Market Research: Um Sistema Automatizado para Prospeção de Clientes usando Web Crawling e Large Language Models

Marco Cunha[1][0009-0004-7097-0923] and Emanuel Silva[2][0000-0002-5046-8942]

[1,2] Instituto Superior de Engenharia do Porto
{1212243,ecs}@isep.ipp.pt

Abstract. A pesquisa de mercado manual para identificar novos clientes business-to-business é um processo demorado e trabalhoso. Este artigo apresenta o design, implementação e avaliação de um sistema automatizado de market research desenvolvido para a SISTRADE, uma empresa de consultoria de software. O sistema visa otimizar a prospeção comercial através da automação das tarefas de pesquisa e análise de potenciais clientes na web. Utiliza técnicas de web crawling para identificar e extrair dados de websites de empresas e integra-se com Large Language Models para classificar automaticamente a sua relevância com base numa pesquisa inicial. A solução adota uma arquitetura Onion e é implementada com recurso a um stack tecnológico moderno, incluindo Vue.js para o frontend, uma camada de integração em .NET/C# e um backend em Python para a lógica de processamento principal, com a persistência de dados a ser gerida pelo SQL Server. O sistema oferece uma interface intuitiva para iniciar pesquisas parametrizadas e rever um histórico completo de resultados. A avaliação através de testes unitários, de integração e de aceitação confirmou que o sistema cumpre com sucesso todos os requisitos, fornecendo uma ferramenta robusta e eficiente para a geração de leads.

Keywords: Market Research, Web Crawling, Web Scraping, Large Language Models, Classificação Automática de Empresas, Inteligência Artificial.

1 Introdução

No cenário empresarial competitivo atual, a capacidade de identificar e atrair novos clientes de forma eficiente é crucial para o crescimento. Para empresas business-to-business (B2B) como a SISTRADE, este processo, conhecido como prospeção comercial, envolve tradicionalmente uma pesquisa manual, que é frequentemente ineficiente e não escala bem com o crescimento exponencial da web. A necessidade estratégica de otimizar este processo levou ao desenvolvimento deste projeto.

O principal problema abordado é a identificação manual de novos clientes B2B. O objetivo foi criar um sistema robusto, capaz de automatizar a recolha, análise e armazenamento de dados relevantes de websites de empresas. Esta abordagem automatizada permite a análise detalhada de websites para identificar oportunidades de negócio,

descobrir informações-chave como contactos e localização da empresa e fortalecer a presença da empresa no mercado. O foco principal deste trabalho é apresentar uma solução arquitetural, mas também a integração de inteligência artificial moderna, especificamente Large Language Models (LLMs), para a classificação e extração de dados, visando aumentar a eficácia das campanhas de marketing da empresa com uma abordagem mais personalizada e estratégica.

2 Estado da arte

O desenvolvimento de um sistema automatizado de prospeção de clientes assenta na interseção de três domínios principais: o Web Crawling para a descoberta de conteúdo, o Web Scraping para a extração de dados, e a Classificação de Relevância para a análise e atribuição de relevância a esses dados. Esta secção apresenta uma análise do estado da arte nestas áreas, contextualizando a abordagem do projeto.

2.1 Descoberta de dados (Web Crawling)

A base de qualquer sistema de análise de mercado na web é a sua capacidade de obter dados de forma autónoma tal como exemplificado na Fig. 1. O Web Crawling é o processo de descoberta, onde um agente automatizado (crawler) navega pela web para encontrar novos conteúdos [1]. Existem várias estratégias, como o Focused Crawling, que direciona a navegação para domínios ou temas específicos, aumentando a pertinência dos dados recolhidos [2]. Outra abordagem também utilizada neste contexto é o Incremental Crawling, que otimiza o processo ao atualizar apenas as páginas que sofreram alterações desde a última visita, reduzindo custos computacionais.

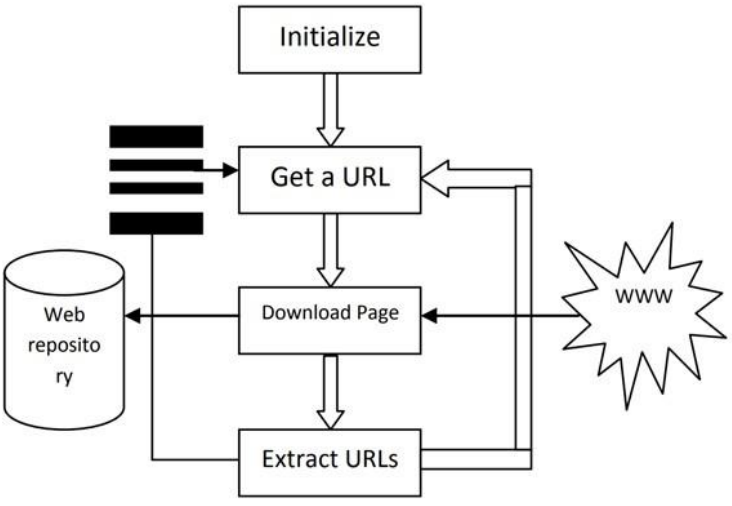

Fig. 1. – Exemplo de um processo de Web Crawling

2.2 Extração de dados (Web Scraping)

Este é o processo de extração de dados não estruturados de páginas web e a sua transformação em formatos estruturados [3]. As técnicas variam, incluindo o DOM Parsing, que utiliza um navegador embutido para interagir com páginas dinâmicas, e o Uso de Expressões Regulares (Regex Matching), que se baseia na identificação de padrões de texto específicos no código da página para extrair os dados desejados [4]. Embora alguns websites disponibilizem APIs que permitem o acesso direto e estável aos seus dados de forma controlada, essas interfaces nem sempre abrangem todas as informações visíveis no site ou podem impor restrições de uso. O web scraping oferece maior flexibilidade, permitindo a extração de dados de praticamente qualquer página da web.

2.3 Classificação de Relevância com Inteligência Artificial

A classificação de registos é o processo que atribui uma pontuação de relevância aos dados. Técnicas como a Aprendizagem Supervisionada, que utiliza algoritmos como Support Vector Machines (SVM) treinados com dados previamente rotulados para aprender padrões, demonstram alta precisão nestes processos.

No entanto, a vanguarda da classificação de textos é atualmente representada pelos LLMs. Utilizando arquiteturas transformer, os LLMs conseguem compreender nuances linguísticas e contextos complexos com uma capacidade sem precedentes [5]. As abordagens baseadas em IA, demonstram resultados consistentemente superiores em precisão e adaptabilidade, quando comparadas com as abordagens mais tradicionais, tornando-se o padrão para sistemas avançados de classificação de registos [4].

2.4 Ferramentas e Trabalhos Relacionados

Existem no mercado ferramentas que abordam partes isoladas deste problema. O BuiltWith[1], por exemplo, é útil para visualizar informações de contacto e outros dados relevantes de uma empresa presentes num website, para além de identificar as tecnologias que este utiliza [6]. Já o Firecrawl [2]especializa-se num workflow de crawling e scraping otimizado para extrair conteúdo limpo, pronto a ser processado por LLMs [7]. O sistema desenvolvido neste projeto diferencia-se por integrar estas várias funcionalidades num workflow coeso e de ponta a ponta, adaptado às necessidades específicas da prospeção de clientes.

2.5 Síntese Comparativa

O Web Crawling trata da descoberta eficiente de páginas, enquanto o Web Scraping se dedica à extração estruturada dos seus dados. A Classificação de Relevância, hoje dominada por modelos de IA e LLMs, avalia a importância desses dados com elevada

[1] https://builtwith.com/
[2] https://www.firecrawl.dev/

precisão. Ferramentas existentes cobrem apenas partes do processo, ao passo que o sistema proposto integra todas as etapas num único workflow.

3 Arquitetura e design do sistema

Para garantir a escalabilidade, manutenibilidade e uma clara separação de responsabilidades, o design do sistema segue os princípios da Onion Architecture. Este estilo arquitetural enfatiza uma abordagem domain-centric, colocando a lógica de negócio principal no centro e garantindo que as camadas externas dependem das camadas internas, e não o contrário [8]. A arquitetura é representada usando uma abordagem do modelo C4 para descrever o sistema em diferentes níveis de abstração.

3.1 Visão de alto nível do sistema

A um nível alto de granularidade (C4 Nível 2), o sistema é decomposto em quatro containers principais, como mostrado na Fig. 2. Esta estrutura isola responsabilidades e facilita o desenvolvimento e deployment independentes.

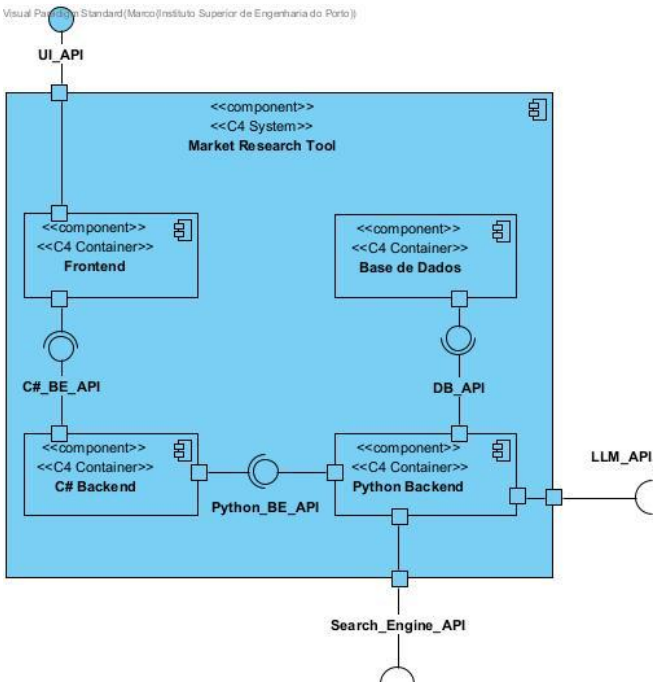

Fig. 2. - Arquitetura de Alto Nível do Sistema

Frontend: Uma single-page application (SPA) em Vue.js que fornece a interface gráfica para os utilizadores configurarem e iniciarem pesquisas, bem como visualizarem os resultados.

C# Backend: Este container atua como uma camada de passthrough ou ponte. Recebe pedidos do frontend e encaminha-os para o backend Python. Este design foi um requisito específico para garantir a compatibilidade e integração futuras com o sistema ERP existente da empresa, baseado em .NET.

Python Backend: Este é o núcleo do sistema, responsável por toda a lógica de processamento principal. Gere os pedidos de pesquisa iniciais, executa os processos de web crawling e scraping, comunica com APIs externas (Search Engines, LLMs) e interage com a base de dados.

Base de Dados: Uma instância do Microsoft SQL Server é usada para a persistência de dados. Armazena toda a informação relacionada com as pesquisas, incluindo os parâmetros definidos pelo utilizador, o estado e os dados extraídos e classificados das empresas.

3.2 Lógica Principal: Python Backend

O Backend Python está estruturado internamente de acordo com as camadas da Onion Architecture (Fig. 3), garantindo que a lógica principal permaneça independente de frameworks e tecnologias externas.

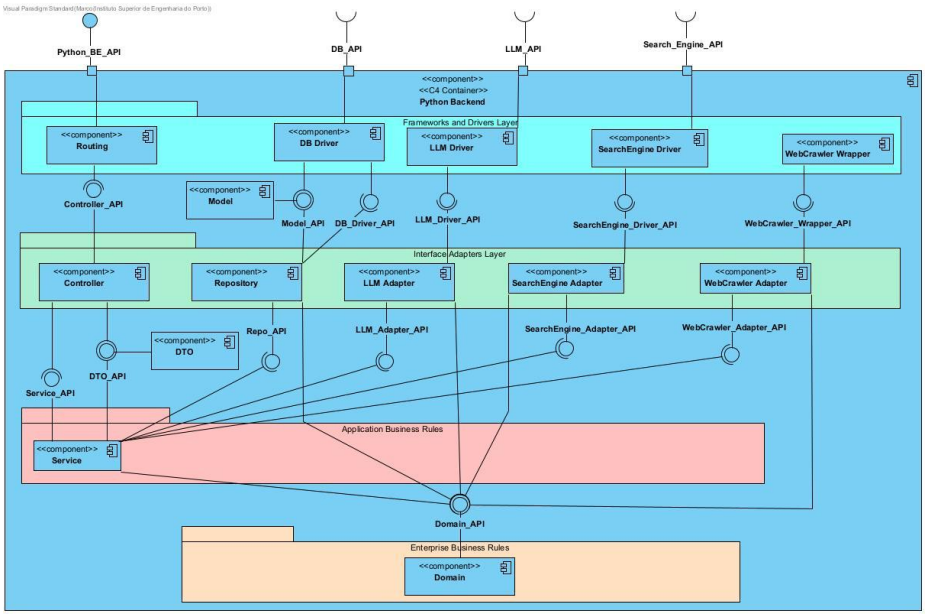

Fig. 3. - Arquitetura em Camadas do Backend Python

Enterprise Business Rules (Dominio): No núcleo encontram-se as entidades de domínio, que encapsulam a lógica de negócio e as regras essenciais da aplicação.

Application Business Rules (Serviços): Esta camada contém os serviços específicos da aplicação que orquestram as funcionalidades, como iniciar uma nova pesquisa ou

obter o histórico de pesquisas. Depende da camada de domínio e define interfaces para as camadas externas.

Interface Adapters: Esta camada adapta os dados do formato mais conveniente para as agências externas (como a base de dados ou APIs web) para o formato mais conveniente para as camadas internas (domínio e serviços). Inclui controladores, repositórios e adaptadores para serviços externos como o LLM e o web crawler.

Frameworks and Drivers: A camada mais externa consiste nas tecnologias que interagem com o mundo exterior, como a web framework, os drivers da base de dados e os wrappers para bibliotecas externas como o Scrapy.

4 Metodologia de implementação

A funcionalidade do sistema é construída em torno de um workflow de várias etapas que transforma uma simples query do utilizador numa lista de potenciais clientes classificados. Este processo é suportado por um stack tecnológico cuidadosamente selecionado.

4.1 Workflow de Aquisição e processamento de Dados

O processo principal começa quando um utilizador inicia uma pesquisa parametrizada e termina com o armazenamento dos dados classificados. As etapas-chave são:

- Obtenção Inicial de URLs: Com base na query do utilizador (ex: "empresas de artes gráficas em Portugal"), o sistema consulta um ou mais motores de busca (ex: Google, Bing) para obter uma lista de URLs iniciais. Isto é conseguido através de APIs oficiais ou simulando a interação de um utilizador;
- Web Crawling: O sistema usa a framework Scrapy para realizar um crawl focado a partir das URLs iniciais. O crawler é configurado para navegar dentro dos domínios permitidos e identificar páginas relevantes, como páginas "Sobre Nós" e "Contactos", que provavelmente contêm informações empresariais valiosas;
- Extração de Dados: Uma vez identificadas as páginas relevantes, o sistema faz o scrape do seu conteúdo para extrair dados não estruturados, incluindo nome da empresa, emails de contacto, números de telefone, moradas e texto descritivo;
- Classificação baseada em LLM: A informação extraída de cada website é compilada e enviada para um LLM interno da empresa através de uma API. O LLM é instruído a analisar o conteúdo no contexto da query original do utilizador e a devolver uma pontuação de relevância (0-100), juntamente com dados limpos e estruturados (ex: uma lista de emails válidos);
- Persistência de dados: Os dados processados e classificados de cada empresa são armazenados na base de dados e ligados ao histórico da pesquisa inicial para recuperação posterior.

4.2 Stack Tecnológico

A seleção das tecnologias baseou-se tanto no ecossistema existente na SISTRADE como nos requisitos específicos do projeto.

Frontend: O Vue.js foi escolhido pela sua baixa curva de aprendizagem e arquitetura baseada em componentes, ideal para criar interfaces de utilizador reativas e de fácil manutenção.

Backend: Foi utilizada uma abordagem com dois backends. O .NET (C#) serve como uma ponte segura para os sistemas internos da empresa. O Python foi selecionado para a lógica principal devido ao seu vasto ecossistema de bibliotecas para web crawling e análise de dados, destacando-se a framework Scrapy.

Base de Dados: O Microsoft SQL Server foi utilizado por ser o sistema de gestão de base de dados padrão na empresa.

5 Resultados e Avaliação

O sistema implementado fornece uma interface web totalmente funcional e foi rigorosamente testado para garantir que cumpria todos os objetivos do projeto.

5.1 Interface e Funcionalidade do Sistema

O frontend apresenta aos utilizadores duas funcionalidades principais: (1) iniciar uma nova pesquisa e (2) visualizar o histórico de pesquisas anteriores. O painel de pesquisa (Fig. 4) permite aos utilizadores definir a query de pesquisa e parâmetros como a profundidade do crawl e que motores de busca utilizar.

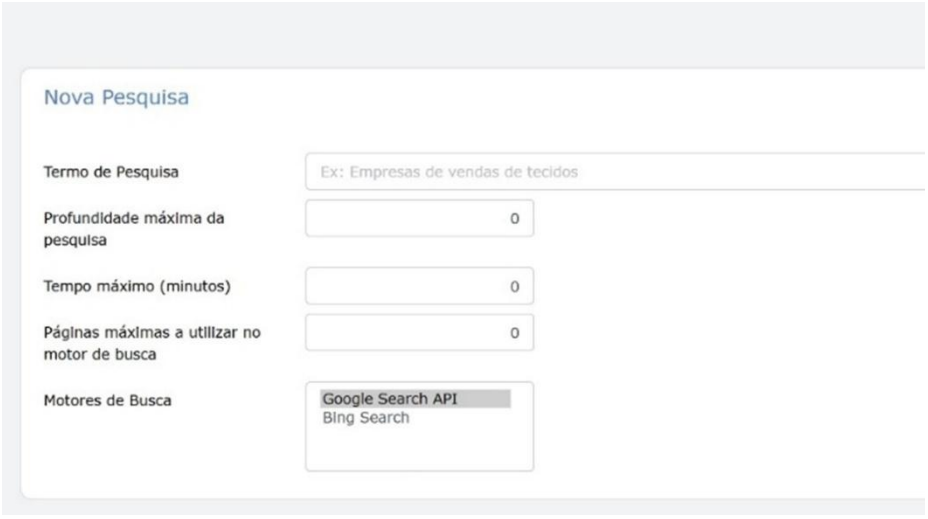

Fig. 4. - Interface do Utilizador (nova pesquisa)

Assim que a pesquisa é concluída, os resultados são exibidos na secção de histórico. Os utilizadores podem ver uma lista das empresas encontradas, cada uma com a sua pontuação de relevância calculada (Fig. 5). Clicar num resultado revela informações detalhadas, incluindo o URL do website da empresa, emails, números de telefone e moradas extraídos. Esta interface também inclui opções de filtragem avançada para o histórico de pesquisas.

Fig. 5. - Interface do Utilizador (histórico de pesquisa)

5.2 Validação do Sistema

A qualidade e correção do sistema foram validadas através de uma estratégia de testes multinível:

- Testes Unitários: Foram escritos testes automatizados em Python para verificar componentes individuais de forma isolada, alcançando uma cobertura de código de 94%. Foram usados Mocks para simular dependências externas, como a base de dados e a API do LLM;
- Testes de Integração: A ferramenta Postman foi utilizada para testar as interações entre o frontend, o backend C# e o backend Python, garantindo que os endpoints da API se comportavam como esperado sob várias condições (ex: filtros combinados);
- Testes de Aceitação: Foram realizadas sessões de teste semanais com os supervisores da empresa para validar que a funcionalidade da aplicação cumpria os requisitos de negócio e as expectativas dos utilizadores.

A avaliação concluiu que todos os objetivos pré-definidos foram totalmente alcançados. O sistema demonstrou um desempenho eficiente e fiável, integrando com sucesso todas as tecnologias especificadas.

6 Conclusões e trabalho futuro

Este artigo apresentou um sistema automatizado de pesquisa de mercado projetado para otimizar a prospeção de clientes. O desenvolvimento e a implementação foram bem-sucedidos, resultando numa aplicação funcional que responde às necessidades identificadas pela SISTRADE. O projeto demonstra a viabilidade técnica da combinação de técnicas de web crawling com Large Language Models para criar ferramentas robustas de business intelligence.

Uma limitação significativa do sistema atual é a sua dependência de uma API de LLM interna que só consegue processar um pedido em paralelo. Isto limita a escalabilidade do sistema, tornando-o adequado principalmente para uso individual, em vez de um uso concorrente e intensivo.

O desenvolvimento futuro focar-se-á em melhorar a experiência do utilizador e a eficiência do sistema. As melhorias planeadas incluem a implementação de uma funcionalidade para interromper pesquisas em andamento; permitir que os utilizadores refaçam uma pesquisa em websites específicos; adicionar uma funcionalidade para gerar automaticamente rascunhos de emails de contacto com base no termo de pesquisa e nos dados da empresa; e introduzir um sistema de notificação para alertar os utilizadores quando pesquisas mais longas terminarem. Estas melhorias tornarão a aplicação mais interativa, escalável e alinhada com as necessidades dinâmicas dos seus utilizadores.

Agradecimentos. O autor gostaria de agradecer à SISTRADE - Software Consulting, S.A. pela oportunidade de estágio e por proporcionar um ambiente acolhedor e à equipa de supervisão, pela sua orientação inestimável, feedback e apoio fundamental ao longo deste projeto.

References

1. Khder, M.: Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. International Journal of Advances in Soft Computing and its Applications 13(3), 145–168 (2021)
2. M. Ahuja, J. Singh, and V. nica, “Web Crawler: Extracting the Web Data,” *International Journal of Computer Trends and Technology*, vol. 13, pp. 132–137, Jul. 2014, doi: 10.14445/22312803/IJCTT-V13P128.
3. Sandeep Shreekumar, Satyan Mundke, and Murlidhar Dhanawade, “IMPORTANCE OF WEB SCRAPING IN E- COMMERCE BUSINESS,” *NCRD’s Technical Review : e-Journal*, vol. 7, no. 1, 2022.
4. Manushi Weerasinghe, “Enhancing Web Scraping with Artificial Intelligence,” 2024.
5. Kostina, A., Dikaiakos, M., Stefanidis, D., Pallis, G.: Large Language Models For Text Classification: Case Study And Comprehensive Review. arXiv:2501.08457 (2025)
6. BuiltWith Technology Lookup. https://builtwith.com/, last accessed 2025/10/21
7. Firecrawl Playground. https://www.firecrawl.dev/playground, last accessed 2025/10/21
8. Martin, R.C.: Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall (2017)