POLITÉCNICO DE LEIRIA

Open Source Intelligence – Redes Sociais

Mestrado em Cibersegurança e Informática Forense

Carlos Eduardo Silva Nascimento Gomes

Leiria, setembro de 2020

POLITÉCNICO DE LEIRIA

Open Source Intelligence – Redes Sociais

Mestrado em Cibersegurança e Informática Forense

Carlos Eduardo Silva Nascimento Gomes

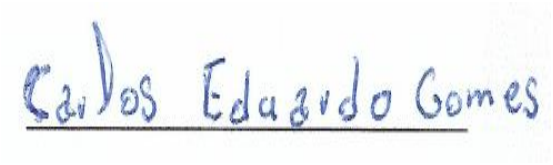
Trabalho de Projeto realizado sob a orientação do Professor Especialista Carlos Manuel Gonçalves Antunes

Leiria, setembro de 2020

Originalidade e Direitos de Autor

O presente relatório de projeto é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para o elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor/a e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Curso de Mestrado em Cibersegurança e Informática Forense, no ano letivo 2019/2020, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.



Carlos Eduardo Silva Nascimento Gomes

Agradecimentos

Queria agradecer ao politécnico de Leiria pela oportunidade de realizar o meu mestrado em Cibersegurança e informática forense. Queria agradecer também ou meu orientador Carlos Manuel Gonçalves Antunes por estar sempre presente e por ajudar em tudo que foi preciso ao decorrer deste projeto. Também queria agradecer a todos os meus amigos que me acompanharam ao longo do mestrado e que me ajudaram sempre que possível. O por último agradecer os meus pais pela oportunidade de realizar o meu mestrado em Portugal e pela confiança deles em min.

Resumo

O presente relatório foi elaborado no contexto da unidade curricular de projeto do mestrado em Cibersegurança e Informática forense.

Atualmente, cada vez mais as pessoas colocam os seus dados na Internet e os mesmos são armazenados. O projeto tem como base o Open Source Intelligence. Open Source Intelligence, ou OSINT, retrata a recolha de dados de fontes abertas na qual não é preciso efetuar qualquer pedido formal para tal. Como fonte aberta denomina-se qualquer fonte onde a informação é guardada e disponibilizada de forma aberta no sentido de que qualquer pessoa pode aceder aos dados guardados, como por exemplo relatórios públicos, redes sociais entre outros. Existem inúmeras técnicas e ferramentas de recolha de informação OSINT, dos quais iremos retratar os principais.

No decorrer deste projeto foram realizados estudos as técnicas e ferramentas de recolha de dados das fontes abertas. Também foi possível efetuar o desenvolvimento de uma solução que possa recolher dados de fontes abertas e mostrar os mesmos ao utilizador os dados guardados na internet sobre os mesmo de uma forma sumarizada.

Após a realização deste projeto foi possível estudar as diferentes ferramentas já existente, bem como o desenvolvimento de uma solução, que permitiu ver a quantidade de informação que pode ser encontrada na Internet.

Palavras-chave: OSINT, fontes abertas, informações

Abstract

This report was prepared in the context of the master's degree in Cybersecurity and Computer Forensics.

Nowadays, more and more people put their data on the Internet, and it is stored. The project is based on Open Source Intelligence. Open Source Intelligence, or OSINT, portrays the collection of data from open sources in which it is not necessary to make any formal request for this. An open source is any source where information is stored and made available in an open way in the sense that anyone can access the data stored, such as public reports, social networks, among others. There are numerous techniques and tools for collecting OSINT information, of which we will portray the main ones.

Throughout the project, studies were carried out on techniques and tools for obtaining data from open sources. It was also possible to carry out the development of a solution that collected data from open sources and show them to the user the data saved on the Internet about them in a summarized way.

After carrying out this project, it was possible to study how different tools that already exist, as well as the development of a solution, which consumes the amount of information that can be found on the Internet.

Keywords: OSINT, Open Source, Information

Índice

Lista de Figuras

Lista de Tabelas

Lista de Siglas e Acrónimos

API	Application Programming Interface
BD	Base de Dados
CRUD	Create Read Update Delete
CSV	Comma-separated Values
ESTG	Escola Superior de Tecnologia e Gestão
EUA	Estados Unidos da América
IP	Internet Protocol
JS	JavaScript
JSON	JavaScript Object Notation
OSD	Open Source Data
OSI	Open Source Information
OSINT	Open Source Intelligence
OTAN	Organização do Tratado Atlântico Norte
PHP	PHP Hypertext Preprocessor
SQL	Structured Query Language
URL	Uniform Resource Locator
XML	Extensible Markup Language

1. Introdução

Atualmente, a Internet faz parte do quotidiano das pessoas. Nas últimas duas décadas assistiu-se ao estabelecimento gradual da Internet como um pilar da sociedade contemporânea, em grande parte devido à sua simplicidade de uso e à utilidade proporcionada por esta. Cada vez mais as pessoas estão a colocar os seus dados nesta e, os mesmos são armazenados pelas suas estruturas. Os utilizadores da Internet deixam um rastro, isto é, informação relativa a estes nas páginas que visitam.

A Organização do Tratado do Atlântico Norte (OTAN, mais conhecido pelas siglas em inglês NATO), desde 2001 definiu os conceitos de "Open Source Data" (OSD) e "Open Source Information" (OSI) referentes a informação antes de ser recolhida e tratada (Antunes & Rodrigues, 2018). Assim, existem técnicas capazes de extrair informações de fontes abertas, estas denominadas de técnicas Open Source Intelligence (OSINT). Open Source Intelligence ou OSINT, como é conhecido, consiste na recolha de informação de fontes abertas ao qual não é necessário efetuar um pedido formal para tal. A NATO define OSINT como "… a informação não classificada que foi deliberadamente descoberta, discriminada, destilada e disseminada para uma audiência selecionada, de modo a responder a uma questão especificas" (Kernan, 2001). As técnicas OSINT surgiram nos EUA (Estados Unidos da Améria) em meados da década de 70 e na Europa na década de 80 do seculo XX.

A importância destas técnicas é hoje reconhecida pelos peritos criminais e consequentemente pelos peritos de informática forense, podendo assim ajudá-los no combate ao crime organizado. Desde esse momento, com a globalização da Internet e o aparecimento de serviços de pesquisa de conteúdos na mesma, nomeadamente os motores de busca, como é o exemplo da Google, as técnicas OSINT tem sido fortemente desenvolvidas e aparecem cada vez mais ferramentas que implementam as mesmas. É de realçar que a maioria das técnicas ou ferramentas são facilmente acessíveis para os utilizadores da Internet. As técnicas OSINT surgiram com o objetivo de recolha de informação pelos serviços secretos e de inteligência, para obter informação acerca dos cidadãos e das suas atividades no território, como forma de antecipar atividades ilícitas para poder garantir a segurança do país.

1.1. Objetivo

A análise de informação obtida através de fontes abertas pode ser um elemento importante na realização de relatórios digitais forenses e na antecipação de potenciais perigos. Por isso, com o desenvolvimento deste projeto, pretende-se estudar as ferramentas já existentes de pesquisa de informação OSINT e desenvolver uma solução capaz de recolher informações de fontes abertas, consumir as mesmas e mostrar de uma forma sumarizada ao utilizador as

informações relativas à pesquisa efetuada. O que distingue esta solução das demais será a possibilidade de pesquisar com base num histórico, isto é, pesquisar os dados com base em datas.

1.2. Planeamento

Figura 1 - Planeamento

O início do projeto consistirá em estudar o que já existe acerca de OSINT. Nesta primeira fase será feito um levantamento e estudo de ferramentas já existentes.

A segunda fase do projeto será o planeamento de uma arquitetura para o desenvolvimento do projeto. Nesta fase será planeado qual a linguagem a ser usada e ainda será definido o esquema da BD (Base de dados) entre outras ações.

A terceira fase do projeto consistirá no desenvolvimento da página web. Nesta fase será seguido a arquitetura previamente feita e será dado o início a desenvolvimento da página web em si.

A quarta fase do projeto consistiu na conclusão do mesmo. Nesta fase será feita a correção de alguns *bugs* que possam surgiu e fazer algumas alterações previstas com base nos testes executados e resultados obtidos.

A quinta e última fase foi a fase de ajustes no relatório. Nesta fase será feito os últimos ajustes no relatório. É de realçar que ao longo das diferentes etapas sempre será feita a escrita do relatório.

1.3. Estrutura

Este relatório está dividido em 5 capítulos. O primeiro capítulo retrata a introdução onde será feito uma breve contextualização e também os objetivos esperados com o desenvolvimento do projeto. Seguidamente o segundo capítulo trata-se do Estado da Arte, onde será efetuado um estudo do que já existe acerca de OSINT. De seguida no terceiro capítulo será retratada a arquitetura da página web, e definidas as tecnologias utilizadas no desenvolvimento da solução. A seguir no quarto capítulo será retratada a solução em si, onde serão especificados os detalhes da solução bem como as fontes utilizadas e a linguagem de programação. O quinto capítulo será a fase de testes onde serão realizados testes a solução. Por último, o sexto capítulo será a conclusão onde será retratado as conclusões retiradas com o desenvolvimento do projeto e trabalhos futuros.

2. Estado da Arte

OSINT refere-se a recolha de informação de fontes abertas sem necessidade de qualquer pedido formal ou autorização para tal.

Quando se trata a recolha de informações OSINT, isto é, informações de fontes abertas existem diversas técnicas e ferramentas capazes de obter as mesmas. As técnicas podem ser divididas por módulos, por exemplo, um módulo para redes sociais, módulos para websites entre outros. Como exemplo de técnicas de recolha de informação OSINT existem os motores de busca, redes sociais, fóruns, repositórios públicos, etc.

Existem um conjunto infindável de soluções e plataformas de recolha de informação, das quais se destaca as que se seguem como base de análise.

2.1. OSINT Framework

A plataforma OSINT Framework (OSInt Framework, n.d.), é uma solução open source que tem um repositório publico no github, onde é possível efetuar contribuições para o mesmo (lockfale, n.d.).

A solução mostra diferentes técnicas e ferramentas, com base no que o utilizador pretende pesquisar, no qual o mesmo pode utilizar para realizar a pesquisa pretendida. Abaixo será dado alguns exemplos de tópicos no qual a página web mostra técnicas para realizar as pesquisas:

- Nome de utilizador
- Endereço email
- Nome de domínio
- Endereço IP (Internet Protocol)
- Redes Sociais

Para cada tópico a página web mostra ainda, para alguns casos, subtópicos como é o caso do endereço de IP. Segue em baixa alguns exemplos de subtópicos para o caso de endereço IP:

- Geolocalização
- Endereços IPv4
- Endereços IPv6
- Descoberta de portos

De seguida para cada um dos subtópicos, ou tópicos no caso de não existirem subtópicos, a solução apresenta então as diferentes técnicas para pesquisar a informação pretendida. Como por exemplo de descoberta de portos, neste caso, existe a técnica Shodan, que será retratado

posteriormente. O Anexo A – OSINT Framework representa a página web OSINT Framework.

2.2. Técnicas de recolha de informação OSINT

Neste subcapítulo serão retratadas algumas técnicas de recolha de informação OSINT.

2.2.1. Motores de busca

Através dos motores de busca, pode ser obtida muita informação através das pesquisas avançadas das mesmas.

Como exemplo dessas pesquisas avançadas existe *o Google Hacking*. O *Google Hacking* diz respeito ao motor de busca Google e com o mesmo podem ser realizadas vários tipos de pesquisas. Com o auxílio do *Google Hacking*, pode ser efetuada pesquisa em que um determinado texto tenha de estar presente nos resultados das pesquisas, pode ser feito uma pesquisa apenas numa determinada página web ou até pesquisar por um determinado tipo de ficheiro, como por exemplo, por pdf. No Anexo I – Google Hacking é possível ver um exemplo de *google Hacking*.

2.2.2. Redes Sociais

Em termos de redes sociais, estas são uma fonte inesgotável de informações OSINT. Cada vez mais, as pessoas publicam detalhes pessoais das mesmas, e muitas outras informações nas redes sociais. Como exemplo de rede sociais existem: o Facebook, o Instagram entre outros.

- Social Searcher

Falando agora da plataforma web Social-Searcher (Social Searcher, n.d.), permite pesquisar referências, por utilizadores ou então por tendências. A página web usa API (Application Programming Interface) abertas para realizar as pesquisas. Como fontes de pesquisa a página web utiliza o Facebook, Linkedin, Instagram, Vimeo e o Dailymotion. O utilizador tem a possibilidade de escolher das fontes mencionadas acima qual delas quer realizar a pesquisa. A página web ainda permite exportar o resultado da pesquisa para um ficheiro CSV (*Comma-Separated Values*). O website ainda disponibiliza aos utilizadores uma API para poder ser usado consoante a necessidade dos utilizadores. A API possui uma versão livre que permite realizar até 100 pesquisas diárias e versões pagas que permitem realizar até 800 pesquisas diárias consoante o preço que os utilizadores estiverem dispostos a pagar. O Anexo B – Social Searcher permite ver a um exemplo da página web.

- 4k Stogram

Falando do 4k Stogram (4kStogram, n.d.), este permite descarregar todas as fotos e vídeos de um utilizador, de uma só vez do Instagram. Esta ferramenta poderá ser útil caso utilizado juntamente com algum outra ferramenta de analise dos metadados das imagens.

2.2.3. Obter informação acerca de um nome de utilizador

Existem páginas web que permitem saber se um determinado nome de utilizador existe ou não em diversas outras páginas. Como exemplo existe o *Check UserNames* e o *User Search*.

- Check Usernames

Começando pelo *Check UserNames* (Check UserNames, n.d.), este como já referido anteriormente permite pesquisar se um determinado nome de utilizador existe em cerca de 160 redes sociais diferentes. Este por sua vez não disponibiliza nenhuma API que pode disponibilizar para os utilizadores. O Anexo C – CheckUserName permite ver um exemplo da execução da página web.

- User Search

A plataforma *User Search* (User Search, n.d.) idêntica ao *Check UserNames* no sentido de permitir pesquisar por um nome de utilizador, mas este apresenta o URL (Uniform Resource Locator) para o perfil do nome de utilizador encontrado. A página web também permite pesquisar por endereço email e também por número de telefone.

2.2.4. Números de Telefone

Existem ainda páginas web capazes de saber informação através do número de telefone. Algum exemplo dessas páginas são o *Numbering Plans* e o *Who calld*.

- Who Calld

O *Who calld* (Who Calld, n.d.) após fornecer o número de telefone fornece dados acerca de onde é o número e também da hora no local onde se encontra o número.

- Numbering Plans

De seguida falando do *Numbering Plans* (Numbering Plans, n.d.), este assim como o anterior, permite obter dados acerca de um número de telefone. O que difere esta página web da outra acima mencionada é que esta também disponibiliza a operadora no qual o cartão pertence. A página web também permite pesquisar pelo *imei* de um telefone. O Anexo D – Numbering Plans, permite ver um exemplo da execução.

2.2.5. Metadados

Metadados são dados sobre os dados (ICANN, n.d.), isto é, dados que permitem descrever a informação armazenada digitalmente. Como por exemplo temos uma fotografia como

metadados seriam dados da fotografia, como por exemplo, a câmara que tirou a foto, a localização de onde foi tirada a foto entres outros dados. Já existem páginas web que permitem fazer o upload de imagens, e a página, extrai os metadados das imagens.

2.2.6. Domínios

Para registar um endereço de domínio é necessário a informação do dono e consequentemente um endereço de e-mail associado ao registo, entre outros dados específicos da pessoa. Já existem serviços na internet e ferramentas que usam essas informações recolhidas de fontes abertas. A informação acerca de quem fez o registo do domínio poderá vir a ser útil pois quando há mudança do endereço de e-mail de um domínio, este poderá ser considerada uma forma de ocultação de evidências. A seguir é retratado alguns exemplos.

- ViewDns Whois

Este serviço (ViewDns, n.d.) gratuito, e permite pesquisas ilimitadas sobre domínios ou endereço IP. A página principal apresenta várias ferramentas que podem ser usadas. Após efetuar a pesquisa por um domínio é possível saber quando é que o domínio foi criado quem é o dono do domínio, o endereço de email de quem criou o domínio entre várias outras informações.

- Whoxy

Este serviço (Whoxy Domain Search Engine, n.d.), assim como o ViewDns, permite saber informações acerca de um domínio. Após a análise de um domínio, este permite saber o endereço de e-mail no qual efetuou o registo do domínio, bem como a que instituição pertence, a data que o domínio foi registado entre outras informações. Este serviço permite ainda, escolher a forma em que os dados são mostrados, podendo este serem mostrados em JSON (JavaScript Object Notation) ou XML (Extensible Markup Language). Este possui uma API só que paga.

2.2.7. Endereço IP

Como base no endereço IP é possível saber várias informações tais como a localização do endereço IP, *torrents* baixados por determinado endereço IP bem como muitas outras informações. Já existem ferramentas que recolhem informação de fontes abertas através do endereço de IP. Na Secção 2.3 serão retratadas algumas ferramentas que realizam pesquisa através do endereço IP.

- I know what you download

Este serviço (I know what you download, n.d.) na internet permite saber os *torrents* baixados por um determinado endereço IP, Este serviço pode vir a ser útil no caso de violação de

direitos de autor. Ao realizar a pesquisa, a página web apresenta quando foi realizada a descarga, que tipo de ficheiro foi descarregado e o nome do ficheiro descarregado.

2.3. Ferramentas de recolha de Informação OSINT

Em termos de ferramentas existe o Maltego, o Metagoofil, o Hypersight e o Spiderfoot, Shodan, Hunter, MailDB, Censys.

2.3.1. Shodan

A solução web Shodan (Shodan, n.d.) (Geng &˘achescu, 2016), é uma solução web capaz de indexar a Internet, tentando descobrir dispositivos ou serviços disponíveis na mesma. Após a indexação e descoberto informações, essas mesmas informações, nomeadamente o endereço IP, os portos e os serviços, são guardados numa BD. O Shodan disponibiliza um API para realizar pedidos a sua BD de forma a poder ser utilizada por outros programas. Para utilizar a API do Shodan o utilizador precisa de uma chave para poder realizar os pedidos, após realizar o pedido, os dados são devolvidos em formato JSON. O Anexo E – Shodan, permite ver um exemplo do Shodan em execução.

2.3.2. Hunter

A página web Hunter (Hunter, n.d.), permite descobrir endereços de email de um determinado domínio. O Hunter interroga os servidores de email se um determinado email existe e caso exista são armazenados na BD todos os registos encontrados nesse domínio. A página também mostra as fontes onde os registos de email foram encontrados. O Hunter web, assim como o Shodan, também disponibiliza uma API para que possam ser realizados consultas as suas BD. O Hunter também possui uma parte *premium*, paga, que é possível exportar o resultado da busca para um ficheiro CSV. O Anexo F – Hunter.io, permite ver em execução o Hunter.

2.3.3. MailDB

A página MailDB (MailDB, n.d.), assim como o Hunter, tenta descobrir os registos email de um determinado domínio. Após encontrar os registos os mesmo são guardados numa BD. Assim como a maioria da ferramenta, esta também disponibiliza uma API para que se possam realizar consultas as suas BD.

2.3.4. Censys

Seguidamente existe a página web Censys (Censys, n.d.), esta permite saber informações acerca de um endereço IP, de um domínio ou de certificados. Pesquisando por um endereço IP o website permite a localização do endereço IP, bem como os portos abertos na máquina e também os serviços correndo nesses portos. Assim como os outros websites, o Censys possui um API só que paga.

2.3.5. Maltego

O Maltego (Paterva, n.d.), é uma ferramenta poderosa capaz de obter várias informações OSINT, como por exemplo informações acerca de um domínio. Esta possui várias versões consoante a necessidade do utilizador, sendo a versão que disponibiliza mais informações paga. A ferramenta funciona com base em transformações, no sentido que cada transformação tenta descobrir informações consoante a transformação realizada. A Figura 2, representa um exemplo da ferramenta Maltego, sobre o domínio ipleiria.pt. O Anexo G – Maltego mostra um relatório gerado pela execução da ferramenta Maltego

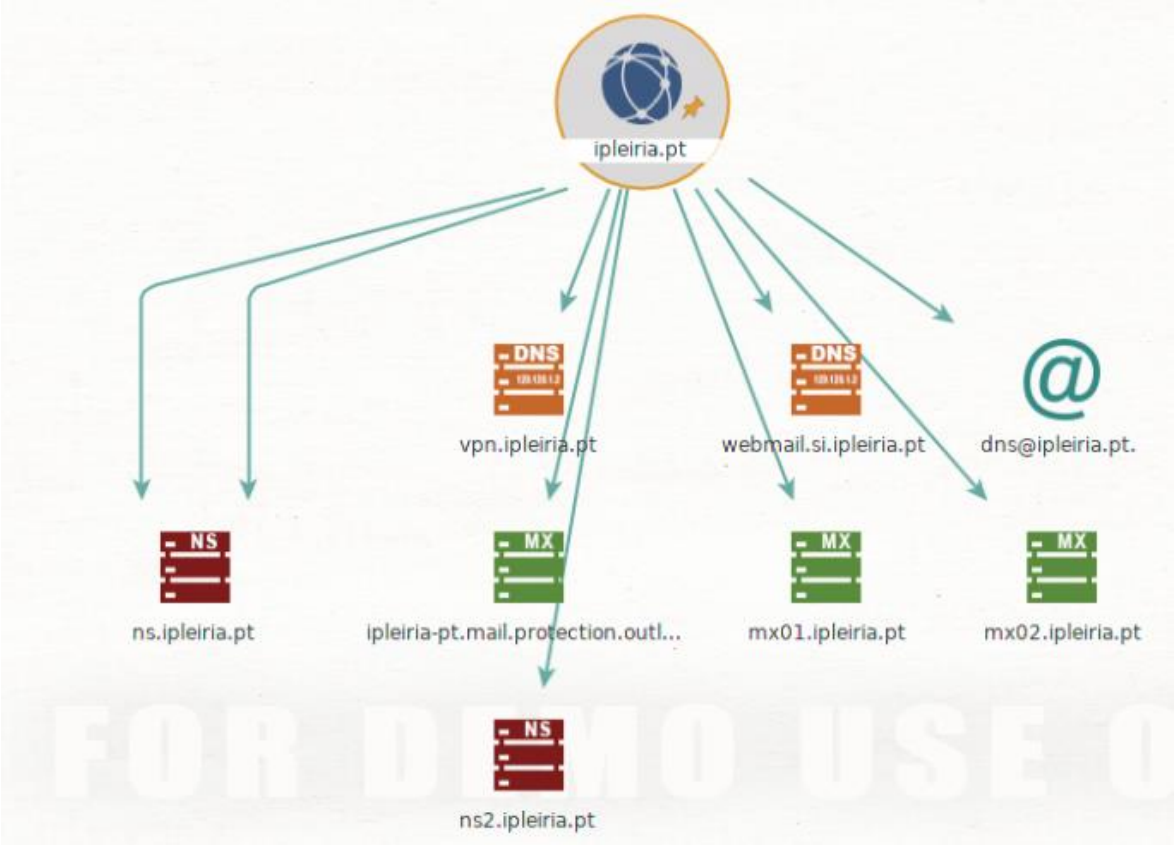


Figura 2 – Maltego

2.3.6. Metagoofil

O metagoofil (Kali Linux Tools, n.d.), é também uma ferramenta poderosa, capaz de extrair metadados de documentos públicos. A ferramenta permite especificar um domínio e o tipo de ficheiros que o utilizador quer extrair os metadados.

2.3.7. Hypersight

O *Hypersight* (Hypersight, n.d.) é mais uma ferramenta, que permite extrair várias informações OSINT. A ferramenta permite com base em informações OSINT monitorizar ameaças de segurança, prevenção de fraude entre outras funcionalidades. Esta é uma ferramenta muito completa, só que é uma ferramenta paga.

2.3.8. Spiderfoot

Como ferramenta existe também o *Spiderfoot* (Spiderfoot, n.d.), está uma ferramenta bastante completa na recolha de informações OSINT. O objetivo da ferramenta é automatizar o processo de coleta de informações acerca de um determinado alvo, podendo este ser um endereço IP, um domínio, endereço de email entre outros. A ferramenta funciona por módulos sendo cada módulo responsável pela obtenção de informação de um determinado tipo de dados. Para obtenção das informações, os módulos vão pesquisar informação através de API disponibilizadas por outras ferramentas. Como exemplo de API que a ferramenta usa estão o *Shodan*, VirusTotal, *Hunter*, *Censys* entre outros que já foram falados. Para obter as informações acerca de um alvo é possível criar casos para cada alvo. Dentro de cada caso é possível escolher os módulos que o utilizador pretende correr. No fim da recolha de informação é possível exportar a analise para um ficheiro CSV.

Existem vários outros exemplos de técnicas e ferramentas de recolha de informação OSINT. A desvantagem da maioria é que as diferentes soluções que já existem dão mais foco apenas a fontes dos Estados Unidos da América. O Anexo H – Spiderfoot.

2.3.9. iKy

Retratando agora da ferramenta *I Know You (iKy)* (iKy, n.d.). Esta recente surgida em 2019, no qual também possui um repositório GitHub (kennbroorg, n.d.). Esta ferramenta recolhe informações com base num endereço de email fornecido pelo utilizador. A ferramenta, para o devido funcionamento, necessita de chaves de algumas API's, como por exemplo, a API do Twitter e do FullContact. A ferramenta também precisa de uma conta do Linkedin. A ferramenta tenta encontrar as contas de redes sociais associadas a esse endereço de email e caso encontre é mostrado numa interface ao utilizador. No caso de ser encontrado algum perfil no GitHub associado a esse endereço de email, este mostra, se possível, os seguidores, o número de repositórios e muitas outras informações. Também mostra o histórico dos *commits* efetuados pelo utilizador. Quanto ao Linkedin, no caso de ser encontrado algum perfil associado ao endereço de email tenta mostrar as capacidades do utilizador bem como todas as informações publicas. No caso do Twitter, a ferramenta mostra os números de seguidores, o número de pessoas a seguir a quantidade de publicações entre outras informações. A ferramenta ainda apresenta uma linha temporal das atividades do endereço de email, caso consiga encontrar. Permite ainda exportar o resultado da pesquisa em formato JSON. O Anexo N - iKy, permite ver a ferramenta.

2.4. Quadro Comparativo

Nesta secção será feita um quadro comparativo entre algumas ferramentas previamente retratadas acima. Nesta Tabela 1, constara a dificuldade, se é grátis, a maneira de funcionamento e caso a mesma possui API.

Tabela 1 - Quadro Comparativo

	Dificuldade	Grátis	Funcionamento	API
Motores de Busca	Difícil	Sim	Funciona com base em pesquisas avançadas	Não possui
Shodan	Medio	Sim	Saber os portos e serviços a correr e as respetivas vulnerabilidades	Possui
Hunter	Fácil	Sim	Interroga os servidores de email, guardando numa BD	Possui, 50 pesquisas/mês
MailDB	Fácil	Sim	Necessário criar conta. Guarda as respostas numa BD	Possui
Censys	Medio	Sim	Saber os portos e serviços a correr	Possui, mas paga
Spiderfoot	Medio	Sim	Funciona por módulos, buscando dados através de API's	Não possui
Maltego	Medio	Sim e possui versão paga	Funciona por transformações	Não possui
Social Searcher	Fácil	Sim	Informação em tempo real de 5 redes sociais diferentes	Não possui

3. Arquitetura da Página Web

Com vista ao desenvolvimento da solução, a primeira etapa a realizar será o desenho da arquitetura. A arquitetura facilitará a fase de desenvolvimento, uma vez que define componentes da solução bem com as suas funções. Nesse capítulo será descrito a arquitetura da solução, explicado os seus componentes. A Figura 3 representa a arquitetura que servirá de base para o desenvolvimento da solução.

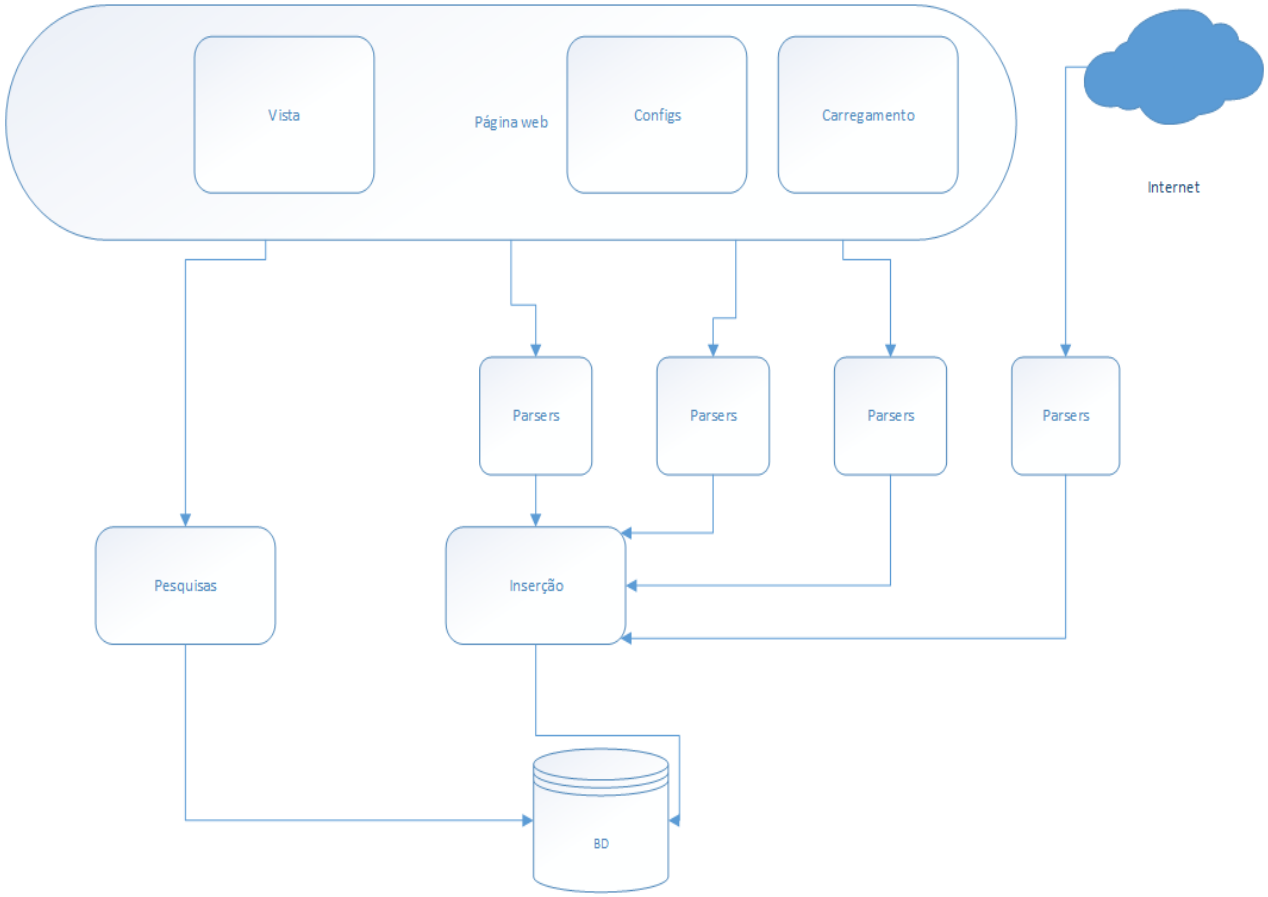


Figura 3 - Arquitetura da página web

3.1. Módulos

3.1.1. Página Web

A página web será responsável pela interface do utilizador, interligação de componentes e apresentação dos dados consoante a pesquisa realizada pelo utilizador. Este módulo subdivide-se em 3 submódulos os quais serão retratados com maior detalhe nos subcapítulos a seguir.

- *Vista*

Este submódulo é responsável pela apresentação, na página web, das informações da mesma, bem com as informações das pesquisas realizadas pelo utilizador. Este submódulo será responsável pela parte gráfica da página web.

- *Configurações*

Este submódulo será responsável, como o nome indica, pelas configurações da página web. Nesta secção poderá ser escolhido todas as configurações como por exemplo se o utilizador pretende fazer a pesquisa em tempo real, consoante os dados da BD ou então por defeito, que neste caso, será pelos dados da BD.

- *Carregamento*

O submódulo do carregamento será responsável pelo carregamento de ficheiros externos, retirados de fontes abertas ou ferramentas de extração de dados de fontes abertas, que possuem dados úteis que possam vir a ser apresentado aos utilizadores.

3.1.2. Parsers

Os *Parsers* são a base do tratamento dos dados recolhidos pelas fontes. Os *parsers* recebem os dados das fontes, trataram dos mesmo e posteriormente armazenará os dados na BD. É de referir que cada fonte consumida pela página web terá o seu próprio *parser* devido à maneira como cada fonte disponibiliza os dados. Os *Parsers* subdividem-se em 3 submódulos, que serão especificados mais detalhes nos subcapítulos a seguir.

- *Carregamento*

Os *parsers* de carregamento serão responsáveis por receber e tratar dados vindo de ficheiros. É de realçar que esses parsers só aceitam determinados tipos de ficheiros com determinados padrões, e caso não seja um padrão reconhecido o mesmo não será consumido.

- *API*

Falando dos *parsers* de API, estes consumiram dados provenientes de API externas. Estes *parsers* consoante os dados recebidos das API tratam os dados e os armazenaram na BD da aplicação de forma a que estes possam vir a ser usados nas pesquisas realizadas pelos utilizadores.

- *Páginas web*

Por últimos os *parsers* de páginas web. Estes serão responsáveis por tratar páginas web e recolher informações dessas páginas, tratar dessas informações e posteriormente guardar essas mesmas informações na BD da aplicação.

3.1.3. Inserção

Retratando agora o módulo de validação e Inserções de dados na BD. Este módulo será responsável para receber os dados já tratados pelos *parsers* e armazenar esses dados na BD.

3.1.4. Pesquisa

O módulo de pesquisas servirá de ligação entre a página web e a BD no sentido que permitirá realizar consultas a BD. Quando o utilizador pretender pesquisar alguma informação, o módulo da pesquisa tratará os dados introduzidos pelo utilizador e realizará a pesquisa pretendida na BD. De seguida o módulo da pesquisa envia os dados para serem representados na página web e mostrar ao utilizador os resultados.

3.1.5. Base de dados

O componente de BD será responsável por armazenar os dados recolhidos e tratados provenientes das fontes abertas. A BD terá de ser uma capaz de suportar grande quantidades de dados, permitir realizar consultas e inserir dados.

3.2. Arquitetura Implementada

Com base na arquitetura planeada anteriormente, foram feitas alterações ao modo como o utilizador poderá efetuar pesquisas, apenas aos dados guardados na BD, ou então pesquisar por dados em tempo real.

Numa primeira abordagem a solução web inicialmente efetuava a pesquisa na BD e caso não encontrasse os dados pretendidos efetuava a pesquisa online. Esse método possuía a desvantagem de os dados devolvidos poderem estar desatualizados. Por isso a nova arquitetura permite efetuar pesquisas diretamente na Internet se essa for a vontade do utilizador. Os restantes processos continuam iguais à arquitetura anteriormente planeada como pode ser consultado na Figura 4.

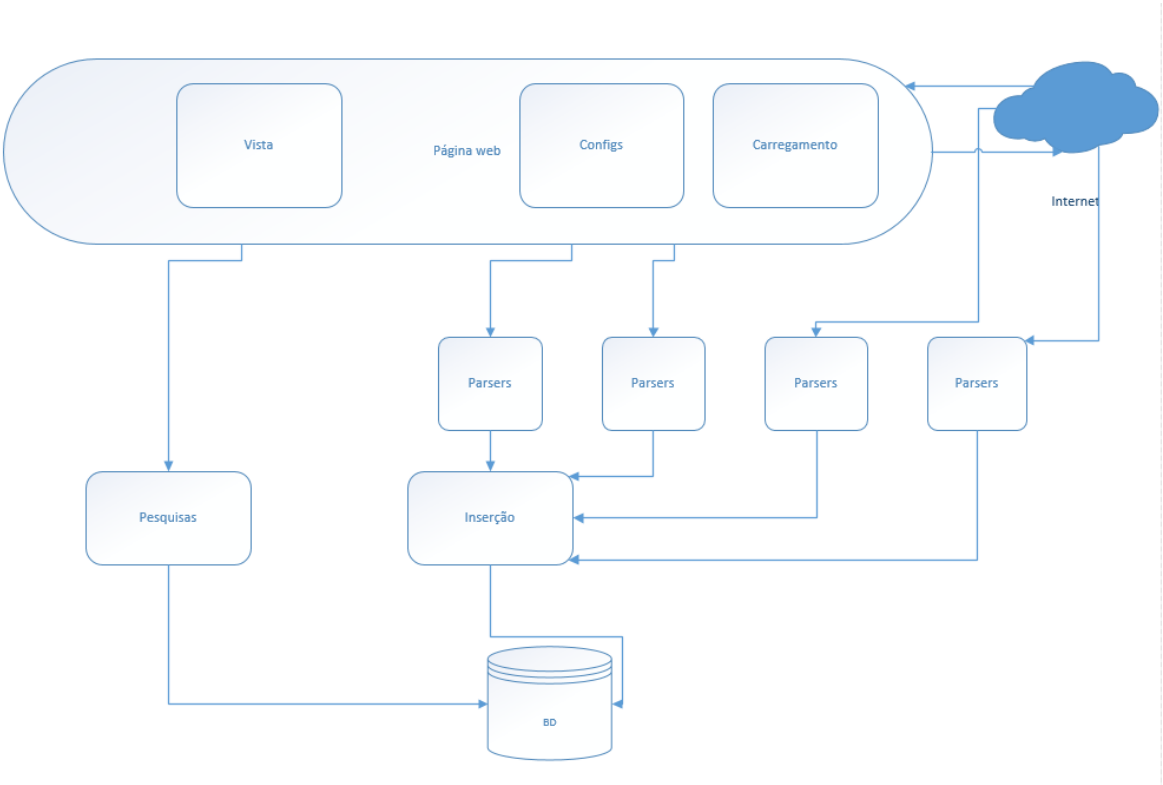


Figura 4 - Arquitetura Implementada

3.3. Fluxo de Dados

Quando uma nova fonte for consumida pela página web, esta poderá ser recolhida de três formas distintas, sendo que para utilização de API's externas há necessidade de configurações adicionais. Seguidamente os dados serão tratados pelos parsers e enviados ao módulo de inserção responsável para guardar esses mesmos dados na BD de forma relacional.

Quando um utilizador pretender realizar uma pesquisa a página web enviará a pesquisa do utilizador para o módulo de pesquisa, onde neste por sua vez será tratado. Seguidamente o módulo da pesquisa, depois de tratado os dados, realizará a pesquisa na BD. Depois de receber os dados da BD, a informação é enviada para a página web, nomeadamente a vista onde os dados serão mostrados ao utilizador.

O utilizador poderá optar por realizar a pesquisa online ou offline mediante a sua necessidade. O processo começa sempre pela inserção dos dados a serem pesquisados.

- Caso o utilizador opte por realizar a pesquisa offline a solução web enviará os dados introduzidos pelo utilizador ao módulo da pesquisa: De seguida o módulo da pesquisa é responsável pela busca dos dados na BD e a retorna dos dados devolvidos para a solução web que pode ser consultada pelo utilizador.
- No caso da pesquisa online, a solução web procura na internet os dados introduzidos pelo utilizador. De seguida os dados são enviados para parsers, onde os mesmos serão

processados e tratados para serem guardados na BD. Para finalizar os dados, após serem guardados na BD são enviados a solução web para serem apresentados ao utilizador. Os detalhes poderão ser observados na Figura 5.

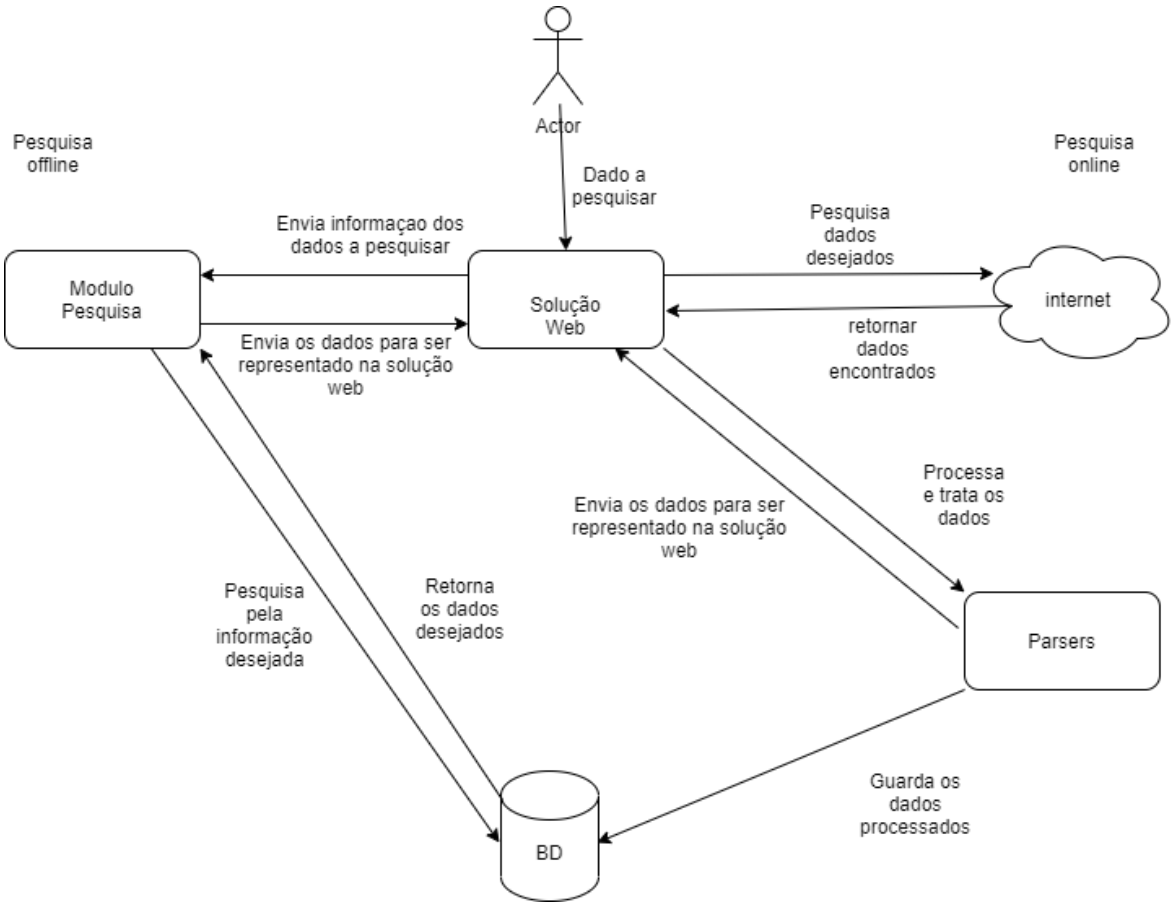


Figura 5 - Fluxo dos dados

3.4. Requisitos funcionais / tecnológicos

Neste subcapítulo serão retratados os requisitos funcionais e tecnológicos da aplicação web.

3.4.1. Requisitos Funcionais

Neste subcapítulo serão retratados os requisitos funcionais da aplicação web.

Tabela 2 - Requisitos Funcionais

Requisito Funcional	Descrição
Poder pesquisar por informações através de um endereço email	Como utilizador da página web será permitido realizar pesquisas por informações OSINT através de um endereço de email

Poder pesquisar por informações através de um nome de utilizador	Como utilizador da página web será permitido realizar pesquisas por informações OSINT através de um nome de utilizador
Poder gerar combinações de nomes de utilizador através de um nome	Como utilizador da página web será permitido com base num nome gerar possíveis combinações de nomes de utilizador
Poder pesquisar por informações com base nos nomes de utilizadores gerados	Como utilizador da página web será permitido realizar pesquisar por informação OSINT com base em nomes de utilizadores gerados
Poder pesquisar em tempo real por informações	Como utilizador da página web será possível efetuar pesquisas em tempo real
Poder pesquisar por informações guardadas na base de dados	Como utilizador da página web será possível efetuar pesquisas aos dados guardados na base de dados
Poder submeter ficheiros com informações	Como utilizador da página web será possível submeter ficheiros para os dados serem guardados na base de dados
Poder pesquisar pelo histórico de um determinado nome de utilizador na data corrente	Como utilizador da página web será possível pesquisar pelo histórico de um determinado perfil através do nome de utilizador na data corrente
Poder pesquisar pelo histórico de um determinado nome de utilizador entre datas	Como utilizador da página web será possível pesquisar pelo histórico de um determinado perfil através do nome de utilizador entre 2 datas
Poder pesquisar pelo histórico de um determinado nome de utilizador antes de uma data	Como utilizador da página web será possível pesquisar pelo histórico de um determinado perfil através do nome de utilizador antes de uma data
Poder pesquisar pelo histórico de um determinado nome de utilizador depois de uma data	Como utilizador da página web será possível pesquisar pelo histórico de um determinado perfil através do nome de utilizador depois de uma data

A Tabela 2 representa todos os requisitos funcionais da aplicação web.

3.4.2. Requisitos tecnológicos

Neste subcapítulo serão retratados os requisitos tecnológicos da aplicação web.

Tabela 3 - Requisitos Tecnológicos

Requisitos tecnológicos	Descrição
Acessibilidade multiplataformas	A aplicação web deverá ser acessível em todas as plataformas quer sejam computadores quer sejam telemóveis.
Permitir consultas online e offline	A aplicação deverá permitir realizar pesquisas online e offline consoante a vontade do utilizador

A Tabela 3 representa todos os requisitos tecnológicos da aplicação web.

3.5. Metodologias, Ferramentas e linguagens utilizadas

Com vista a desenvolver um demonstrador da solução baseada na arquitetura anteriormente retratada na secção 3, iremos iniciar o trabalho avaliando as técnicas, ferramentas e tecnologias disponibilizadas para cada um dos módulos.

Neste subcapítulo será feita a escolha das diferentes metodologias, ferramentas e linguagens a serem usadas no desenvolvimento do projeto.

3.5.1. Armazenamento de dados

Para o armazenamento de dados será usado uma BD. Para tal irão ser testados 3 tipos de BD para saber qual será a melhor opção para a BD. Esta fase consistirá na escolha da BD, armazenar dados na mesma e por fim a fase de teste. Após isso, então será escolhido a BD que melhor servirá para o desenvolvimento do projeto. Serão testadas as bases de dados Structured Quey Language (SQL) ou BD relacionais, as bases de dados NoSQL ou BD não relacionais e as bases de dados transacionais ou gráficas.

- **Base de dados relacionais**

Uma BD relacional (Oracle, n.d.)é um tipo de BD que armazena e fornece acesso a pontos de dados relacionados entre si. Este possui a vantagem de os dados poderem relacionar-se entre si, o que será útil por causa das relações entre os vários dados que serão recolhidos de fontes abertas. Como exemplo de uma BD relacional existe o PostgreSQL. O PostgreSQL (PostgreSQL, n.d.) (Milani, 2008), é uma BD relacional usada para armazenar dados e administrar o acesso às informações. Esta tem a vantagem das bases de dados relacionais, permite que os dados tenham relações entre si.

- **Base de dados não relacionais**

BD não relacionais (Nayak, Poriya, & Poojary, 2013), ou NoSQL, como também são conhecidas, são BD criados para modelos de dados específicos. Estas BD são mais escaláveis, na medida em que estas podem crescer com muita facilidade. Estes também têm a vantagem de serem mais rápidas. Como exemplo deste tipo de BD existe o MongoDB. O MongoDB (MongoDB, n.d.) (Nayak, Poriya, & Poojary, 2013), possui um alto desempenho

e eficiência. Este também possui grande escalabilidade e flexibilidade, tendo em conta que este tem espaço para crescer e é flexível pois permite armazenar qualquer tipo de dado.

- **Base de dados gráficas**

As bases de dados gráficas (Robinson, Webber, & Eifren, 2013), são caracterizadas por permitirem realizar operações Create Read Update Delete (CRUD) e que expõem os dados num modelo gráfico. Esta BD possui entidades, representadas por nós, ligados por uma relação. O modelo final é uma estrutura simples composta por nós relacionados entre si, muito mais simples que as BD relacionais e as não relacionais. Como exemplo deste tipo de BD temos o neo4j. O neo4j (Bruggen, 2014) permite criar entidades, representando cada entidade como sendo um nó do gráfico. Também permite criar relações entre os nós, podendo dar o nome desejado a relação. A BD também permite as entidades e as relações possuírem atributos. Esta BD também é escalável pelo que permite adicionar uma nova entidade sempre.

3.5.2. Dados guardados

Nesta secção serão abordados alguns dados que poderão vir a ser recolhidos de fontes abertas e posteriormente guardados.

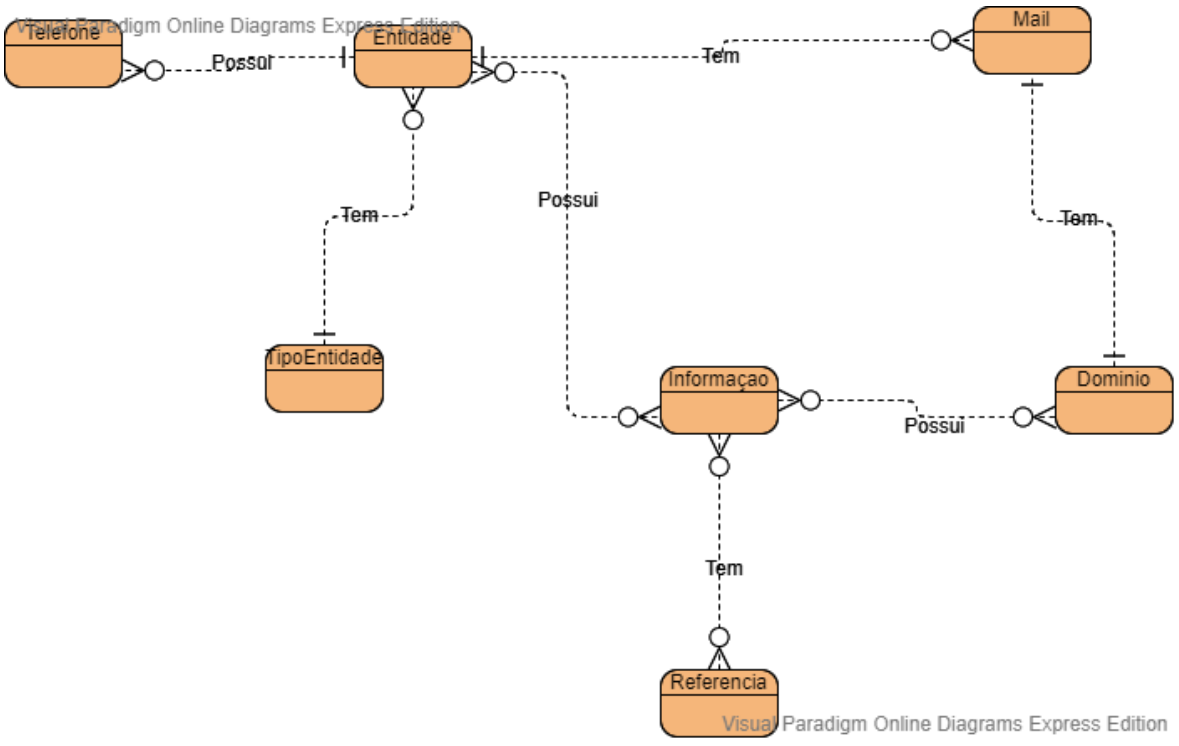


Figura 6 - Estrutura dos dados

A Figura 6 representa uma possível estrutura dos dados que serão extraídos de fontes abertas e guardadas na BD. O telefone consiste nos diferentes números de telefones que poderão ser encontrados. O tipo da entidade, como o nome indica, corresponde ao tipo da entidade, como por exemplo, pessoas, empresas associações entre outros tipos que poderão surgir. As entidades poderão assumir vários tipos como já dito anteriormente. Os Mail correspondem

aos diferentes endereços de email. Os domínios representam os domínios no qual serão extraídas informações. É de realçar que as informações extraídas deverão sempre ser de fontes abertas. As informações consistirão nas informações encontradas das diferentes fontes. As referências serão as referências de onde foram encontradas as informações. É de realçar que as diferentes tabelas estão relacionadas entre si.

3.5.3. Teste as bases de dados

Neste subcapítulo serão realizados testes às diferentes bases de dados com o propósito de saber qual será utilizada no desenvolvimento do projeto. Serão testadas as seguintes bases de dados: mySQL, MongoDB e neo4j.

- **MongoDB**

O Primeiro teste realizado foi ao MongoDB. Este como já tinha sido referido anteriormente, é uma BD não relacional. Esta BD funciona com base em coleções invés de tabelas, como as clássicas bases de dados SQL. Para o teste foram inseridos vários dados acerca de pessoas. A BD conseguiu suportar a inserção de vários dados, o que é bom. Mas como sendo OSINT e os dados possivelmente estarão relacionados entre si este tipo de BD não será o ideal para o projeto, visto que, não são permitidas relações entre as tabelas. Em anexo segue o código usado para realização dos testes e os dados inseridos BD Anexo K – MongoDB.

- **SQL**

O segundo teste realizado foi a uma BD SQL. Este tipo é uma BD relacional, isto é, permite criar relações entre as diferentes tabelas que possam existir. Para o teste foram inseridos vários dados acerca de pessoas. A BD, assim como anteriormente, conseguiu suportar a inserção de vários dados. Este tipo de BD tem uma "desvantagem", que sempre que pretende-se alterar a estrutura de uma tabela, esta terá de ser adicionado o novo atributo. Neste tipo de BD também é necessário criar as tabelas. Em anexo segue o código usado para realização dos testes e os dados inseridos na BD Anexo L - SQL.

- **Neo4j**

O último teste realizado foi a uma BD gráfica. Para o teste foram inseridos vários dados acerca de pessoas. Esta BD foi capaz de lidar com a inserção de grande quantidade de dados. Diferente do teste realizado anteriormente, este tipo de BD altera-se automaticamente, sempre que pretende alterar a estrutura de uma tabela. Em anexo segue o código usado para realização dos testes e os dados inseridos na BD Anexo M – Neo4j

3.5.4. Quadro Comparativo entre os diferentes tipos de base de dados

Nesta secção será feita um quadro comparativo entre as diferentes bases de dados retratadas acima. Na Tabela 4 será feita a comparação tendo como base se permite relações, se é de fácil alteração, o tempo para inserir dados, o tempo para consultar dados entre outras medidas. Para a comparação foram usados aproximadamente 2600 dados.

Tabela 4 - Quadro comparativo das diferentes bases de dados

BD	Tipo	Relações diretas	Fácil alteração	Tempo Inserção (ms)	Tempo Consulta (ms)	Armazenamento de dados
Neo4j	BD gráfica	Permite	sim	646	56	Nós
SQL	BD SQL	Permite	não	283	88	Tabelas
mongoDB	BD no-SQL	permite	sim	422	55	Coleções

3.6. Ferramentas escolhidas

Após a fase de testar as diferentes bases de dados resolveu-se escolher a BD neo4j para o desenvolvimento do projeto. Apesar da BD SQL demostrar ser mais rápida na inserção de dados foi escolhido a BD neo4j devido a facilidade de alteração da estrutura de dados e a facilidade de criar relações. Também a BD neo4j permite suportar a inserção de vários volumes de dados e é de mais fácil uso. Também será escolhida a framework Laravel, Vue.js e Node.js para o desenvolvimento da página web, bem como as a linguagens de programação PHP Hypertext Preprocessor (PHP) e JavaScript (JS).

4. Desenvolvimento

Neste capítulo será abordado o desenvolvimento do demonstrador e da página web de suporte, bem como todas as funcionalidades requeridas para obtenção de informação nas diversas fontes.

4.1. Funcionamento

A página web permitira ao utilizador pesquisar por informações de diferentes fontes abertas agrupando-as e mostrando-as ao utilizador tudo de uma maneira sumarizada. O website também permitirá ao utilizador pesquisar em tempo real ou então pesquisar somente com os dados guardados na BD. A página web funcionara por módulos.

A página web funciona no modelo cliente-servidor, em que o cliente é responsável por obter as informações introduzidas pelo utilizador e o servidor responsável pela busca das informações em fontes abertas, guardando os resultados numa BD e de seguida mandando as mesmas informações para o cliente para serem mostradas ao utilizador. O cliente foi desenvolvido com base na framework Laravel, com o auxílio da framework Vue.js e o servidor utilizou-se o Node.js. A comunicação entre o cliente e servidor é feito através de sockets. A página web também permite realizar pesquisas em tempo real ou também pesquisas à BD consoante a vontade do utilizador.

De seguida será retratada a implementação dos diferentes módulos apresentados na arquitetura no capítulo 3.

4.1.1. Base de Dados

Como já referido anteriormente no capítulo 3.6, a BD foi escolhida foi a Neo4j, pois esta permite de facilmente relacionar os diferentes dados encontrados e também permite alterar facilmente a estrutura dos dados. Para realizar a conexão a BD foi utilizado um driver do neo4j para node.js.

```
var neo4j = require('neo4j-driver');
var driver = neo4j.driver('bolt://[redacted]',neo4j.auth.basic(apiKey["db_username"],apiKey["db_pass2"]));
```

Figura 7 - Conexão à base de dados

A Figura 7 representa a ligação ao módulo da BD. Primeiro é necessário instalar um módulo do neo4j, de seguida declarar o modo e por fim fazer a configuração de acesso à BD. Para realizar a conexão a BD é preciso o URL da BD e das respetivas credenciais para a realização da autenticação.

4.1.2. Inserção

O módulo da inserção é responsável por recolher os dados tratados pelos parsers e inseri-los na BD.

```
var session = driver.session();
session
    .run('CREATE (n:teamsOSIF {name:{nameParam}}) RETURN n',{nameParam:record})
    .then(function(result){
        console.log("team osif created");
        session.close();
    })
    .catch(function (err){
        console.log(err);
    });
```

Figura 8 – Inserção

A Figura 8 representa um exemplo de código para inserir dados na BD. Primeiro é aberta uma sessão à BD, de seguida é executada a query responsável por criar a entidade e guardá-la na BD. Após a operação é terminada a sessão.

Caso a entidade possua relações com outras entidades é efetuada a relação entre as mesmas.

```
var session = driver.session();

session
       .run('MATCH (a:userOSIF {username:$usernameParam}), (b:teamsOSIF {name:$nameParam})'+
        'MERGE (a)-[r:GOSTA]->(b) RETURN a,b',{usernameParam:userFace.username,nameParam:record})
       .then(function(result){
           console.log("Relationship Created  between user OSIF and  teams");
           session.close();

       })
       .catch(function (err){
           console.log(err);
       });
```

Figura 9 - Criação de relações entre as entidades

A Figura 9 representa um exemplo de como é criada uma relação entre duas entidades. Neste caso é criado uma relação entre um utilizador e um desporto, praticado por este.

4.1.3. Pesquisa

O módulo da pesquisa é responsável por receber os dados da vista introduzidas pelo utilizador e efetuar a pesquisa à BD ou a fontes externas.

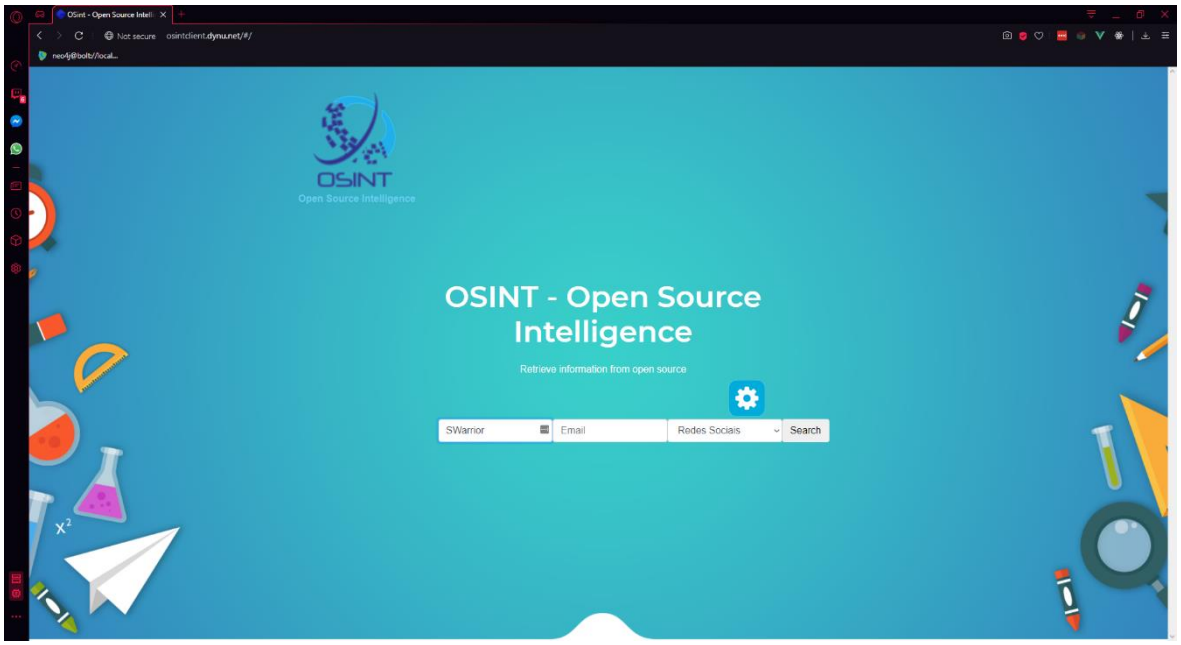


Figura 10 - Pesquisa pelo nome de utilizador

A Figura 10 representa uma pesquisa efetuada pelo nome de utilizador SWarrior. O módulo da pesquisa recebe o nome de utilizador introduzido pelo utilizador, e efetua o pedido respetivo, quer seja em tempo real, quer seja pela BD.

```
socket.on("search_social",data => {
    linkFound = [];
    let user = data;
    let url;
    if(data.username != ''){
        for (link in links) {
            retrieveInfo(links[link]['url'].replace('{}', user.username), link);
        }

        setTimeout(function(){
                socket.emit("retrieve_social",linkFound);
        }, 10000);
        fetchFaceData(data.username,socket);
        fetchLinkedin(data.username,socket);
        fetchUserSearch(data.username,socket);
        fetchTikTokUser(data.username,socket);
    }else if(data.email != ''){
        //
        fetchEmailEmailRep(data.email,socket);
        fetchEmail(data.email,socket);
        findSocialEmail(data.email,socket);
        fetchLinkedin(data.email,socket);
    }


});
```

Figura 11 - Pesquisa em tempo real

A Figura 11 representa uma pesquisa realizada pelo utilizador em tempo real. Em que é recebido um nome de utilizador e de seguida efetuado uma busca às diferentes fontes abertas.

Ao nível de funcionalidades disponibiliza pesquisas online e offline, sendo também possível inserção de conteúdo baseado em ficheiros.

- **Online**

Baseado na tecnologia node, foi desenvolvido um módulo que permite consumir API's externas e possibilita a busca de conteúdo de páginas especificas. Ao introduzir um nome de utilizador, email ou nome esse mesmo dado é enviado para o servidor que por sua vez consoante o dado recebido trata a informação. No caso de ter recebido um endereço de email o mesmo é validado para saber se é um endereço de email válido, depois são realizadas buscas em tempo real às diferentes fontes abertas de obtenção de informação. Após a recolha da informação, a mesma é filtrada pelos parsers que tratarão do processamento da respetiva informação. De seguida os dados são guardados na BD e o servidor envia os dados ao cliente para poderem ser mostrados ao utilizador.

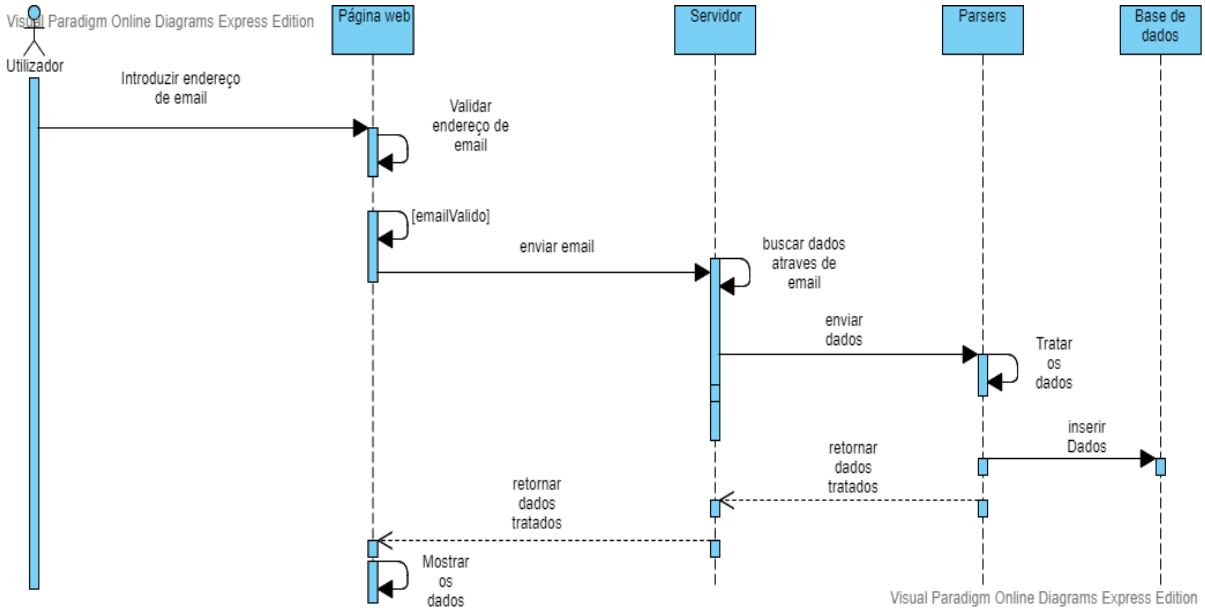


Figura 12 - Diagrama de sequência email

A Figura 12 representa o diagrama de sequência quando introduzido um endereço de email.

No caso de ter sido recebido um nome, são gerados possíveis nomes de utilizadores com base no nome recebido. De seguida os nomes gerados são enviados para o cliente, no qual o utilizador pode escolher quais combinações de nomes de utilizador pretende pesquisar. Após a escolha do utilizador, é pesquisado nas diferentes fontes consoante a escolha realizada pelo utilizador.

A Figura 13 representa o diagrama de sequência quando introduzido um nome.

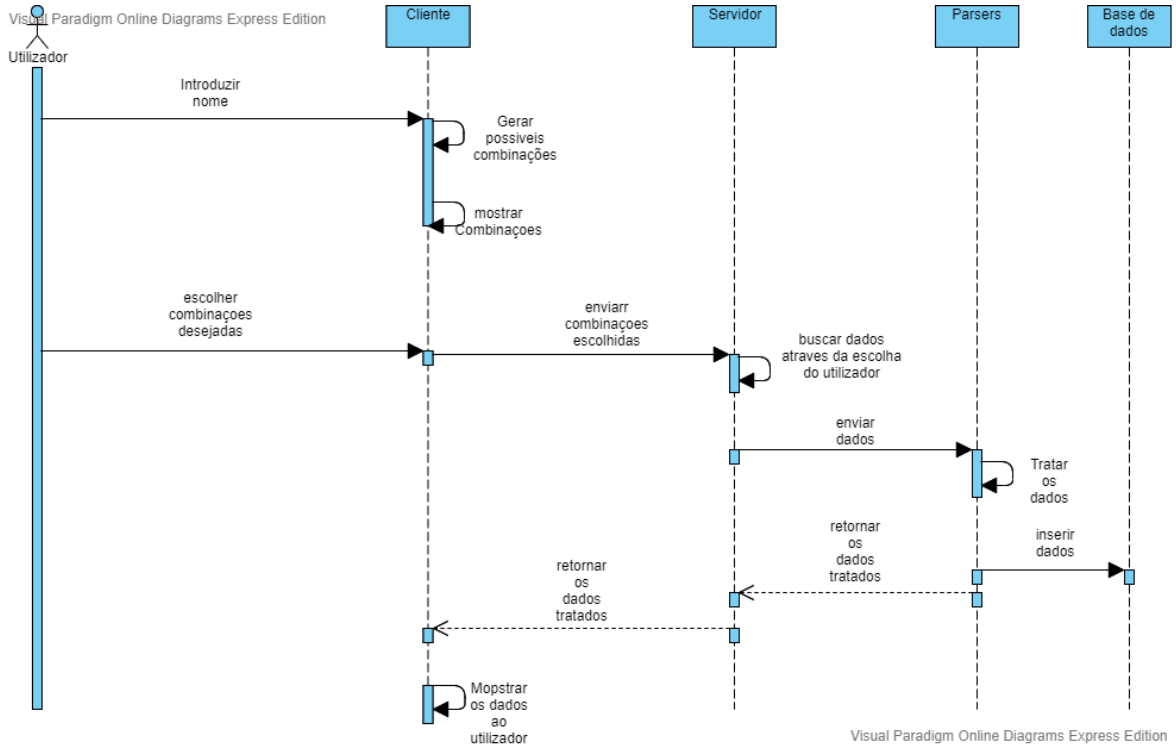


Figura 13 - Diagrama de sequência nome

No caso de ter recebido um nome de utilizador, este é enviado para o servidor que fará a busca da informação. Seguidamente os parsers tratarão dos dados, filtrando apenas as informações relevantes. Após o filtro da informação, a mesma é guardada numa BD e enviada para a página web para que possa ser mostrada ao utilizador.

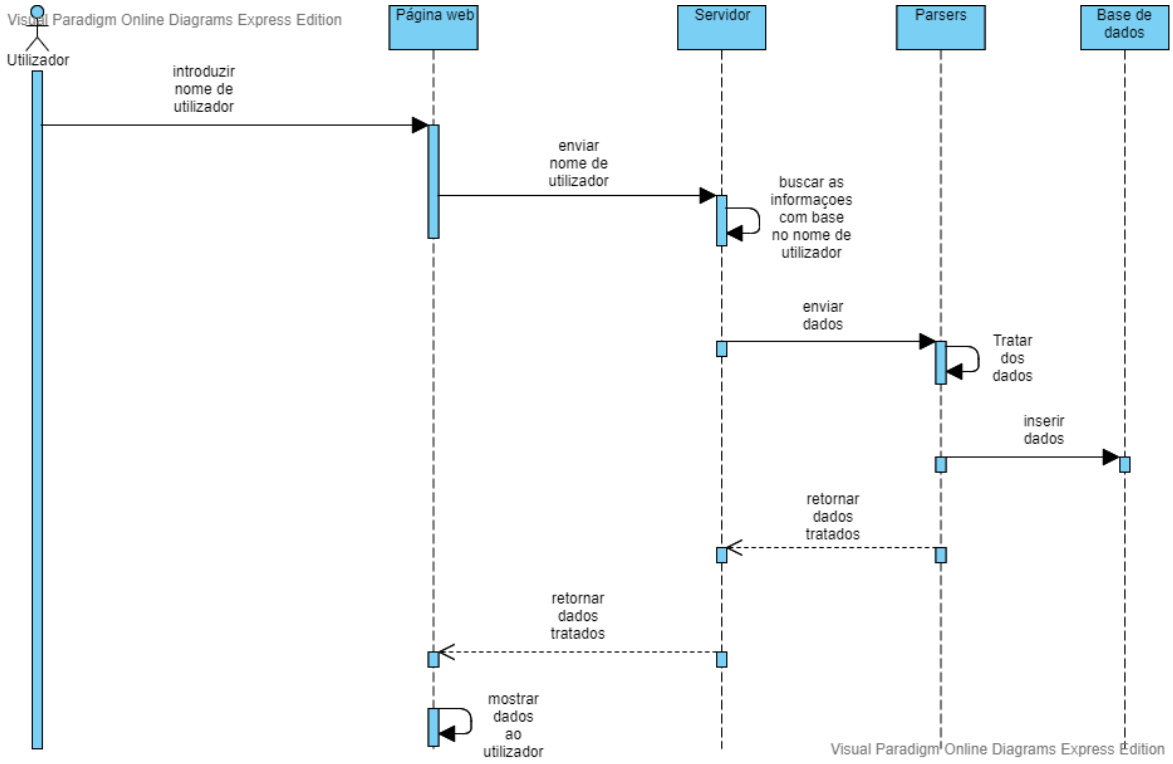


Figura 14 - Diagrama de sequência nome de utilizador

A Figura 14 representa o diagrama de sequência para quando introduzido um nome de utilizador.

- **Offline**

Uma das opções que o utilizador poderá escolher, ao contrário da pesquisa em tempo real, é a pesquisa apenas na informação guardada na BD.

Assim como na pesquisa online ou em tempo real, o utilizador poderá pesquisar por nome, nome de utilizador ou então por endereço de email.

Na pesquisa por nome de utilizador, este é enviado ao servidor que por sua vez procura na BD, todos os dados relacionados com o nome de utilizador fornecido. Seguidamente todos os dados encontrados são enviados para a página web para poderem ser mostrados ao utilizador.

Na pesquisa por email, assim como na pesquisa por nome de utilizador é enviado o email ao servidor que realiza todas as buscas pelo endereço de email. Seguidamente todos os dados são enviados para a página web para serem mostrados ao utilizador.

Na pesquisa por nome, distinto da pesquisa em tempo real, não são gerados nomes de utilizadores que posteriormente são mostrados ao utilizador para ser realizada a escolha. Neste caso quando fornecido o nome, o mesmo é enviado ao servidor, e este é responsável por encontrar todos os dados guardados na BD com o nome fornecido. Seguidamente os

dados encontrados serão enviados para a página web, para poderem ser mostrados ao utilizador.

- **Ficheiros**

Assim como referido anteriormente a página web permite a inserção de ficheiros com dados para poderem ser guardados na BD. Para facilitar a forma que os dados estão no ficheiro, para poderem ser reconhecidos pela página web, foram criados alguns scripts em python para poderem facilitar a geração dos ficheiros.

- OSIF

  OSIF (Open Source Intelligence Facebook) foi um script adaptado de um projeto já existente no GitHub (CiKu370, s.d.). Este projeto permite a obtenção de informações na rede social Facebook. Para o correto funcionamento a ferramenta precisa de um *token* de autenticação. Possuindo o *token*, a ferramenta consegue obter informações dos perfis amigos da conta no qual pertence o *token*.

- SocialScan

  SocialScan (Foundation, s.d.) refere-se a uma biblioteca em python que permite pesquisar por informação em várias redes sociais através de email ou através de nome de utilizador. Tendo como base esta biblioteca foi então desenvolvida um script em python que permite receber um nome de utilizador, um endereço de email ou então um ficheiro contendo vários nomes de utilizador ou endereço de email. No final da execução do script é gerado um ficheiro de texto de output.

- Sherlock

  Sherlock (sherlock-project, s.d.), assim como o OSIF, este também foi um script adaptado de um projeto do GitHub. Este projeto permite pesquisar por nomes de utilizador em várias redes sociais e no fim dizer se determinado nome de utilizador está ou não presente nas várias redes sociais pesquisadas.

- **Histórico**

Outra das opções no qual o utilizador poderá optar, é pesquisar com base em um histórico. O histórico permitirá pesquisar por dados entre datas, antes de uma data e depois de uma data. Para ter acesso ao histórico o utilizador devera realizar uma pesquisa com base num nome de utilizador. Como teste do histórico foi apenas utilizado as redes sociais Instagram e GitHub.

4.1.4. Parsers

Os parsers são responsáveis por recolher os dados das diferentes fontes, ou ficheiros introduzidos pelo utilizador, processar os dados e de seguida enviar os mesmos para o módulo da inserção que guardara os dados na BD.

```
function fetchPinterestData(url,socket){
    fetch(url,{method: 'GET'})
            .then(function(response) {
                return response.text()
            })
            .then(function(html) {
                var parser = new DOMParser();

                var doc = parser.parseFromString(html, "text/html");
                var script = doc.getElementsByTagName('script');
                var title = doc.getElementsByTagName('title')[0].innerHTML;
                var user = {name:'',username:'',avatar:'',followers:'',following:'',collectionCount:'',
                            country:'',description:'',imageCount:'',location:'',collection:[]};
                 for(let i = 0; i < script.length; i++){
                     if (script[i].getAttribute('id') == 'initial-state') {
                        var jsonObject = JSON.parse(script[i].innerHTML);
                        var content = jsonObject.resourceResponses;
                        var userPinterest = content[0].response.data.user;
                        user.name = userPinterest.full_name;
                        user.username = userPinterest.username;
                        user.avatar = userPinterest.image_xlarge_url;
                        user.followers = userPinterest.follower_count;
                        user.following = userPinterest.following_count;
                        user.collectionCount = userPinterest.board_count;
                        user.country = userPinterest.country;
                        user.description = userPinterest.about;
                        user.imageCount = userPinterest.pin_count;
                        user.location = userPinterest.location;

                        var urls = content[1];
                        var allCollection = urls.response.data;
                        for(var j in allCollection){
                            var collection = {name:'',owner:'',imageCount:'',url:'',images:[]};
                            collection.name = allCollection[j].name;
                            collection.owner = allCollection[j].owner.username;
                            collection.imageCount = allCollection[j].pin_count;
                            collection.url = 'https://www.pinterest.pt'+allCollection[j].url;
                            user.collection.push(collection);
                        }
                     }
                 }
                fetchPinterestImages(user,socket);
            })
            .catch(function(err) {
                console.log('Failed to fetch page: ', err);
            });
}
```

Figura 15 - Pesquisar dados a uma fonte aberta

A Figura 15 representa um exemplo de código que procura informações a uma fonte aberta, trata os dados que posteriormente será enviado para a BD para ser armazenado.

.

4.1.5. Página Web

A página web permitirá mostrar aos utilizadores as informações pesquisadas. Também permite escolher as configurações pretendidas, definir as chaves das API's e também permite fazer o upload de ficheiros.

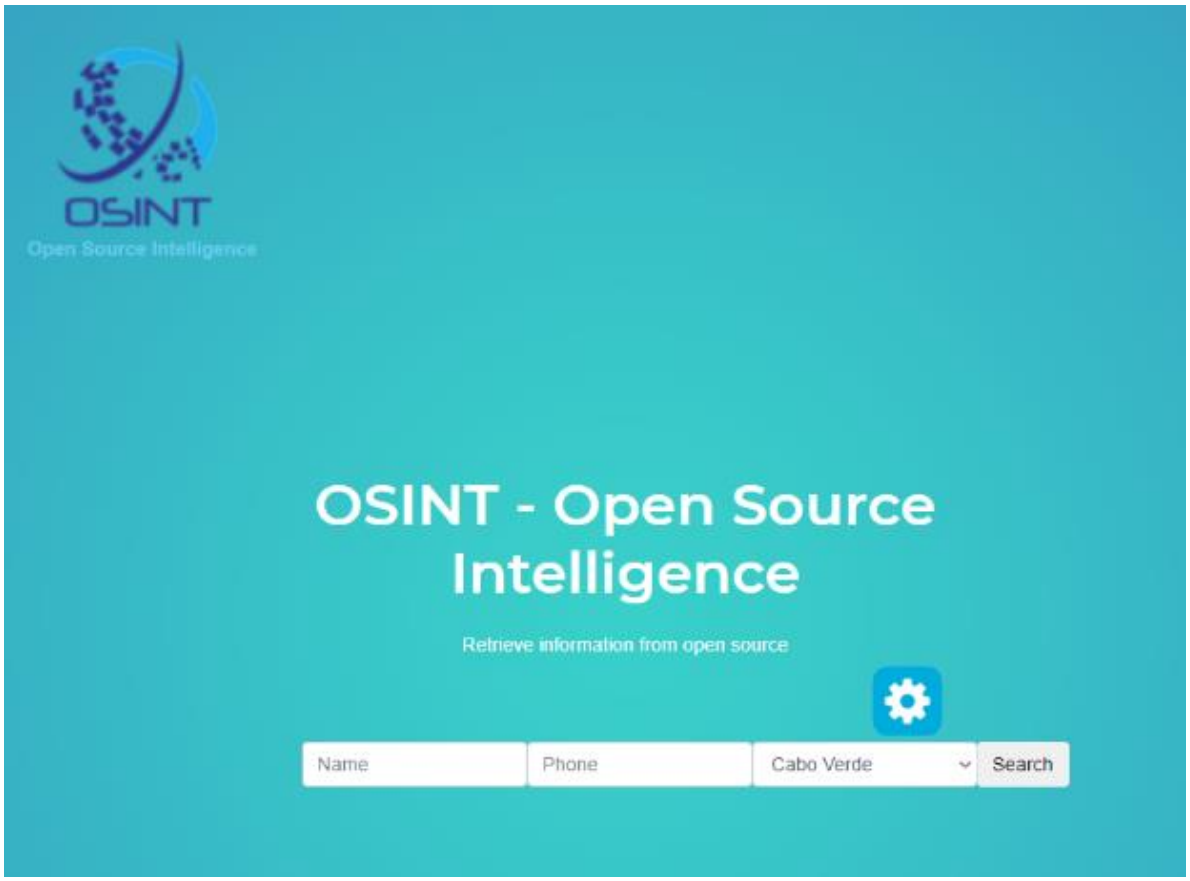


Figura 16 - Página inicial

A Figura 16 representa a página inicial da página web. Como já foi já referido anteriormente é possível definir o módulo pretendido para a realização da pesquisa e de seguida realizar a pesquisa pretendida.

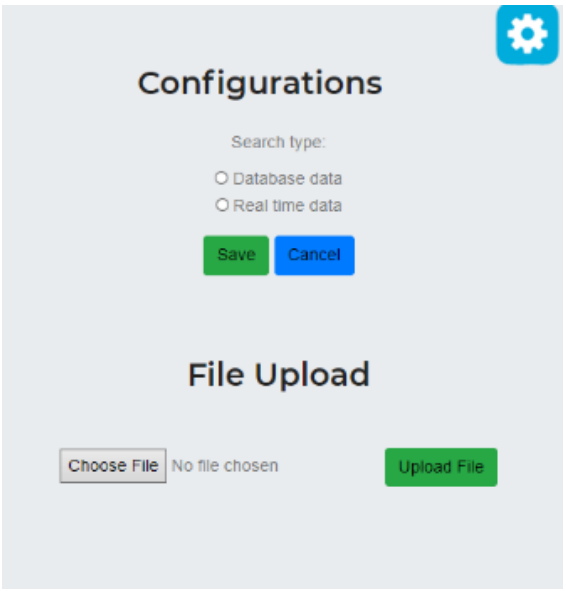


Figura 17 - Configurações da página web

A Figura 17 representa a parte de configurações da página web. É possível fazer upload de ficheiros, definir onde será realizada a pesquisa, isto é, na BD ou em tempo real e também permite definir as chaves das API's.

4.2. Módulos de Utilização da página web

A página Web funciona com base em módulos de pesquisa, isto é, escolhe em que âmbito o utilizador pretende pesquisar as informações pretendidas. Existem dois módulos, o módulo de Cabo Verde e o módulo de redes sociais.

4.2.1. Módulo Cabo Verde

O primeiro módulo corresponde ao módulo Cabo Verde. Neste módulo são consumidos fontes abertas cabo-verdianas, guardada a informação da mesma numa BD e mostrada ao utilizador de forma sumarizada a informação.

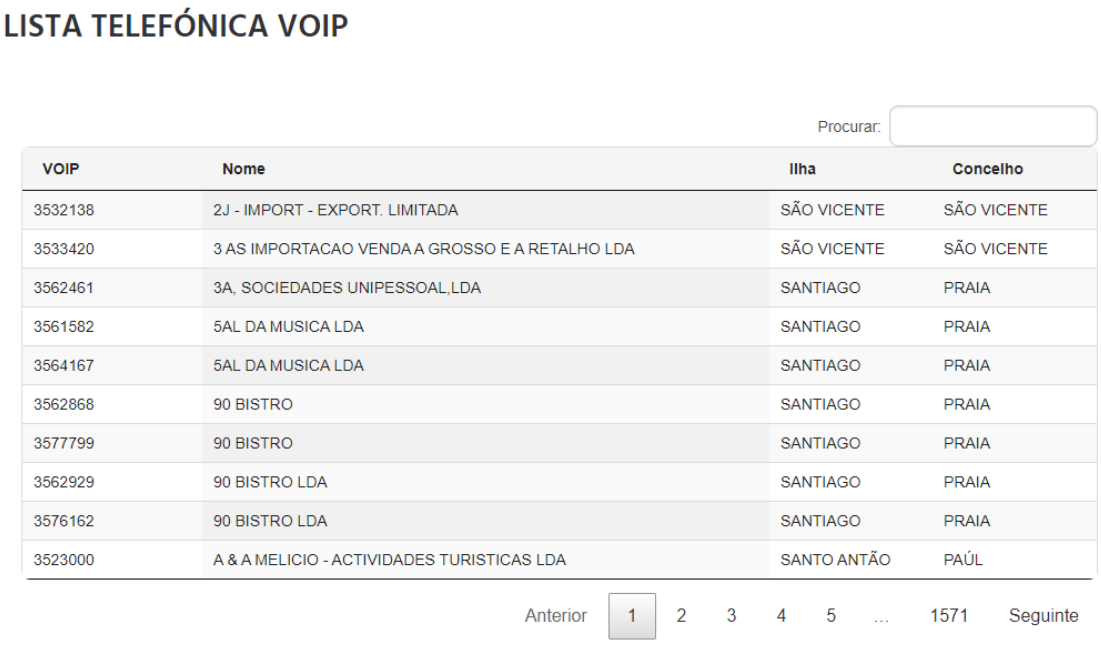
4.2.2. Módulo Redes Sociais

O segundo módulo corresponde ao módulo de redes sociais. Este módulo consume informações de várias redes sociais, trata os dados e depois armazena-os numa BD, podendo depois mostrar ao utilizador os dados de forma sumarizada.

4.3. Fontes utilizadas

Neste subcapítulo serão retratadas as várias fontes utilizadas para o desenvolvimento do projeto.

4.3.1. Módulo Cabo Verde – Fontes Cabo-Verdianas



LISTA TELEFÓNICA VOIP

Procurar:

VOIP	Nome	Ilha	Concelho
3532138	2J - IMPORT - EXPORT. LIMITADA	SÃO VICENTE	SÃO VICENTE
3533420	3 AS IMPORTACAO VENDA A GROSSO E A RETALHO LDA	SÃO VICENTE	SÃO VICENTE
3562461	3A, SOCIEDADES UNIPESSOAL,LDA	SANTIAGO	PRAIA
3561582	5AL DA MUSICA LDA	SANTIAGO	PRAIA
3564167	5AL DA MUSICA LDA	SANTIAGO	PRAIA
3562868	90 BISTRO	SANTIAGO	PRAIA
3577799	90 BISTRO	SANTIAGO	PRAIA
3562929	90 BISTRO LDA	SANTIAGO	PRAIA
3576162	90 BISTRO LDA	SANTIAGO	PRAIA
3523000	A & A MELICIO - ACTIVIDADES TURISTICAS LDA	SANTO ANTÃO	PAÚL

Anterior 1 2 3 4 5 … 1571 Seguinte

Figura 18 - Fonte Cvmultimédia

A Figura 18 representa a fonte aberta da Cvmultimédia (Zap, s.d.), esta uma página web que permite consultar números de telefones *Voice Over Internet Protocol* (VoIP). A página permite saber o número de telefone VoIP da pessoa, ou empresa, a ilha em que a empresa/pessoa se situa, bem como o concelho da mesma. Ao analisar a pagina foi possível descobrir que a página efetua um pedido a um link no qual devolve em formato JSON os dados. Então foi possível também obter o ficheiro JSON na própria página web e consumir a mesma.

4.3.2. Módulo Redes Sociais

Neste subcapítulo serão retratadas as diferentes fontes utilizadas para o desenvolvimento do módulo de pesquisa em redes sociais.

O Instagram (Instagram, s.d.) foi uma das várias fontes utilizadas. Este uma rede social que permite seguir outros pessoas e compartilhar fotos e vídeos entre os diferentes utilizadores.

A Figura 19 representa a rede social Instagram. Para obter as informações foi feito um pedido *get* e de seguida foi utilizado parsers para obter apenas as informações relevantes.

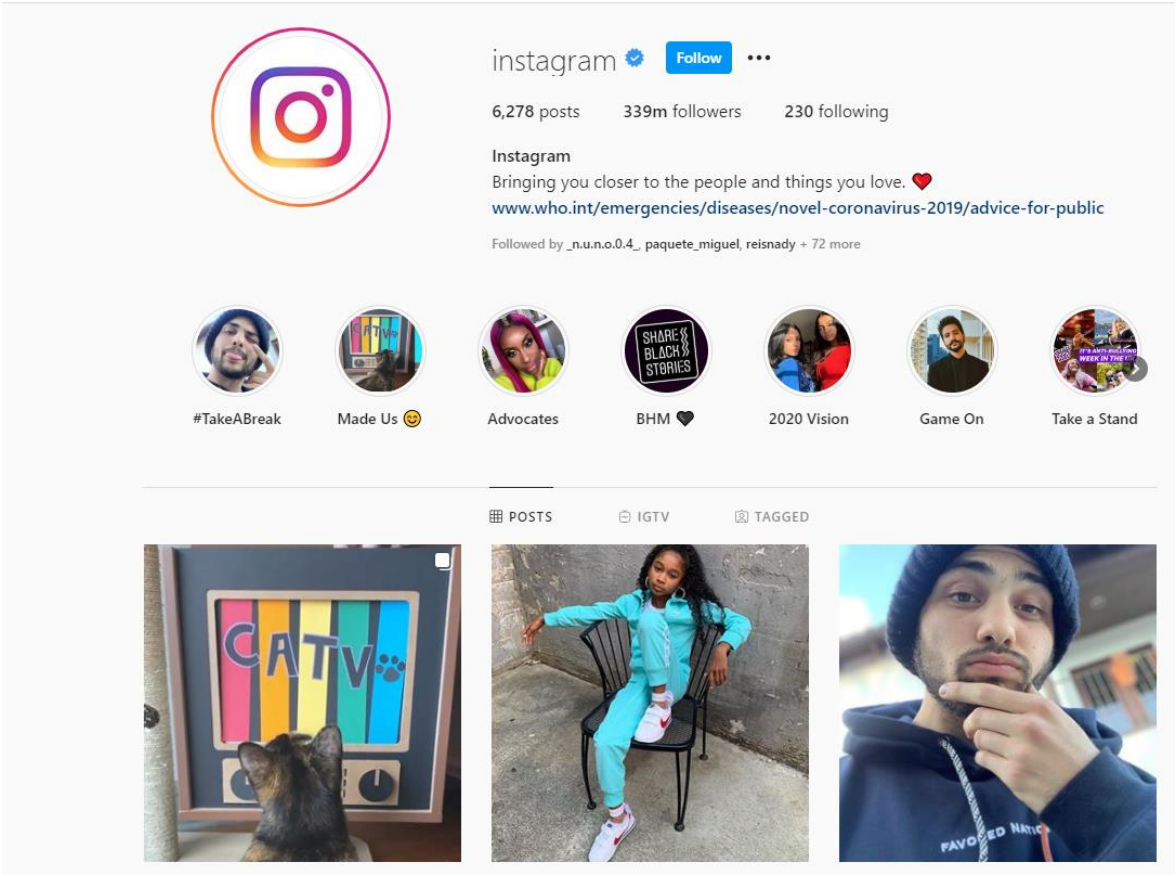


Figura 19 - Instagram

O GitHub (GitHub, s.d.) foi outra fonte utilizada. Este uma plataforma que permite armazenar código desenvolvidos.

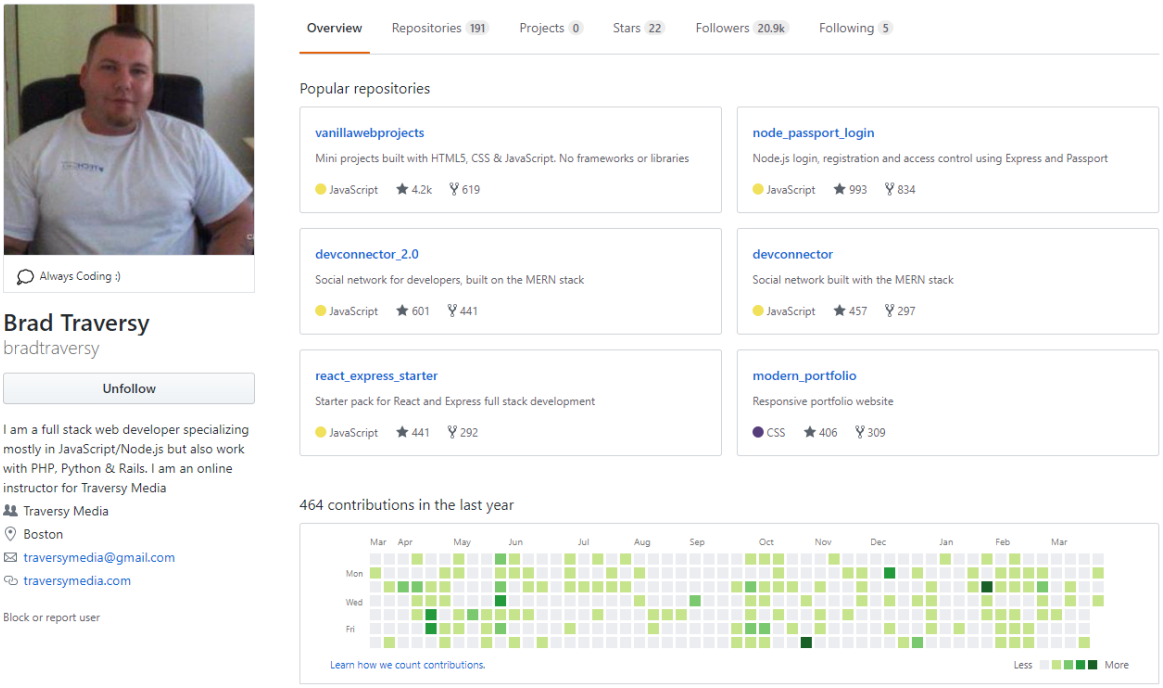


Figura 20 – GitHub

A Figura 20 representa a rede social GitHub. Para a obtenção de informação do GitHub, assim como no Instagram, foi feito um pedido *get* e de seguida usado parsers para obter apenas as informações relevantes. A rede social GitHub também possui uma API, mas, no entanto, foi usado um pedido *get* para reduzir o número de chaves de API's necessárias.

Foi também usada a API da página web EmailRep (EmailRep, s.d.). Esta página permite saber várias informações acerca de um endereço de email, tais como, se um endereço email é suspeito ou não e também os diferentes perfis associados a esse endereço de email.

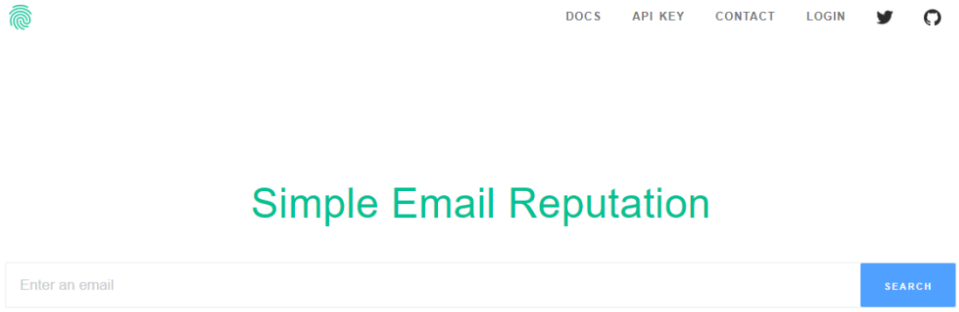


Figura 21 – EmailRep

A Figura 21 representa a página inicial da página web EmailRep. Para a utilização desta API, foi necessário pedir uma chave da API. De seguida foi usado os parsers para obtenção das informações devolvidas pela API.

O Twitter (Twitter, s.d.), assim comos os outros, é uma rede social no qual permite seguir pessoas, fazer publicações entre outras funcionalidades.

Figura 22 – Twitter

A Figura 22 representa a rede social Twitter. Para a obtenção das informações, foi realizado o mesmo processo que foi utilizado para a rede social Instagram e o GitHub. De realçar que o Twitter também possui uma API, mas, pelo mesmo motivo que o GitHub foi preferível realizar um pedido *get*, para diminuir o número de chaves API necessárias.

A página web Ask.fm (Ask.fm, s.d.) é uma rede social que permite aos utilizadores enviar perguntas uns aos outros.

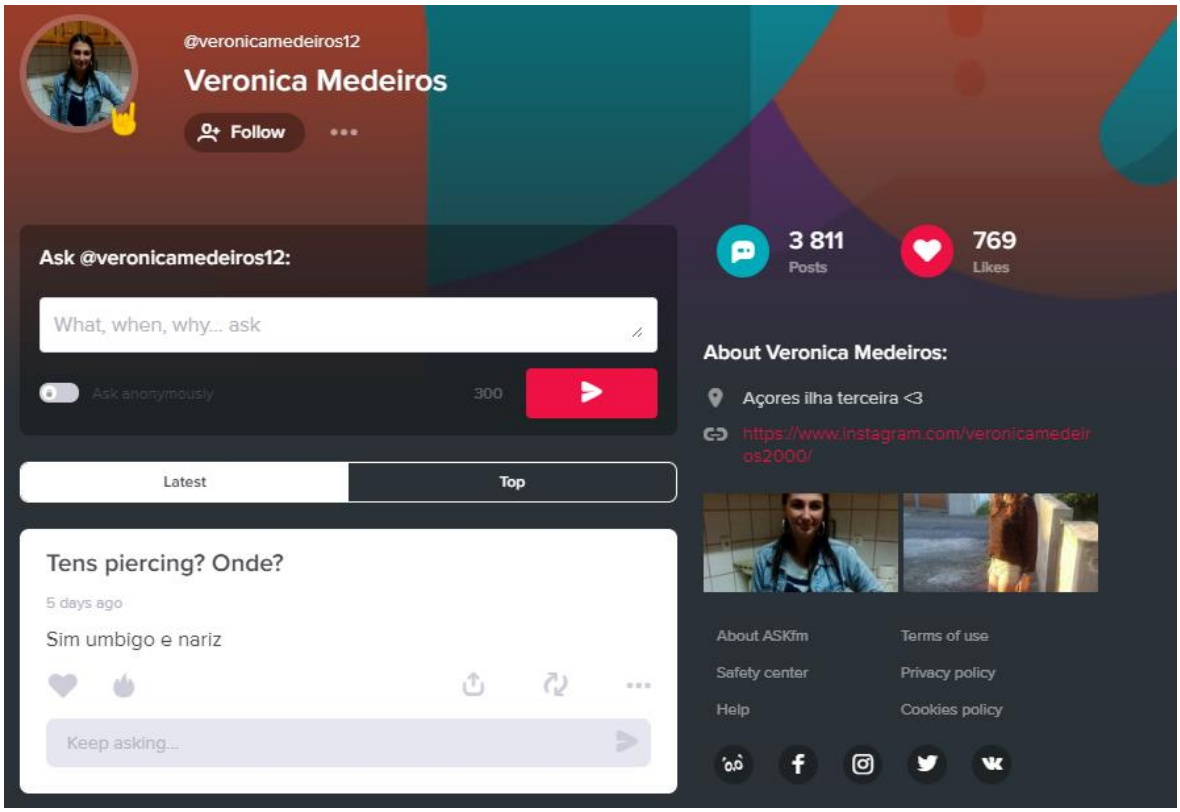


Figura 23 - Ask.fm

A Figura 23 representa a rede social Ask.fm. Para a obtenção de dados, foi utilizado o mesmo processo realizada na obtenção de dados da rede social Instagram.

Foi usada a API da empresa FullContact (FullContact, s.d.). A API permite com base no endereço de email ou outros dados pesquisar por perfis associados.

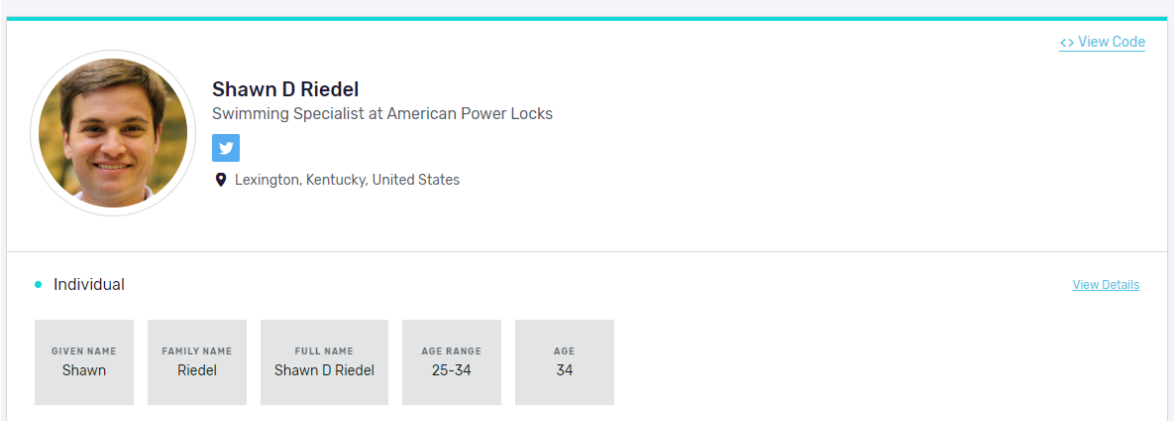


Figura 24 – FullContact

A Figura 24 representa um exemplo de utilização da API da página web FullContact. Para a utilização, foi pedido uma chave da API deles, e posteriormente com os parsers tratados os dados devolvidos pela API.

O Facebook (Facebook, s.d.), outra rede social utilizada como fonte de obtenção de dados, é uma rede social que permite conectar pessoas de diferentes partes do mundo, podendo partilhar publicações e mensagens entre os mesmos.

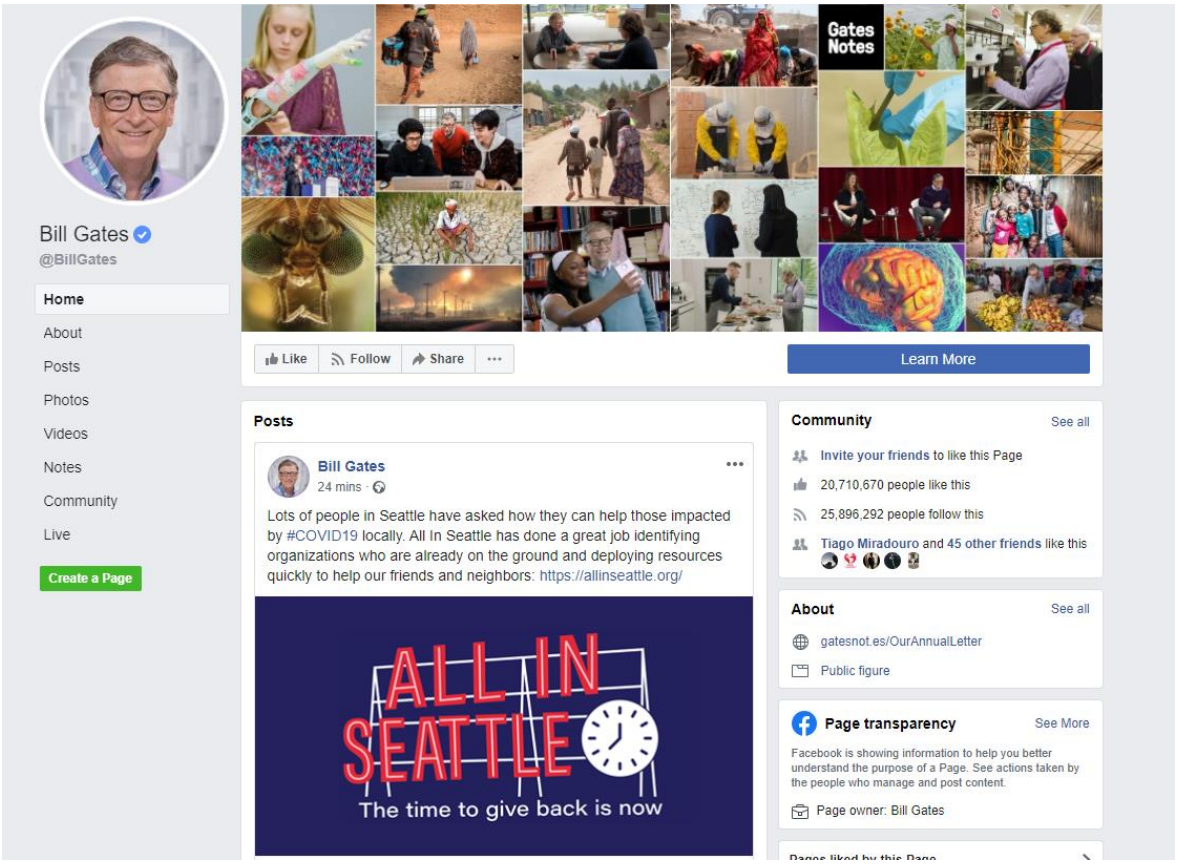


Figura 25 – Facebook

A Figura 25 representa um exemplo de um perfil da rede social Facebook. Para a extração de informação foi realizado o mesmo processo da rede social Instagram. É de realçar que o Facebook limita os pedidos a página web a pedidos sem autenticação.

O Letterboxd (Letterboxd, s.d.), esta outra rede social, difere-se das restantes por ser uma rede social para partilha de opiniões acerca de filmes.

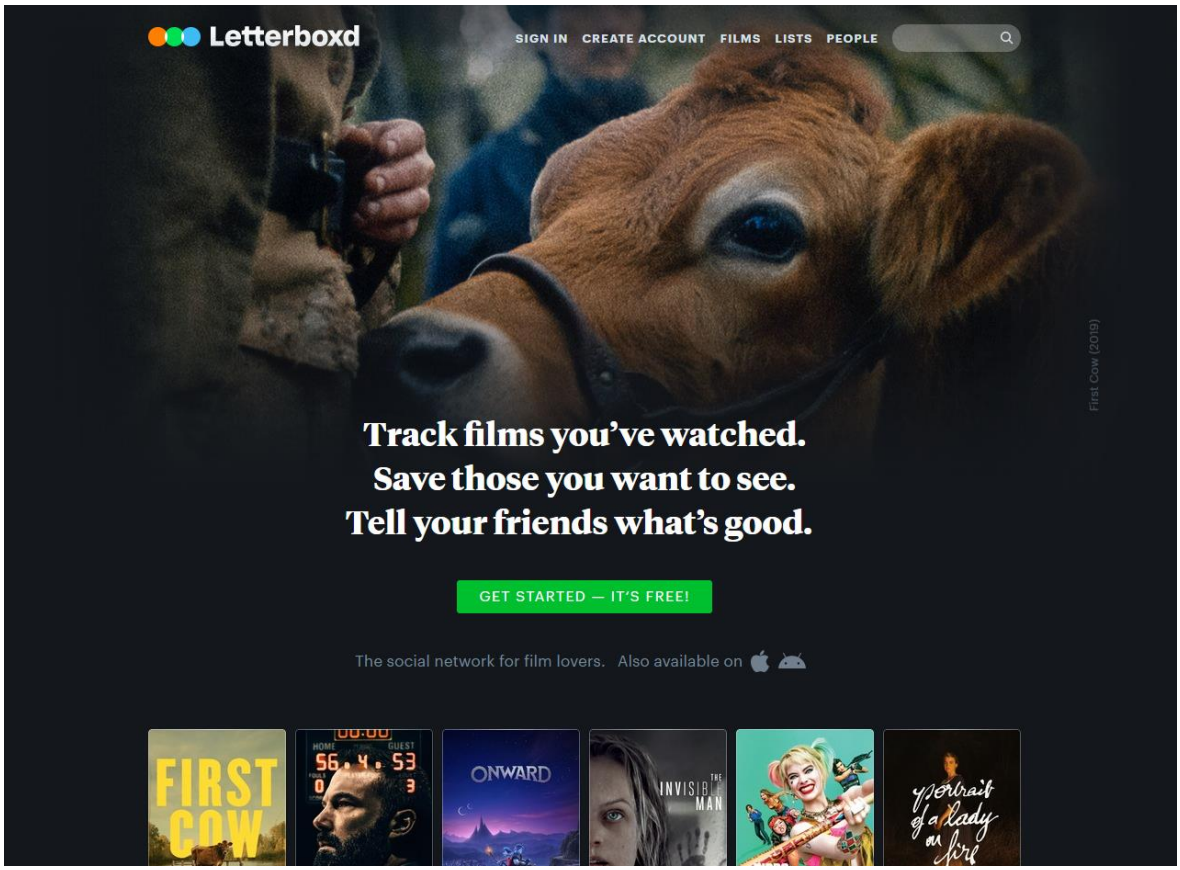


Figura 26 – Letterboxd

A Figura 26 representa a página inicial da rede social Letterboxd. Para a obtenção de informação desta rede social, assim como no Instagram foi realizado um pedido *get* e de seguida foram usados os parsers para extração das informações relevantes.

O Linkedin (Linkedin, s.d.), esta mais uma rede social, conecta pessoas, mas numa perspetiva profissional.

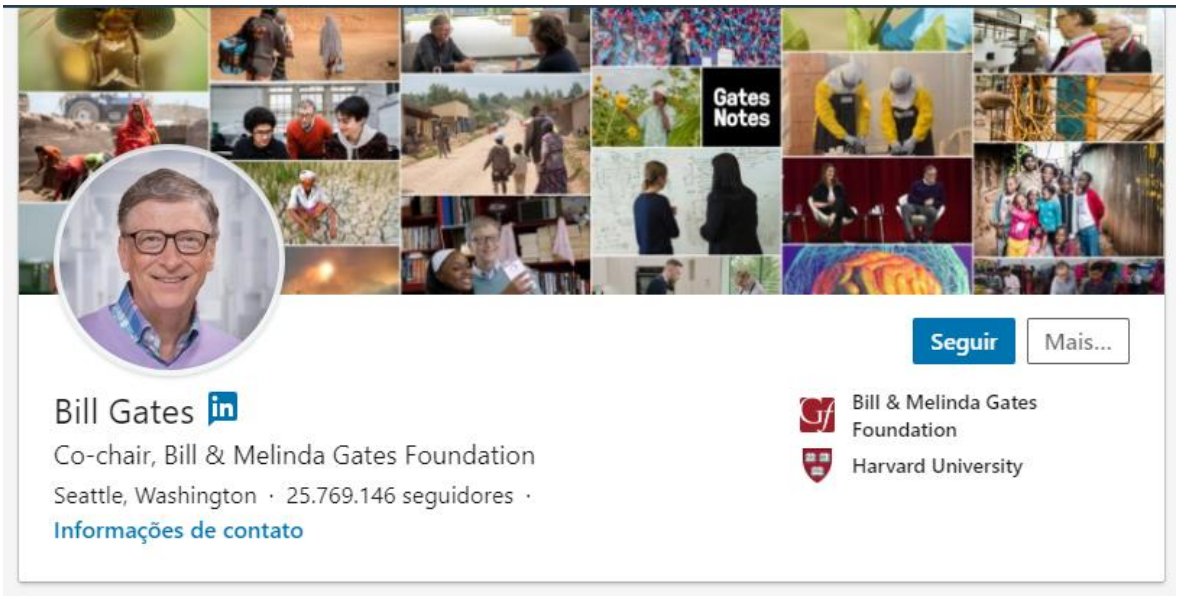


Figura 27 – Linkedin

A Figura 27 representa um exemplo de um perfil do Linkedin. Para a extração das informações da rede social, foi adaptado um projeto encontrado no GitHub (Cooya, s.d.) para a extração dos dados da rede social.

A página web UserSearch (User Search, s.d.) permite ao utilizador encontrar uma pessoa através de um endereço de email, nome de utilizador ou número de telefone.

Figura 28 – UserSearch

A Figura 28 representa a página inicial da página web UserSearch. Para a extração dos dados foi realizada um pedido *get* a página e de seguida utilizado parsers para a obtenção dos dados relevantes.

A ferramenta Sherlock (sherlock-project, s.d.), este um projeto do GitHub, que permite pesquisar por um nome de utilizador em várias redes sociais diferentes.

Figura 29 – Sherlock

A Figura 29 representa a ferramenta do GitHub Sherlock. Para a obtenção de dados desta ferramenta foi adaptado o script em python para poder ser integrado no projeto.

Foi também utilizada a biblioteca do python SocialScan, que permite pesquisar por email ou nome de utilizador em várias redes sociais diferentes, como por exemplo, o Instagram, Twitter, GitHub entre vários outros.

A Figura 30 representa um exemplo de execução da biblioteca SocialScan (Foundation, s.d.). Para a extração dos dados, foi desenvolvida um script em python que permite gerar um ficheiro com a informação e de seguida submetido o ficheiro na página web para a obtenção de resultados.

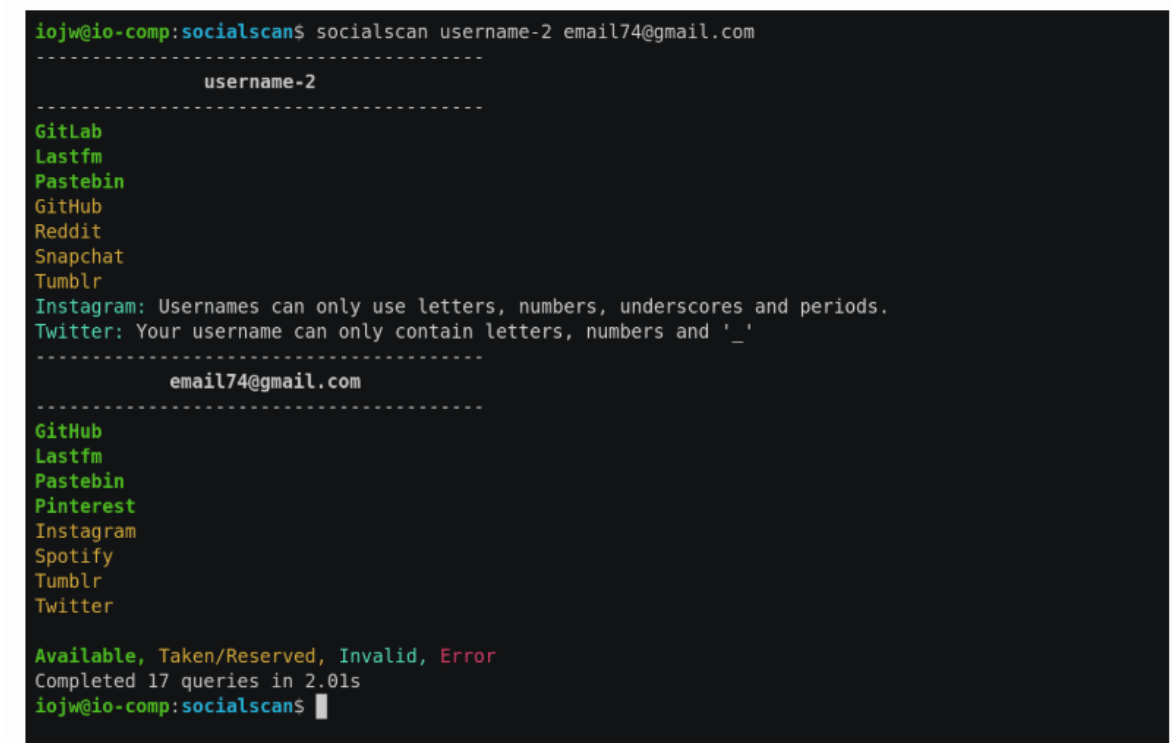


Figura 30 - Biblioteca SocialScan

Foi também utilizado um outro projeto do GitHub (CiKu370, s.d.), nomeadamente o OSIF (Open Source Intelligence Facebook). Este projeto foi adaptado e reformulado para poder ser integrado como fonte da página web.

A Figura 31 representra o script OSIF em execuçao. Para a obtenção das informações, foi adaptado o script de forma a que o mesmo fosse mais simples e de seguida podesse ser gerado um ficheiro que de seguida será submitido para pagina web. Seguidamente os dados são tratados pelos parsers que obtêm os dados relevantes.

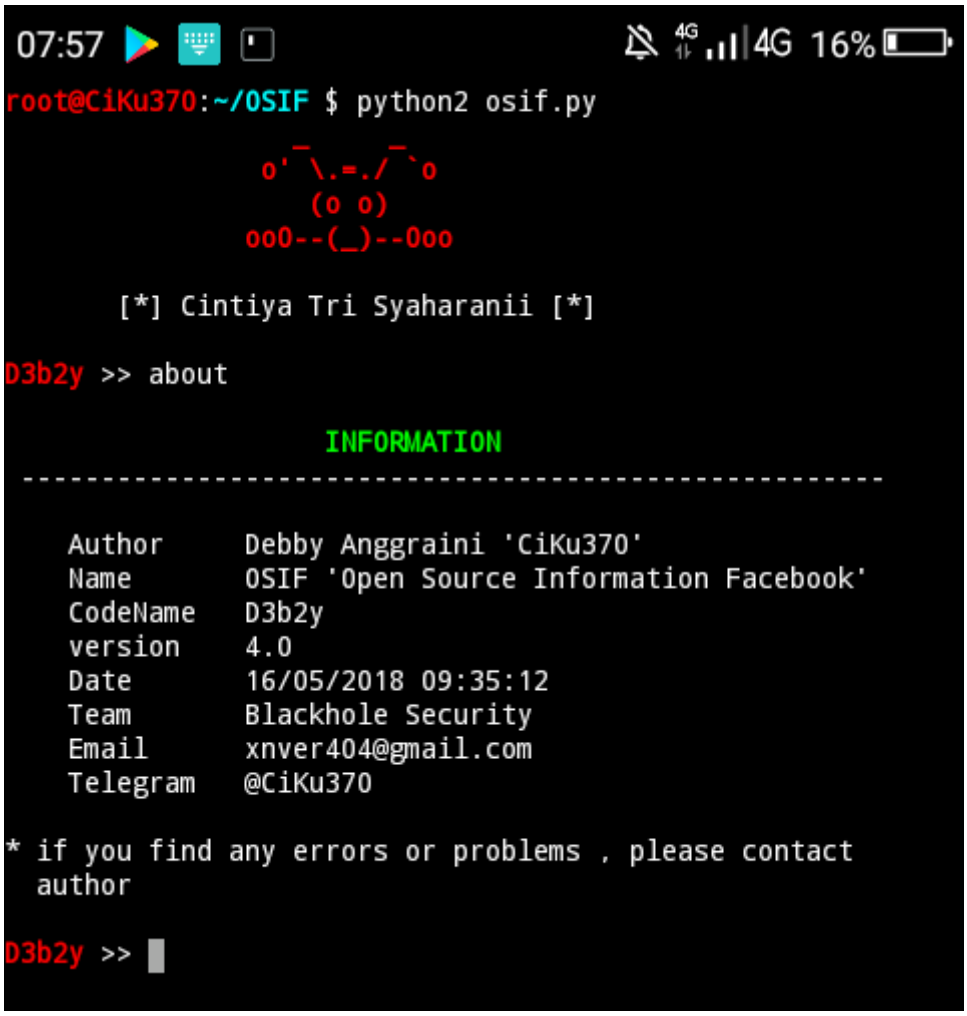


Figura 31 - OSIF

Posteriormente também foi usada a rede social TikTok (TikTok, s.d.), esta é uma rede social que permite aos utilizadores partilharem vídeos de curta duração.

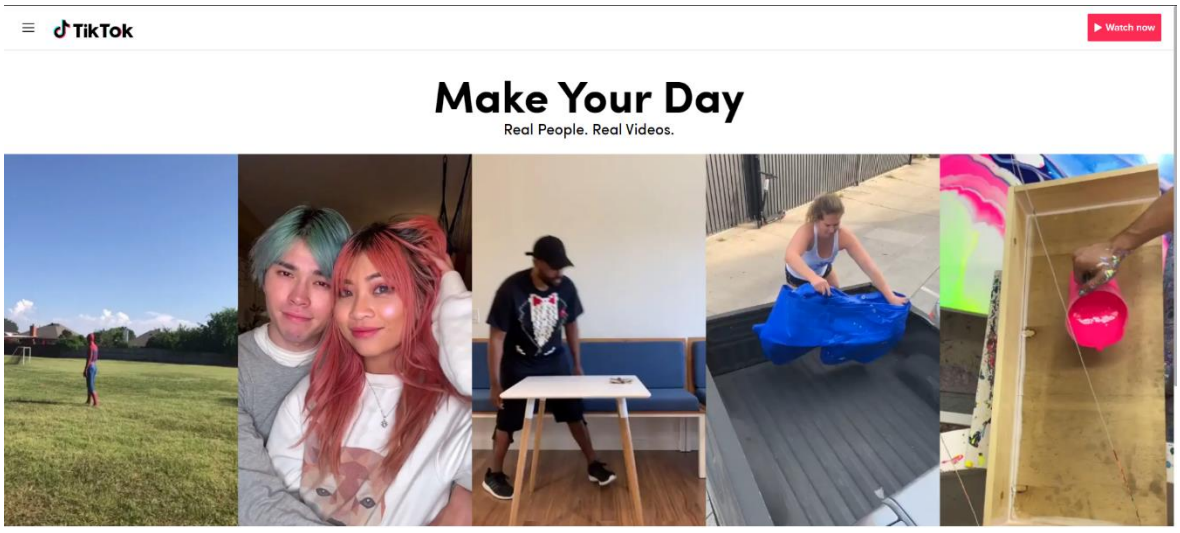


Figura 32 – TikTok

A Figura 32 representa a página inicial da rede social TikTok. Para a obtenção dos dados foi utilizado o mesmo processo que a rede social Instagram.

Parecido com o GitHub, também foi utilizado, como fonte de recolha de informação, o GitLab (GitLab, s.d.). Este assim como o GitHub permite a partilha de código.

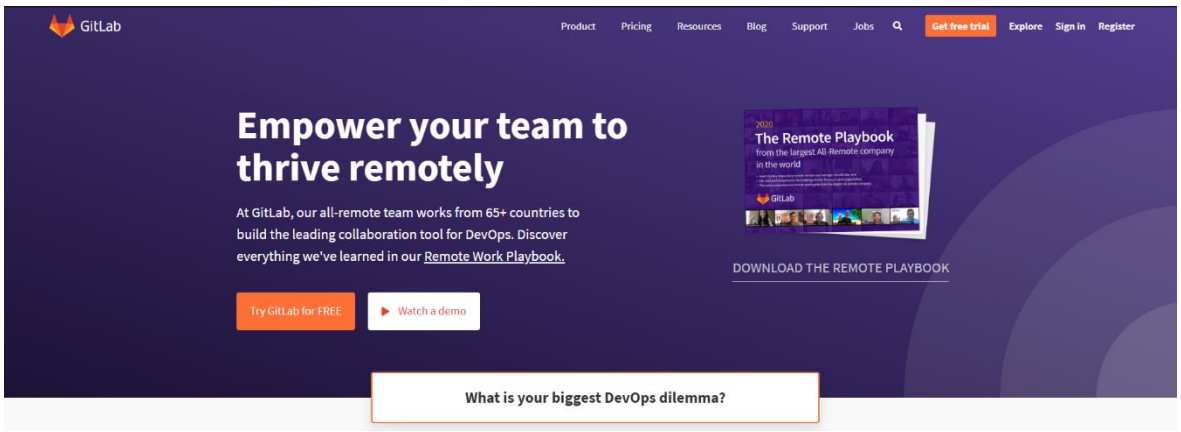


Figura 33 – GitLab

A Figura 33 representa a página inicial da rede social GitLab. Para a obtenção de dados desta rede social, assim como o GitHub foi realizado um pedido *get* para a obtenção de dados e seguidamente utilizado os parsers para a filtragem dos dados obtidos.

Também foi utilizado a rede social Pinterest (Pinterest, s.d.). Este uma rede social que permite a partilha de fotos.

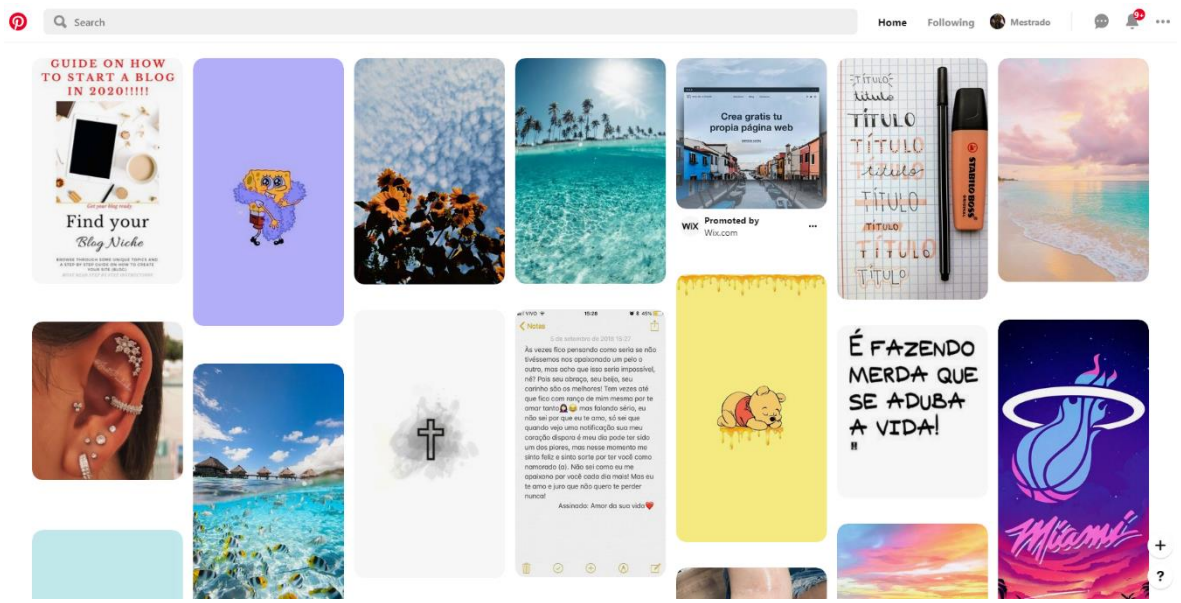


Figura 34 – Pinterest

A Figura 34 representa a página inicial da rede social Pinterest. Para a obtenção das informações foi realizado o mesmo processo da rede social Instagram.

Foi também utilizado a rede social Dev.to (Dev.to, s.d.). Esta uma rede social mais utilizada por programadores que permite partilhar de ideias e ajuda entre os seus vários utilizadores.

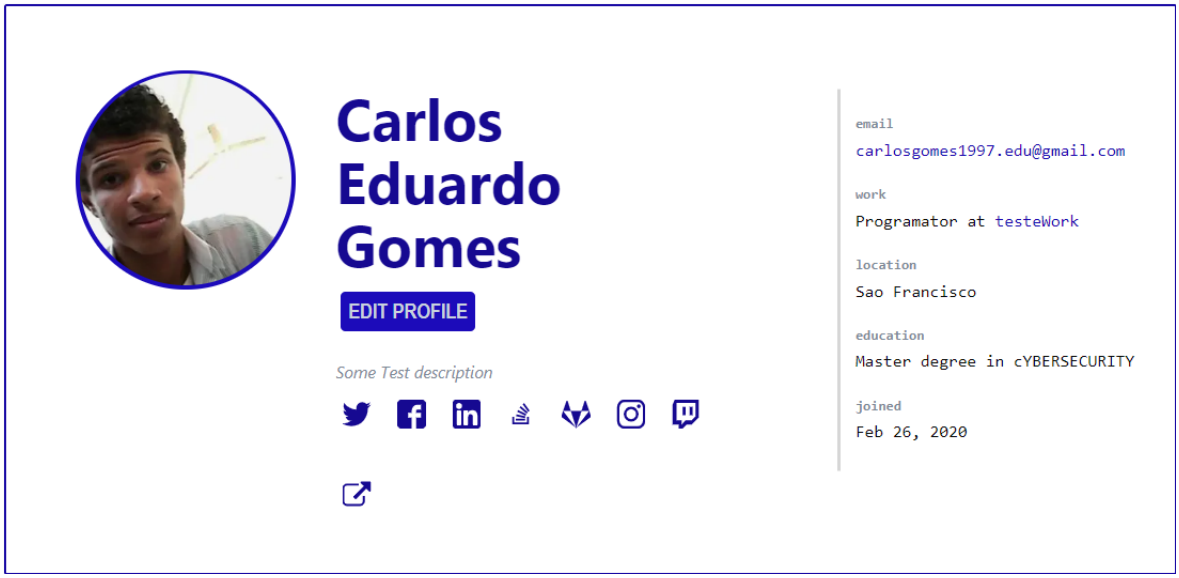


Figura 35 - Dev.to

A Figura 35 representa um exemplo de um perfil da rede social dev.to. Para a obtenção de informação foram utilizados os mesmos processos das redes sociais GitHub e Instagram.

4.4. API de acesso remoto a dados

Foi iniciado o desenvolvimento de uma API que permite a consulta dos dados armazenados na BD da aplicação web. Como teste foram criadas consultas aos dados relativos a rede social Instagram.

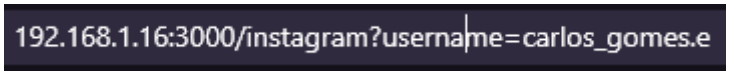


Figura 36 - Exemplo consulta API

A Figura 36 representa um exemplo de uma consulta à API. A aplicação web recebe através do URL o username que posteriormente é efetuada a pesquisa aos dados da BD. Após realizada a pesquisa são devolvidos os dados em formato JSON.

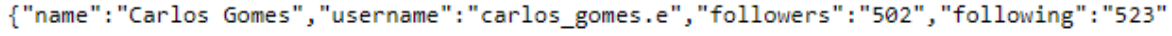


Figura 37 - Dados devolvidos da API

A Figura 37 representa um exemplo de dados retornados após realizado a pesquisa.

5. Testes

Neste capítulo serão feitos teste a aplicação web, tanto teste funcionais como teste a plataforma.

5.1.1. Testes funcionais

Neste subcapítulo será abordado os diferentes testes realizadas a página web para verificação dos requisitos funcionais retratados em 3.4.

- Pesquisar por informações através de endereço de email

Como teste foi pesquisado o endereço de email carlosgomes1997.edu@gmail.com.

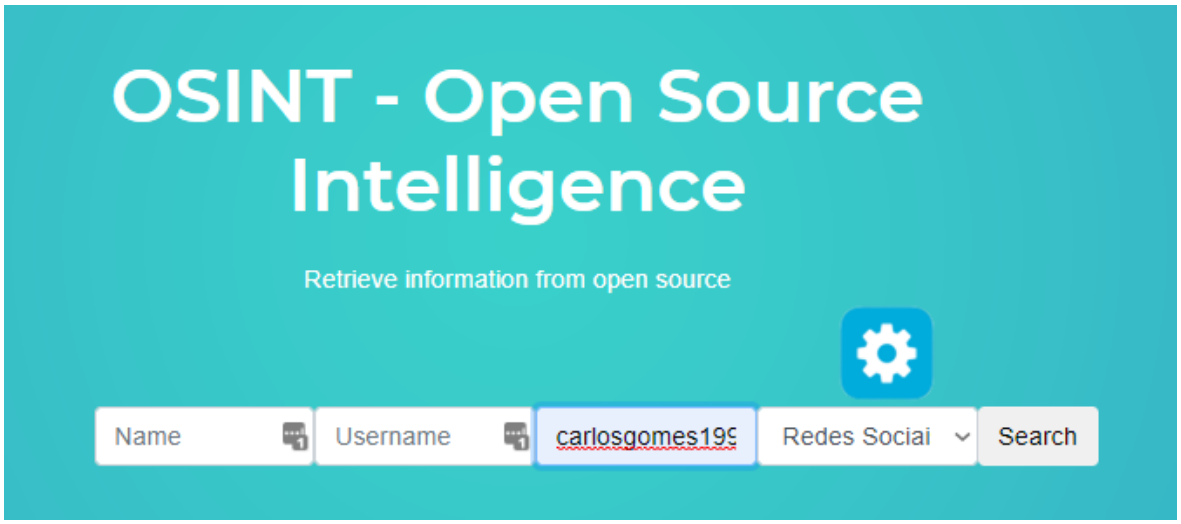


Figura 38 - Pesquisa por email

A Figura 38 representa um exemplo de uma pesquisa por endereço de email.

Figura 39 - Resultados email

A Figura 39 representa a resultado da pesquisa efetuada anteriormente.

- Pesquisar por informações através de um nome de utilizador

Como teste foi pesquisado pelo nome de utilizador SWarrior97.

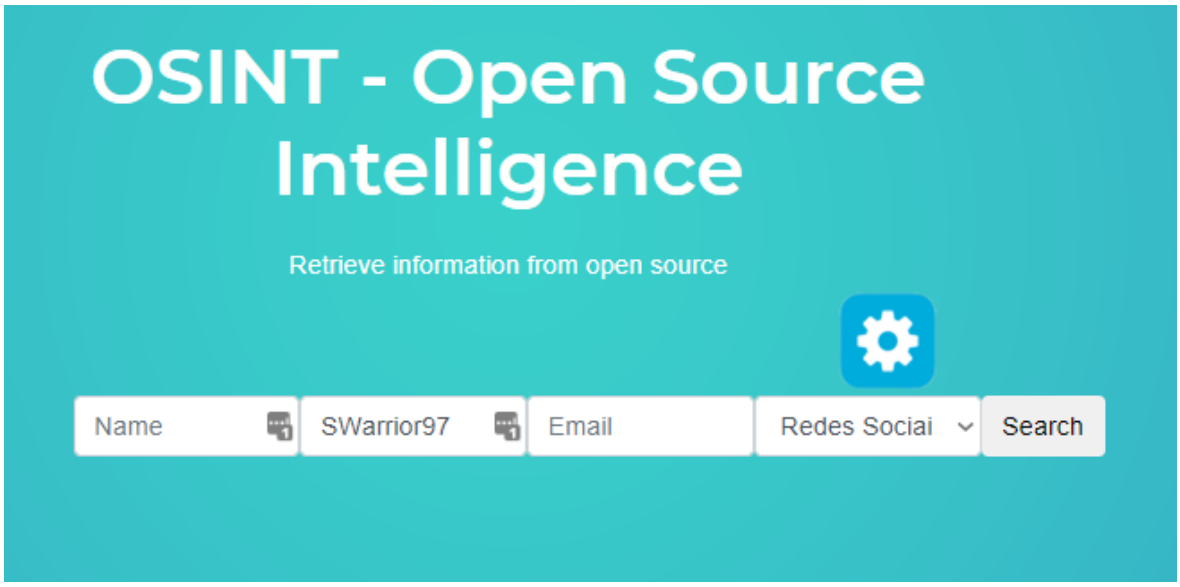


Figura 40 - Teste nome de utilizador

A Figura 40 representa a pesquisa pelo nome de utilizador SWarrior97.

A Figura 41 representa o resultado do teste realizado por nome de utilizador.

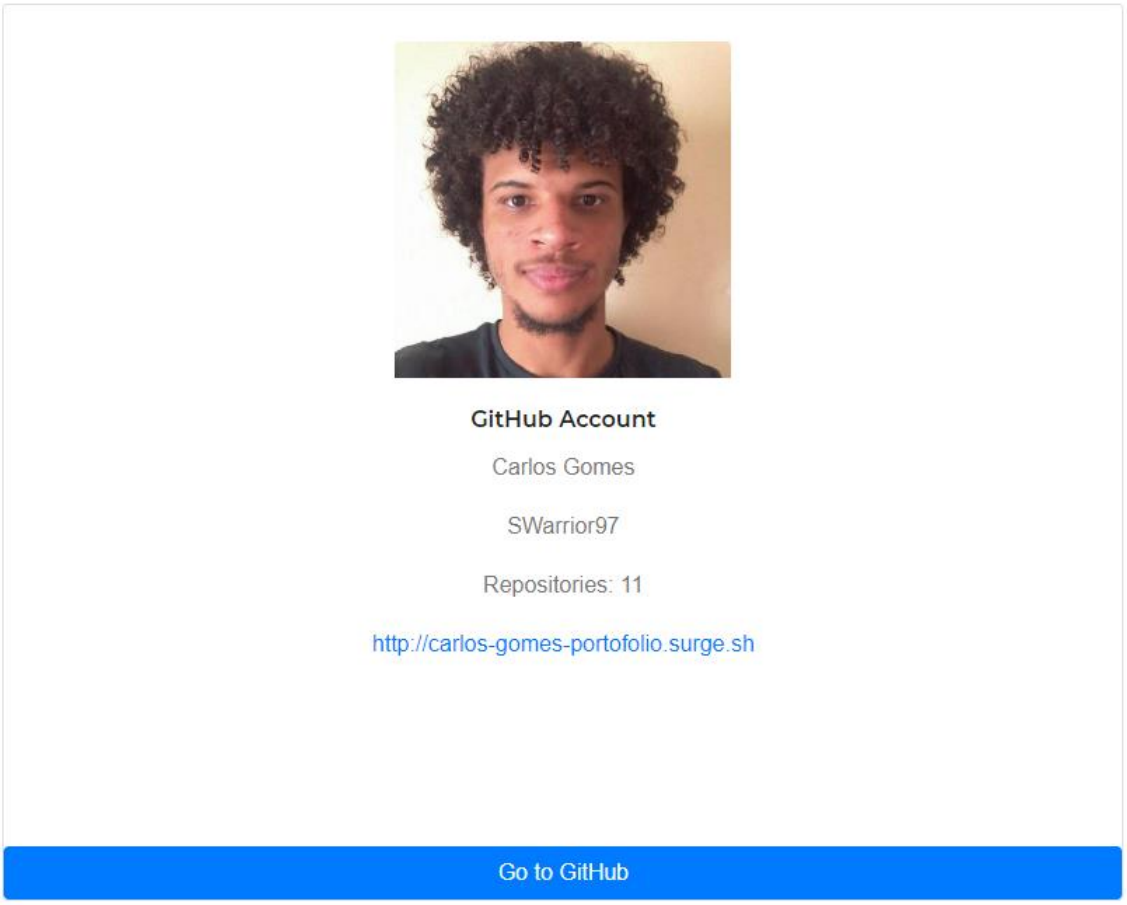


Figura 41 - Teste realizado por nome de utilizador

- Gerar combinações de nomes de utilizador através de um nome

Para a realização deste teste foi escolhido o nome Carlos Gomes.

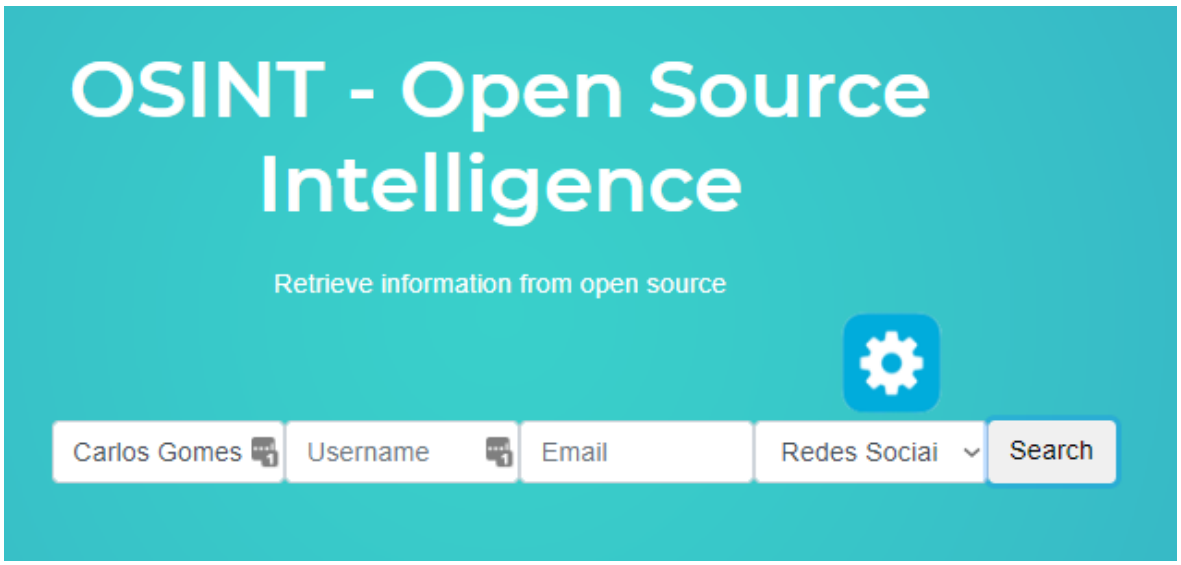


Figura 42 - Pesquisa por nome

A Figura 42 representa a pesquisa realizada pelo nome. Após pesquisar pelo nome são retornados possíveis nomes de utilizador gerados com base no nome.

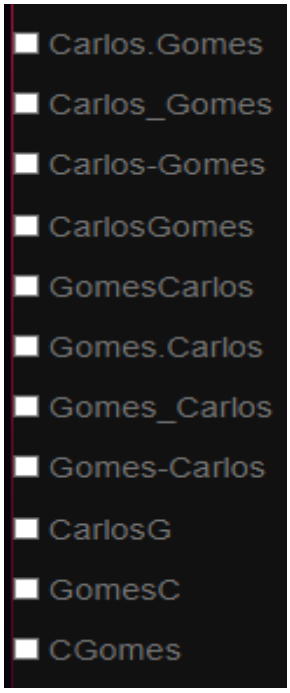


Figura 43 - Nomes de utilizadores gerados

A Figura 43 representa os nomes de utilizadores gerados através do nome Carlos Gomes.

- Pesquisar por informações com base nos nomes de utilizadores gerados

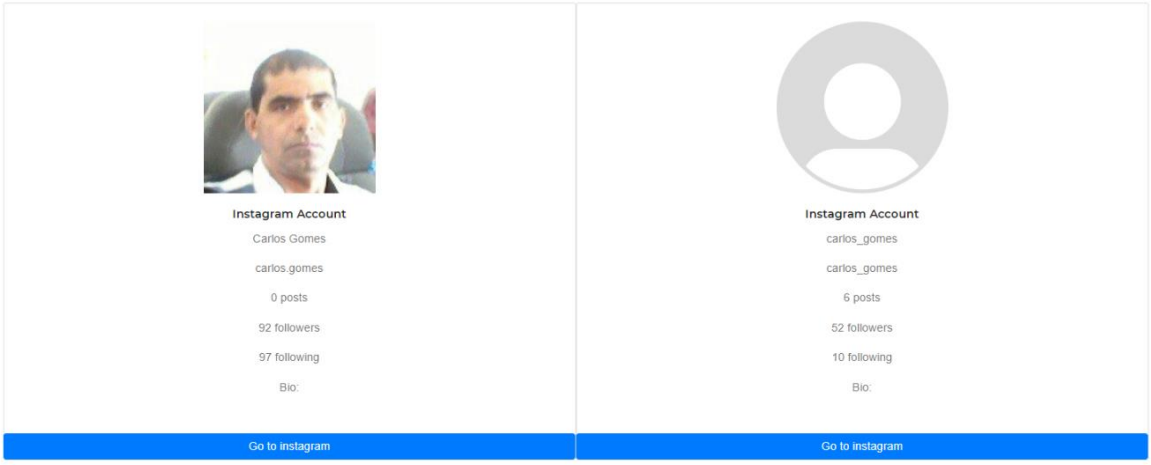


Figura 44 - Resultado da pesquisa realizada por múltiplos nomes de utilizador

A Figura 44 representa o resultado da pesquisa realizada pelos nomes de utilizadores escolhidos pelo utilizador.

- Pesquisar em tempo real por informações

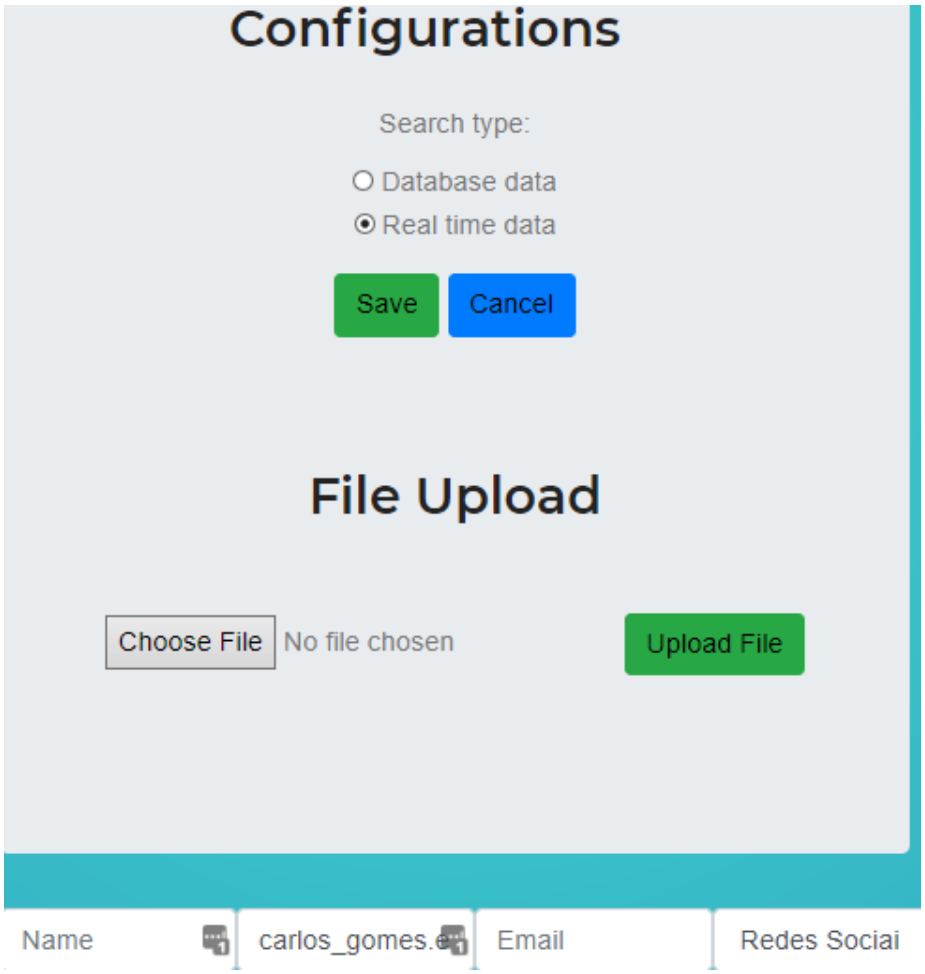


Figura 45 - Pesquisa realizada em tempo real

A Figura 45 representa a pesquisa realizada em tempo real.

Instagram Account

Carlos Gomes

carlos_gomes.e

Gomes's posts

500 followers

543 following

Bio:

Go to instagram

Figura 46 - Resultado pesquisa tempo real

A Figura 46 representa o resultado da pesquisa em tempo real.

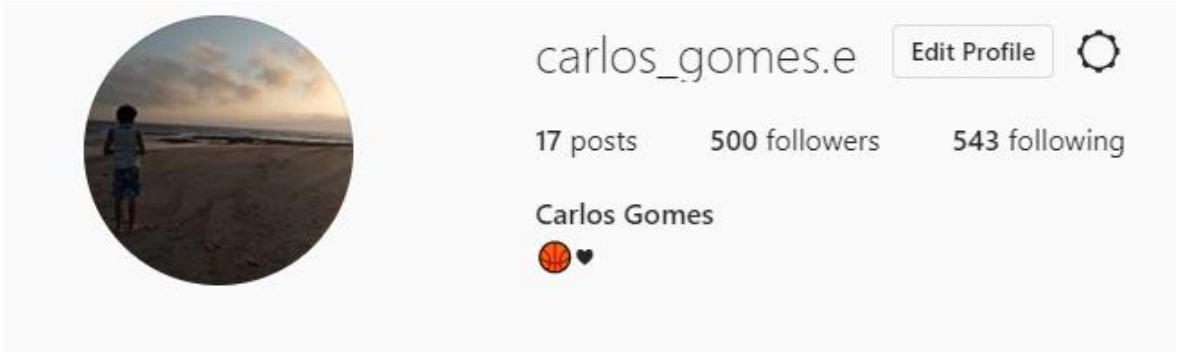


Figura 47 - Prova pesquisa tempo real

A Figura 47 representa a prova da pesquisa realizado em tempo real.

- Pesquisar por informações guardadas na base de dados

Para a realização deste teste foi realizada a pesquisa pelo nome de utilizador SWarrior97.

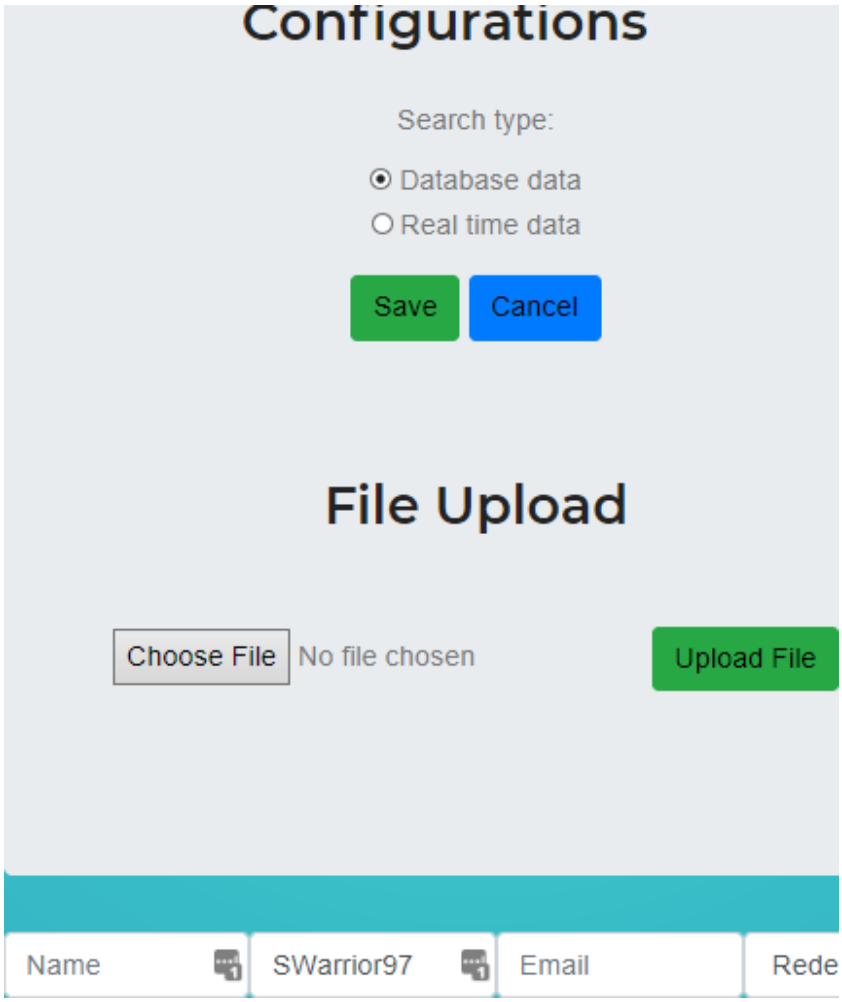


Figura 48 - Pesquisa por dados na base de dados

A Figura 48 representa a pesquisa realizada por dados na BD.

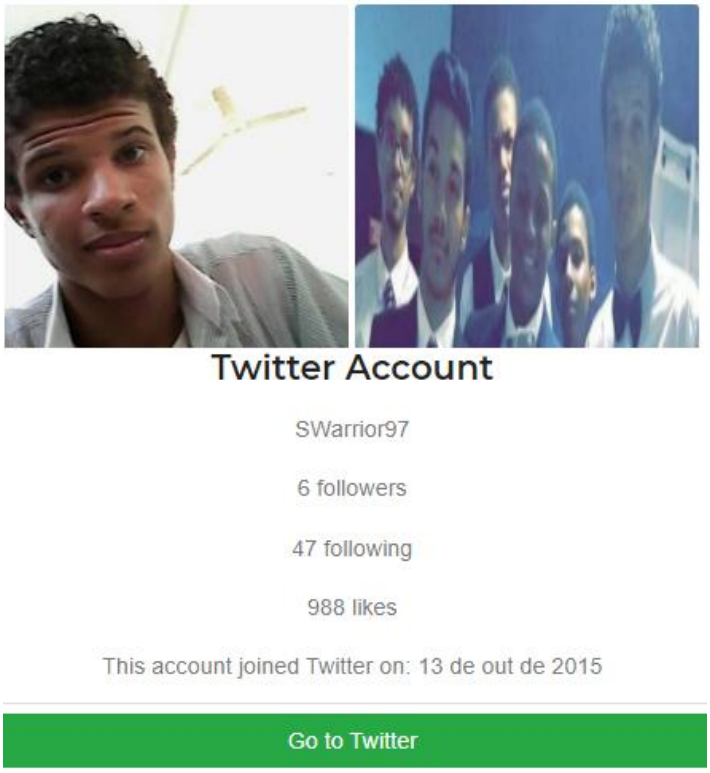


Figura 49 - Resultado da pesquisa na base de dados

A Figura 49 representa o resultado da pesquisa realizada pelos dados na BD.

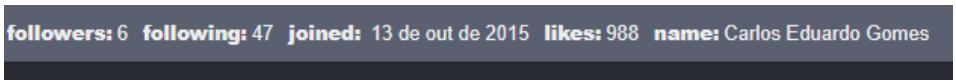


Figura 50 - Prova pesquisa por dados na base de dados

A Figura 50 representa a prova da pesquisa realizada por dados na BD.

- Poder submeter ficheiros com informações

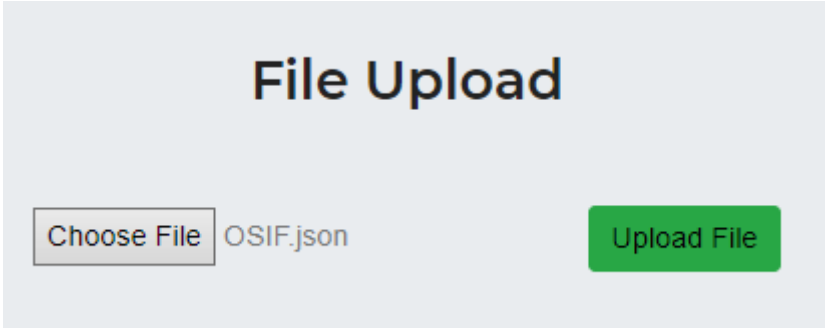


Figura 51 - Upload ficheiros

A Figura 51 representa o exemplo de upload de ficheiro de nome OSIF. Após o upload do ficheiro, são filtrados os dados e mostrados ao utilizador de uma forma simples.

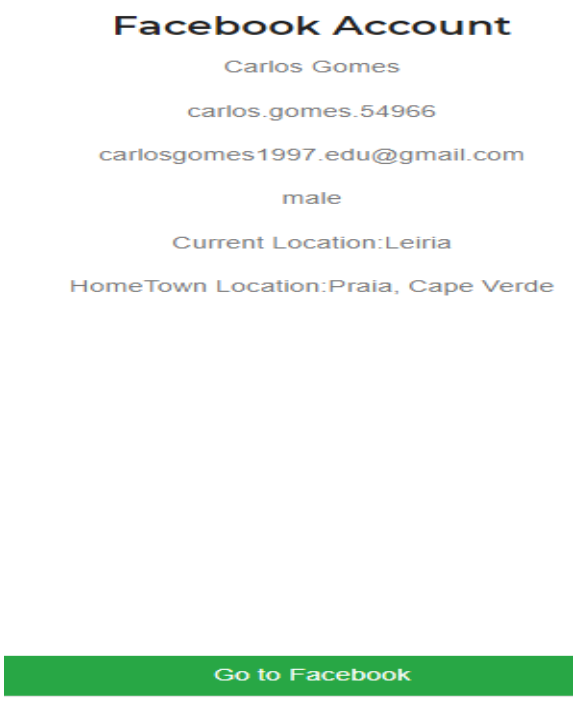


Figura 52 - Resultado upload ficheiro

A Figura 52 representa o resultado do upload de ficheiro.

- Poder pesquisar pelo histórico de um determinado nome de utilizador na data corrente

A Figura 53 representa a escolha pela pesquisa através do histórico na data atual.

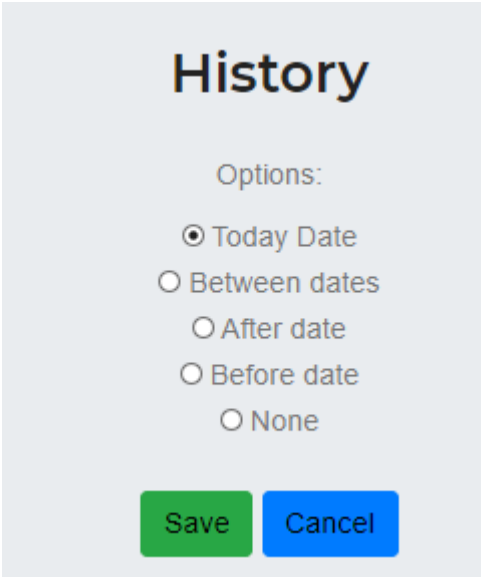


Figura 53 - Pesquisa histórico dia atual

Para realizar a pesquisa do histórico foi utilizado o nome de utilizador *carlos_gomes.e*.

A Figura 54 representa o resultado da pesquisa efetuada pelo histórico na data atual.

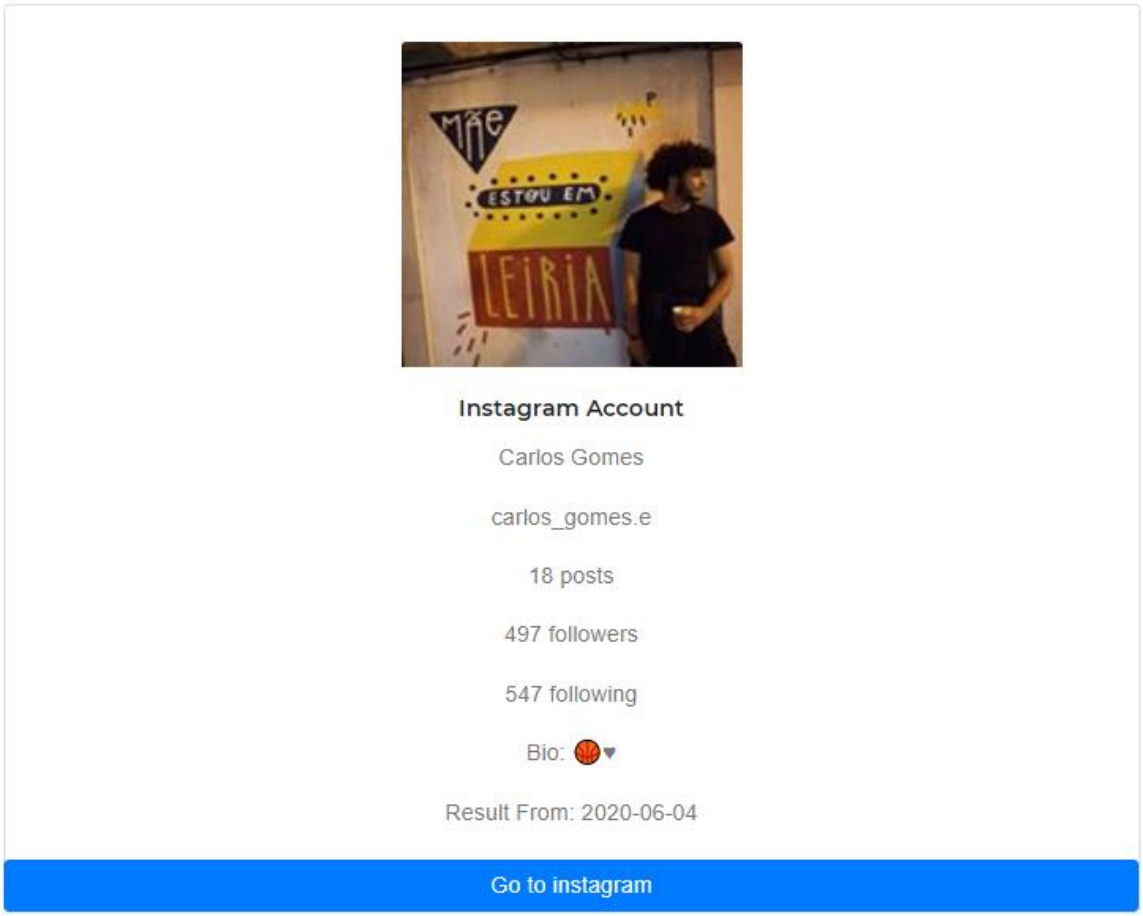


Figura 54 – Resultado da pesquisa pelo histórico do dia atual

- Poder pesquisar pelo histórico de um determinado nome de utilizador entre datas

A Figura 55 representa a escolha da opção de pesquisa entre duas datas.

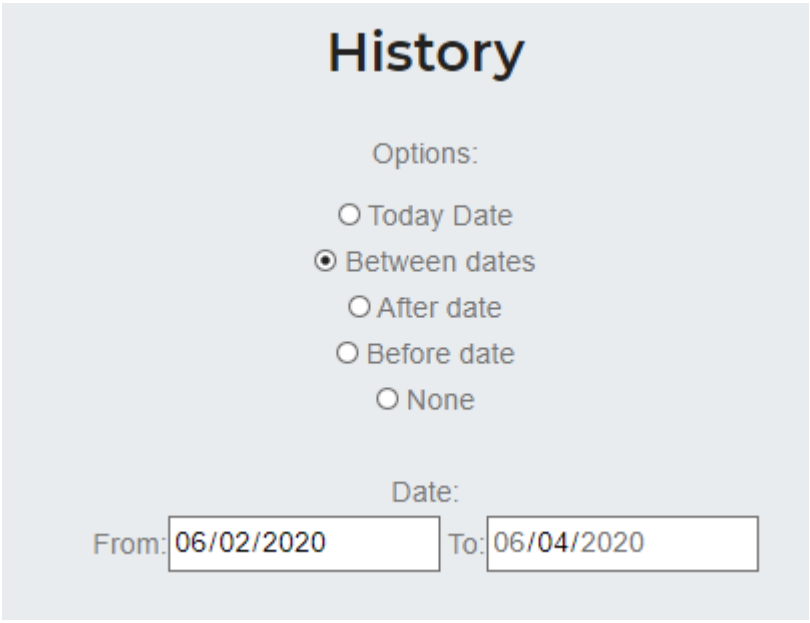


Figura 55 - Escolha da opção de pesquisa pelo histórico entre duas datas

Para realizar a pesquisa do histórico foi utilizado o nome de utilizador *carlos_gomes.e*.

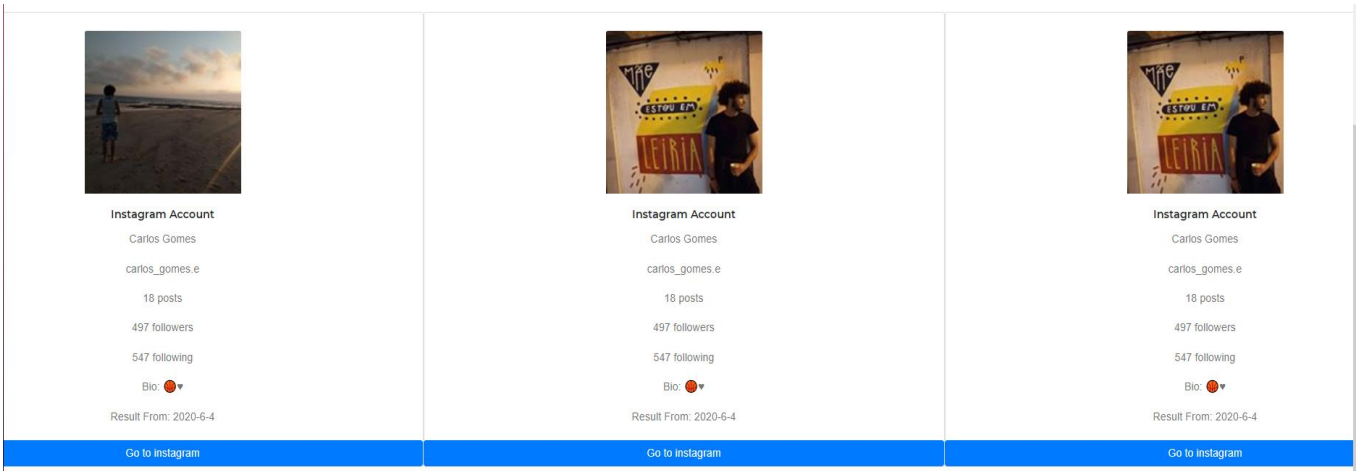


Figura 56 – Resultado da pesquisa entre duas datas

A Figura 56 representa a resultado após efetuada a pesquisa entre duas datas.

- Poder pesquisar pelo histórico de um determinado nome de utilizador depois de uma data

A Figura 57 representa a seleção da opção de pesquisa pelo histórico depois de uma data.

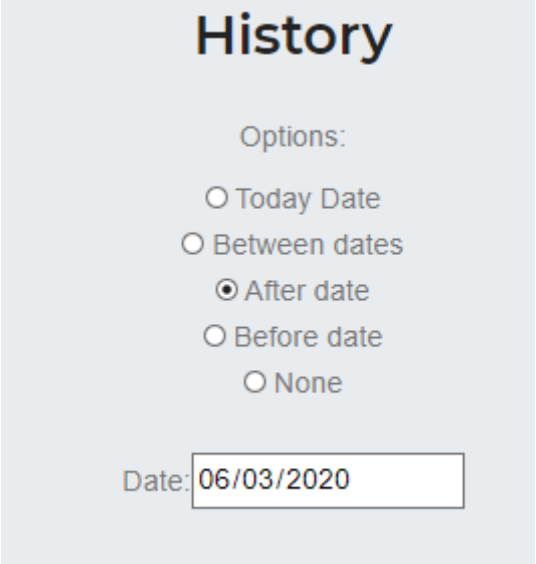


Figura 57 - Escolha opção pesquisa pelo histórico depois de uma data

Para a realização do teste da pesquisa ao histórico depois de uma data foi utilizador o nome de utilizador *carlos_gomes.e.*

A Figura 58 representa o resultado após efetuada a pesquisa pelo histórico depois de uma data escolhida pelo utilizador.

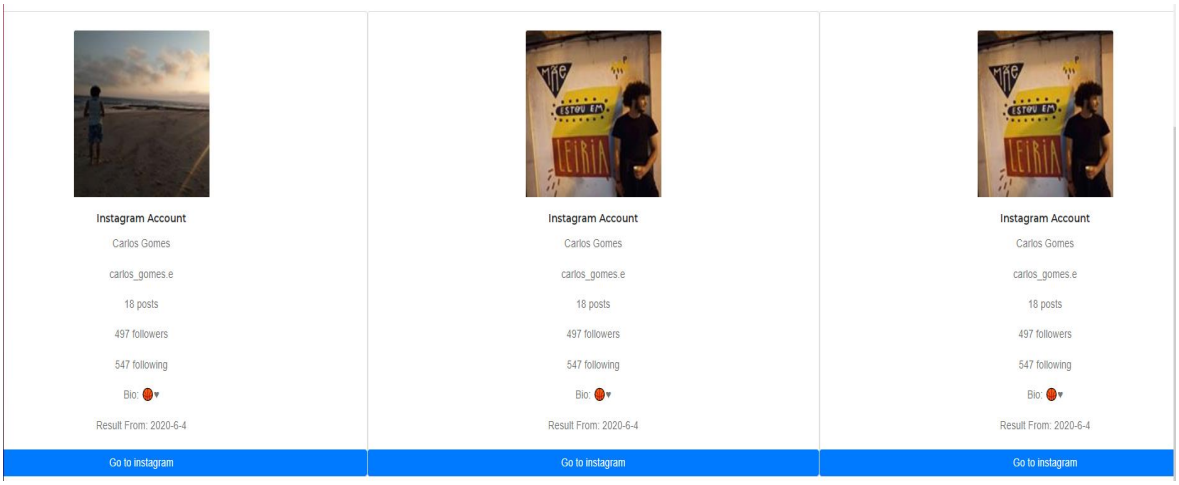


Figura 58 - Resultado pesquisa pelo histórico depois de uma data

- Poder pesquisar pelo histórico de um determinado nome de utilizador antes de uma data

A Figura 59 representa a escolha da opção de pesquisa pelo histórico antes de uma data selecionada pelo utilizador.

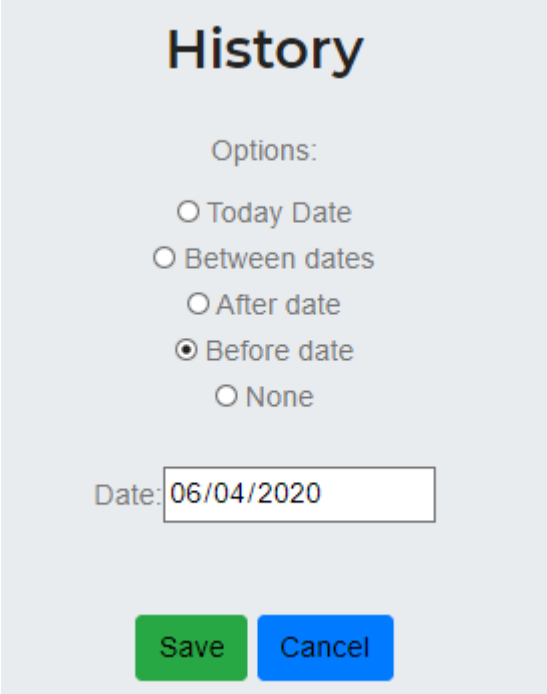


Figura 59 - Escolha opção pesquisa pelo histórico antes de uma data

Para a realização do teste da pesquisa ao histórico depois de uma data foi utilizador o nome de utilizador *carlos_gomes.e*.

A Figura 60 representa o resultado após efetuada a pesquisa pelo histórico antes de uma data escolhida pelo utilizador.

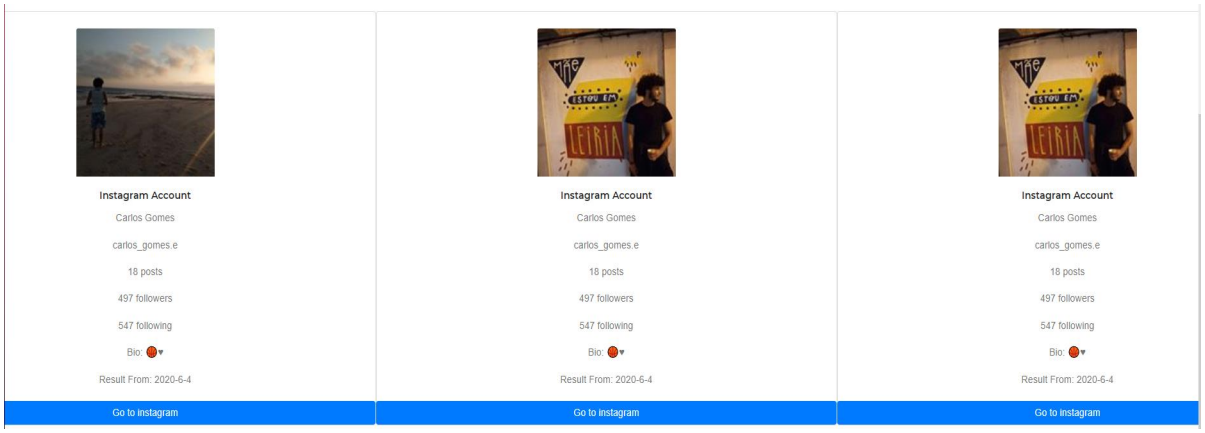


Figura 60 - Resultado pesquisa pelo histórico antes de uma data

5.1.2. Testes à plataforma

Neste subcapítulo serão abordados alguns testes realizados à plataforma web.

- Teste de inserção de dados não standard

Como já referido anteriormente, a plataforma web permite a inserção de ficheiros com de dados a fim de serem processados e ser guardados na BD e mostrados ao utilizador. Mas nem sempre os dados estão num formato reconhecido pela página web. Por isso um dos testes realizados foi a inserção de dados num formato não standard.

Para o primeiro teste foi tentado inserir um ficheiro num formato não aceite pela plataforma web. A Figura 61 representa o teste realizado.

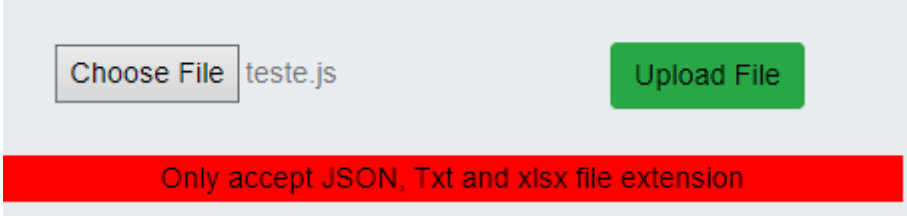


Figura 61 - Extensão de ficheiro não permitido

De seguida, para a realização de mais testes foram gerados ficheiros com um formato não standard para os diferentes tipos de scripts feitos: OSIF, SocialScan, Sherlock.

- OSIF

Para a realização deste teste foi gerado um ficheiro com falta de dados e com dados insuficientes e com ordem não standard. Após a realização do carregamento do ficheiro foi possível observar que alguns dados foram guardados de forma errada, enquanto outros dados estavam em falta como pode ser observado na Figura 62. Através da figura pode-se observar que os dados "nome de utilizador" e "nome" estão em falta e, consequentemente não são guardados na BD, sendo apenas apresentado mostrado ao utilizador.

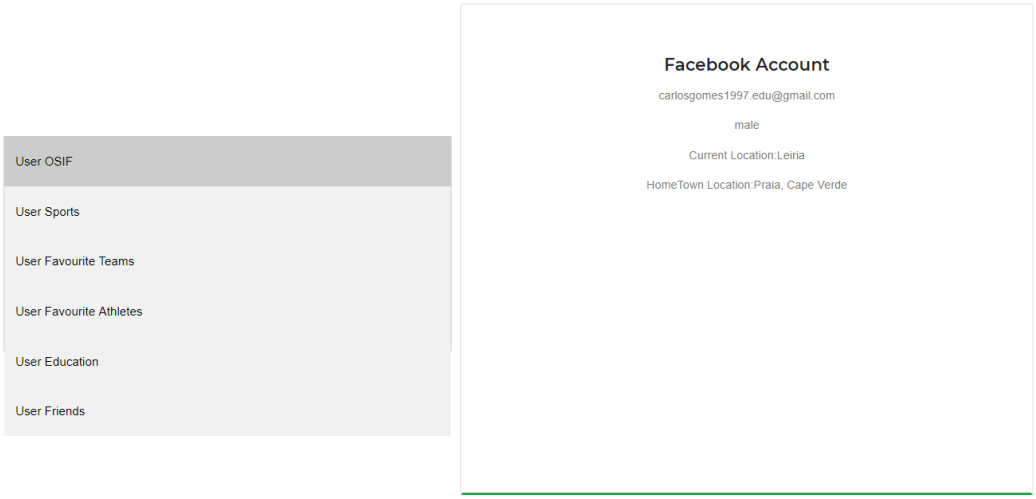


Figura 62 - Teste inserção dados formato não standard OSIF

- SocialScan

Para a realização deste teste foi criado ficheiros com dados diferentes do qual a plataforma web espera, bem como dados inexistentes.

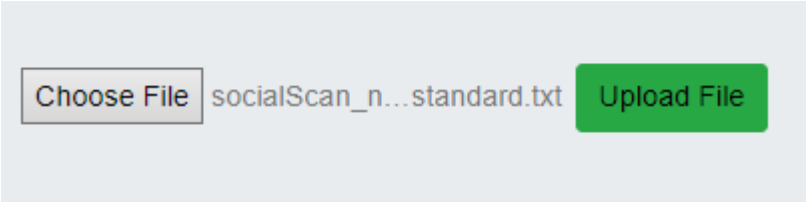


Figura 63 - Teste inserção dados não standard SocialScan

A Figura 63 representa a inserção do ficheiro para a realização do teste. Após a realização deste teste a plataforma web não conseguiu tratar o ficheiro e assim não consegui mostrar dados acerca do ficheiro.

- Sherlock

Para a realização deste teste foi criado ficheiros com falta de informação e em ordem trocada.

Após a realização deste teste observou-se que os dados foram processados pela plataforma web de forma errada, sendo que não foi possível determinar o nome de utilizador, logo os dados não foram guardados na BD, como pode ser observado na Figura 64.

Figura 64 - Teste inserção dados não standard Sherlock

6. Conclusões e Trabalho Futuro

Nos dias de hoje cada vez mais pessoas partilham os seus dados na internet sem terem a noção disso. Com o desenvolvimento deste projeto foi possível estudar as diversas técnicas e ferramentas já existentes e também foi possível desenvolver uma solução para a pesquisa de dados em fontes abertas, nomeadamente as redes sociais que são ricas em informações.

O resultado obtido com o desenvolvimento deste projeto foi um sucesso, no ponto de terem sido concluídos todos os objetivos definidos. Foi um trabalho gratificante podendo não só pôr em prática os conhecimentos de web developer, mas também os conhecimentos de OSINT.

A solução ainda poderá ser melhorada em diversos aspetos, nomeadamente:

- Acrescentando cada vez mais fontes onde podem ser realizadas pesquisas;
- O design da página web poderá ser melhorado;
- Automatização da BD, isto é, periodicamente a solução gerar nomes e efetua a pesquisas automáticas armazenando os dados na BD para futuras consultas.

7. **Bibliografia**

4kStogram. (s.d.). Acedido em 05 de 10 de 2019, de https://www.4kdownload.com/products/product-stogram

Antunes, M., & Rodrigues, B. (2018). *Introdução À Cibersegurança.* FCA.

Ask.fm. (s.d.). Acedido em 16 de 10 de 2019, de https://ask.fm

Bruggen, R. V. (2014). *Learning neo4j*. Birmingham B3 2PB, UK.: Packt.

Censys. (s.d.). Acedido em 15 de 10 de 2019, de https://censys.io

Check UserNames. (s.d.). Acedido em 15 de 10 de 2019, de https://checkusernames.com

CiKu370. (s.d.). *GitHub*. Acedido de https://github.com/CiKu370/OSIF

Cooya. (s.d.). *GitHub*. Acedido em 15 de 03 de 2020, de https://github.com/Cooya/Linkedin

Dev.to. (s.d.). Acedido em 15 de 02 de 2020, de https://dev.to

EmailRep. (s.d.). Acedido em 18 de 02 de 2020, de https://emailrep.io

Facebook. (s.d.). Acedido em 17 de 02 de 2020, de https://www.facebook.com

Foundation, P. S. (s.d.). *pypi*. Acedido em 15 de 03 de 2020, de https://pypi.org/project/socialscan/

FullContact. (s.d.). Acedido em 24 de 02 de 2020, de https://www.fullcontact.com/developer-portal/

Geng, B., & ̆achescu, C. ̆. (Novembro de 2016). *ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services, 9*(15). Acedido em 12 de 10 de 2019

GitHub. (s.d.). Acedido em 01 de 03 de 2020, de https://github.com

GitLab. (s.d.). Acedido em 20 de 02 de 2020, de https://about.gitlab.com

Hunter. (s.d.). Acedido em 20 de 10 de 2019, de https://hunter.io

Hypersight. (s.d.). Acedido em 15 de 10 de 2019, de https://www.hypersight.com/

I know what you download. (s.d.). Acedido em 12 de 12 de 2019, de https://iknowwhatyoudownload.com/

ICANN. (s.d.). *Parte I: o que são metadados?* Acedido em 18 de 11 de 2019, de https://www.icann.org/news/blog/parte-i-o-que-sao-metadados

iKy. (s.d.). Acedido em 10 de 01 de 2020, de https://kennbroorg.gitlab.io/ikyweb/

Instagram. (s.d.). Acedido em 20 de 02 de 2020, de https://www.instagram.com

Kali Linux Tools. (s.d.). Acedido em 15 de 12 de 2020, de https://tools.kali.org/information-gathering/metagoofil

kennbroorg. (s.d.). *iKy*. Acedido em 12 de 02 de 2020, de Github: https://github.com/kennbroorg/iKy

Kernan, W. F. (2001). *Nato Open Source Intelligence HandBook.* Obtido de NATO Open Source Intelligence Handbook.

Letterboxd. (s.d.). Acedido em 23 de 02 de 2020, de https://letterboxd.com

Linkedin. (s.d.). Acedido em 01 de 03 de 2020, de https://www.linkedin.com

lockfale. (s.d.). *GitHub*. Acedido em 01 de 10 de 2019, de https://github.com/lockfale/OSINT-Framework

MailDB. (s.d.). Acedido em 15 de 12 de 2019, de https://dash.maildb.io/login

Milani, A. (2008). *PostgreSQL Guia do Programador*. NOVATEC EDITORA Lta. ISBN:978-85-7522-157-0

MongoDB. (s.d.). Acedido em 03 de 01 de 2020, de https://www.mongodb.com/what-is-mongodb

Nayak, A., Poriya, A., & Poojary, D. (2013, Março). *Type of NOSQL Databases and its Comparison with Relational Databases.* International Journal of Applied Information Systems *5*(4), pp. 16-19.

Numbering Plans. (s.d.). Acedido em 3 de 12 de 2019, de https://www.numberingplans.com/?page=analysis&sub=phonenr

Oracle. (s.d.). Acedido em 05 de 01 de 2020, de What a Relational Database Is: https://www.oracle.com/pt/database/what-is-a-relational-database/

OSInt Framework. (s.d.). Acedido em 01 de 10 de 2019, de https://osintframework.com

Paterva. (s.d.). *Maltego*. Acedido em 20 de 11 de 2020, de https://www.paterva.com/web7/buy/maltego-clients/maltego-ce.php

Pinterest. (s.d.). Acedido em 20 de 02 de 2020, de https://www.pinterest.pt

PostgreSQL. (s.d.). Acedido em 15 de 02 de 2020, de https://www.postgresql.org

Robinson, I., Webber, J., & Eifren, E. (2013). *Graph Database* (1ª ed.). O'Reilly.

sherlock-project. (s.d.). *GitHub*. Acedido em 12 de 01 de 2020, de https://github.com/sherlock-project/sherlock

Shodan. (s.d.). Acedido em 19 de 12 de 2020, de https://www.shodan.io

Social Searcher. (s.d.). Acedido em 28 de 12 de 2020, de https://www.social-searcher.com

Spiderfoot. (s.d.). *Spiderfoot*. Acedido em 12 de 01 de 2020, de https://www.spiderfoot.net

TikTok. (s.d.). Acedido em 15 de 02 de 2020, de https://www.tiktok.com

Twitter. (s.d.). Acedido em 18 de 02 de 2020, de https://twitter.com

User Search. (s.d.). Acedido em 16 de 10 de 2019, de https://usersearch.org/index.php

ViewDns. (s.d.). Acedido em 12 de 12 de 2019, de https://viewdns.info/whois/

Who Calld. (s.d.). Acedido em 12 de 12 de 2019, de https://whocalld.com

Whoxy Domain Search Engine. (s.d.). Acedido em 10 de 12 de 2019, de https://www.whoxy.com/reverse-whois/demo.php

Zap. (s.d.). Acedido em 15 de 02 de 2020, de http://zap.cvmultimedia.cv/?page_id=50099

Glossário

Application Programming Interface: conjunto de funções e protocolos aplicacionais que permite aceder aos dados guardados num sistema de informação.

Arquitetura Cliente-Servidor: é uma estrutura de aplicativos distribuídos que particiona tarefas ou cargas de trabalho entre os provedores de um recurso ou serviço, chamados servidores, e solicitantes de serviço, chamados clientes

Vue.js: framework de JavaScript focado no desenvolvimento de interfaces de utilizador.

Node.js: Interpretador JavaScript focado no desenvolvimento de servidores.

Sockets: ponto que serve para receber ou enviar dados.

Driver neo4j para node.js: driver que permite conectar a base de dados neo4j através do node.js

Anexos

Anexo A – OSINT Framework

O primeiro anexo retrata a página web OSINT framework retratada no estado da arte em 2.1

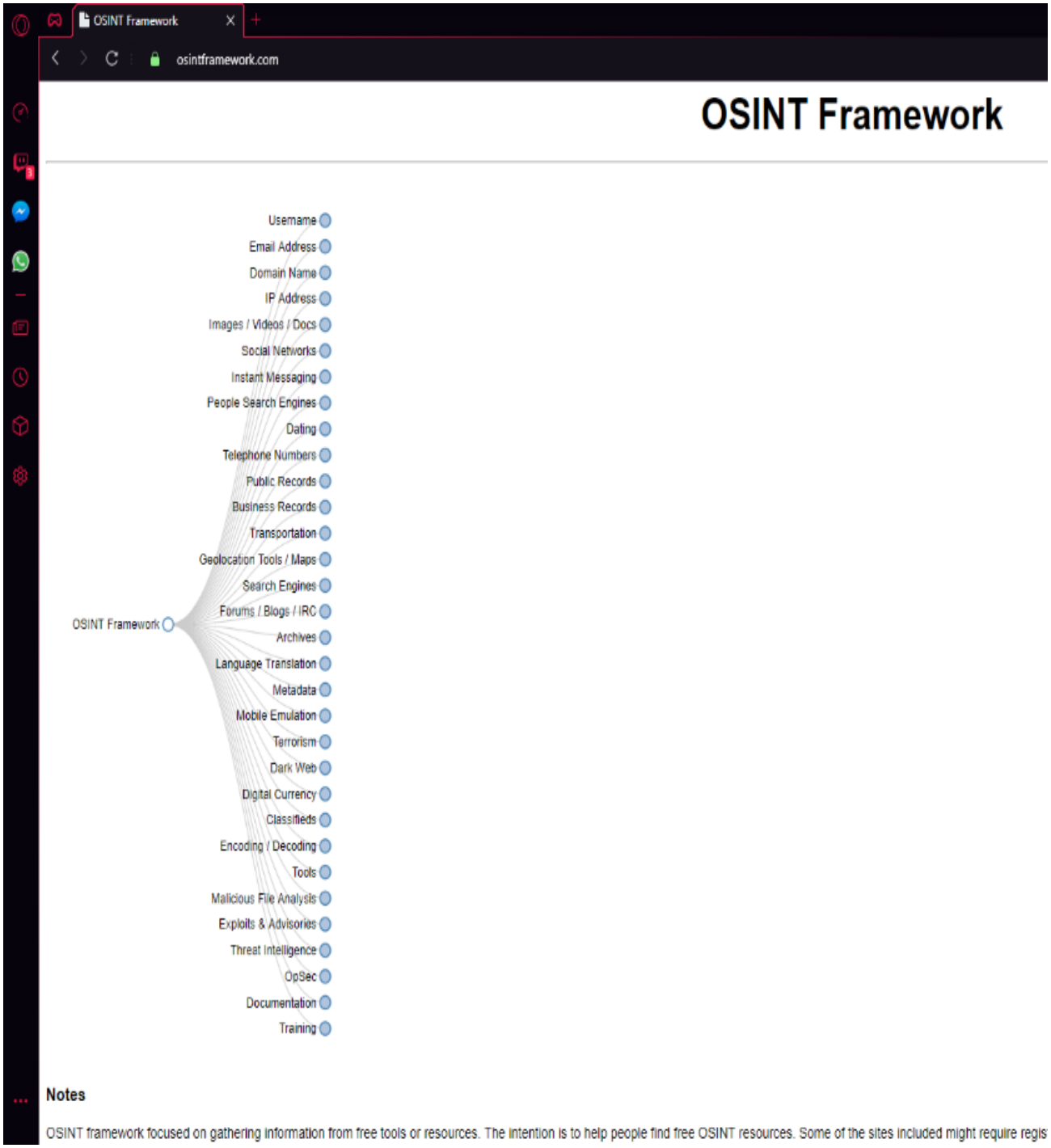


Figura 65 - OSINT Framework

Anexo B – Social Searcher

Ferramenta Social Searcher, também retratada no estado da arte em 2.2.2.

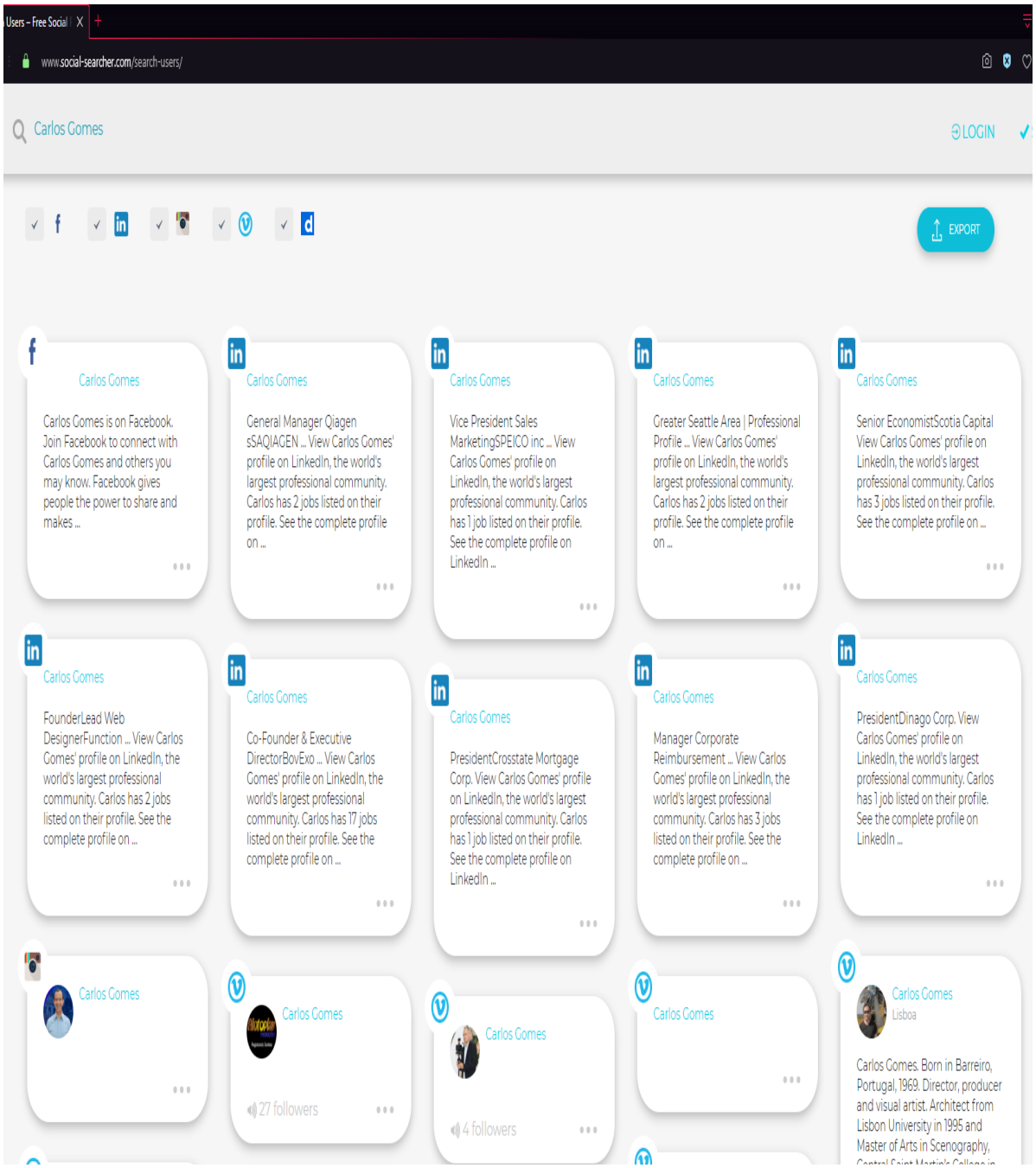


Figura 66 - Social Searcher

Anexo C – CheckUserName

Ferramenta retratada no estado da arte em 2.2.3.

Figura 67 - CheckUserName

Anexo D – Numbering Plans

Ferramenta retratada no estado da arte em 2.2.4.

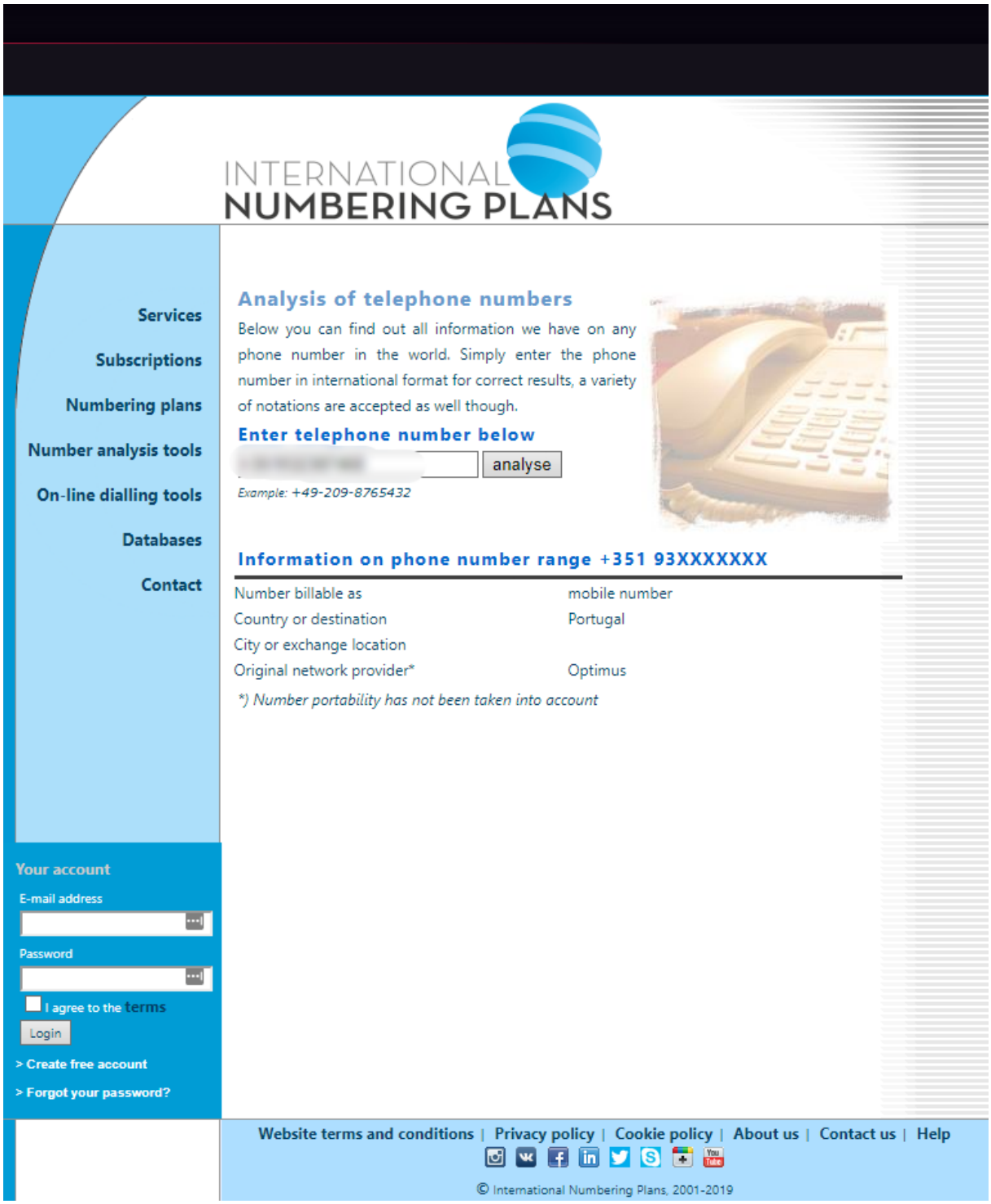


Figura 68 - Numbering Plans

Anexo E – Shodan

Retratado no estado da arte em 2.3.1.

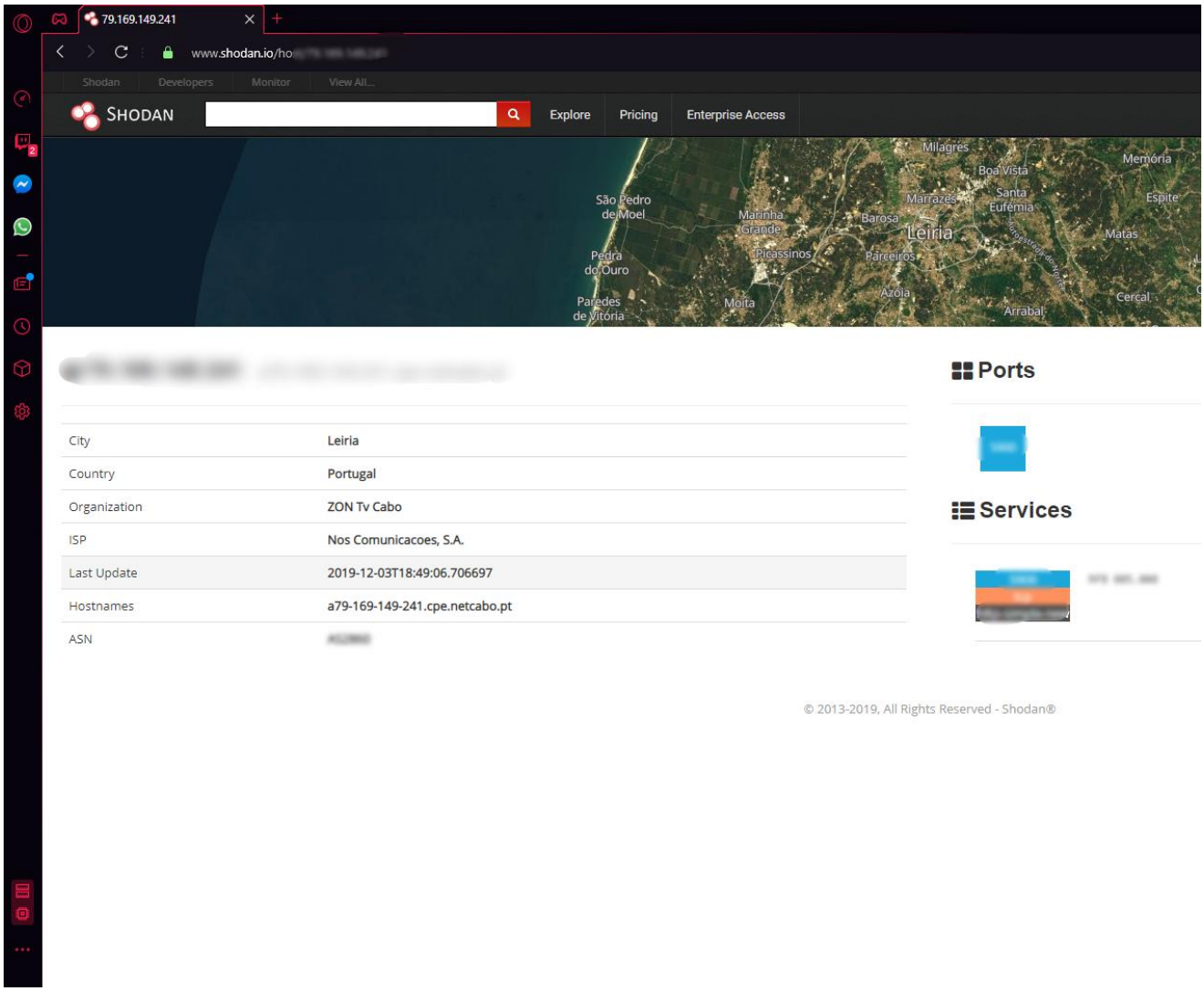


Figura 69 - Shodan

Anexo F – Hunter.io

Retratado no estado da arte em 2.3.2.

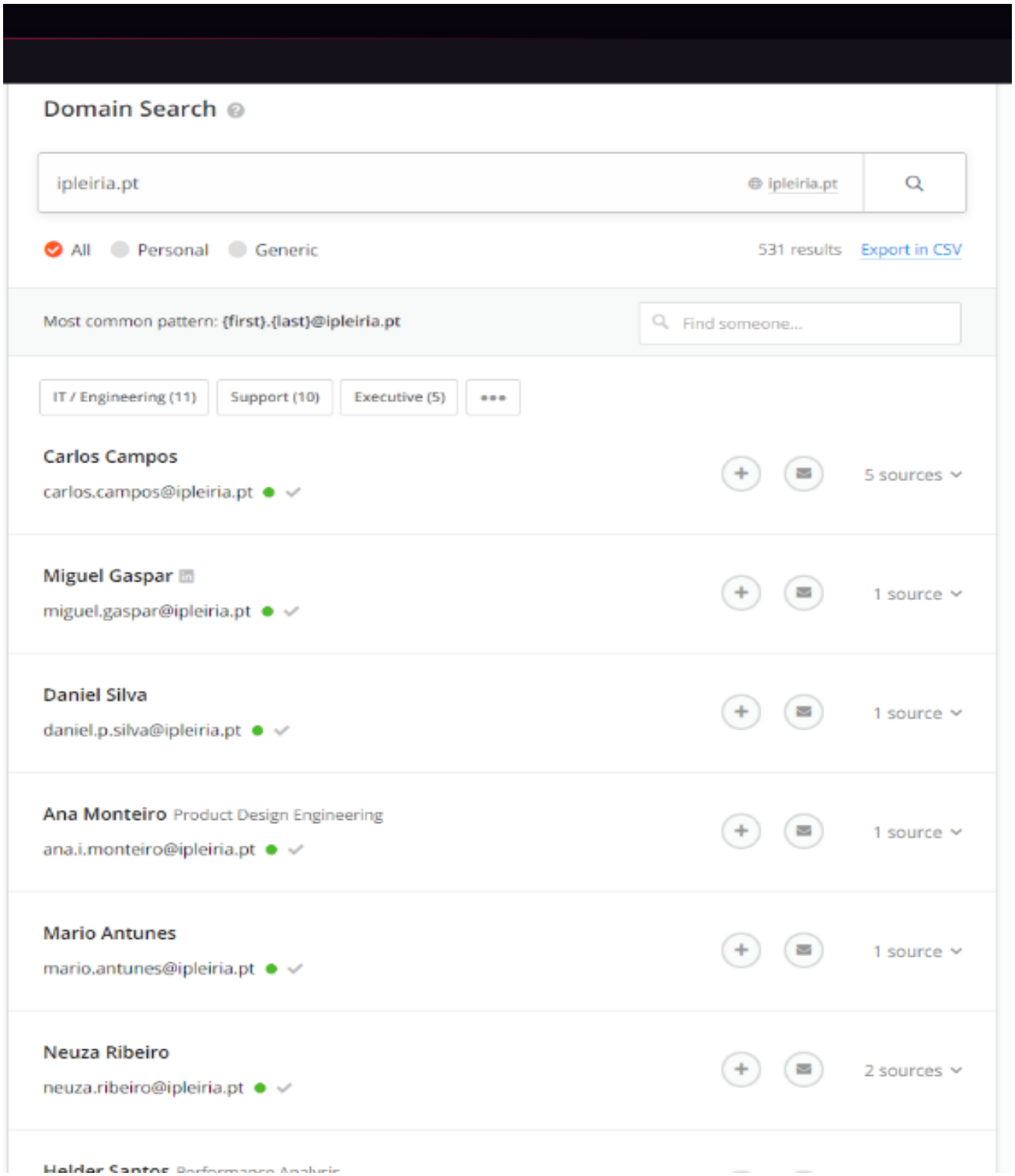
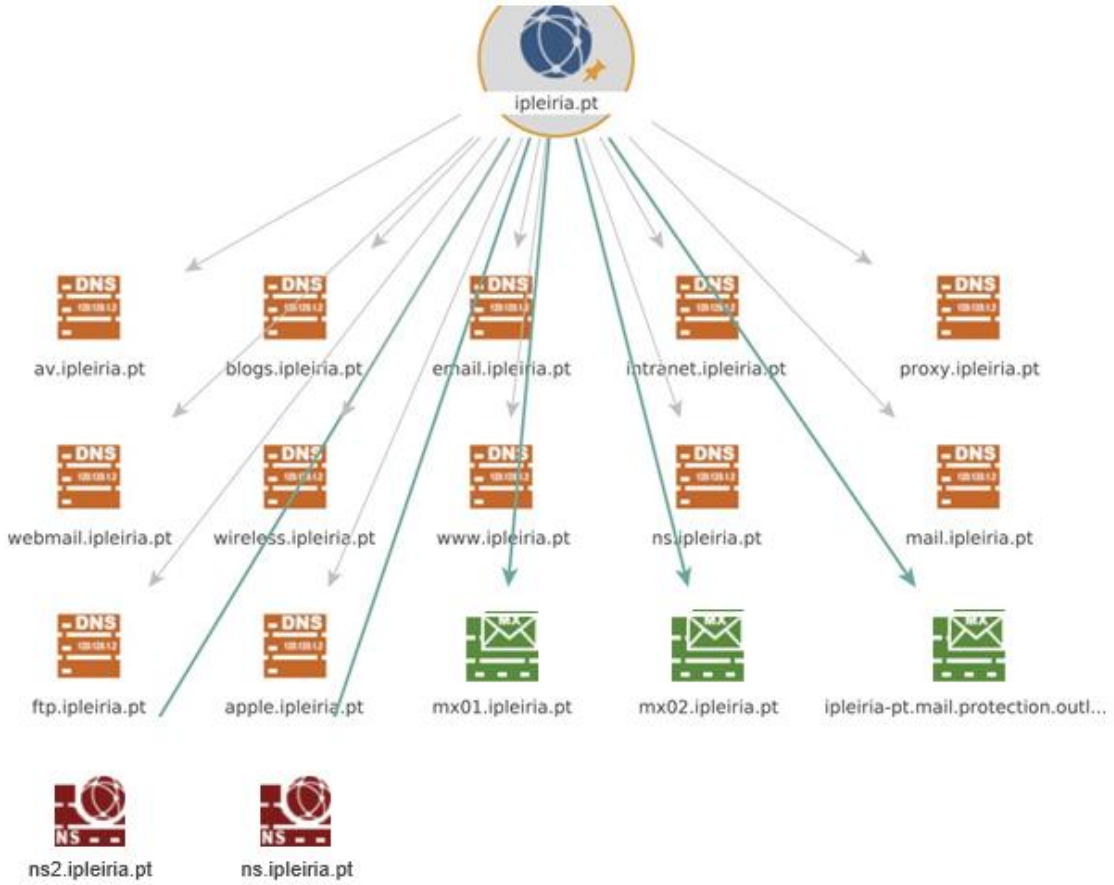


Figura 70 - Hunter.io

Anexo G – Maltego



Top 10 Entities

Total number of entities	18
Total number of links	17

Ranked by Incoming Links

Rank	Type	Value		Incoming links
1	DNS Name	apple.ipleiria.pt	1	
2	NS Record	ns.ipleiria.pt	1	
3	NS Record	ns2.ipleiria.pt	1	
4	DNS Name	email.ipleiria.pt	1	
5	DNS Name	blogs.ipleiria.pt	1	
6	DNS Name	av.ipleiria.pt	1	
7	DNS Name	wireless.ipleiria.pt	1	
8	DNS Name	webmail.ipleiria.pt	1	
9	DNS Name	proxy.ipleiria.pt	1	
10	DNS Name	intranet.ipleiria.pt	1	

Ranked by Outgoing Links

Rank	Type	Value		Outgoing links
1	Domain	ipleiria.pt		17
2	DNS Name	apple.ipleiria.pt	0	
3	NS Record	ns.ipleiria.pt	0	
4	NS Record	ns2.ipleiria.pt	0	
5	DNS Name	email.ipleiria.pt	0	
6	DNS Name	blogs.ipleiria.pt	0	
7	DNS Name	av.ipleiria.pt	0	
8	DNS Name	wireless.ipleiria.pt	0	
9	DNS Name	webmail.ipleiria.pt	0	
10	DNS Name	proxy.ipleiria.pt	0	

Ranked by Total Links

Rank	Type	Value	Total	links
1	Domain	ipleiria.pt		17
2	DNS Name	apple.ipleiria.pt	1	
3	NS Record	ns.ipleiria.pt	1	
4	NS Record	ns2.ipleiria.pt	1	

5	DNS Name	email.ipleiria.pt	1
6	DNS Name	blogs.ipleiria.pt	1
7	DNS Name	av.ipleiria.pt	1
8	DNS Name	wireless.ipleiria.pt	1
9	DNS Name	webmail.ipleiria.pt	1
10	DNS Name	proxy.ipleiria.pt	1

Entities by Type

DNS Names (12)	
apple.ipleiria.pt	av.ipleiria.pt
blogs.ipleiria.pt	email.ipleiria.pt
ftp.ipleiria.pt	intranet.ipleiria.pt
mail.ipleiria.pt	ns.ipleiria.pt
proxy.ipleiria.pt	webmail.ipleiria.pt
wireless.ipleiria.pt	www.ipleiria.pt

Domains (1)
ipleiria.pt

MX Records (3)	
ipleiria-pt.mail.protection.outlook.com	mx01.ipleiria.pt
mx02.ipleiria.pt	

NS Records (2)	
ns.ipleiria.pt	ns2.ipleiria.pt

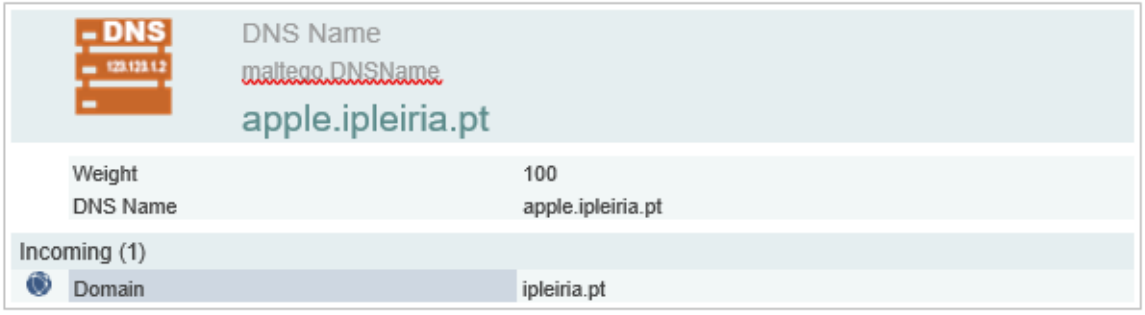
Entity Details

Domain
maltego.Domain
ipleiria.pt

Weight	0
Domain Name	ipleiria.pt
WHOIS Info	

Outgoing (17)

DNS Name	apple.ipleiria.pt
DNS Name	av.ipleiria.pt
DNS Name	blogs.ipleiria.pt
DNS Name	email.ipleiria.pt
DNS Name	ftp.ipleiria.pt
DNS Name	intranet.ipleiria.pt
DNS Name	mail.ipleiria.pt
DNS Name	ns.ipleiria.pt
DNS Name	proxy.ipleiria.pt
DNS Name	webmail.ipleiria.pt
DNS Name	wireless.ipleiria.pt
DNS Name	www.ipleiria.pt
MX Record	ipleiria-pt.mail.protection.outlook.com
MX Record	mx01.ipleiria.pt
MX Record	mx02.ipleiria.pt
NS Record	ns.ipleiria.pt
NS Record	ns2.ipleiria.pt



DNS Name
maltego.DNSName
apple.ipleiria.pt

Weight	100
DNS Name	apple.ipleiria.pt

Incoming (1)

Domain	ipleiria.pt

NS Record
maltego.NSRecord
ns.ipleiria.pt

Weight	100
NS Record	ns.ipleiria.pt


Incoming (1)
Domain
ipleiria.pt
NS Record
maltego.NSRecord
ns2.ipleiria.pt
Weight
100
NS Record
ns2.ipleiria.pt
Incoming (1)
Domain
ipleiria.pt
DNS Name
maltego.DNSName
email.ipleiria.pt
Weight
100
DNS Name
email.ipleiria.pt
Incoming (1)
Domain
ipleiria.pt
DNS Name
maltego.DNSName
blogs.ipleiria.pt
Weight
100
DNS Name
blogs.ipleiria.pt
Incoming (1)
Domain
ipleiria.pt
DNS Name
maltego.DNSName
av.ipleiria.pt
Weight
100
DNS Name
av.ipleiria.pt
Incoming (1)
Domain
ipleiria.pt

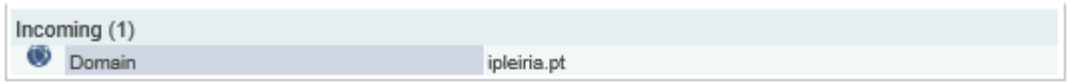

DNS Name
maltego.DNSName
wireless.ipleiria.pt
Weight 100
DNS Name wireless.ipleiria.pt
Incoming (1)
Domain ipleiria.pt


DNS Name
maltego.DNSName
webmail.ipleiria.pt
Weight 100
DNS Name webmail.ipleiria.pt
Incoming (1)
Domain ipleiria.pt

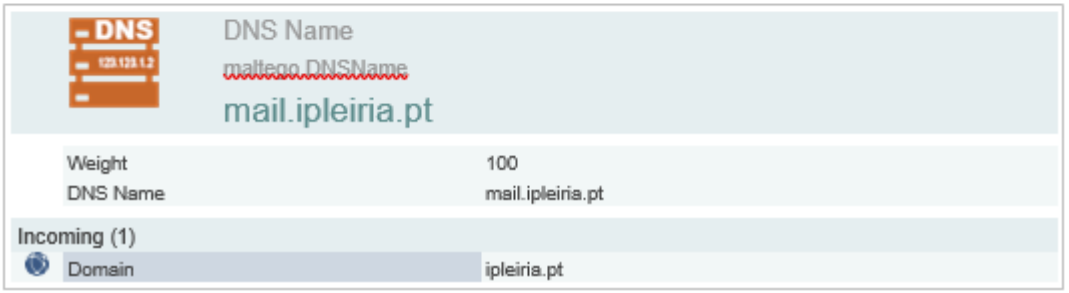

DNS Name
maltego.DNSName
proxy.ipleiria.pt
Weight 100
DNS Name proxy.ipleiria.pt
Incoming (1)
Domain ipleiria.pt


DNS Name
maltego.DNSName
intranet.ipleiria.pt
Weight 100
DNS Name intranet.ipleiria.pt
Incoming (1)
Domain ipleiria.pt


DNS Name
maltego.DNSName
ftp.ipleiria.pt
Weight 100
DNS Name ftp.ipleiria.pt

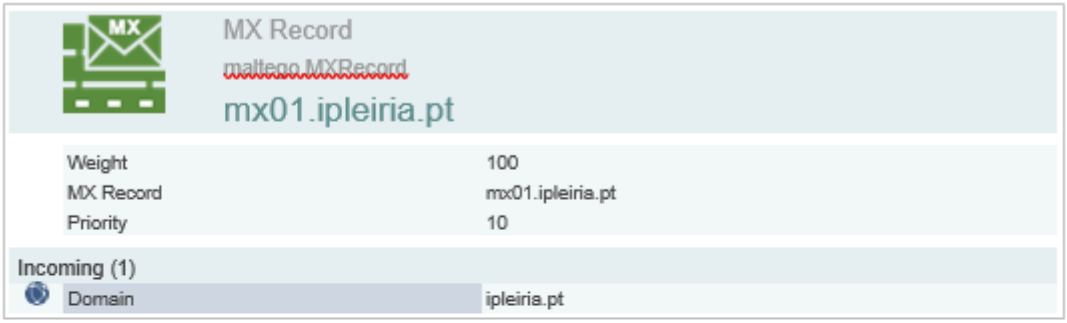

Incoming (1)
Domain
ipleiria.pt


MX Record
maltego.MXRecord
ipleiria-pt.mail.protection.outlook.com
Weight
100
MX Record
ipleiria-pt.mail.protection.outlook.com
Priority
0
Incoming (1)
Domain
ipleiria.pt


DNS Name
maltego.DNSName
mail.ipleiria.pt
Weight
100
DNS Name
mail.ipleiria.pt
Incoming (1)
Domain
ipleiria.pt


MX Record
maltego.MXRecord
mx02.ipleiria.pt
Weight
100
MX Record
mx02.ipleiria.pt
Priority
10
Incoming (1)
Domain
ipleiria.pt


DNS Name
maltego.DNSName
ns.ipleiria.pt
Weight
100
DNS Name
ns.ipleiria.pt
Incoming (1)
Domain
ipleiria.pt


MX Record
maltego.MXRecord
mx01.ipleiria.pt
Weight
100
MX Record
mx01.ipleiria.pt
Priority
10
Incoming (1)
Domain
ipleiria.pt


DNS Name
maltego.DNSName
www.ipleiria.pt
Weight
100
DNS Name
www.ipleiria.pt
Incoming (1)
Domain
ipleiria.pt

Anexo H – Spiderfoot

Retratado no estado da arte em 2.3.8.

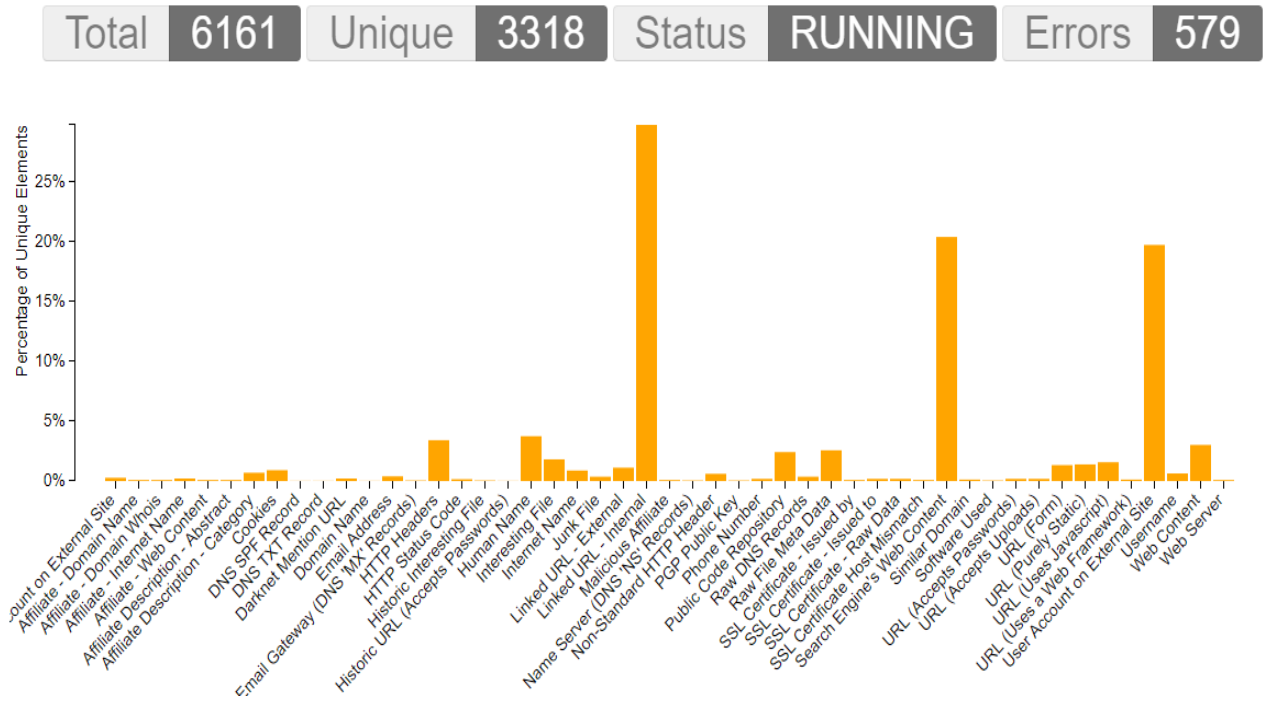


Figura 71 - Spiderfoot

A Figura 72 representa um trecho de um relatório gerado pela ferramenta Spiderfoot. Na primeira coluna conseguimos saber a data em que a analise foi realizada, na coluna 2, o módulo usado para descobrir a informação, A terceira coluna permite saber onde a ferramenta foi pesquisar a informação e na última coluna conseguimos saber a informação encontrada pelo Spiderfoot.

11-12-19 17:02	PHONE_NUMBER	sfp_phone	<!DOCTYPE html>	0 +351244829400
11-12-19 17:02	PHONE_NUMBER	sfp_phone	<!DOCTYPE html><!--[if IE 7 | IE 8]><html class="ie" lang="en-US"><![endif]--><!--[if (gte IE 9	0 +351244829471
11-12-19 17:02	PHONE_NUMBER	sfp_phone	<!DOCTYPE html><!--[if IE 7 | IE 8]><html class="ie" lang="en-US"><![endif]--><!--[if (gte IE 9	0 +351244829499
11-12-19 15:02	PHONE_NUMBER	sfp_phone	<!DOCTYPE html><html lang="en-US"> <head> <meta http-equiv="Content-Type" conte	0 +351244830010
11-12-19 15:02	PHONE_NUMBER	sfp_phone	<!DOCTYPE html><html lang="en-US"> <head> <meta http-equiv="Content-Type" conte	0 +351244845050
11-12-19 13:12	PHONE_NUMBER	sfp_phone	<!DOCTYPE html><html class="IPLeiria no-bootstrap no-js no-svg" lang="en-GB"> <head>	0 +351244845052
11-12-19 13:17	PHONE_NUMBER	sfp_phone	<!DOCTYPE html><html class="IPLeiria no-bootstrap no-js no-svg" lang="pt-PT"> <head>	0 +351244845052

Figura 72 - Spiderfoot *report*

Anexo I – Google Hacking

Anexo referente ao estado da arte no ponto 2.2.1. Neste caso a pesquisa vai pesquisar no domínio ipleiria.pt e no resultado da pesquisa tem de estar presente a palavra “Carlos Gomes”.

Figura 73 - Google Hacking

Anexo J – ViewDns

Retratado no estado da arte em 2.2.6. Neste exemplo foi usado o domínio ipleiria.pt para testar a ferramenta.

```
WHOIS Information for ipleiria.pt
==========

Domain: ipleiria.pt
Domain Status: Registered
Creation Date: 18/04/2001 00:00:00
Expiration Date: 09/11/2022 23:59:27
Owner Name: Instituto Politecnico de Leiria
Owner Address: Rua General Norton de Matos, Apartado 4133
Owner Locality: Leiria
Owner ZipCode: 2411-901
Owner Locality ZipCode: Leiria
Owner Country Code: PT
Owner Email: ipleiria@ipleiria.pt,dsi@ipleiria.pt
Admin Name: Instituto Politecnico de Leiria
Admin Address: Rua General Norton de Matos, Apartado 4133
Admin Locality: Leiria
Admin ZipCode: 2411-901
Admin Locality ZipCode: Leiria
Admin Country Code: PT
Admin Email: ipleiria@ipleiria.pt,dsi@ipleiria.pt
Name Server: ns.ipleiria.pt | IPv4: 193.137.239.226 and IPv6:
Name Server: ns2.ipleiria.pt | IPv4: 193.137.239.231 and IPv6:
```

Figura 74 - ViewDns

Anexo K – MongoDB

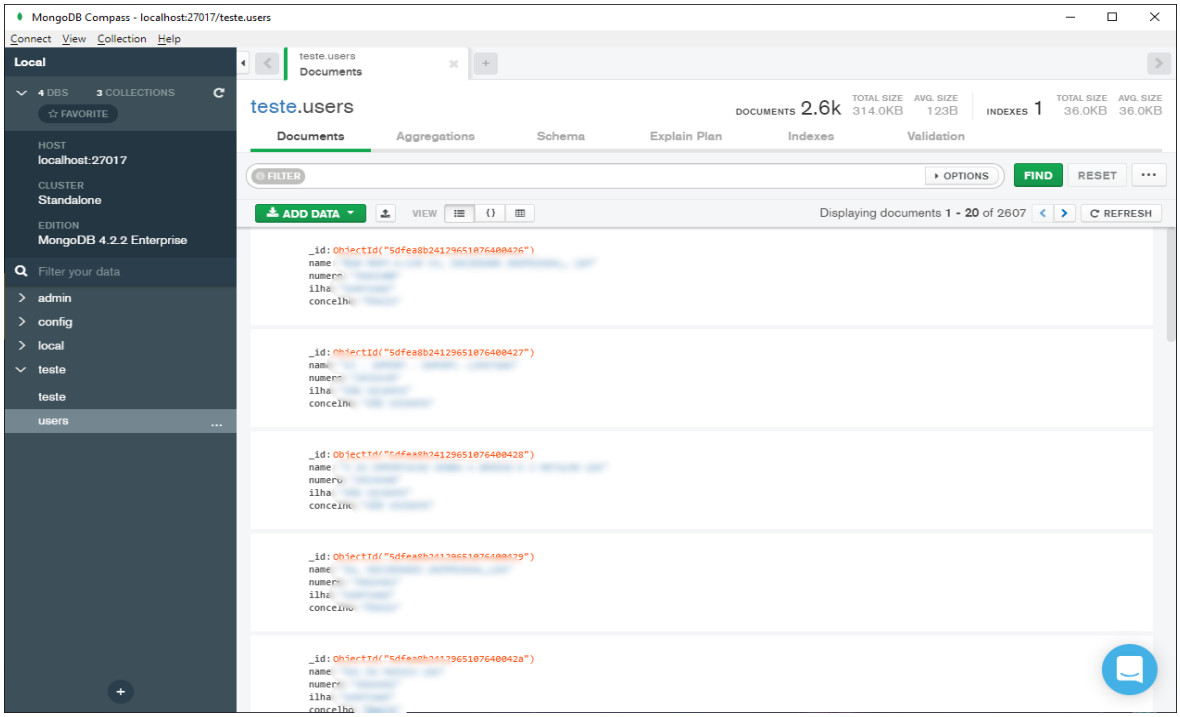


Figura 75 - MongoDB

```
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("teste");

 //console.log(nome);
    fs.readFile('dados.txt', 'utf8', function(err, data) {
    if (err) throw err;
    //console.log(data);
     var splitedData = data.split("[");
     var splitted = splitedData[1].split("}");

    for (var i = 0; i < splitted.length; i++) {
            var data = splitted[i].split('"');
            var numero = data[3];
            var nome = data[7];
            var ilha = data[11];
            var concelho = data[15];

            var user = { name: nome, numero:numero,ilha:ilha,concelho:concelho };
              dbo.collection("users").insertOne(user, function(err, res) {
                if (err) throw err;

                console.log("Inserted");
                //db.close();
              });
    }
});
```

Figura 76 - Inserir dados na base de dados MongoDB

A Figura 75 mostra os dados inseridos na base de dado MongoDB pelo código representado na Figura 76.

Anexo L - SQL

	id	name	concelho	ilha	numero
Edit Copy Delete	1	010 RENT-A-CAR CV, SOCIEDADE UNIPESSOAL, LDª	PRAIA	SANTIAGO	3561100
Edit Copy Delete	2	2J - IMPORT - EXPORT. LIMITADA	SÃO VICENTE	SÃO VICENTE	3532138
Edit Copy Delete	3	3 AS IMPORTACAO VENDA A GROSSO E A RETALHO LDA	SÃO VICENTE	SÃO VICENTE	3533420
Edit Copy Delete	4	3A, SOCIEDADES UNIPESSOAL,LDA	PRAIA	SANTIAGO	3562461
Edit Copy Delete	5	5AL DA MUSICA LDA	PRAIA	SANTIAGO	3561582
Edit Copy Delete	6	90 BISTRO	PRAIA	SANTIAGO	3577799
Edit Copy Delete	7	90 BISTRO	PRAIA	SANTIAGO	3562868
Edit Copy Delete	8	90 BISTRO LDA	PRAIA	SANTIAGO	3562929
Edit Copy Delete	9	90 BISTRO LDA	PRAIA	SANTIAGO	3576162
Edit Copy Delete	10	A & A MELICIO - ACTIVIDADES TURISTICAS LDA	PAUL	SANTO ANTÃO	3523000
Edit Copy Delete	11	A & A MELICIO - ACTIVIDADES TURISTICAS LDA	PAUL	SANTO ANTÃO	3523007
Edit Copy Delete	12	A & A MELICIO - ACTIVIDADES TURISTICAS LDA	PAUL	SANTO ANTÃO	3523009
Edit Copy Delete	13	A GRANDE MURALHA, LDA	PRAIA	SANTIAGO	3561486

Figura 77 - SQL

```
fs.readFile('dados.txt', 'utf8', function(err, data) {
    if (err) throw err;
     var splitedData = data.split("[");
     var splitted = splitedData[1].split("}");

    for (var i = 0; i < splitted.length; i++) {
            var data = splitted[i].split('"');
            var numero = data[3];
            var nome = data[7];
            var ilha = data[11];
            var concelho = data[15];

            var user = { name: nome, numero:numero,ilha:ilha,concelho:concelho };
            let sql = 'INSERT INTO Persons SET ?';
            let query = db.query(sql, user, (err, result) => {
                if(err) throw err;
                console.log(result);
                console.log("Inserted");
            });
    }
});
```

Figura 78 - Código SQL

A Figura 77 mostra os dados inseridos na base de dado SQL pelo código representado na Figura 78.

Anexo M – Neo4j

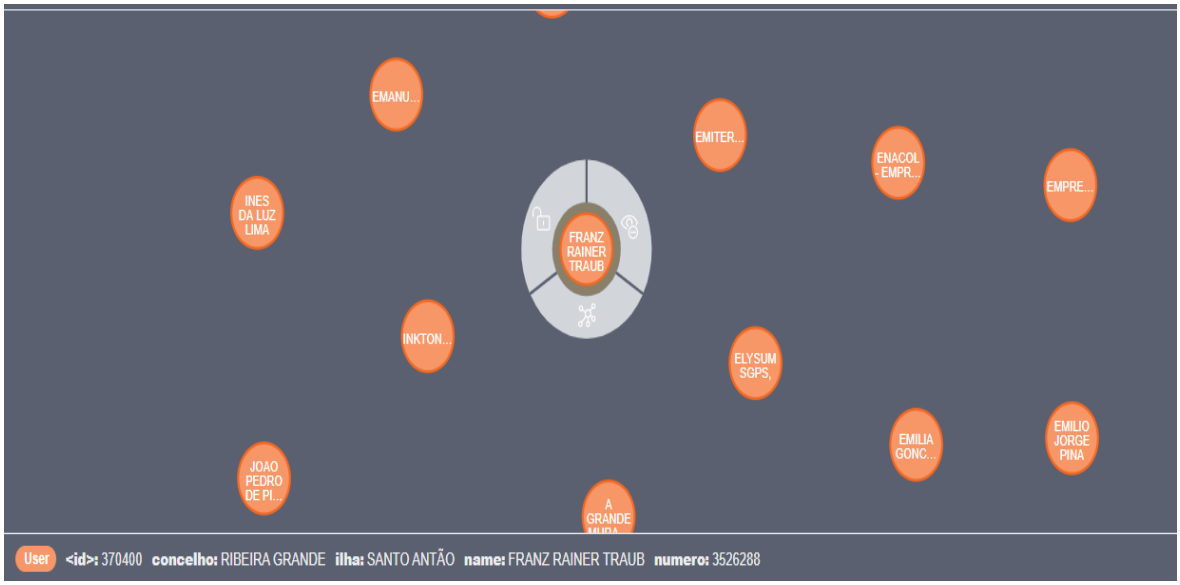


Figura 79 . Neo4j

```
const express = require ("express");
const http = require("http").Server(express);
var neo4j = require('neo4j-driver');
const fs = require('fs');
var driver = neo4j.driver('bolt://localhost',neo4j.auth.basic('neo4j','secret'));



fs.readFile('dados.txt', 'utf8', function(err, data) {
    if (err) throw err;
     var splitedData = data.split("[");
     var splitted = splitedData[1].split("}");

    for (var i = 0; i < splitted.length; i++) {
          var data = splitted[i].split('"');
          var numero = data[3];
          var nome = data[7];
          var ilha = data[11];
          var concelho = data[15];
          var session = driver.session();
          session
              .run('CREATE (n:User {name:{nameParam},numero:{numeroParam},ilha:{ilhaParam},concelho:{concelhoParam}}) RETURN n',
                   {nameParam:nome,numeroParam:numero,ilhaParam:ilha,concelhoParam:concelho})
              .then(function(result){
                  console.log("User Created");
                  session.close();
              })
              .catch(function (err){
                  console.log(err);
          });
    }
});
```

Figura 80 - Código Neo4j

A Figura 79 mostra os dados inseridos na base de dado gráfica, neste caso o neo4j, pelo código representado na Figura 80.

Anexo N - iKy

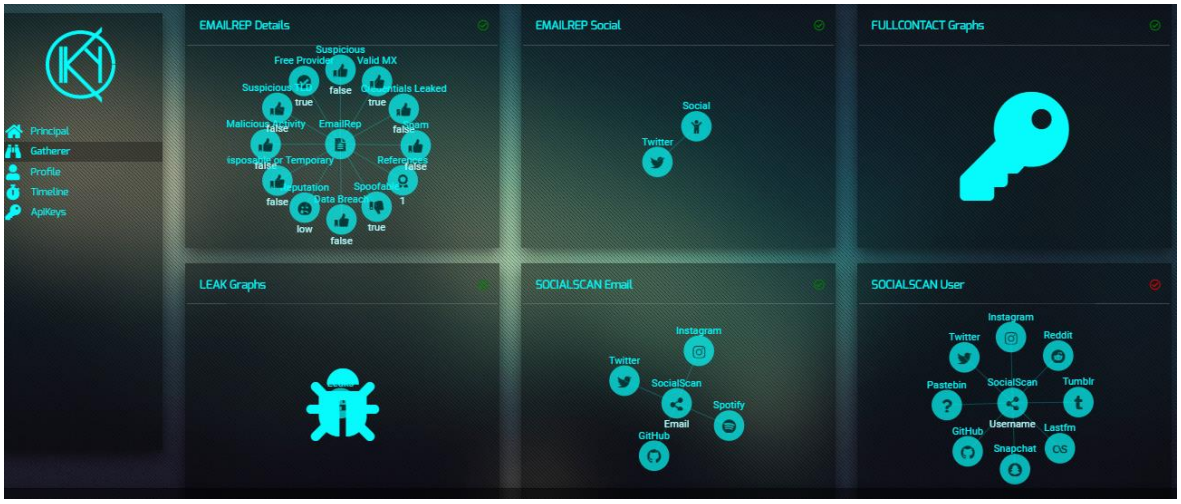


Figura 81 – iKy

A Figura 81 representa a ferramenta *I Know You*. Neste exemplo nem todos os dados são mostrados por causa da falta da chave da API do FullContact e também por falta da chave do Twitter.