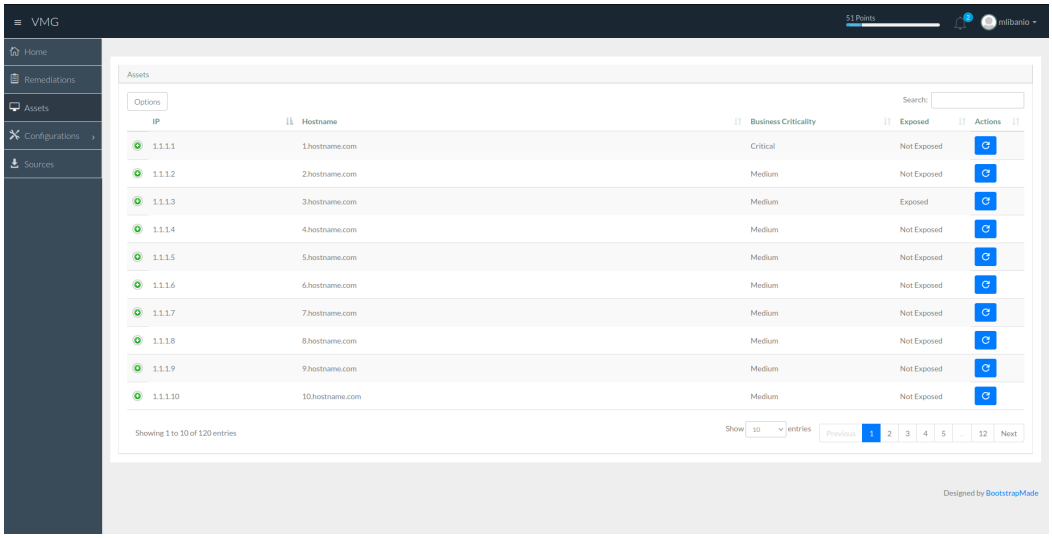



Figura 18: Página dos assets.

Cada dispositivo tem também a opção de lançar uma nova análise de vulnerabilidades com uma das políticas disponíveis, um conjunto de *scripts* de deteção de vulnerabilidades, esta opção, irá lançar uma nova análise através do HIAB para a máquina escolhida e lançar também uma tarefa para seguidamente importar os resultados e atualizar os resultados antigos com os novos dados.

## 6.7 SISTEMAS DE GAMIFICAÇÃO

A gamificação foi implementada através de três sistemas e interligada com os vários gráficos de atividade na página principal da plataforma e os vários relatórios de atividade. Nesta secção serão explorados os três principais sistemas de gamificação:

### 6.7.1 *Pontuação*

De modo a incentivar a utilização da plataforma, os utilizadores são recompensados através de um sistema de atribuição de pontos. Esta recompensa é atribuída à medida que os utilizadores completam ações predestinadas.

Para a implementação do sistema de pontos e níveis, foi utilizada o Pinax Points, que proporcionou à plataforma a capacidade de associar pontos aos utilizadores da aplicação através de um conjunto de funções e tabelas para a gestão de pontuação de cada utilizador.

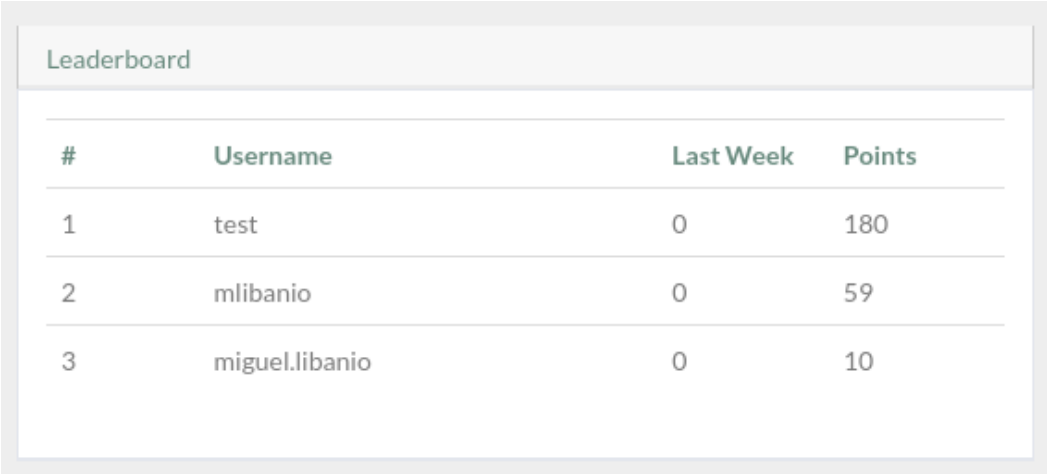



Figura 19: Leaderboard.

Além disso, esta *app* também permitiu à plataforma disponibilizar aos utilizadores as suas posições num *leadersboard*, apresentada na figura 19 na página principal através das *templatetags* disponibilizadas para esse efeito.

Tabela 7: Tabela do Cálculo da Pontuação

| **Acção** | **Pontuação** | **Mínimo** | **Máximo** |
|---|---|---|---|
| Verificar vulnerabilidade | 5 | 5 | 5 |
| Resolver Vulnerabilidade | Likelyhood * 2.5 | 30 | 100 |
| Declarar Falso Positivo | Likelyhood * 2 | 20 | 80 |
| Aceitar Vulnerabilidade | Likelyhood * 0.5 | 1 | 20 |
| Resolver projeto de Remediação | Priority * 2.5 | 30 | 100 |
| Comentar | 10 | 10 | 10 |
| Faltar data de Expiração | Priority * -5 | -60 | -200 |
| Remediações em Atraso Fechado | Priority * 7.5 | -90 | -300 |
| Criar projeto de Remediação | 50 | 50 | 50 |
| Torneios | Definido na Criação | N/A | N/A |

Como referido anteriormente, este sistema permitiu também a implementação de níveis, outro mecanismo destinado mensurar e recompensar a interação que o utilizador tem com a aplicação. A cada nível, a complexidade e dificuldade das metas aumenta, desafiando o utilizador a ir mais longe.

Os níveis são representados através de 4 badges, cada um com uma dificuldade acrescida relativamente à anterior, requerendo assim que os pontos necessários para atingir o nível seguinte seja maior que o anterior.

### 6.7.2 *Badges*

Estas representações visuais dos objetivos atingidos pelos utilizadores, como indicado no capítulo 3, representam um método mais impactante que o anterior em recompensar e encorajar os utilizadores a resolver as vulnerabilidades do seu ambiente.

Este sistema foi também implementado através de uma das apps disponibilizadas pela plataforma Pinax, a *app* "Pinax Badges", permite à plataforma a atribuição de *badges* a um utilizador por completar determinadas tarefas, incluindo *badges* com vários níveis para tarefas de crescente dificuldade como a acumulação de pontos ou resolução de vulnerabilidades. A tabela 8 apresenta os *badges* implementados e as metas necessárias para desbloquear cada um:

Tabela 8: Tabela dos Badges

| **Titulo do Badge** | **Acção** |
|---|---|
| Bronze | Atingir 300 Pontos |
| Silver | Atingir 1000 Pontos |
| Gold | Atingir 2000 Pontos |
| Platinum | Atingir 5000 Pontos |
| Bronze Finds Resolver | Resolver 1 Vulnerabilidade |
| Silver Finds Resolver | Resolver 10 Vulnerabilidades |
| Gold Finds Resolver | Resolver 30 Vulnerabilidades |
| Platinum Finds Resolver | Resolver 50 Vulnerabilidades |
| Bronze Rps Resolver | Resolver 1 projeto |
| Silver Rps Resolver | Resolver 5 projetos |
| Gold Rps Resolver | Resolver 10 projetos |
| Platinum Rps Resolver | Resolver 20 projetos |
| Bronze Commenter | Comentar Pela Primeira Vez |
| Silver Commenter | Comentar 5 vezes |
| Gold Commenter | Comentar 10 vezes |
| Platinum Commenter | Comentar 20 vezes |

O Pinax Badges permite definir os *badges* em subclasses de uma classe principal. Por exemplo, nos badges de pontuação, são atribuídas quatro classificações que tem por base, os pontos que um utilizador tem associados a si. Sendo a diferença entre o nível mais baixo, o de bronze, e mais elevado, o de platina, a quantidade de pontos necessária alcançar.

### 6.7.3 *Torneios*

Finalmente, temos a página de torneios, demonstrada na figura 20, onde podemos criar ou editar torneios. Um torneio é um evento que ocorre dentro do prazo estipulado, finalizado esse prazo, o utilizador ou utilizadores com mais pontos atribuídos durante o torneio recebem um prémio definido pela organização. Os torneios aparecerão no calendário na página inicial um mês antes da sua realização com as informações sobre os torneios em curso para que os utilizadores sejam alertados com o devido tempo.

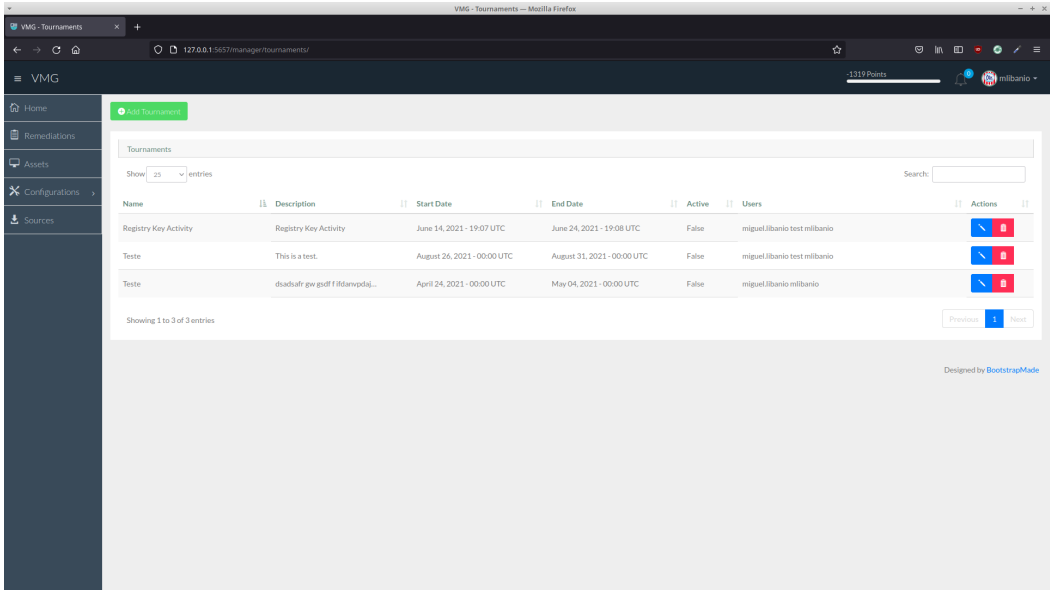



Figura 20: Página dos torneios.

## 6.8 HISTÓRICO E NOTIFICAÇÕES

O sistema de notificações da plataforma foi implementado através da aplicação "*Pinax Notifications*"disponibilizada pela plataforma Pinax.

Através desta *app* de gestão de notificações a utilizadores, foi possível criar e gerir notificações para os utilizadores quando eventos são despoletados, como por exemplo, a sua posição final de um determinado torneio, quando um projeto de remediação é alocado ao utilizador ou quando um episódio de deteção pedido pelo utilizador foi terminado.

As notificações são apresentadas aos utilizadores através de um pequeno *widget* no formato de um sino no canto superior da interface gráfica. Neste *widget* o utilizador

é apresentado as últimas 5 notificações e um atalho para a página de notificações onde estão listados todas as notificações, cada uma com o seu titulo, descrição e data de criação.

O registo das atividades geradas pelos utilizadores na plataforma foi implementado através da *app* "Django Activity Stream"[49]. Esta aplicação foi concebida de modo a possibilitar a criação e exibição de ações geradas numa plataforma Django.

Tal como no sistema de notificações, sempre que uma de certas ações são disputadas, como a criação de um novo projeto de remediação, um pedido de uma nova análise de vulnerabilidades, um projeto ou vulnerabilidade com a qual um utilizador está envolvido é comentada, uma vulnerabilidade é remediada ou um torneio finaliza, essa atividade será registada no *activity stream* dos utilizadores envolvidos.

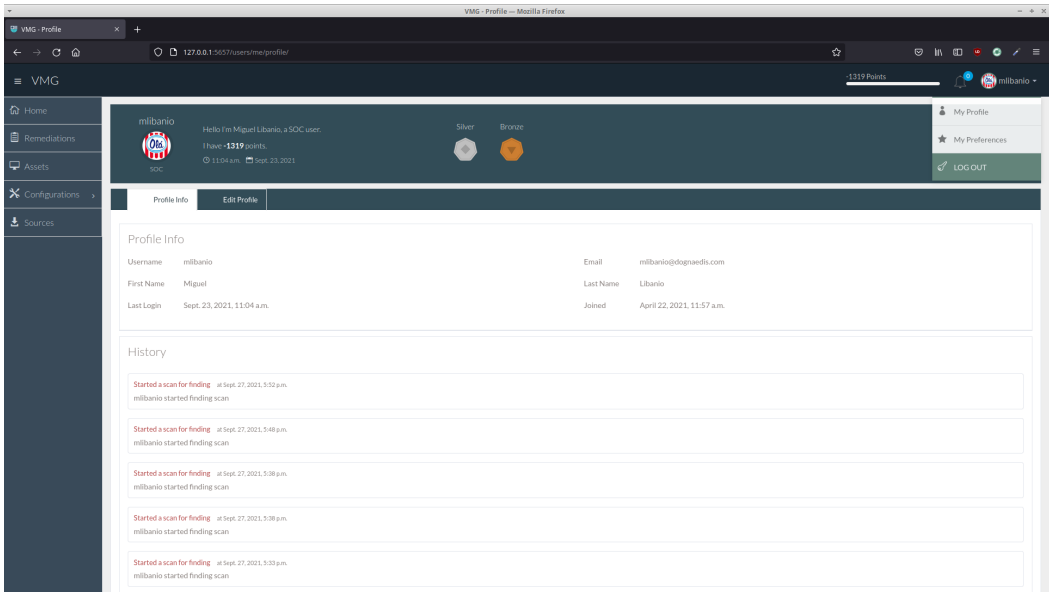



Figura 21: Página de perfil.

Estes registos podem então depois ser observados na página de perfil do utilizador, demonstrada na figura 21, na qual um utilizador para alem de ver o seu histórico de ações também é capaz de verificar os detalhes da sua conta. Nesta página é possível ao utilizador alterar a sua password e os detalhes mais básicos da sua conta como o email, nome, imagem de perfil, etc. Caso o utilizador da sessão aberta pertença ao grupo *Admin*, o utilizador poderá ainda alterar os clientes associados à conta do perfil e caso necessário alterar também a password.

## 6.9 GRÁFICOS E TENDÊNCIAS

A última etapa do desenvolvimento focou-se nas representação gráfica da informação e nos entregáveis da plataforma. Nesta secção será abordada a implementação dos gráficos e tabelas e as várias estatísticas obre a sua atividade e a gestão da vulnerabilidade na plataforma.

Para a implementação dos vários gráficos disponibilizados na aplicação web foi utilizado o Chart.js[50], uma biblioteca JavaScript gratuita e *open-source* para visualização de dados.

Sendo que os gráficos são partes constituintes da página principal e dos relatórios da plataforma, optou-se apenas por descrever a página principal visto que esta os engloba a todos agrupando-os pelas suas pretendidas funções.

Ao entrar na página principal o utilizador é apresentado um conjunto de gráficos e estatísticas de modo a proporcionar aos utilizadores uma visão geral sobre o estado da sua infraestrutura e o seu desempenho na remediação de vulnerabilidades.

O primeiro destes gráficos, representado na figura 22, é denominado como "Points Awarded"e visa facultar o desempenho dos cinco melhores utilizadores e o do utilizador, caso este não faça parte do grupo, através de um gráfico de linhas onde são representados o total de pontos que cada utilizador possuiu ao longo do tempo e quando é que um determinado *badge* foi alcançado.

Este gráfico procura fomentar o espírito de competição de cada utilizador comparando o seu desempenho relativamente aos melhores da organização.

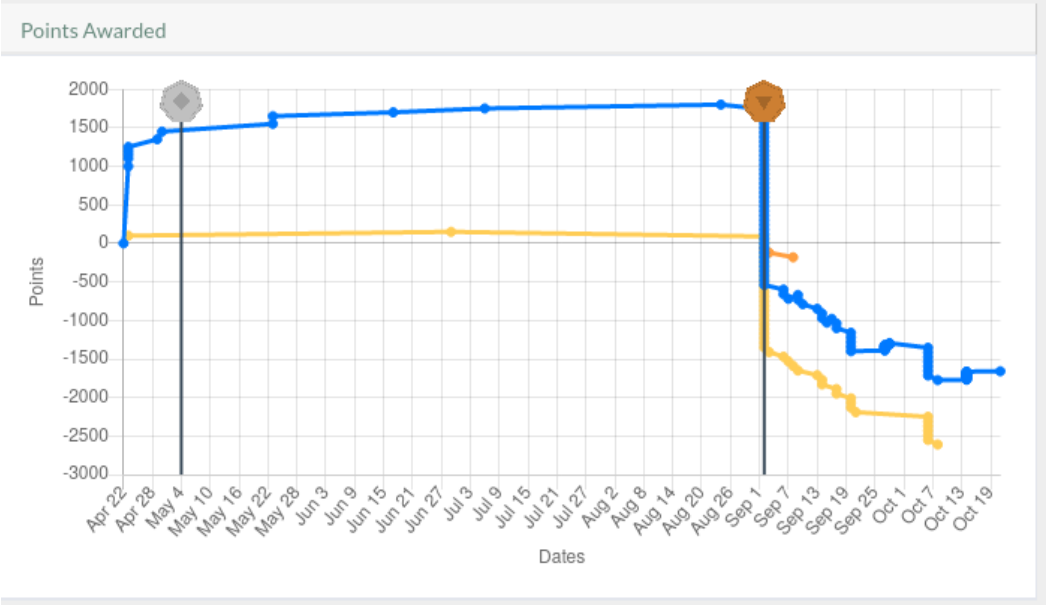



Figura 22: Gráfico da pontuação

Ainda no topo da página, e seguindo e mesmo raciocínio, é apresentada uma pequena tabela como os dez utilizadores com as melhores pontuações e a posição do utilizador com a sessão aberta. Para finalizar a secção dedicada à gamificação existe uma janela com o calendário, implementado através da plugin FullCalendar[51], onde são assinalados todos os eventos e datas limite para projetos do utilizador.

Para consultar os detalhes dos torneios ou obter uma vista rápida do estado dos seus projeto é necessário apenas retornar à janela do calendário e alternar a *tab* selecionada na pequena janela.

Para apresentar a visão geral sobre o estado atual do primeiro cliente do utilizador e o seu desempenho na remediação de vulnerabilidades nesse cliente foi também implementado um conjunto de gráficos e estatísticas com esse intuito.

Os primeiros destes gráficos estão representados na figura 23, apresentam a quantidade de vulnerabilidades detetadas pelo Outpost24, o número de remediações efetuadas nessas vulnerabilidades e a diferença entre este dois, representado pelo valor Growth, ao longo das semanas nos últimos três meses. Para complementar estes valores é também calculada a tendência que estes valores vão seguir no próximo mês, as tendências são calculadas através da soma da média de todos os valores anteriores com o valor atual, incentivando-se assim um maior esforço na remediação e diminuição de vulnerabilidades dos sistemas monitorizados.

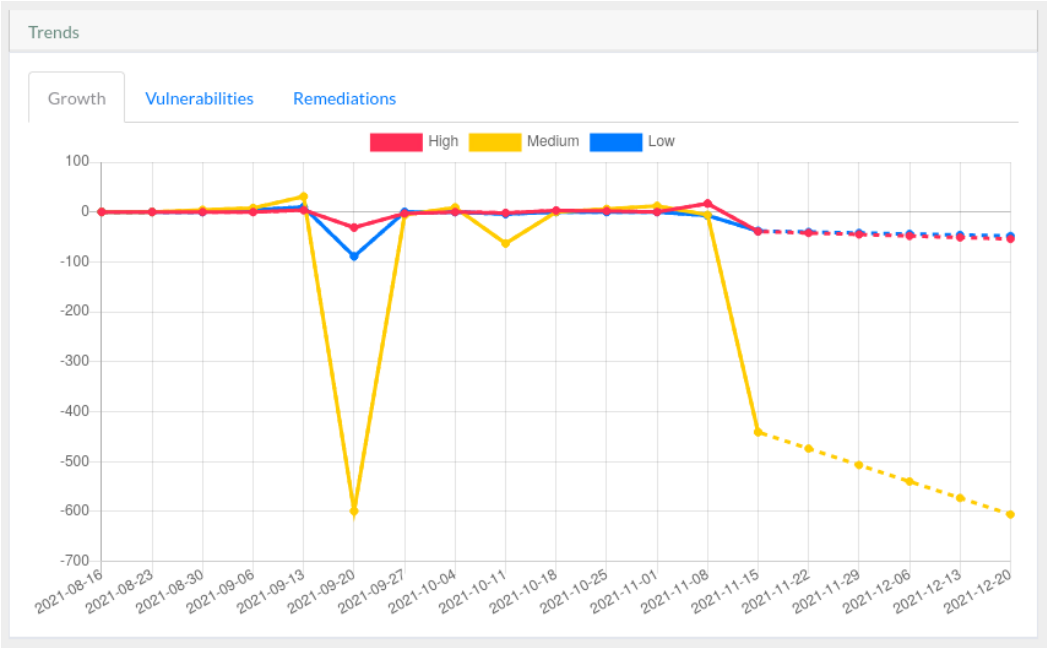



Figura 23: Gráfico de tendências.

De seguida é disponibilizada a tabela das vulnerabilidades mais importantes, onde são listadas, por importância, as dez vulnerabilidades que apresentam um risco mais imediato aos sistemas, cada uma detalhada com a sua posição, nome, CVE, risco e número de *assets* afetados.

Existe ainda a tabela das melhores soluções, demonstrada na figura 24, é apresentada uma lista das dez melhores soluções com o seu nome, produto, número de vulnerabilidades e *assets* envolvidos, cada uma destas entradas possui ainda uma opção que leva o utilizador para a página de criação de projetos de remediação com os filtros e opções já preenchidos com a solução correspondente.

A acompanhar esta tabela está um gráfico de sectores com a percentagem de vulnerabilidades resolvidas pelas soluções apresentadas.

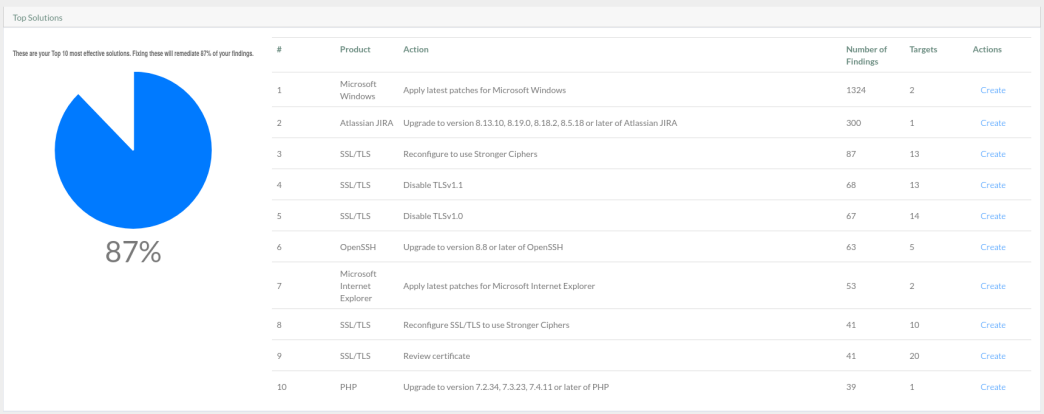



Figura 24: Tabela e gráfico das melhores soluções.

Por fim é apresentado o sumario do risco de todos os *assets* monitorizados, na janela do "Risk Summary". Nesta janela, apresentada na figura 25, o utilizador pode inspecionar que risco é que os assets da sua organização estão expostos, com um pequeno sumário das ações e cuidados o utilizador deve tomar para remediar e priorizar os seus dispositivos.

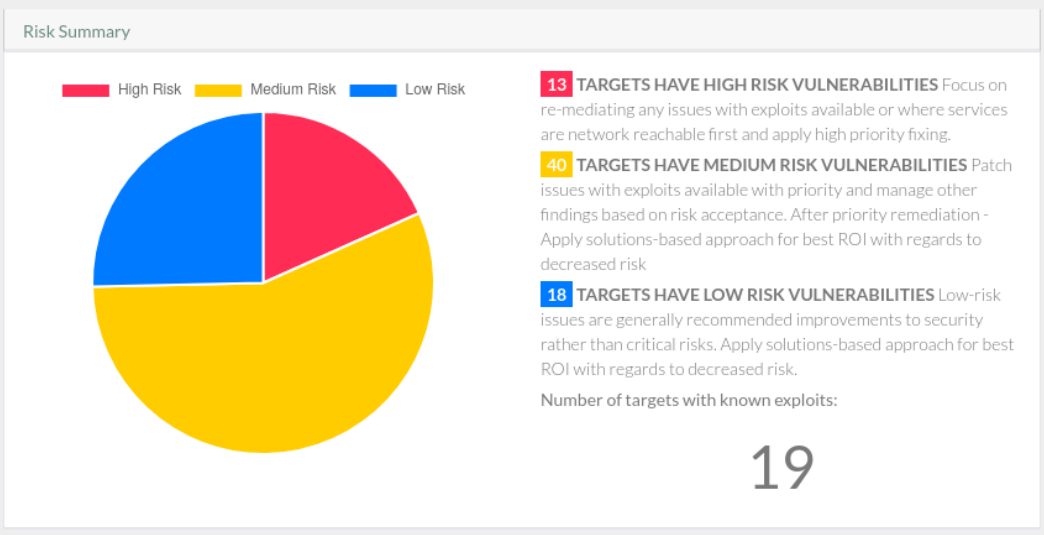



Figura 25: Sumario do risco de todos os assets.

A acompanhar esta janela está a janela com os detalhes dos cinco *assets* em maior risco. Enumerando a sua posição, IP, quantidade de vulnerabilidades altas, medias e baixas, a soma de todas as suas Likelyhoods e a maior Likelyhood que as suas vulnerabilidades possuem.

## 6.10 RELATÓRIOS DE ATIVIDADE

De modo a disponibilizar relatórios relativos ao desempenho global ou singular dos utilizadores na remediação de vulnerabilidades foi implantado três distintos tipos de relatórios, cada um com o seu objetivo e âmbito.

Na página principal é possível descarregar dois tipos de relatórios sobre a atividade do utilizador e organização.

Sendo o primeiro destes o relatório administrativo sobre a remediação de vulnerabilidades no contexto do cliente, que recorrendo às tecnologias enumeradas no capítulo 5, nomeadamente o Numpy, Matplotlib e Pandas é criado um ficheiro no formato PDF onde os gráficos apresentados na página inicial são novamente inseridos. Um exemplo deste relatório pode ser encontrado no apêndice C.

O segundo tipo de relatório visa mostrar ao utilizador o seu desempenho na plataforma e fazer uma comparação com o desempenho dos outros utilizadores. Como para o relatório anterior este relatório pode ser visualizado através da apêndice D que disponibiliza um exemplo de como este tipo de relatórios são constituídos.

Na primeira página do documento é disponibilizado um resumo do trabalho feito pelo utilizador através de tabelas como os *badges* alcançados, um gráfico "Points Awarded"e a tabela Activity onde informação como o numero de vulnerabilidades e projetos resolvidos, comentários realizados, pontos ganhos e *scans* realizados. Ainda nesta página o utilizador pode consultar a tabela *leadersboard* e também a tabela com as remediações mais importantes a serem resolvidas e o seu progresso. Na página seguinte o utilizador pode consultar os seus comentários e os projetos que conseguiu resolver.

Além disso, no final de cada mês, os utilizadores recebem um email com um resumo mensal sobre a sua actividade do mês e várias sugestões para os próximos projetos de remediação que o utilizador se deve focar. Este email, está disponível no formato PDF no apêndice E.

## 6.11 SÍNTESE

No decurso de este capítulo foi descrito em pormenor a implementação da plataforma VMG e na forma como os componentes e sistemas estão relacionados entre si.

A descrição da implementação começa com a exposição de como os vários sistemas de integração com o HIAB estão implementados. De seguida são detalhados os vários componentes da plataforma, nomeadamente os clientes, preferências, utilizadores e as suas equipas, chegando por fim a um dos componentes cruciais para a gestão de vulnerabilidades, os projetos de remediação e como a gestão de vulnerabilidades e *assests* é realizada.

Foram ainda, explicados os vários sistemas de gamificação implementados e como estes estão relacionados com os outros vários componentes da aplicação.

Para concluir, e depois de descrito como notificações, histórico e os vários gráficos da aplicação foram implementados, são apresentados os vários relatórios de atividade.

# 7

# VERIFICAÇÃO E VALIDAÇÃO

Com o desenvolvimento concluído, procedeu-se à verificação e validação da plataforma desenvolvida. Assim, ao longo deste capítulo será validado que a plataforma se encontra devidamente desenvolvida, analisando primeiro quais os requisitos funcionais que foram cumpridos e os que não foram e seguidamente, se os atributos de qualidade estavam satisfeitos e o seu impacto global no resultado final.

Esta verificação será efetuada através de testes funcionais e validações relacionadas com os atributos qualitativos estipulados ao longo capítulo 4.

## 7.1 TESTES FUNCIONAIS

De modo a aferir o cumprimento dos requisitos funcionais foram implementados vários testes funcionais, organizados como definido no método MoSCoW. Através deles foi possível averiguar e avaliar o cumprimento dos requisitos propostos.

Os testes funcionais podem ser consultados nas tabelas 9 a 11. Cada tabela contém o título do teste e raciocínio para a validação do requisito.

A tabela 9 descreve os teste funcionais implementados de modo avaliar e validar o cumprimento dos requisitos críticos para o projeto, descritos no grupo "Must Have".

Tabela 9: Testes Funcionais - Must Have

| | |
|---|---|
| **Título** | Importação de vulnerabilidades |
| **Validação** | A importação de vulnerabilidades deverá ocorrer com sucesso sempre que a sua Log Source esteja configurada corretamente. |
| **Título** | Verificar remediação de vulnerabilidades |
| **Validação** | Sempre que necessário deverá ser lançado um pedido de verificação ao *scanner* de vulnerabilidades e os dados da plataforma atualizados depois do episódio de deteção concluir. |
| **Título** | Declarar uma vulnerabilidade como falso positivo |
| **Validação** | Vulnerabilidades podem ser consideradas como falso positivo e declaradas resolvidas e a plataforma HIAB atualizada com esses detalhes. |
| **Título** | Declarar uma vulnerabilidade como aceite |
| **Validação** | Vulnerabilidades podem ser aceites e consideradas como resolvidas conforme necessário e a plataforma HIAB atualizada com esses detalhes. |
| **Título** | Criar um projeto de remediação manualmente |
| **Validação** | Um projeto de remediação deverá ser criado manualmente através dos filtros selecionados e as pré-visualizações selecionadas. |
| **Título** | Criar um projeto de remediação automaticamente |
| **Validação** | Deverá ser possível efetuar um pedido para a criação automática de projetos de remediação. Este pedido deverá devolver uma lista de pré-visualizações com equipas entre três e cinco elementos. |
| **Título** | Editar um projeto de remediação |
| **Validação** | Caso necessário deverá ser possível alterar o nome, descrição, solução, prioridade, vulnerabilidades ou os utilizadores encarregues de um projeto. |
| **Título** | Remover um projeto de remediação |
| **Validação** | Caso necessário deverá ser possível remover um projeto de remediação. |
| **Título** | Resolver um projeto de remediação |
| **Validação** | Quando todas as vulnerabilidades de um projeto são resolvidas esse projeto deverá ser designado como resolvido e os utilizadores envolvidos recompensados com a devida pontuação. |
| **Título** | Criar um Torneio |
| **Validação** | Um utilizador do grupo "Priveleged"ou "Admin"deverá poder criar um novo Torneio. |
| **Título** | Acabar com um Torneio |
| **Validação** | Quando um torneio atinge a data de finalização os utilizadores envolvidos deverão ser notificados da sua posição e dos resultados finais. |
| **Título** | Remover um Torneio |
| **Validação** | Caso necessário deverá ser possível remover um torneio. |

Na tabela 10 são apresentados os teste funcionais implementados para avaliar o cumprimento dos requisitos não críticos, mas ainda assim são importantes definidos no grupo "Should Have".

Tabela 10: Testes Funcionais - Should Have

| | |
|---|---|
| **Título** | Pedir um episódios de deteção a uma máquina e obter os resultados |
| **Validação** | Quando um pedido para um episódio é efetuado a tarefa que verifica a sua conclusão e importa os resultados deverá também ser lançados. |
| **Título** | Criar um utilizador |
| **Validação** | Um utilizador do grupo "Priveleged"ou "Admin"deverá poder criar um novo utilizador. |
| **Título** | Editar um utilizador |
| **Validação** | Um utilizador deverá ser capaz de alterar os seus dados, nomeadamente o seu *username*, primeiro e último nome, email e avatar. Caso o utilizador seja do grupo "Admin", para alem dos dados enumerados anteriormente, também deverá ter acesso aos clientes do utilizador. |
| **Título** | Remover um utilizador |
| **Validação** | Caso necessário deverá ser possível remover um utilizador. |

A tabela 11 são apresentados os teste funcionais implementados para avaliar o cumprimento dos requisitos menos importantes definidos pelo grupo "Could Have".

Tabela 11: Testes Funcionais - Could Have

| | |
|---|---|
| **Título** | Faltar uma data limite para a resolução de um projeto e escalar esse projeto |
| **Validação** | Se uma data limite de um projeto de remediação for atingida, esse projeto deve ser atualizado com a nova data, como estipulado no cliente desse projeto e os pontos dos utilizadores associados devem ser penalizados. |
| **Título** | Fechar um projeto em atraso |
| **Validação** | Caso um projeto de remediação tenha sido criado à mais de três meses, um utilizador no grupo "Admin"poderá encerrar este projeto debitando a quantidade correta da pontuação dos utilizadores responsáveis. |
| **Título** | Comentar uma vulnerabilidade |
| **Validação** | Quando um utilizador comenta numa vulnerabilidade, esse comentário deve ser criado tanto na plataforma VMG como no HIAB. |
| **Título** | Comentar um projeto de remediação |
| **Validação** | Um utilizador deve ser capaz de comentar num projeto e esse comentário ser criado e associado corretamente. |

Todos os testes "Must Have"foram cumpridos, o que significa que a plataforma possui o conjunto mínimo de capacidades exigido pelo proprietário do projeto para que o projeto seja sucesso.

Os testes dos grupos "Should Have"e "Could Have"também foram cumpridos embora não tenha sido possível implementar testes para todos os requisitos destes grupos.

As verificações das tecnologias e componentes escolhidos, como as *apps* Pinax, foram validados através dos seus próprios testes unitários que disponibilizavam para testar o seu funcionamento.

## 7.2 TESTES A ATRIBUTOS DE QUALIDADE

Com os testes sobre as funcionalidades da plataforma concluídos, procedeu-se então à validação dos atributos de qualidade enumerados e descritos no capítulo 4. Nas próximas secções serão abordados e descritos os testes aplicados e as conclusões deles retiradas.

### 7.2.1 *Desempenho e Escalabilidade*

Para avaliar o desempenho foi utilizada a Django App Silk. Esta ferramenta serviu para a recolha de métricas e características de pedidos a uma aplicação Django. O Silk intercepta e armazena os pedidos HTTP e consultas à base de dados antes de apresentar métricas como, o tempo do pedido, o número de consultas à base de dados, tempo gasto em consultas e apresenta estas métricas através de um relatório HTML. Depois de concluídos os testes a ferramenta foi removida.

Com a ferramenta para a recolha de métricas implementada procedeu-se à configuração de uma máquina para realizar os testes. Esta máquina foi configurada com sistema operativo Ubuntu Server 18.04.5, 2 CPUs, 2GB de RAM e 40 GB de memoria.

Para que fosse verificado o desempenho da plataforma foram selecionados um conjunto de *url paths* mais relevantes, apresentados na tabela 12 onde com a sua avaliação foi possível medir o desempenho da plataforma nas suas tarefas mais cruciais e obter uma visão geral do cumprimento, ou falta, dos requisitos estipulados anteriormente. Os *url paths* e as suas justificações foram os seguintes:

Tabela 12: Definição dos testes de desempenho

| | |
|---|---|
| **Url** | vmg.dognaedis.com/manager/ |
| **Justificação** | A página principal da plataforma é responsável por apresentar um conjunto crucial de informação ao utilizador e como tal é essencial que isto seja feito numa forma expedita. |
| **Url** | vmg.dognaedis.com/manager/remediations/.../add_comment |
| **Justificação** | A criação de comentários apresenta como a plataforma se desempenha com um pedido relativamente simples. |
| **Url** | vmg.dognaedis.com/manager/remediations/create |
| **Justificação** | A criação de projetos de remediação é uma funcionalidade crucial da plataforma. |
| **Url** | vmg.dognaedis.com/manager/remediations/ |
| **Justificação** | A janela de remediações é onde muitas das principais funcionalidades são desempenhadas e como tal requer um desempenho que não impeça os utilizadores de fazer o que necessitam. |
| **Url** | vmg.dognaedis.com/manager/assets/ |
| **Justificação** | A página de assets devido a quantidade de dados apresentados, possui um dos maiores tempos de resposta, se não o maior, no entanto, a sua importância para a gestão de vulnerabilidades requer que estes tempos sejam controlados de modo a não prejudicar a usabilidade da plataforma. |
| **Url** | vmg.dognaedis.com/manager/customers/.../due_dates/ |
| **Justificação** | A configuração de SLAs é uma funcionalidade importante da plataforma. |
| **Url** | vmg.dognaedis.com/manager/users/ |
| **Justificação** | A página de utilizadores e equipas é uma das mais utilizadas devido ao seu papel para as atribuições de projetos e como tal devia ter um tempo de resposta pequeno. |
| **Url** | vmg.dognaedis.com/manager/users/teams/create |
| **Justificação** | A criação de equipas é uma funcionalidade crucial para as atribuições de projetos. |

De seguida serão apresentados os resultados obtidos pela ferramenta utilizada. Nesta tabela serão apresentados os tempos que cada pedido precisou, as queries à base de dados e os seus tempos.

Tabela 13: Resultados dos testes de desempenho

| | | | |
|---|---|---|---|
| **Url** | vmg.dognaedis.com/manager/ | | |
| **200 GET** | 290ms em geral | 69ms em queries | 38 queries |
| **Url** | vmg.dognaedis.com/manager/remediations/.../add_comment | | |
| **200 GET** | 128ms em geral | 22ms em queries | 11 queries |
| **Url** | vmg.dognaedis.com/manager/remediations/create | | |
| **200 GET** | 293ms em geral | 17ms em queries | 10 queries |
| **200 POST** | 19048ms em geral | 5808ms em queries | 6598 queries |
| **Url** | vmg.dognaedis.com/manager/remediations/ | | |
| **200 GET** | 46ms em geral | 6ms em queries | 3 queries |
| **200 POST** | 40ms em geral | 5ms em queries | 3 queries |
| **Url** | vmg.dognaedis.com/manager/assets/ | | |
| **200 GET** | 63946ms em geral | 14672ms em queries | 13694 queries |
| **Url** | vmg.dognaedis.com/manager/customers/.../due_dates/ | | |
| **200 GET** | 267ms em geral | 23ms em queries | 11 queries |
| **Url** | vmg.dognaedis.com/manager/users/ | | |
| **200 GET** | 196ms em geral | 30ms em queries | 18 queries |
| **Url** | vmg.dognaedis.com/manager/users/teams/create | | |
| **200 GET** | 217ms em geral | 23ms em queries | 12 queries |

Os resultados dos testes de desempenho, apresentados na tabela 13, revelaram que a grande maioria de pedidos efetuados á aplicação encontram-se com tempos de resposta aceitáveis, no entanto, foi também observado que os pedidos de criação de projetos de remediação e de acesso à página de *assets* possuem tempos de resposta muito maiores do que pretendido.

Isto deve-se aos sistemas implementados na criação de projetos, como o cálculo da prioridade ou a associação automática de uma equipa conforme as suas preferência. Relativamente á página de assets, o seu tempo de resposta deve-se à grande quantidade nela presente.

Visto que estes pedidos são operações pouco frequentes no uso habitual da plataforma e que embora os seus tempos sejam notavelmente maiores o facto da gestão de vulnerabilidades ser desempenhada conforme os episódios de deteção faz com que tempos de resposta maiores sejam aceitáveis.

### 7.2.2 *Usabilidade*

Para que fosse aferida a usabilidade da aplicação foi distribuído um questionário baseado na System Usability Scale (SUS)[52] com o objetivo de fazer uma avaliação global da usabilidade de um sistema. Este questionário foi implementado através do Google Forms e contempla dois conjuntos de questões, o primeiro é composto por 9 perguntas onde é solicitado que seja seguido um conjunto de instruções e ainda *sugestões*, melhorias e comentários relativamente ás principais funcionalidades e aspetos da plataforma. O segundo grupo consiste num conjunto de 10 perguntas em que pedido ao utilizador para avaliar numa escala de 1 a 5 se o utilizador concorda com a frase utilizada. Quanto mais alto o valor mais o utilizador está de acordo.

Como este foi o primeiro contacto dos utilizadores com a aplicação, a acompanhar o questionário, foi entregue um pequeno guia da aplicação a explicar as funcionalidades e os passos necessários para completar o questionário. Este guia pode ser encontrado no apêndice F.

Este estudo foi realizado por 6 utilizadores da equipa operacional da entidade acolhedora, sendo 3 destes parte da equipa de gestão de vulnerabilidades.

De seguida será apresentado uma síntese das sugestões obtidas no primeiro grupo de 9 perguntas:

- **Página principal**: Nesta pergunta foi sugerido adicionar uma pequena descrição aos gráficos de tendências e um modo de expandir as entradas da tabela a fim de verificar que *assets* são afetados. Foi pedida ainda uma maneira de verificar onde a pontuação, tanto do utilizador como dos outros, é ganha (para alem do histórico na página de perfil).
- **Clientes**: Foi pedido uma melhor explicação de como os SLAs funcionam.
- **Importação e log sources**: Existe uma falta de feedback durante a importação, alguns utilizadores pensaram que tinha parado e foi pedido uma opção para criar clientes.
- **Perfil**: Funciona como pretendido.
- **Preferências**: Funciona como pretendido, contudo, houve uma sugestão para adicionar novas preferências com a prioridade mais baixa e não em primeiro lugar. A necessidade do botão "Update"pode ser questionável.

- **Remediações**: O botão para comentar devia estar visível sem necessitar de abrir os detalhes da remediação e existe uma falta de informação nas notificações dos episódios de verificação a uma vulnerabilidade.
- **Assets**: Foi indicado um erro quando são pedidos dois *scans* seguidos, na mesma máquina. O HIAB recusa e a mensagem de erro e dá a entender que não foi efetuado.
- **Utilizadores e equipas**: Nada foi adicionado. Funciona como pretendido.
- **Torneios**: Foi sugerido que os torneios deviam ter mais objetivos do que apenas pontuação, por exemplo, resolver todas as vulnerabilidades com CVSS maior que 7.5, algo que para a segurança da organização seria mais benéfico.

Para alem das sugestões e pedidos apontados anteriormente, também foram indicados vários bugs na aplicação que depois foram corrigidos.

As restantes 10 perguntas foram feitas seguindo a escala SUS, com o intuito de obter uma visão, que embora subjetiva, permitisse verificar como os utilizadores consideraram a utilidade da aplicação. A seguir serão listadas as perguntas deste grupo e a média dos resultados obtidos, numa escala de 1 a 5, arredondada ao número mais próximo.

- **Gostaria de utilizar esta aplicação frequentemente**: 4
- **A aplicação é desnecessariamente complexa**: 1
- **A aplicação é fácil de usar**: 4
- **Vou precisar de ajuda técnica para poder utilizar a aplicação**: 3
- **As várias funções da aplicação estão bem integradas.**: 5
- **A aplicação é demasiado inconsistente.**: 2
- **A maioria dos utilizadores aprende a utilizar esta aplicação rapidamente**: 4
- **A aplicação é muito confusa de utilizar**: 2
- **Senti-me confiante ao utilizar a aplicação**: 4
- **Precisava de aprender sobre algumas coisas antes de poder usar a aplicação**: 4

Como se pode verificar, os utilizadores consideraram a plataforma fácil de utilizar. Contudo, foi apontado que a sua complexidade e a do seu tema tornam a curva de aprendizagem para a sua utilização alta. Algo possivelmente remediado, como sugerido, com a adição de mais descrições das funcionalidades e uma alteração do aspeto visual de modo a realçar as opções mais pertinentes ao utilizador.

### 7.2.3 *Segurança*

Relativamente à segurança da aplicação, esta foi avaliada através do *scanner* de vulnerabilidades HIAB por meio de dois episódios de deteção de vulnerabilidades, o primeiro à máquina e o segundo à aplicação web. O primeiro episódio à máquina não encontrou nenhuma vulnerabilidade no entanto o episódio à plataforma web revelou 8 vulnerabilidades, todas relacionadas com o uso de HTTP e não HTTPS. No entanto, optou-se por não remediar a situação dado que o servidor da aplicação é incapaz de comunicar com a Internet e foi disponibilizado apenas para testes.

### 7.2.4 *Robustez*

Os testes de robustez têm o intuito de assegurar que a plataforma, através dos seus mecanismos de controlo, é capaz, quando sobrecarregada, reagir corretamente a quebras na rede ou problemas de desempenho.

Durante estes testes foram postos á prova os controlos de robustez, um destes testes comprovou que quando a ligação com o HIAB é perdida, por exemplo, quando uma importação está a correr e perde-se a conexão, verificou-se que esta operação era abortada corretamente. Portanto, que os novos dados eram descartados e este erro era registado tanto na página de importação como nos ficheiros para o *logging* de erros.

## 7.3 SÍNTESE

Ao longo deste capítulo foram descritos os testes funcionais desenvolvidos para avaliar o cumprimento dos requisitos funcionais estipulados anteriormente no capítulo 4. Seguidamente foi verificado se os atributos de qualidade traçados anteriormente foram satisfeitos e o seu impacto no resultado final.

# 8

# CONCLUSÕES

## 8.1 TRABALHO FUTURO

Esta aplicação foi concebida para responder às preocupações relacionadas com a falta de uma ferramenta que permita uma melhor gestão de vulnerabilidades e a falta incentivo observada na remediação e mitigação de vulnerabilidades por parte das organizações afetadas. Como tal, as funcionalidades desenvolvidas focaram-se nestes dois âmbitos, no entanto, com estes objetivos atingidos é possível então começar a expandir as funcionalidades e a implementar novas.

Visto que a plataforma foi propositadamente planeada para ser extensível, e a sua natureza modular permite a adição de novos componentes, como a adição de uma nova *app* que possibilita-se a integração com novos *scanners* de vulnerabilidades.

Outras áreas de possíveis melhorias seriam os relatórios disponibilizados pela plataforma através da expansão dos já existentes com novos gráficos, tabelas ou textos com novos dados.

Também é possível o desenvolvimento de outros relatórios como um documento que contenha o conjunto mínimo de vulnerabilidades necessárias serem remediadas para que a organização possa ser considerada minimamente segura.

Por fim, como sugerido nos testes de usabilidade também seria essencial uma melhoria da interface da aplicação, quer com a adição de novas descrições e dicas de como a plataforma funciona, quer com uma alteração do aspeto visual da plataforma com novas cores e realçando as opções mais pertinentes.

## 8.2 CONCLUSÃO

Ao longo deste projeto procurou-se responder à crescente necessidade, por parte da entidade acolhedora, de um sistema de gestão de vulnerabilidades capaz de responder à falta de recursos para responder à totalidade das vulnerabilidades descobertas e também à falta de incentivo por parte das organizações a responder ás remediações propostas. Esta falta de incentivo devem principalmente da pouca consciência do impacto real que a exploração das vulnerabilidades pode ter na infraestrutura de uma organização.

Como tal foi desenvolvida uma plataforma capaz de agregar os resultados dos vários episódios de deteção e de priorizar a resolução de vulnerabilidades tendo em consideração as suas severidades, *assets*, soluções e outros indicadores, com o uso de projetos de remediação com data limite e associados a um grupo de utilizadores.

A mitigação de vulnerabilidades de forma priorizada e continua não advém só graças à utilização clara e intuitiva de projetos de remediação, mas também ao foco dado aos utilizadores responsáveis pelas remediações.

Este foco nos utilizadores expressa-se através de vários sistemas que procuram automatizar ou reduzir os processos necessários para a realização de remediações, tais como a simplificação de acções como os pedidos de verificação de vulnerabilidades ou de máquinas, um sistema de preferências para melhor atribuição de projetos ou novas verificações que diminuem o numero de falsos positivos detetados pelos sistemas de deteção.

Simultaneamente a plataforma VMG consegue também incentivar o seu uso e consequentemente que a remediação de vulnerabilidades seja mantida através de um conjunto de sistemas de gamificação. Como o uso de recompensas atribuindo aos utilizadores pontuação por determinadas ações e removendo pela sua ausência, ou atribuído *badges* por metas atingidas, ou ainda o fomento à competição entre os utilizadores através do uso torneios, *leaderboards* ou mesmo relatórios de atividade.

# BIBLIOGRAFIA


[1] CMU Portugal. *CMU Portugal startup, Dognaedis, began a new chapter under a new Global brand: Cipher, a Prosegur Company*. https://www.cmuportugal.org/media/cmu-portugal-startup-dognaedis-began-a-new-chapter-under-a-new-global-brand-cipher-a-prosegur-company/, Ultimo acesso: Setembro 2021. Mar. de 2021.

[2] Dognaedis. *Dognaedis*. URL: https://www.dognaedis.com/. (Ultimo acesso: 10.2020).

[3] Scrum. *What is Scrum?* URL: https://www.scrum.org/resources/what-is-scrum. (Ultimo acesso: Outubro 2020).

[4] Ken Schwaber e Jeff Sutherland. *The Scrum Guide*. http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=10, Ultimo acesso: Outubro 2020. Jul. de 2016.

[5] OWASP. *Vulnerabilities*. URL: https://owasp.org/www-community/vulnerabilities/. (Ultimo acesso: Outubro 2020).

[6] ICTEA. *What is a vulnerability*. URL: https://www.ictea.com/cs/knowledgebase.php?action=displayarticle&id=2092&language=english. (Ultimo acesso: Outubro 2020).

[7] Eric Dosal. *Top 5 Computer Security Vulnerabilities*. https://www.compuquip.com/blog/computer-security-vulnerabilities, Ultimo acesso: Outubro 2020. Mar. de 2020.

[8] Team Ascend. *Ascend Technologies Website*. URL: https://blog.teamascend.com/stages-of-vulnerability-management. (Ultimo acesso: Fevereiro 2022).

[9] CVE. *CVE Website*. URL: https://cve.mitre.org/index.html. (Ultimo acesso: Fevereiro 2022).

[10] CCE. *CCE Website*. URL: http://cce.mitre.org/. (Ultimo acesso: Fevereiro 2022).

[11] CPE. *CPE Website*. URL: http://cpe.mitre.org/. (Ultimo acesso: Fevereiro 2022).

[12] CWE. *CWE Website*. URL: https://cwe.mitre.org/. (Ultimo acesso: Fevereiro 2022).

[13] CVSS. *CVSS Website*. URL: https://www.first.org/cvss7. (Ultimo acesso: Fevereiro 2022).

[14] Skybox Security. *Skybox Security Website*. URL: https://www.skyboxsecurity.com. (Ultimo acesso: Outubro 2020).

[15] Doug Olenick. *Skybox Security – Security Suite, Vulnerability and Threat Management*. https://www.compuquip.com/blog/computer-security-vulnerabilities, Ultimo acesso: Outubro 2020. Jun. de 2019.

[16] Skybox Vulnerability Center. *Methodology  Data Sources: Skybox Vulnerability Center*. URL: https://www.vulnerabilitycenter.com/methodology. (Ultimo acesso: Outubro 2020).

[17] GetApp. *GetApp - Skybox Vulnerability Control*. URL: https://www.getapp.com.au/software/116303/vulnerability-control. (Ultimo acesso: Outubro 2020).

[18] VMware. *VMware Cloud Management Blog*. URL: https://blogs.vmware.com/management/. (Ultimo acesso: Setembro 2021).

[19] Rhett Glauser. *New SaltStack SecOps Products Automate Vulnerability Remediation and Continuous Security Compliance*. https://saltproject.io/new-saltstack-secops-products-automate-vulnerability-remediation-and-continuous-security-compliance/, Ultimo acesso: Setembro 2021. Nov. de 2019.

[20] VMware. *Introducing VMware vRealize Automation SaltStack SecOps*. URL: https://blogs.vmware.com/management/2021/02/introducing-saltstack-secops.html. (Ultimo acesso: Setembro 2021).

[21] VMware. *VMware vRealize Automation SaltStack SecOps Datasheet*. URL: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vmware-saltstack-secops-ds-final.pdf. (Ultimo acesso: Outubro 2020).

[22] Rapid7. *InsightVM - Product Brief*. URL: https://www.rapid7.com/globalassets/_pdfs/product-and-service-briefs/rapid7-insightvm-product-brief-1.pdf. (Ultimo acesso: Janeiro 2021).

[23] Rapid7. *InsightVM - Solution Brief*. URL: https://www.rapid7.com/globalassets/_pdfs/product-and-service-briefs/rapid7-solution-brief-quantifying-risk-insightvm.pdf. (Ultimo acesso: Janeiro 2021).

[24] Maja Dakić. *Gamification in Mobile Apps*. https://medium.datadriveninvestor.com/gamification-in-mobile-apps-1be65d660af5, Ultimo acesso: Janeiro 2022. Jun. de 2019.

[25] Anastasia Khomych. *Is Gamification the Only Way for Apps to Survive?* https://blog.getsocial.im/is-gamification-the-only-way-for-apps-to-survive/, Ultimo acesso: Janeiro 2022. Nov. de 2021.

[26] Valamis. *What is gamification?* https://www.valamis.com/hub/gamification, Ultimo acesso: Janeiro 2022. Set. de 2021.

[27] ProductPlan. *MoSCoW Prioritization*. URL: https://www.productplan.com/glossary/moscow-prioritization/. (Ultimo acesso: Outubro 2020).

[28] Ruggero Ruggieri. *Análise sobre a ISO 9126 – NBR 13596*. https://www.tiespecialistas.com.br/analise-sobre-iso-9126-nbr-13596/, Ultimo acesso: Janeiro 2022. Out. de 2016.

[29] Django. *Django Website*. URL: https://www.djangoproject.com/. (Ultimo acesso: Novembro 2021).

[30] Youssef Nader. *What is Django? Advantages and Disadvantages*. https://hackr.io/blog/what-is-django-advantages-and-disadvantages-of-using-django, Ultimo acesso: Março 2022. Fev. de 2022.

[31] Pinax. *Pinax Website*. URL: https://pinaxproject.com/. (Ultimo acesso: Março 2022).

[32] Pinax Badges. *Pinax Badges Github Page*. URL: https://github.com/pinax/pinax-badges. (Ultimo acesso: Março 2022).

[33] Pinax Points. *Pinax Points Github Page*. URL: https://github.com/pinax/pinax-points. (Ultimo acesso: Março 2022).

[34] Pinax Notifications. *Pinax Notifications Github Page*. URL: https://github.com/pinax/pinax-notifications. (Ultimo acesso: Março 2022).

[35] Tutorials Point. *NumPy - Matplotlib*. URL: https://www.tutorialspoint.com/numpy/numpy_matplotlib.htm. (Ultimo acesso: Janeiro 2022).

[36] Pandas. *Pandas Website*. URL: https://pandas.pydata.org/. (Ultimo acesso: Janeiro 2022).

[37] PostgreSQL. *PostgreSQL Website*. URL: https://www.postgresql.org/. (Ultimo acesso: Fevereiro 2022).

[38] Brandon Chen. *PostgreSQL vs. MySQL: What You Need to Know*. https://www.fivetran.com/blog/postgresql-vs-mysql, Ultimo acesso: Março 2022. Set. de 2021.

[39] PostgreSQL. *PostgreSQL Documentation*. URL: https://www.postgresql.org/docs/current/datatype.html. (Ultimo acesso: Fevereiro 2022).

[40] Celery Project. *Celery Project Website*. URL: http://www.celeryproject.org/. (Ultimo acesso: Dezembro 2021).

[41] RabbitMQ. *RabbitMQ Website*. URL: https://www.rabbitmq.com/. (Ultimo acesso: Dezembro 2021).

[42] Lovisa Johansson. *Part 1: RabbitMQ for beginners - What is RabbitMQ?* https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html, Ultimo acesso: Dezembro 2021. Set. de 2019.

[43] Redis. *Redis Website*. URL: https://redis.io/. (Ultimo acesso: Dezembro 2021).

[44] Bootstrap. *Bootstrap Website*. URL: https://getbootstrap.com/. (Ultimo acesso: Dezembro 2020).

[45] NiceAdmin. *NiceAdmin - Free bootstrap admin HTML template*. URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/. (Ultimo acesso: Dezembro 2020).

[46] Outpost24. *Outpost24 Knowledge base - XML API Interface Technical Document*. URL: https://kb.outpost24.com/kb/articles/api-s/xml-api-interface-technical-document. (Ultimo acesso: Dezembro 2021).

[47] Outpost24. *Outpost24 Knowledge base - REST API Interface Technical Document*. URL: https://kb.outpost24.com/kb/articles/api-s/rest-api-interface-technical-document. (Ultimo acesso: Dezembro 2021).

[48] Celery Progress. *Celery Progress Bars*. URL: https://github.com/czue/celery-progress. (Ultimo acesso: Janeiro 2021).

[49] Action Streams. *Action Streams*. URL: https://django-activity-stream.readthedocs.io/en/latest/streams.html. (Ultimo acesso: Maio 2021).

[50] Chart.js. *Chart.js Website*. URL: https://www.chartjs.org/. (Ultimo acesso: Maio 2021).

[51] FullCalendar. *FullCalendar Website*. URL: https://fullcalendar.io/. (Ultimo acesso: Junho 2021).

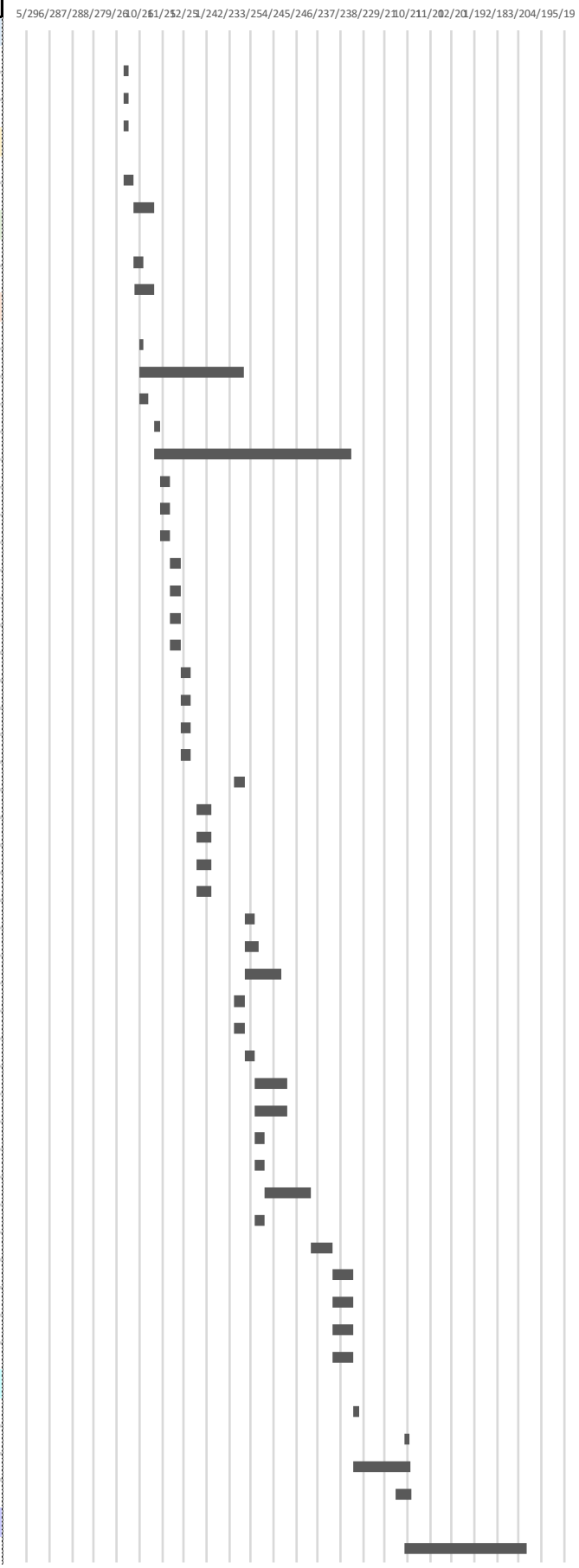
[52] usability.gov. *System Usability Scale (SUS)*. URL: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html. (Ultimo acesso: Dezembro 2021).

# APÊNDICES

# A

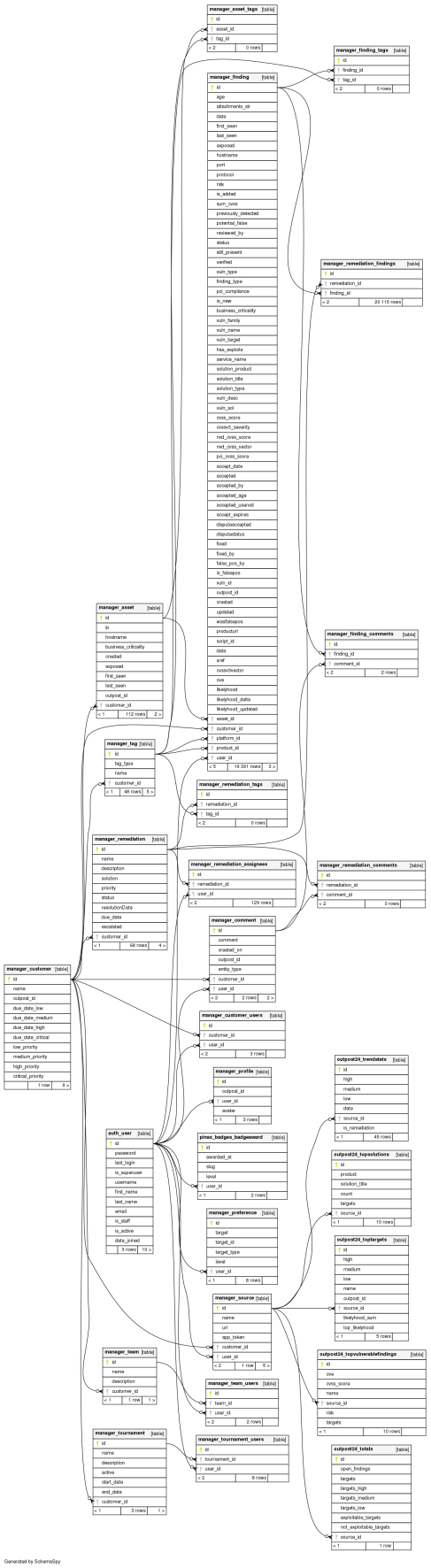
# GANTT CHART

| Tarefas | | | Duração (Dias) |
|---|---|---|---|
| Inicio | Fim | Descrição | |
| Planeamento | | | |
| 10/5/20 | 10/12/20 | Planear as actividades e calendarização | 7 |
| 10/5/20 | 10/12/20 | Estudo das Funcionalidades a Implementar | 7 |
| 10/5/20 | 10/12/20 | Definir requisitos | 7 |
| Estado de Arte | | | |
| 10/5/20 | 10/19/20 | Estudo das ferramentas existentes | 14 |
| 10/19/20 | 11/16/20 | Estudo do Outpost24 | 27 |
| Arquitectura | | | |
| 10/19/20 | 11/2/20 | Design da Base de Dados | 13 |
| 10/20/20 | 11/16/20 | Design da Aquitectura | 26 |
| Desenvolvimento | | | |
| 10/26/20 | 11/2/20 | Criar a Base de Dados | 6 |
| 10/26/20 | 3/17/21 | Importar os dados do Outpost24 | 141 |
| 10/26/20 | 11/9/20 | API Authentication | 13 |
| 11/16/20 | 11/23/20 | Listar Assets | 7 |
| 11/16/20 | 8/10/21 | Logging | 264 |
| 11/23/20 | 12/7/20 | Selecionar as preferencias do User | 14 |
| 11/23/20 | 12/7/20 | Autenticação do User | 14 |
| 11/23/20 | 12/7/20 | Criar os grupos dos Users | 14 |
| 12/7/20 | 12/21/20 | Criar Resolution Projects | 14 |
| 12/7/20 | 12/21/20 | Automatizar a criação de RPs | 14 |
| 12/7/20 | 12/21/20 | Definir as regras de agrupamento dos RPs | 14 |
| 12/7/20 | 12/21/20 | Listar Findings e RPs | 14 |
| 12/21/20 | 1/4/21 | Definição das Due Dates | 13 |
| 12/21/20 | 1/4/21 | Automatic Escalation | 13 |
| 12/21/20 | 1/4/21 | Ver os detalhes de um RP | 13 |
| 12/21/20 | 1/4/21 | Ver os detalhes de uma Finding | 13 |
| 3/2/21 | 3/17/21 | Editar os assignees de uma RP | 15 |
| 1/11/21 | 2/1/21 | Organizar findings e RPs | 20 |
| 1/11/21 | 2/1/21 | Pesquisar findings e RPs | 20 |
| 1/11/21 | 2/1/21 | Ordenar findings e RPs | 20 |
| 1/11/21 | 2/1/21 | Filtrar findings e RPs | 20 |
| 3/17/21 | 3/30/21 | Comentar nas RPs e Findings | 13 |
| 3/17/21 | 4/6/21 | Lançar um re-test | 19 |
| 3/17/21 | 5/6/21 | Resolver uma finding | 49 |
| 3/2/21 | 3/17/21 | Implementação de Roles e Permissões | 15 |
| 3/2/21 | 3/17/21 | Editar o User | 15 |
| 3/17/21 | 3/30/21 | Definir as regras de scoring | 13 |
| 3/30/21 | 5/14/21 | Criar os Scores | 44 |
| 3/30/21 | 5/14/21 | Leaderboard | 44 |
| 3/30/21 | 4/13/21 | Listar e mostrar detalhes dos Tournaments | 13 |
| 3/30/21 | 4/13/21 | Criar Tournaments | 13 |
| 4/13/21 | 6/14/21 | Listar Badges | 61 |
| 3/30/21 | 4/13/21 | Definir os Badges | 13 |
| 6/14/21 | 7/13/21 | Create UI relating to Gamification | 29 |
| 7/13/21 | 8/10/21 | Estatisticas | 27 |
| 7/13/21 | 8/10/21 | Follow-up Periodicos | 27 |
| 7/13/21 | 8/10/21 | Criar os gráficos dos Dashbords | 27 |
| 7/13/21 | 8/10/21 | Exportar relatórios | 27 |
| Verificação e Validação | | | |
| 8/10/21 | 8/17/21 | Definição dos Testes | 7 |
| 10/17/21 | 10/24/21 | Implementação dos testes | 7 |
| 8/10/21 | 10/26/21 | Debugging | 76 |
| 10/5/21 | 10/26/21 | Deployment | 21 |
| Relatório | | | |
| 10/17/21 | 3/31/22 | Escrita do Relatório | 164 |



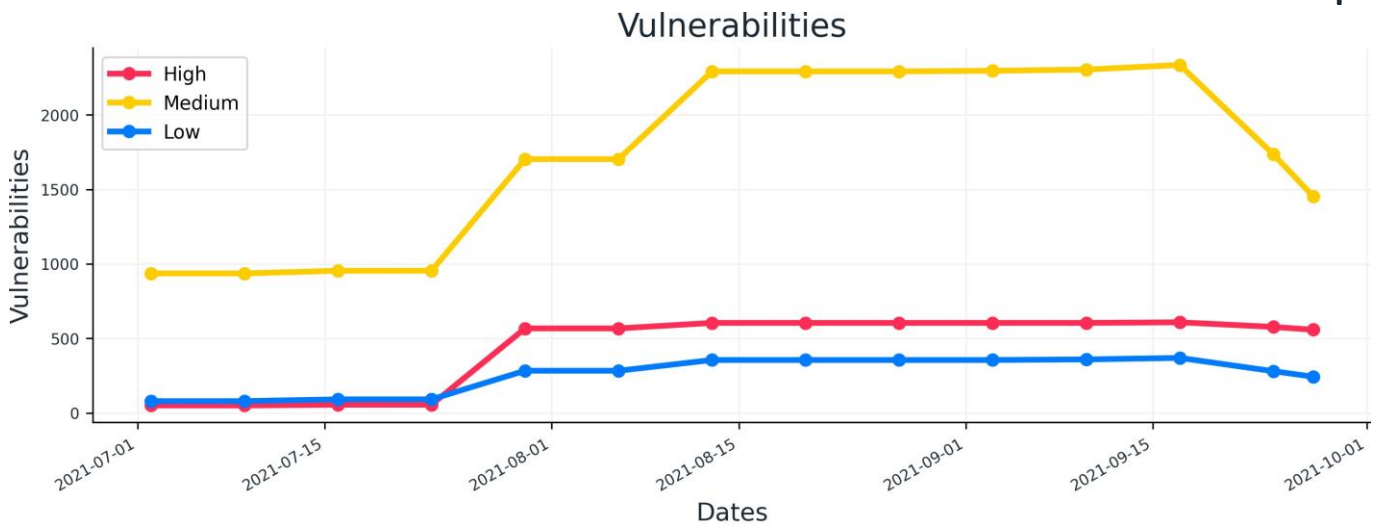
# B

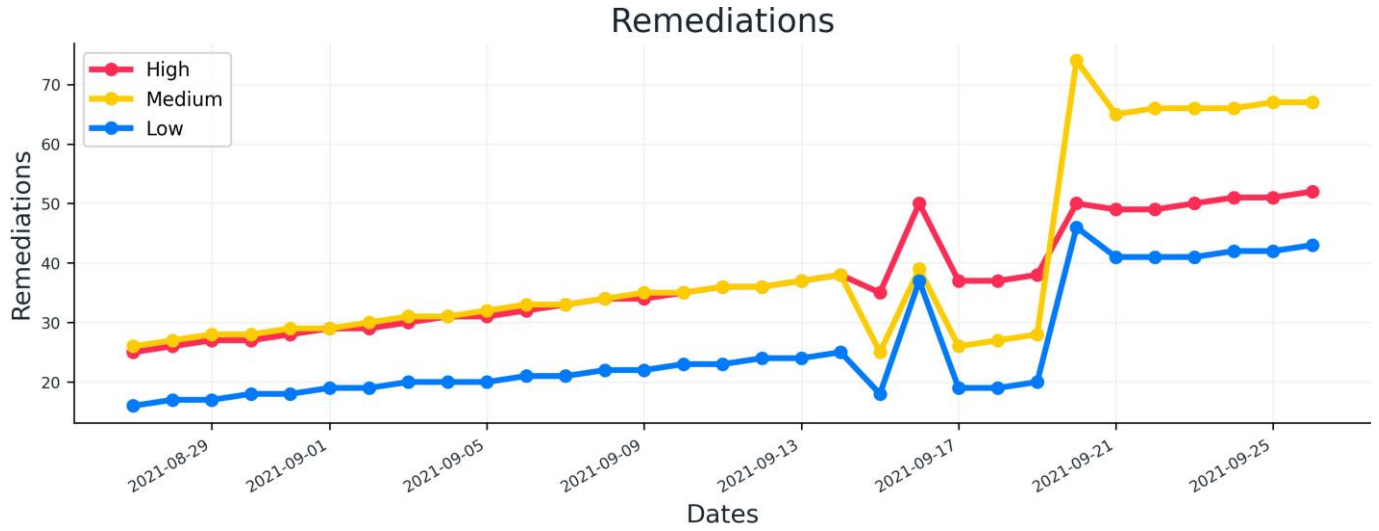
## DIAGRAMA DE ENTIDADE-RELACIONAMENTO

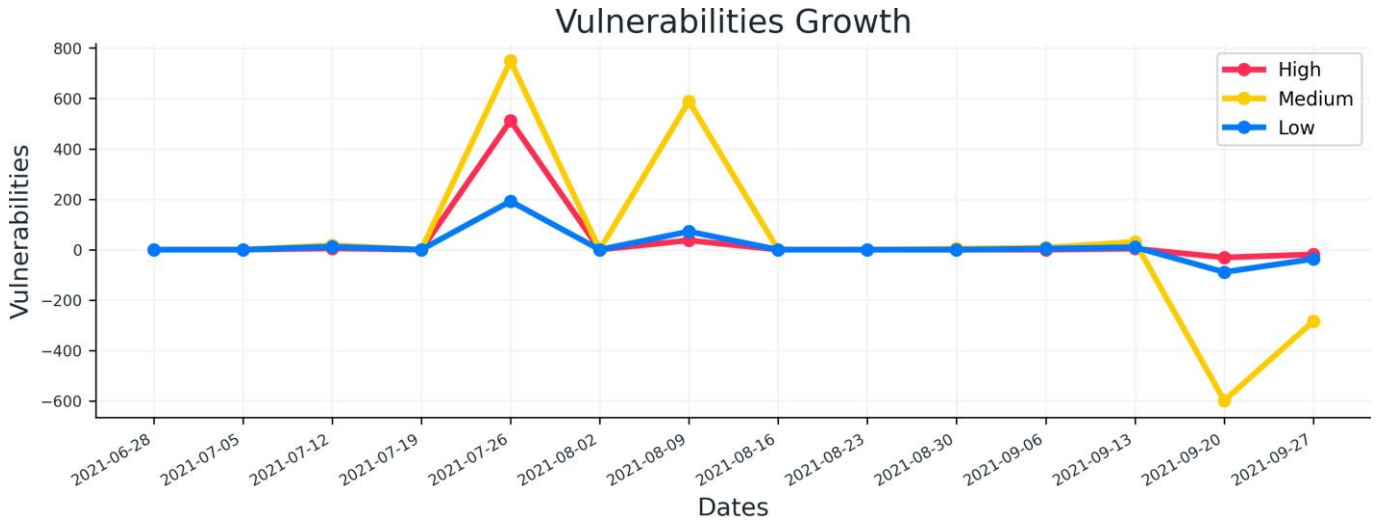

manager_asset_tags [table]
manager_finding_tags [table]
manager_finding [table]
manager_remediation_findings [table]
manager_asset [table]
manager_finding_comments [table]
manager_tag [table]
manager_remediation_tags [table]
manager_remediation [table]
manager_remediation_assignees [table]
manager_remediation_comments [table]
manager_comment [table]
manager_customer [table]
manager_customer_users [table]
manager_profile [table]
outpost24_trendstats [table]
auth_user [table]
pinax_badges_badgeaward [table]
outpost24_topsolutions [table]
manager_preference [table]
outpost24_toptargets [table]
manager_source [table]
manager_team [table]
outpost24_topvulnerablefindings [table]
manager_team_users [table]
manager_tournament [table]
manager_tournament_users [table]
outpost24_totals [table]
Generated by SchemaSpy

# C

# RELATÓRIO ADMINISTRATIVO



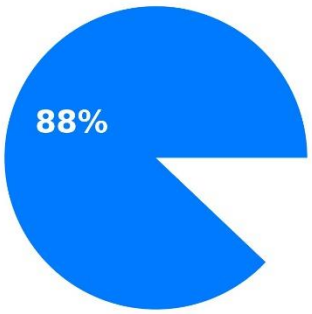

*Activity*

| *Findings Resolved* | *Remediations Closed* | *Comments* | *Points Earned* | *Target Scans* | *Finding Re-Scans* |
|---|---|---|---|---|---|
| *10* | *1* | *3* | *4449* | *0* | *33* |

## Most Vulnerable Findings

| # | Finding | CVE | Risk | Targets |
|---|---|---|---|---|
| 0 | Product End-of-Life (EOL) | None | 4.0 | 8.0 |
| 1 | SMB Anonymous Login Enabled | None | 4.0 | 2.0 |
| 2 | Default SMB Credentials | None | 4.0 | 1.0 |
| 3 | Adobe Flash Player: Use-After-Free Remote Code Execution Vulnerability | CVE-2019-7096 | 4.0 | 1.0 |
| 4 | Adobe Flash Player: Use-After-Free Arbitrary Code Execution Vulnerability | CVE-2018-15982 | 4.0 | 1.0 |
| 5 | Default FTP Credentials | None | 4.0 | 1.0 |
| 6 | Adobe Flash Player: Same Origin Method Arbitrary Code Execution Vulnerability | CVE-2019-8069 | 4.0 | 1.0 |
| 7 | Adobe Flash Player: Use-After-Free Arbitrary Code Execution Vulnerability | CVE-2019-8070 | 4.0 | 1.0 |
| 8 | Adobe Flash Player: Type Confusion Arbitrary Code Execution Vulnerability | CVE-2018-15981 | 4.0 | 1.0 |
| 9 | Microsoft .NET Framework: Remote Code Execution Injection Vulnerability | CVE-2020-0646 | 4.0 | 1.0 |

Findings Solved by Solutions



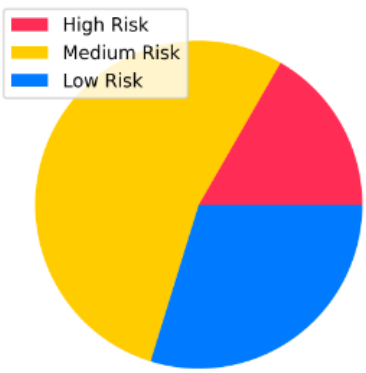

## Top Solutions

| # | Product | Action | Findings | Targets |
|---|---|---|---|---|
| 0 | Windows | Apply latest patches for Microsoft Windows | 1324 | 2 |
| 1 | Atlassian JIRA | Upgrade to version 8.13.10, 8.19.0, 8.18.2, 8.5.18 | 158 | 3 |
| 2 | SSL/TLS | Reconfigure to use Stronger Ciphers | 118 | 14 |
| 3 | SSL/TLS | Disable TLSv1.1 | 100 | 14 |
| 4 | SSL/TLS | Disable TLSv1.0 | 97 | 15 |
| 5 | SSL/TLS | Reconfigure SSL/TLS to use Stronger Ciphers | 97 | 15 |
| 6 | OpenSSH | Upgrade to version 8.6p1 or later of OpenSSH | 80 | 10 |
| 7 | SSL/TLS | Update SSL certificate | 58 | 12 |
| 8 | Microsoft IE | Apply latest patches for Microsoft Internet Explore | r53 | 2 |
| 9 | SSL/TLS | Review certificate | 53 | 22 |

## Most Vulnerable Targets

| # | Target | High Risk | Medium Risk | Low Risk | Likelyhood Sum | Top Likelyhood |
|---|---|---|---|---|---|---|
| 0 | 1.1.1.1 | 477 | 659 | 171 | 21430.3199999999 | 38.46 |
| 1 | 1.1.1.2 | 30 | 88 | 20 | 2104.94 | 38.46 |
| 2 | 1.1.1.3 | 0 | 126 | 1 | 3816.6 | 38.46 |
| 3 | 1.1.1.4 | 0 | 96 | 1 | 3716.6 | 38.46 |
| 4 | 1.1.1.5 | 9 | 59 | 11 | 5357.37 | 38.46 |

## Risk Summary




**14** **TARGETS HAVE HIGH RISK VULNERABILITIES**

*Focus on re-mediating any issues with exploits available or where services are network reachable first and apply high priority fixing.*

**45** **TARGETS HAVE MEDIUM RISK VULNERABILITIES**

*Patch issues with exploits available with priority and manage other findings based on risk acceptance. After priority remediation - Apply solutions-based approach for best ROI with regards to decreased risk.*

**25** **TARGETS HAVE LOW RISK VULNERABILITIES**

*Low-risk issues are generally recommended improvements to security rather than critical risks. Apply solutions-based approach for best ROI with regards to decreased risk.*

*Number of targets with known exploits:*

23

# D

# RELATÓRIO DE UTILIZADOR

# Miguel Libânio

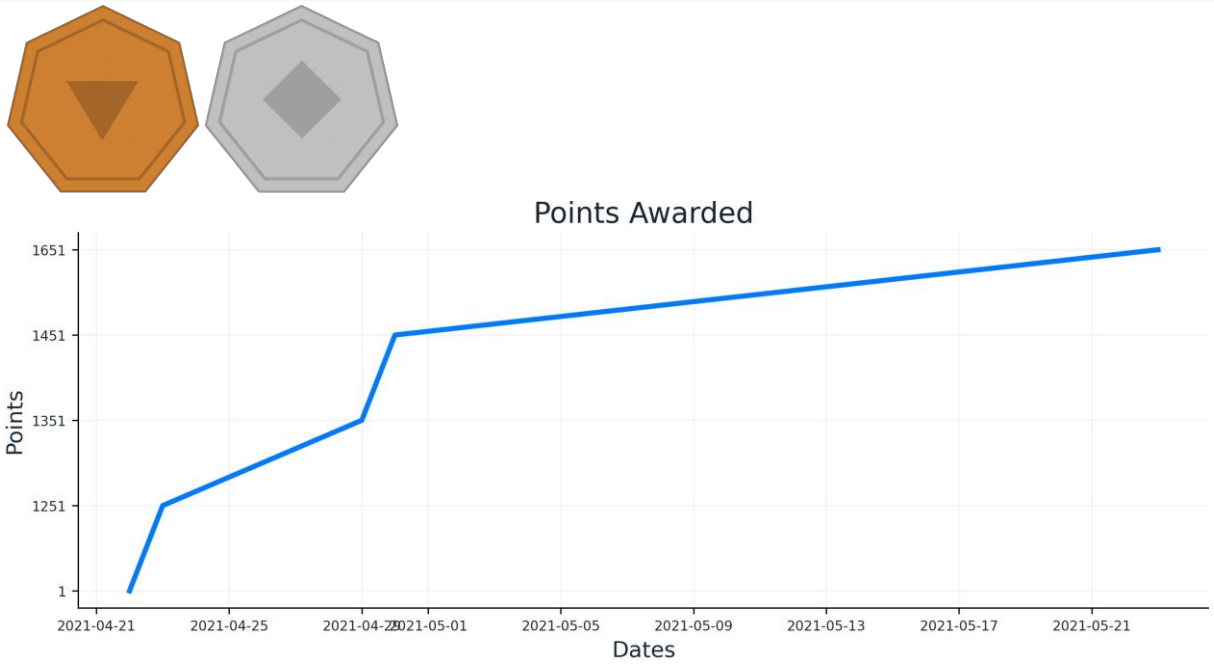
## Leadersboard

| | | |
|---|---|---|
| 1# | Username: test | Points: 2610 |
| 2# | Username: mlibanio | Points: 1659 |
| 3# | Username: miguel.libanio | Points: 180 |

## Activity

| | |
|---|---|
| Findings Resolved | 0 |
| Remediation Prj. Closed | 1 |
| Comments | 3 |
| Points Earned | 1659 |
| Target Scans | 0 |
| Re-Scans on Findings | 33 |

## Badges Earned (Last 7)




## Most Pressing Remediation Projects

| | | |
|---|---|---|
| **Upgrade to version 3.5.0 or later of jQuery.** | | |
| *Priority: 11.0* | *Due Date: 2021-06-22 11:30* | *Progress: 0/78* |
| **172.31.48.4 - git.dognaedis.com** | | |
| *Priority: 11.0* | *Due Date: 2021-07-04 11:40* | *Progress: 0/2* |
| **Upgrade to version 8.16.0, 8.13.5 or later of Atlassian JIRA.** | | |
| *Priority: 11.0* | *Due Date: 2021-07-11 15:31* | *Progress: 0/2142* |
| **Purchase/Generate a new SSL/TLS certificate for this service.** | | |
| *Priority: 3.8* | *Due Date: None* | *Progress: 0/372* |

*On 24-05-21 of next month the tournament Teste is going to start, be sure to login and check it out!*

| Comments | | |
|---|---|---|
| **This is another test.** | | |
| | *Created On: 2021-09-27 17:06:01* | *Extra: 0* |
| **This is a test.** | | |
| | *Created On: 2021-09-27 17:01:43* | *Extra: 0* |

*You haven't resolved any findings.*

| Resolved Remediation Projects | | |
|---|---|---|
| **Upgrade to the latest version of Debian.** | | |
| *Due Date: None* | *Priority: 6.78* | *Closing Date: None* |

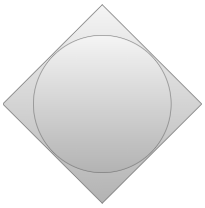
# E

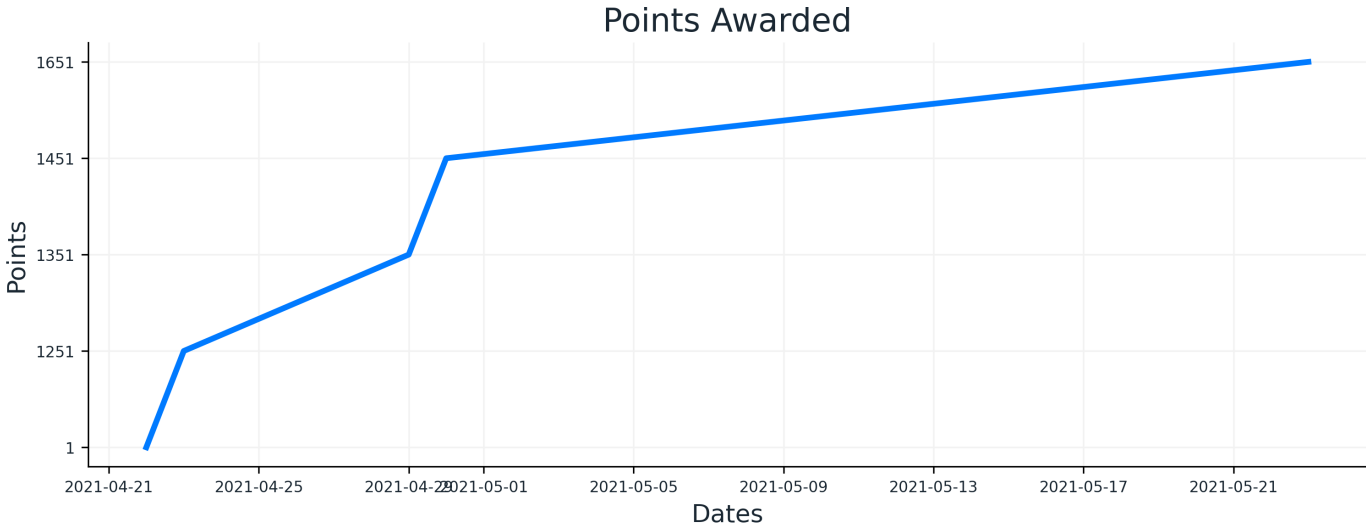
## RESUMO MENSAL

*During the month of May you resolved 0 findings and 0 remediation projects, leaving 1 comments and performing 0 scans. This earned you 200 points and 1 badges.*
*During this month the tournament Teste ended and you managed to obtain 1st place!*

| Activity | |
|---|---|
| Findings Resolved | 0 |
| Remediation Prj. Closed | 0 |
| Comments | 1 |
| Points | 200 |
| Target Scans | 0 |
| Re-Scans on Findings | 0 |

## Badges Earned During Month (Last 7)






## Most Pressing Remediation Projects

| | | |
|---|---|---|
| **Upgrade to version 1.1.1j or later of OpenSSL.** | | |
| *Priority: 4.4* | *Due Date: None* | *Progress: 0/11* |
| **Purchase/Generate a new SSL/TLS certificate for this service.** | | |
| *Priority: 3.8* | *Due Date: None* | *Progress: 0/372* |
| **Reconfigure the SSL/TLS service to not use CBC ciphers in combination with SSL and TLSv1.** | | |
| *Priority: 3.8* | *Due Date: None* | *Progress: 0/97* |

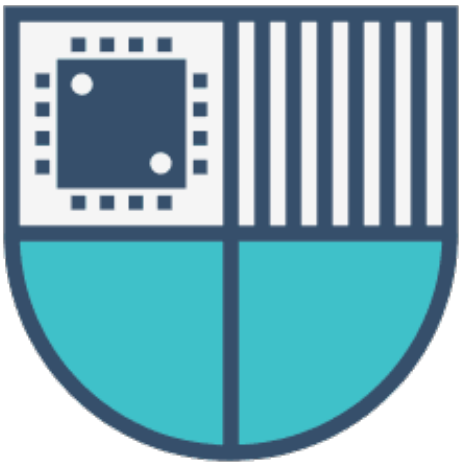
*On 24-04-21 of next month the tournament Teste is going to start, be sure to login and check it out!*

# F

GUIA VMG

# VMG Guide

Miguel Libânio



# Contents

# 1 Introduction

This guide intends to help the user, by explaining or by giving the main guidelines for the usage of the VMG application through a short guide were the several pages and systems of the website will be presented.

# 2 Login

When you access the VMG Web Application you will be presented with a login page and will be requested to insert your login credentials. Like in the image below:

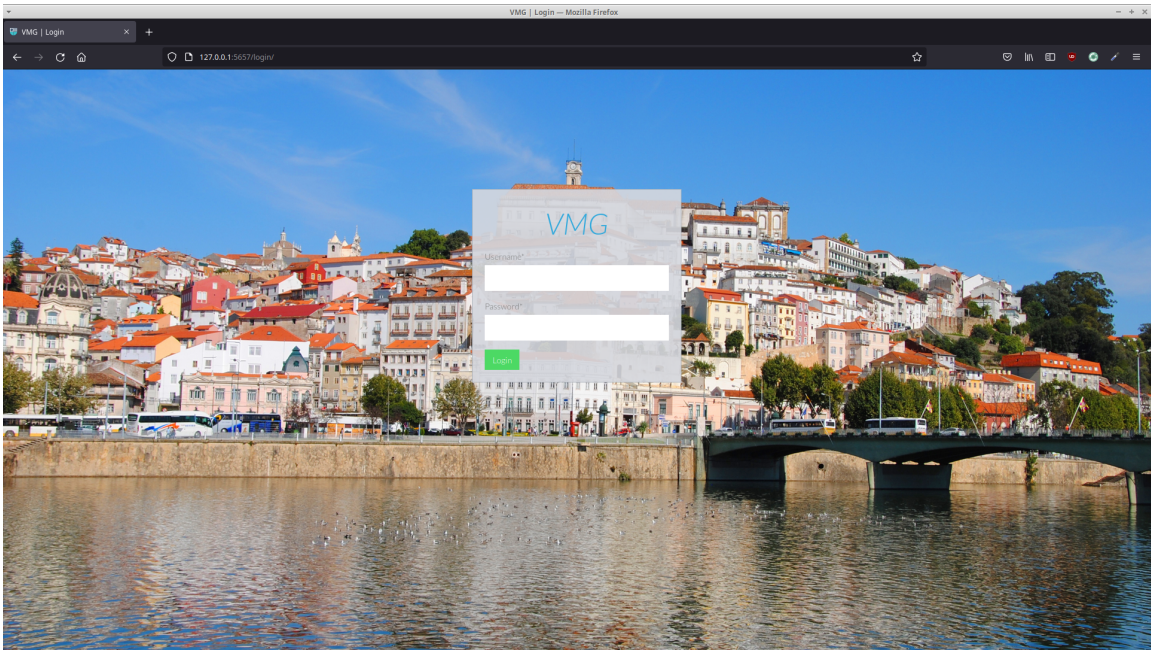



Figure 1: Login Page

# 3 Home Page

Once the login is completed you'll be redirected to the home page of the website. There you will find several statistics about your activity and your organization vulnerability management. In this page it's also possible to download a report about your activity, that will include closed findings, comments, points received, among other relevant statistics, and an organization's report with the graphs presented on the home page. Additionally, at the end of every month you will receive an email with a monthly digest about your activity on the current month and several suggestions for the next remediation projects to focus on.

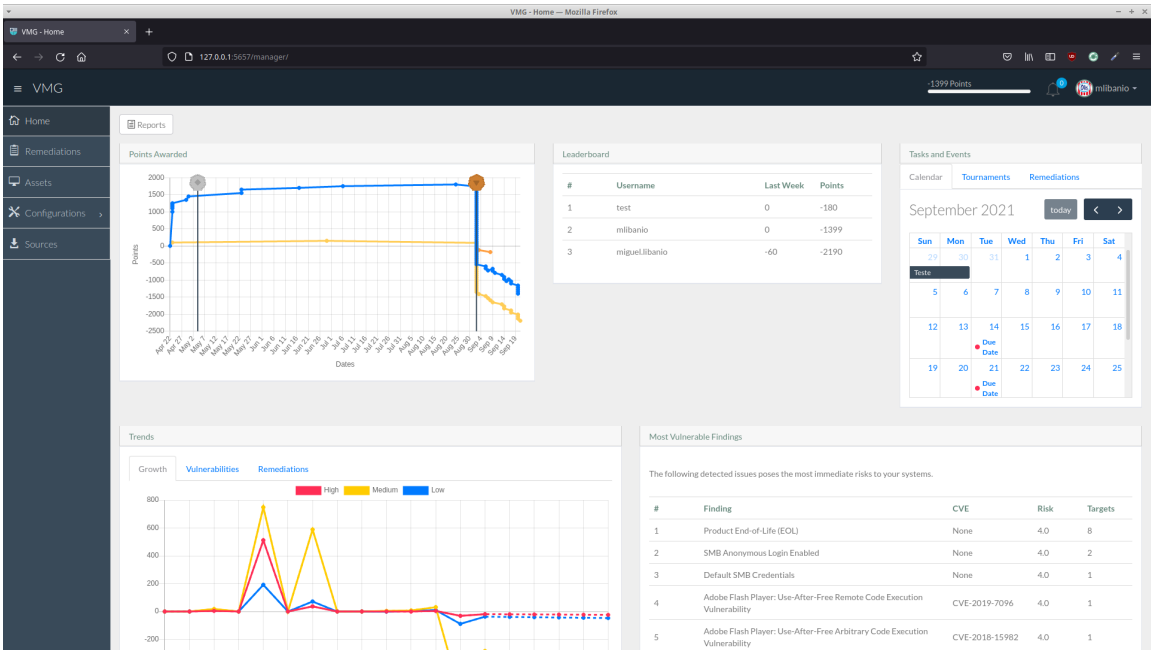



Figure 2: Home Page.

# 4 Customers

To access the data related to costumers you first need to press the "Configurations" option on the side menu and then select "Customers". Please note that this page is only accessible to authorized users. In this page you will be presented with the customers being managed on the application and the deadlines for the corresponding priorities of each client. This deadlines can be changed by selecting the calendar button on the right of each customer, which will redirect you to the due dates configuration page.

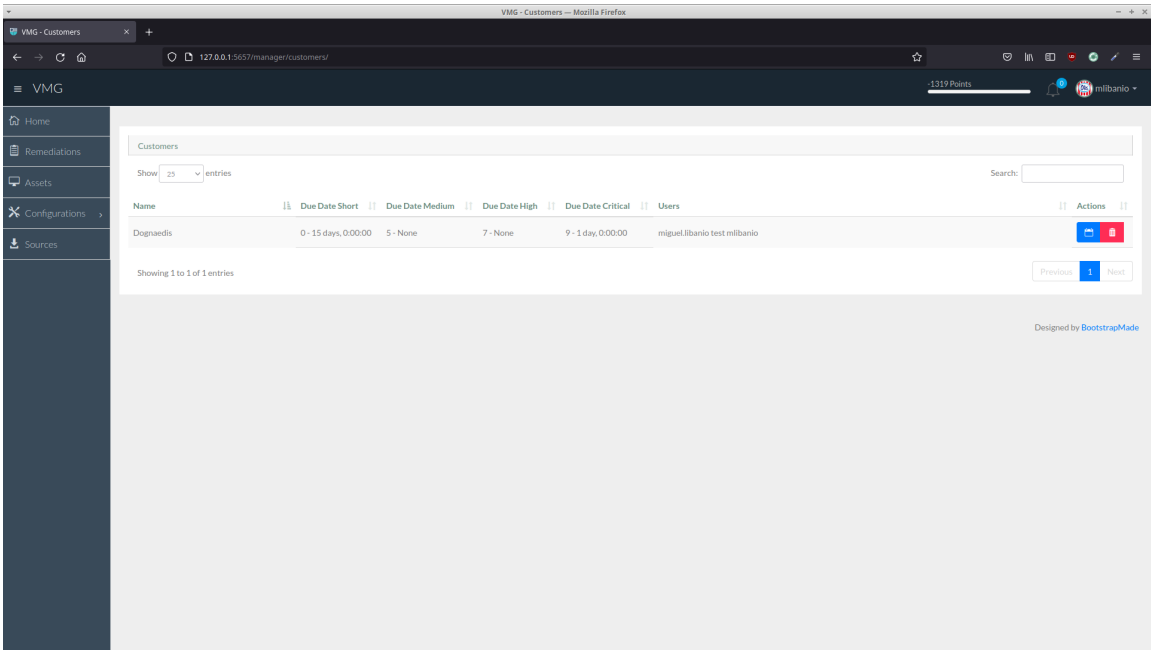



Figure 3: Customers Page.

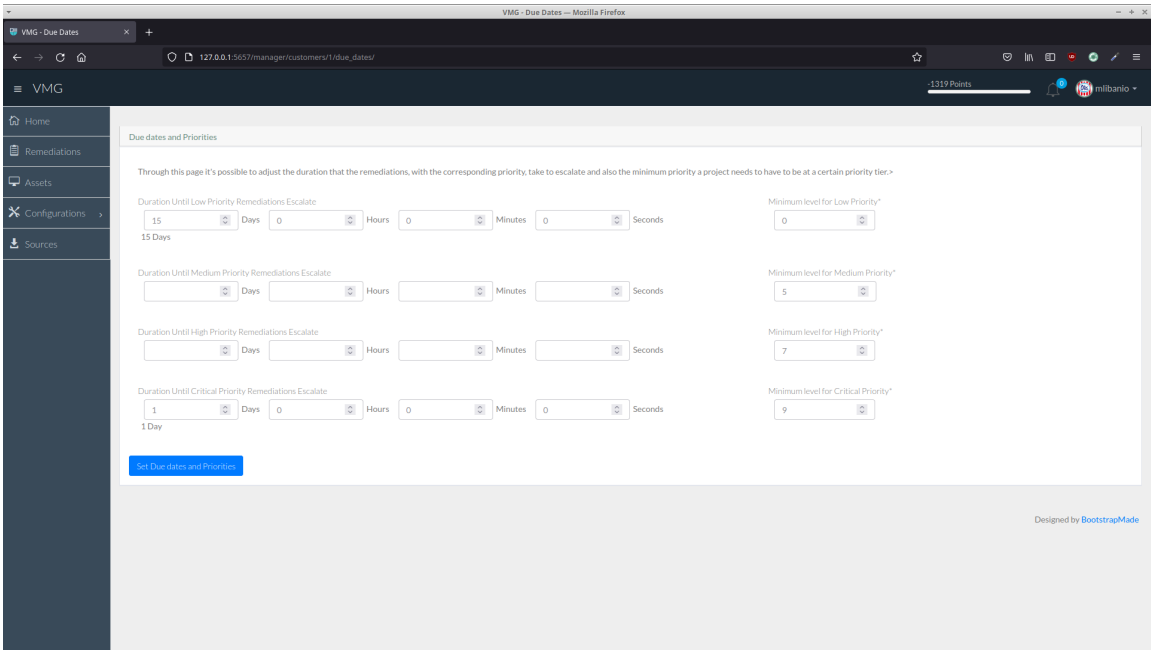



Figure 4: Due Dates Page.

# 5 Sources and Imports

To access the page for the configuration of importation of data from outpost24 you need to select the "Sources" option on the navigation menu. Once again this page is only accessible to authorized users. In this page you can see the configured data sources accessed by the VMG application to download all the necessary data as well as communicate to, for example: request new scans, declare findings has false positives or accepting them, comment on findings, etc. To initiate the importation of data you can press the "Import All" button that will start importing all of the data from each log source or select the import option of a determined source. To configure a new data source you need to first select the "Create Source" button, which will redirect you to the creation page. In this page you'll need to fill out the information about the new source, namely:

- Name - The name of the log source.
- URL - Outpost24's API URL path.
- Customer - The customer that this log source belongs.
- User - The name of user that will show up when VMG post comments on Outpost24.
- API Key - The key of the API.

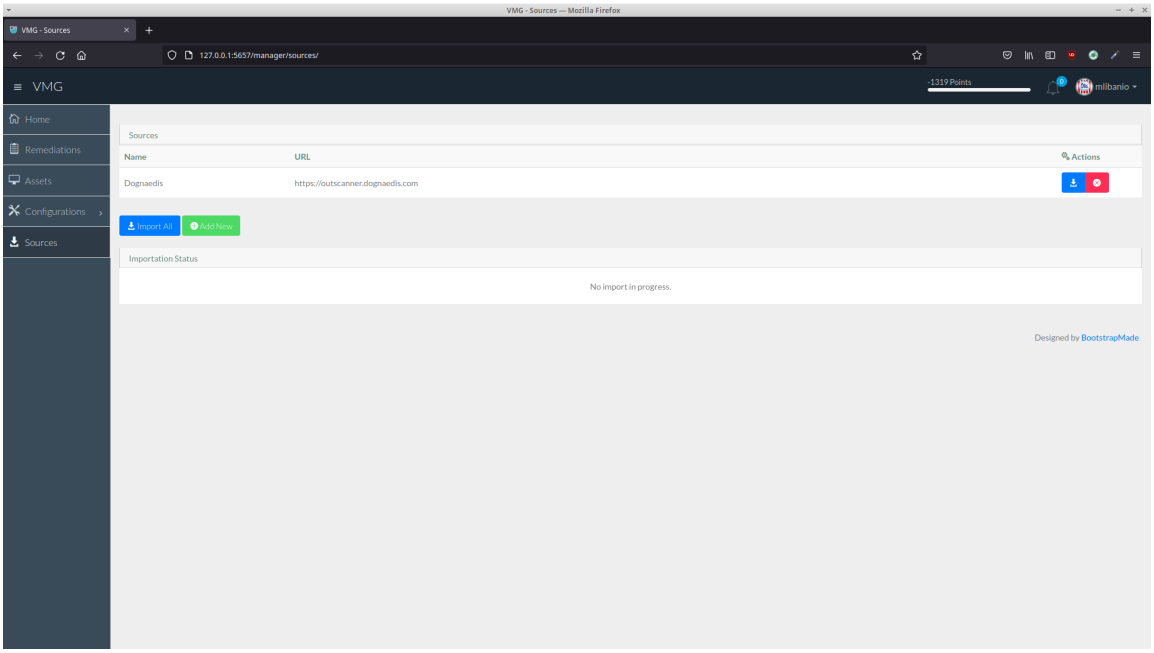



Figure 5: Sources Page.

# 6 Profile

If you press your user name on the right upper corner of the screen a drop down menu will be shown, select the "Profile" option to view your profile. In this page you can change your profile information, view your earned achievements and also a log of your activity.

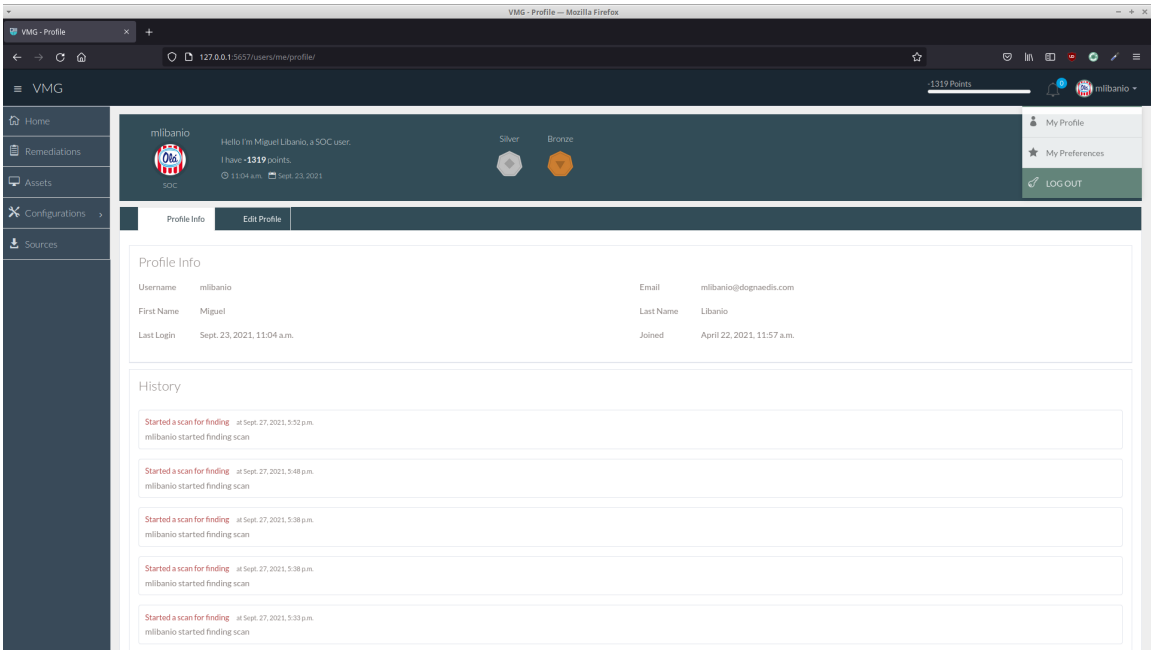



Figure 6: Profile Page.

# 7 Preferences

By selecting the drop down menu again and instead selecting the "Preferences" option you will be redirected to the preferences page where you can tell the technologies and assets you'd prefer to be assigned to you in the automated project assignment.

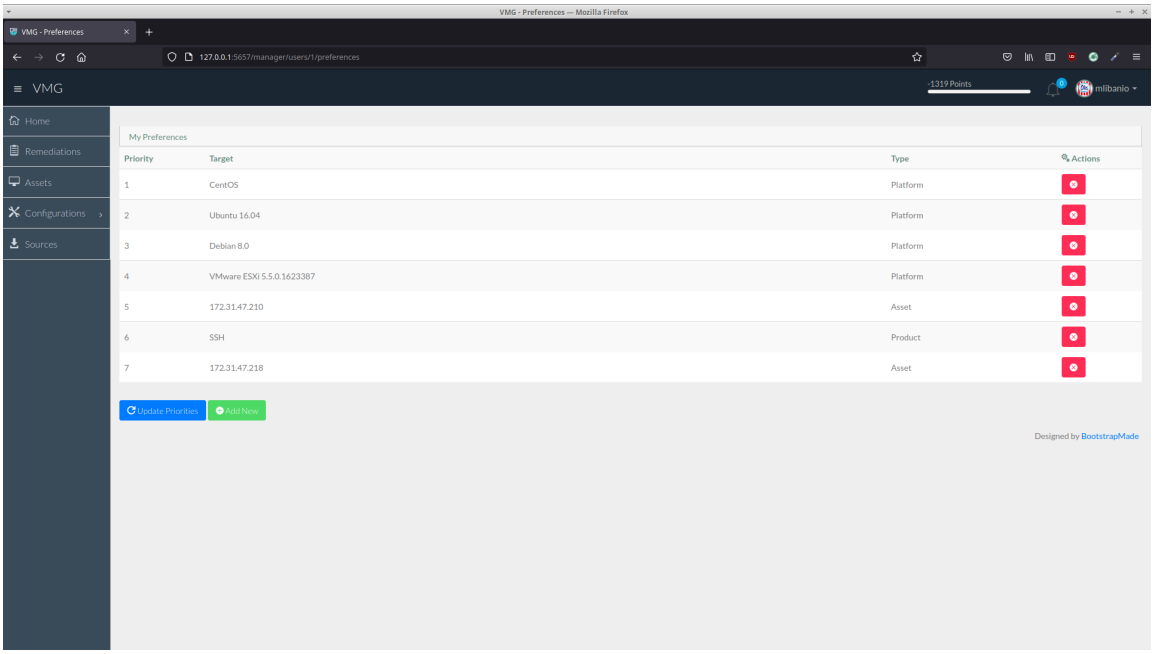



Figure 7: Preferences Page.

# 8 Remediations

In the remediations page you will be presented with all the remediation projects of the customer/s you belong. As seen on the image bellow, besides the usual information such as names, assignees, and due dates, each project has been classified with a priority which is based on the likelihood of its findings and they corresponding severity. Each project, if the max resolution time for each project has been configured, will have a corresponding deadline which upon being reached will removed the assigned user a substantial amount of points based on the priority of the remediation projects. If a project has passed an due date and is more than 3 month old it will show up in the "Overdue Remediations", which can be accessed by selecting the tab of the same name. In this tab the admin users will be able to close these projects, also debiting a large amount of points from the responsible users.

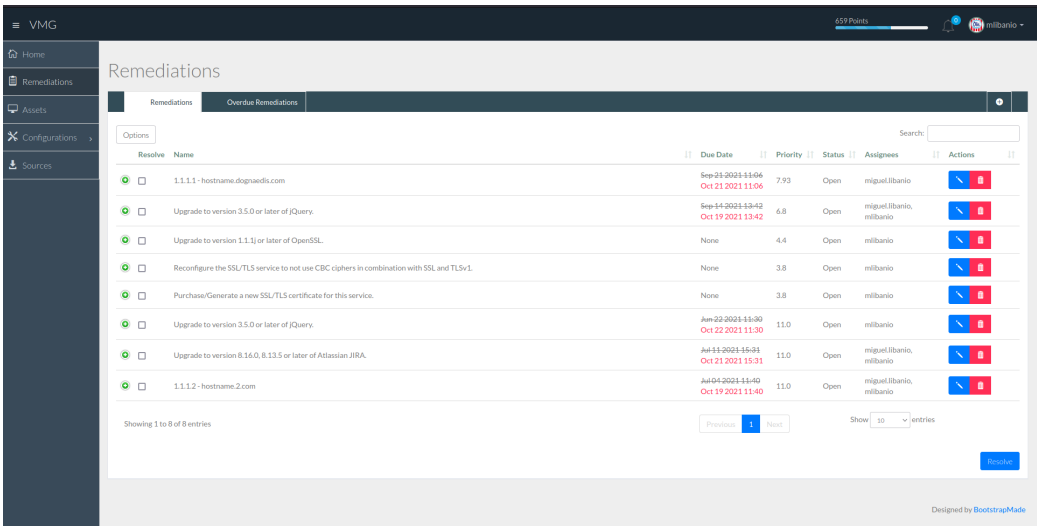



Figure 8: Remediation Projects Page.

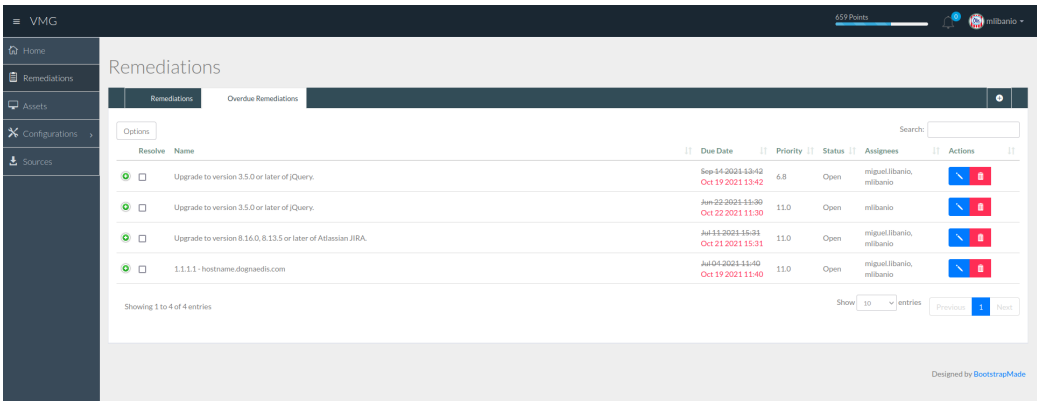



Figure 9: Overdue Projects Tab.

To view the findings press the "+" button on the left side of the project, this will expand the project and show the findings of that project, to view a finding details you only need to select the expand option of the finding again.

To ensure that your points aren't taken and you make points instead, you could start closing projects. This can be done by resolving the findings associated to an project, by selecting the checkbox of each finding or project you wish to close and then selecting the resolve button at the end of the page, this will open a window with three options:

- Verify - This option will launch a scan to verify if the finding as been resolved, if so it will update the information and close the finding awarding the user points.
- False Positive - This will declare the finding as a false positive, this option awards less points than the previous one.
- Accept - This declares the finding as accepted for the stipulated number of days or forever. Also awarding the user points.

Another way to earn points and contribute to the closing of projects is to comment on a finding or project, this can be done by selecting the comment button at the bottom of each finding and project.

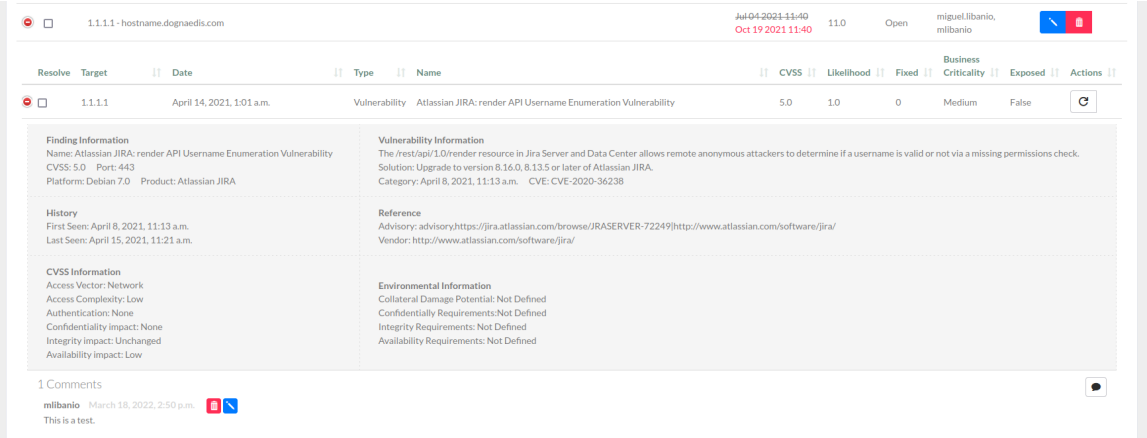



Figure 10: Comment Finding.

But to close remediation projects we must first create them, which is only accessible to authorized users, this could be done by selecting the plus button on the right side of the tabs. You will be presented with a popup window with two options, the "automated creation" automatically creates and assigns remediation projects and then presents the user with previews so you can select the ones you wish to create. The second option allows for the manual creation of projects through filters were you can select the findings with the specified attributes and how you want to group the findings (by solution, asset or even not grouped), the last tab allows for the selection of the user you wish to assign these projects, if you leave it empty it will be assigned automatically.

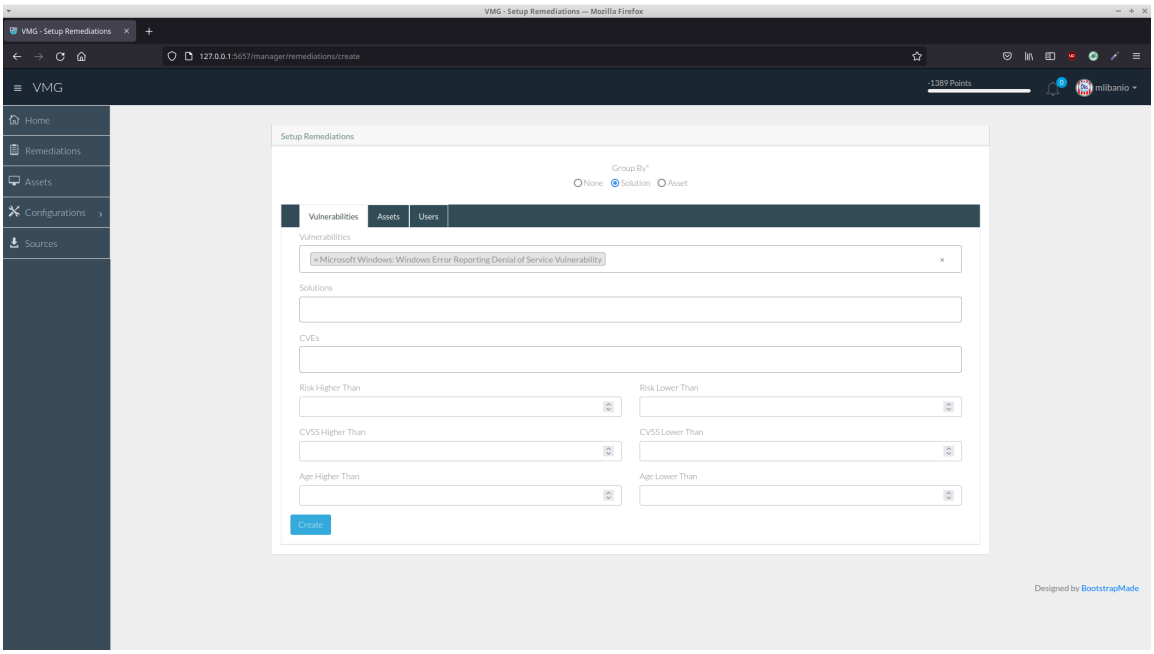



Figure 11: Manual Remediation Project Creation Page.

Once you are ready select the create option to proceed to the previews page where you'll need to select the projects that will be created and then select "finish" to complete the process and be redirected back to the remediations page.

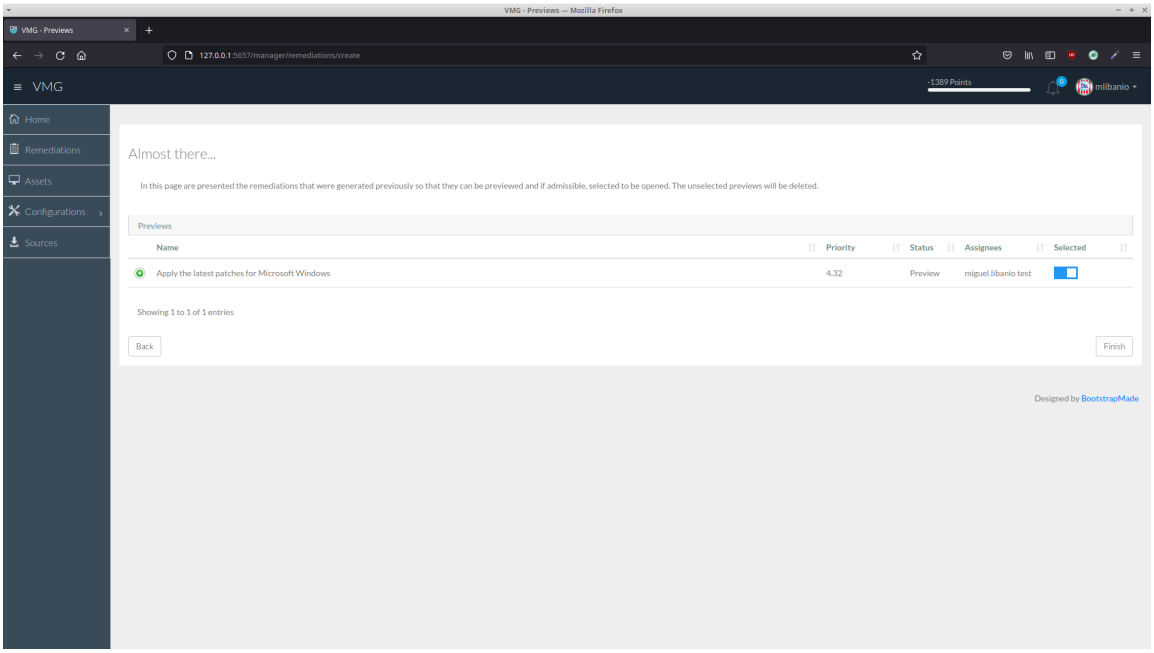



Figure 12: Preview Page.

# 9 Assets

In the assets page you will be presented with all the assets of the customer/s you belong and if you, like in the remediations page, select the expand option of the asset the findings belonging to this device will be displayed. Each asset also has the option to launch a re-scan of the machine by pressing the re-scan action and selecting the intended scan, this will launch a scan for the machine and a task to then import the results and update the old findings with the new data.

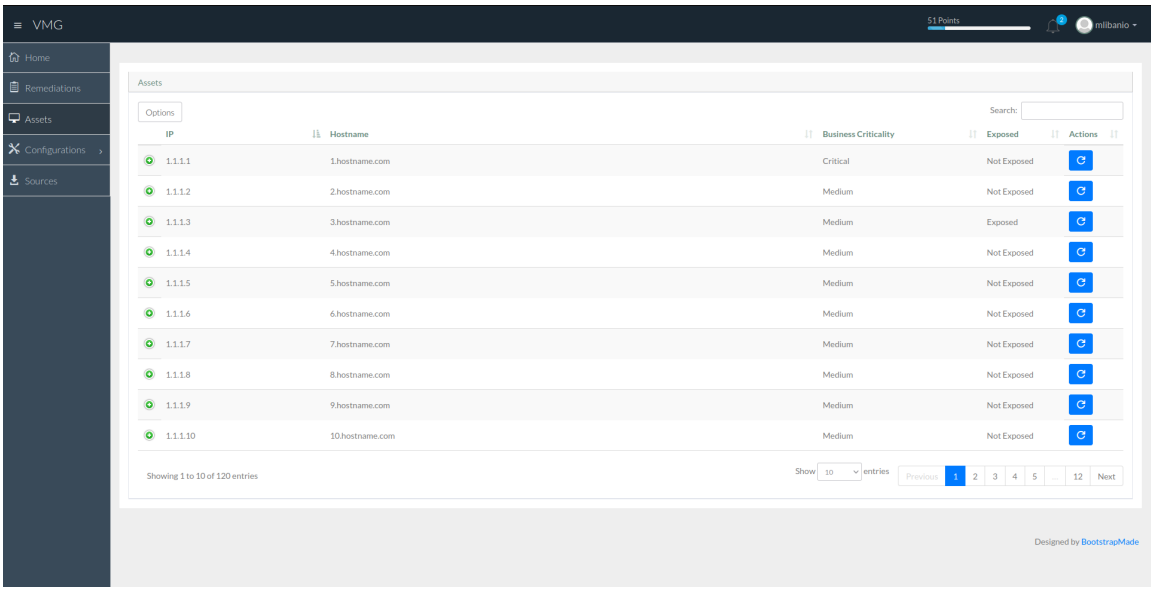



Figure 13: Asset Page.

# 10 Users and Teams

To access the page related to user management you need to once again press the "Configurations" option on the side menu and then select "Users". Like the previous page on the configurations section, this page is only accessible to authorized users. In this page you can create new users or access they profile page and also create teams, teams serve as a way to indicate to the automatic remediation projects assignment systems which users to prioritise group together or assign new projects to.

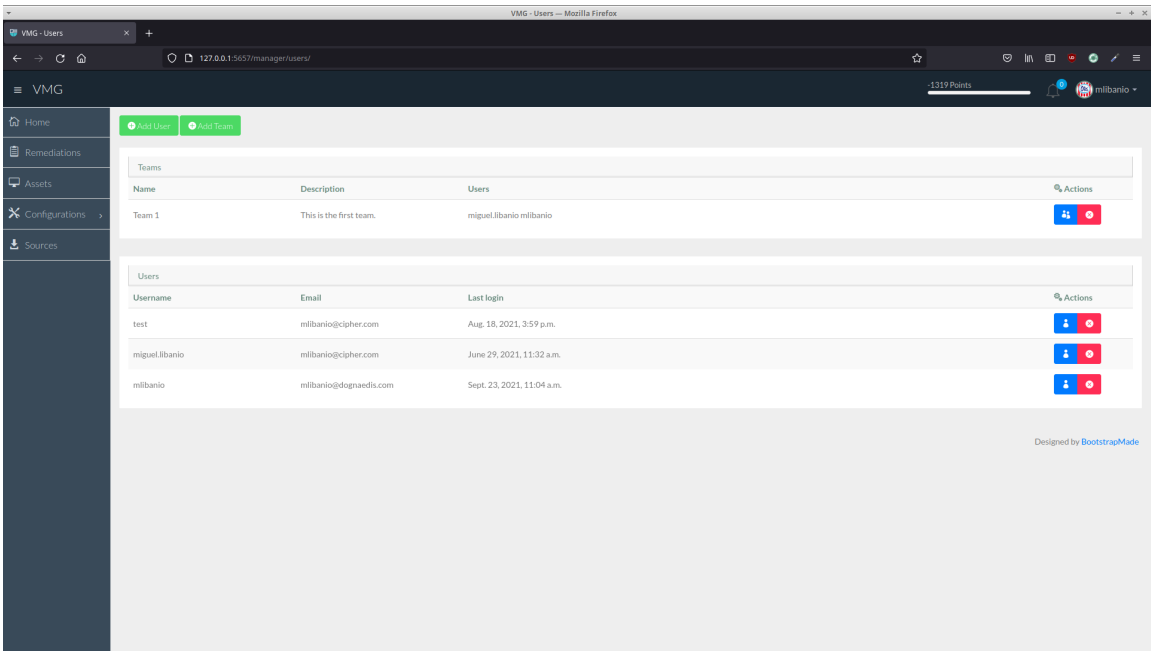



Figure 14: Users and Teams Page.

# 11 Tournaments

Lastly we have the tournaments page where we can create or edit tournaments. A tournament is an event that occurs within the stipulated time frame and at the end, the user or users with the most points awarded during the tournament receive an award such as merch. The tournaments will show up on the calendar on the home page a month before they occur. You can also see more information about ongoing an upcoming tournaments on the "Tournaments" tab of the "Tasks and Events" window on the home page.

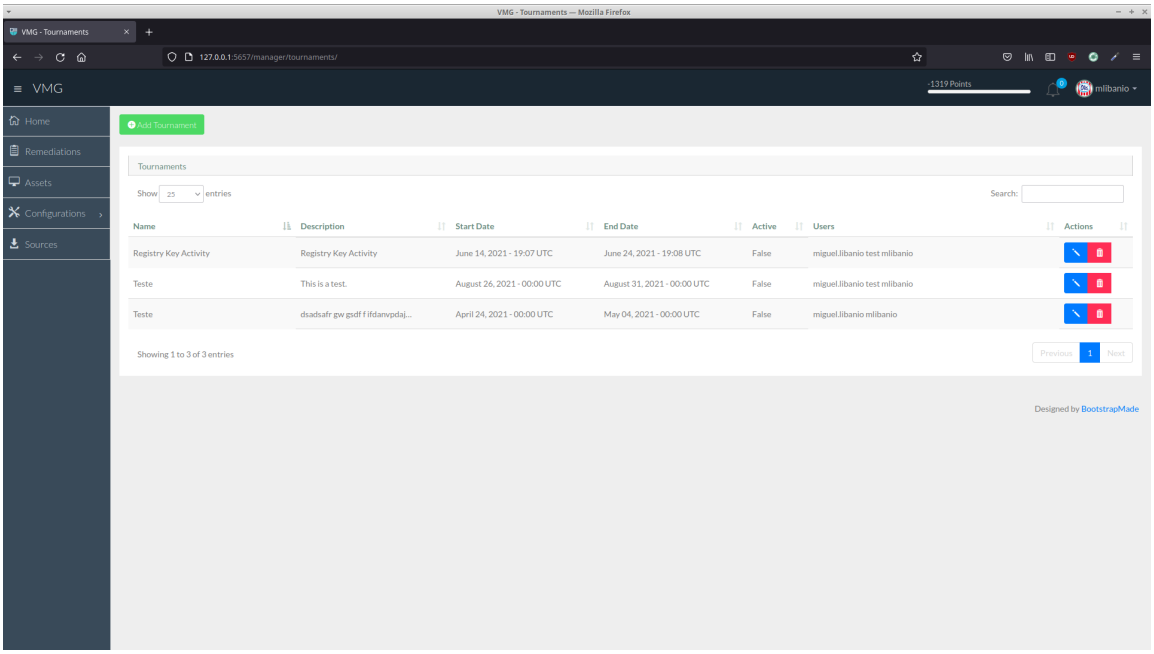



Figure 15: Tournaments Page.

# DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado neste projeto, com o título *"Vulnerability Management Gamification"*, é original e foi realizado por Estudante Miguel de Almeida Martins Libânio (2190378) sob orientação de Professor Doutor Paulo Manuel Almeida Costa paulo.costa@ipleiria.pt.

*Leiria, Março de 2022*

Estudante Miguel de Almeida Martins Libânio