

KR7 PROSPECT - Software Para Gestão De Leads e Equipes de Vendas Diretas utilizando Georreferenciamento

Douglas Kramer ¹, Rogerio Xavier Azambuja ¹

¹Instituto Federal do Rio Grande do Sul (IFRS)
Campus Farroupilha, Farroupilha – RS – Brazil

kramerdouglas@hotmail.com, rogerio.xavier@farroupilha.ifrs.edu.br

Resumo. Os indicadores de desempenho e produtividade são cruciais para avaliar as contribuições dos colaboradores e compreender o seu impacto no sucesso organizacional. Para otimizar essas métricas, as empresas devem adotar práticas estratégicas e baseadas em dados. Este trabalho tem como objetivo desenvolver um software para registro, monitoramento e gestão de equipes de vendas diretas. O mercado de serviços atualmente é muito competitivo, manter uma base de clientes ativa e fornecer acesso contínuo aos serviços é vital para o crescimento sustentável. Embora o fechamento de pedidos continue sendo uma métrica fundamental, outros fatores como o desempenho dos funcionários, a identificação de novos clientes potenciais e a geração de Leads (contatos) qualificados – indivíduos genuinamente interessados em um produto ou serviço – são igualmente importantes. Ao integrar essas diversas métricas, as empresas podem avaliar melhor a produtividade da equipe, aprimorar suas estratégias de vendas e estabelecer um funil de vendas mais resiliente e sustentável.

Palavras-chave: Leads, Funil de vendas, Georreferenciamento

Abstract. Performance and productivity indicators are crucial for evaluating employee contributions and understanding their impact on organizational success. To optimize these metrics, companies must adopt strategic and data-driven practices. This study aims to develop a software to register, monitor and manage direct sales teams. In today's highly competitive services market, retaining an active customer base and providing seamless access to services is vital for sustainable growth. While closing sales remains a cornerstone metric, other factors like employee performance, identifying new prospects, and generating qualified Leads (contacts) — individuals genuinely interested in a product or service — are equally important. By integrating these diverse metrics, businesses can better evaluate team productivity, enhance their sales strategies, and establish a more resilient and sustainable sales funnel.

Key-words: Leads, Sales funnel, Georeferencing

1. Introdução

O canal de venda direta tem como um dos benefícios o baixo investimento já que não necessita de um espaço físico, porém devido a alta rotatividade de trabalho e baixa produtividade dos colaboradores, a falta de registros de clientes interessados prejudica a avaliação de resultados e a entrada de um funil de vendas. O canal abordado no contexto

deste artigo é o canal de venda direta ou PAP (porta a porta), onde as empresas criam equipes de vendas para encontrar novos clientes diretamente em suas casas sem a intermediação de lojas (Sebrae, 2024). As vendas diretas de produtos e serviços foram responsáveis pelo faturamento de R\$ 47 bilhões no Brasil em 2023, segundo a ABEVD (Associação Brasileira de Empresas de Vendas Diretas). Segundo dados desta associação o Brasil é o sétimo colocado mundial e líder entre os países da América Latina.

O Brasil historicamente possui uma alta taxa de rotatividade de trabalho entre as nações com médias comparáveis. Um estudo comparativo entre alguns países quanto aos números de trabalhadores com menos de 2 anos de serviço na mesma empresa aponta que o Brasil possui a maior média, sendo 47% dos entrevistados (Gonzaga, 2014). A rotatividade é um tema essencial em pesquisas sobre economia e trabalho, devido ao seu elevado índice no país e às consequências preocupantes para a produtividade dos trabalhadores. Por conta desse alto índice, as relações de trabalho tendem a ser mais curtas, levando os empregados a se comprometerem menos com suas tarefas e os empregadores a investirem menos na capacitação e desenvolvimento de seus colaboradores (Ferreira, 2018). Possuir ferramentas adequadas, utilizar tecnologia para acompanhar a produtividade, possuir métricas para avaliar o comprometimento com as tarefas e metas estabelecidas é essencial, e pode ser um diferencial competitivo para uma empresa.

Segundo Campos (2018), o trabalho no Brasil possui baixos níveis de produtividade, e também apresentou uma baixa evolução entre o final da década de 1990 e meados de 2010. A qualificação dos colaboradores é essencial para melhorar os resultados, qualificações essas que podem ser adquiridas fora da empresa com incentivos à educação como também internamente com treinamentos contínuos, melhoria de processos e utilização de tecnologias para o desempenho de algumas atividades. Ter bons indicadores de produtividade ou KPIs (Key Performance Indicator) é muito importante para medir o desempenho de uma empresa, equipe, colaborador ou de um negócio. Os KPIs podem ser utilizados para montar estratégias e campanhas, avaliar os resultados, com isso fazer um comparativo com assertividade de cada estratégia montada e fazer melhorias contínuas nas etapas com o intuito de melhorar o desempenho e produtividade da empresa ou processo (Sydle, 2024).

Este trabalho tem como objetivo desenvolver um software para registro, monitoramento e gestão de equipes de vendas diretas, e através do software fornecer dados considerados importantes para a gestão de equipes como também verificar de forma visual quais regiões possuem maior potencial de vendas ou maior número de clientes interessados no produto ou serviço da empresa em questão. Os objetivos específicos do software são:

- Salvar localização e informações relevantes de forma eficiente e prática;
- Apresentar no mapa os *Leads* de cada representante;
- Disponibilizar uma página estilo *dashboard* para apresentar relatórios mensais;
- Apresentar uma tabela comparativa de resultados com novos *Leads*, pedidos, visitas e receita gerada por cada representante;
- Apresentar e agrupar em um mapa a localização de todos os *Leads* captados;

No restante do artigo serão apontados dados, conceitos e metodologias utilizadas no desenvolvimento do KR7 Prospect. No referencial teórico será apresentada a origem das relações comerciais, o conceito de venda direta, métricas nacionais e internacionais deste setor, o conceito de funil de vendas e a justificativa do seu uso, algumas análises sobre marketing e geomarketing. Em sequência do referencial teórico serão apresentadas as ferramentas utilizadas para o desenvolvimento do sistema, linguagens de programação, *frameworks* e alguns padrões de desenvolvimento. Na etapa de desenvolvimento serão apresentados requisitos do sistema, trechos do código fonte de funcionalidades consideradas essenciais e como os requisitos foram implementados. A etapa de desenvolvimento também apresenta algumas interfaces do software, explicações de funcionalidades, resumo das funcionalidades e rotas da API (*Application Programming Interface*), com entradas e saídas de dados separados por entidade. Após a apresentação do desenvolvimento são apresentados os resultados obtidos com o desenvolvimento do software e por último as considerações finais e referências utilizadas.

2. Referencial Teórico

2.1 Origem do Comércio

O comércio surgiu segundo Pereira (2024) após o início das atividades agrícolas, em um primeiro momento os agricultores faziam trocas entre si de produtos e alimentos, essa troca é denominada escambo. Conforme o avanço das atividades agrícolas ocorreu a especialização de algumas comunidades e a produção de excedentes. Com o surgimento da tecnologia de manufatura dos metais, objetos metálicos começaram a ser utilizados nas trocas. Neste período surgiram as primeiras moedas e seu valor era determinado pelo tipo de material da moeda, mas logo passaram a representar valores mais abstratos para facilitar as transações comerciais (Pereira, 2024).

2.2 Vendas diretas B2C (*Business to Consumer*)

Vendas diretas B2C é o termo utilizado para o método de vendas que não utiliza intermediários. Este método possui apenas o consumidor, o vendedor/representante e a marca sem necessitar de uma estrutura física ou ponto de venda. Ao longo dos séculos o universo das vendas evoluiu, se diversificou, ganhou novas e avançadas ferramentas de trabalho. No setor de vendas diretas pode-se citar os casos de sucesso de marcas de cosméticos, que segundo dados da ABEVD, representaram mais de 40% do volume total de vendas direta no Brasil. Muitas empresas ainda consideram essencial o efeito de um bom vendedor, que através das técnicas consegue apresentar o produto, captar a atenção e criar empatia com o cliente e ajudá-lo no processo de decisão de compra. Por este e outros motivos muitas empresas ainda utilizam este o canal de venda. Podemos citar como uma das vantagens a margem de lucro maior e o baixo investimento inicial (Pipedrive, 2024).



Figura 1: Vendas Diretas Globais em Bilhões de Dólares

Fonte: Imagem do site ABEVD, 2024

A ABEVD divulga anualmente os números do setor no país. Em 2023, foram comercializados R\$ 47 bilhões de reais em produtos e serviços por venda direta. Um fato bastante apresentado pela entidade é que o Brasil detém a sétima colocação mundial em vendas diretas e é líder na América Latina, conforme o anúncio da Federação Mundial das Associações de Vendas Diretas (World Federation of Direct Selling Associations – WFDSA), da qual a ABEVD é associada. A Figura 1 apresenta o ranking com a lista das empresas com maior volume em Bilhões de Dólares (ABEVD, 2024).

2.3 Funil de vendas

2.3.1 Origem do funil de vendas

O termo funil de vendas representa as etapas de um processo desde a captação de clientes com interesse até o fechamento de negócios. Esse processo foi criado no intuito de orientar profissionais de vendas e utiliza-se até hoje (Gomes, 2024). Esta divisão de etapas teve origem a partir do método desenvolvido pelo publicitário Elías Elmo Lewis, em 1898 apresentado por Kellogg 1925 (Gomes, 2024). O acrônimo AIDA significa “Atenção”, “Interesse”, “Desejo”, “Ação” como mostra a Figura 2. Esse processo foi criado no intuito de orientar profissionais de vendas e utiliza-se até hoje.

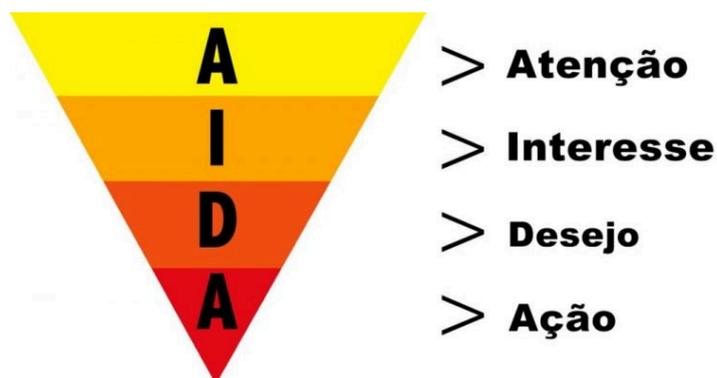


Figura 2: Funil de vendas AIDA
 Fonte: Imagem do site Agendor, pelo escritor Gustavo Gomes, 2024.

2.3.2 O que é um funil de vendas?

O artigo de Gomes (2024) demonstra que o funil de vendas passou por diversos processos de desenvolvimento, outros autores contribuíram para a consolidação do termo. Analisando os diversos materiais disponíveis na literatura e na internet, pode-se concluir que de modo geral trata-se da divisão de processos de venda de produto ou serviço. O padrão apontado pelos autores é a entrada, meio e fundo do funil. Baseado nessas etapas podem-se adicionar subdivisões conforme o segmento e produto. Pode-se encontrar funis com diversas etapas, cada empresa ou representante pode desenvolver uma metodologia adequando as etapas ao tipo de produto ou serviço que comercializa. Entende-se que a entrada é a abordagem ou apresentação do produto com o intuito de gerar interesse do cliente. O meio do funil utiliza-se para apresentar números, ativar gatilhos mentais e criar a necessidade e desejo pelo artefato apresentado. A etapa final do funil trata-se do fechamento do pedido, finalização da compra ou assinatura do serviço. Alguns autores também adicionam a etapa de pós-vendas no final do funil.



Figura 3: Funil de vendas
 Fonte: Imagem do site Agendor, pelo escritor Gustavo Gomes, 2024.

A Figura 3 apresenta um modelo de funil de vendas mais atualizado com as demandas atuais, no topo do funil pode-se observar o item mercado alvo, trata-se do perfil de cliente que utilizará produto ou serviço. No setor de vendas diretas pode-se enviar equipes para determinadas regiões fazer um trabalho de divulgação, pesquisa e panfletagem, conforme o retorno dos clientes abordados deve-se tomar decisões a respeito de rotas. Se em uma determinada região após o trabalho de divulgação não apresentou interesse no artefato apresentado, não houve captação e *leads*, deve-se trabalhar em outras regiões.

O segundo item da Figura 3, “visitantes” pode não se aplicar totalmente a vendas diretas, pois não há um espaço físico ou lojas, por outro lado pode ser aplicado se a empresa utilizar mídias sociais ou site como também utilizar pontos de vendas. O item “*prospects*” é a primeira métrica do funil de vendas, também chamado de *lead* é o cliente que demonstrou interesse no produto ou serviço, quanto maior a quantidade de *prospects* maior a quantidade de vendas. Os próximos itens deste funil são os possíveis compradores que após atender a alguns requisitos de compra específicos, realmente tem possibilidade de se tornar um comprador, e após efetuar a compra se tornam clientes.

2.3.3 Justificativa de uso de funil de vendas

Através do funil de vendas pode se desenvolver processos mais assertivos de vendas utilizando métricas adequadas para aferição de desempenho de equipes de vendas, como também pontos de vendas. Utilizar um funil com um processo e metodologia claro pode ajudar identificar falhas no processo de vendas, apontar onde as melhorias e treinamentos devem ser feitas, identificar e aperfeiçoar os serviços e produtos (Gomes, 2024). Se por exemplo a empresa fecha um negócio ou uma venda a cada dez clientes que o produto é apresentado, e a meta da empresa para o consultor ou representante é vinte, então pode-se chegar a conclusão que ele deve apresentar no mínimo duzentos clientes para alcançar o objetivo.

2.4 Marketing e Geomarketing

2.4.1 - Marketing

O marketing envolve a identificação e satisfação das necessidades humanas e sociais de forma que estejam alinhadas com os objetivos das organizações. Os autores apontam que a Associação Americana de Marketing define o marketing como a atividade, o conjunto de conhecimentos e processos de desenvolvimento, comunicação, entrega e troca de ofertas que possuam valor para consumidores, clientes, parceiros e a sociedade como um todo. A administração de *marketing* ocorre quando uma das partes em uma relação de troca potencial, procura meios de obter a resposta desejada da outra parte (Kotler, Keller, Chernev, 2024). Levando em conta o contexto de aplicação do KR7 Prospect, o conceito de *marketing* apresentado e a metodologia de funil de vendas, o *marketing* entra nas etapas iniciais do funil de vendas, na pesquisa de público alvo, na abordagem e apresentação do produto ou serviço, como também nas etapas do meio do funil, utilizando perguntas adequadas para criar a necessidade e desejo no cliente, com intuito do fechamento do negócio.

Segundo os autores Kotler, Keller, Chernev (2024) pode-se estabelecer definições distintas ao termo *marketing* do ponto de vista social e gerencial. Utilizando a perspectiva social, alguns autores afirmam que o papel do *marketing* é “proporcionar uma vida melhor”. Para Kotler, Keller, Chernev, (2024) a definição social é que trata-se de um processo pelo qual indivíduos e grupos obtêm o que necessitam e desejam por meio da criação, da oferta e da livre troca de bens e serviços de valor entre si. Do ponto de vista gerencial muitas vezes o *marketing* é descrito como “a arte de vender produtos”. Os autores apontam que a venda é o resultado do *marketing*, mas que o principal papel do marketing não é vender. Para Drucker citado em Kotler, Keller, Chernev, (2024) sempre haverá a necessidade de vender, mas o papel do marketing é conhecer e entender tão bem as necessidades do cliente de modo que o bem ou serviço se venda sozinho. No contexto de vendas diretas pode-se adaptar este conceito na abordagem e apresentação dos produtos, antes de apresentar os produtos e serviços disponíveis, fazer uma breve pesquisa com o cliente e entender as reais necessidades dele, e após isso apresentar o produto e serviço que mais adequado para para ele.

O *marketing* atual apresenta uma realidade diferente do que a outras épocas, os consumidores possuem mais potência na palma da mão. O acesso à informações permite que os clientes possam fazer melhores escolhas, comparar preços, características de produtos e serviços. Neste contexto é fundamental desenvolver estratégias, realizar planejamento e definir objetivos claros para conseguir apresentar, engajar, criar desejo e percepção de valor em serviços e produtos. Para atingir as metas estabelecidas, é necessário construir estratégias precisas para cada público-alvo, posicionamento e realizar investimentos apropriados. Os dados de uma pesquisa realizada demonstra que dois terços dos entrevistados relatam não gostar de publicidade. Por este motivo, esses clientes podem ser menos tolerantes à campanha, pular comerciais, esquivar-se de campanhas de marketing via correio ou telefone (Kotler, Keller, Chernev, 2024). Por este motivo muitas empresas recorrem a outras metodologias de vendas, utilizando PAP ou vendas diretas, nestes métodos há um contato real com o cliente, e há uma oportunidade de realmente entender a necessidade do cliente e fazer uma apresentação de um produto ou serviço.

2.4.2 - O que é Geomarketing

A análise de localização identifica as oportunidades de mercado, hábitos de compra, além de desejos e necessidades de clientes potenciais e futuros. A tecnologia do GPS (*Global Positioning System*) pode identificar a localização exata dos consumidores, a oferta através da localização é mais atrativa pois o produto ou serviço já está disponível para aquele cliente, ou há um ponto de venda próximo (Kotler, Keller, Chernev, 2024). Para Aragão (2005, citado em Mangini 2014) geomarketing é um termo utilizado no gerenciamento de dados de natureza geográfica com o intuito de apoiar o processo de escolha baseado no entendimento e compreensão do mercado, identificando as características de uma determinada região e fornecendo informações para uma análise do potencial socioeconômico. Cases de sucesso como Netflix, Amazon, Google ou Alibaba já utilizam a localização em seus algoritmos para alimentar sistemas de recomendações para clientes (Kotler, Keller, Chernev, 2024).

Aranha e Figoli (2004), como citado em Grassi (2010) apontam que o termo teria nascido nos Estados Unidos no início do século passado, e que possui relacionamento com as teorias de localização urbana empreendidas por Applebaum, quando criou técnicas quantitativas para escolha de possíveis pontos comerciais. Para Melo (2003, citado em Mangini 2014) geomarketing é uma área de aplicação focada no tratamento e análise de dados relacionados à geografia, que possibilita a integração, organização e manipulação de dados referentes a usuários e análises, considerando um ponto de vista geográfico. Ele também aponta que a principal função é a análise e visualização em mapas físicos ou virtuais de dados de mercado, cruzando com informações de localização, para campanhas de marketing e vendas.

O Sebrae (2023) aponta algumas vantagens do geomarketing como: alto poder de segmentação, aumento do ROI (*Return on Investment*) no *marketing*, direcionar as estratégias, melhora na geração de *leads* personalizando a comunicação, melhora no relacionamento com os clientes, pode-se enviar mensagens mais convenientes tendo conhecimento de onde está o cliente e como ele se comporta em cada local, apoio à gestão comercial, alinhar a força de vendas e ajustar o mix de produtos por região. O KR7 Prospect através da coleta de dados em campanhas de vendas em rotas específicas, permite uma visualização real de registros de clientes realmente interessados no produto e serviço ofertado. Fornecendo informações relevantes para gestão de equipes de vendas diretas, com apresentação de gráficos com métricas e visualização de mapas com registros e agrupamentos de *leads*.

3. Procedimentos Metodológicos

Para o desenvolvimento do KR7 Prospect, utilizou-se o modelo incremental, este modelo de desenvolvimento é baseado na ideia de desenvolver uma versão inicial, apresentá-la aos usuários e continuar o desenvolvimento de novas versões e funcionalidades até alcançar o sistema necessário conforme Figura 4. As especificações, desenvolvimento e validações são alternadas, e não divididas, com retornos rápidos entre as etapas desenvolvidas (Sommerville, 2018)



Figura 4. Ilustração Desenvolvimento Incremental.
Fonte: Engenharia de Software, Ian Sommerville (2018, p. 49).

3.1 Ferramentas utilizadas

3.1.1 - Lucidchart

Lucidchart é uma ferramenta utilizada para criação de diagramas, fluxos, processos, casos de uso entre outras funcionalidades. Esta ferramenta é online, possui integração com diversos aplicativos como Microsoft Office, Dropbox, Google entre outros. A interface é armazenada em nuvem, então possui compatibilidade com a maioria dos sistemas operacionais e dispositivos, no trabalho em questão utilizou-se para criação da modelagem dos casos de uso (Lucidchart, 2024).

3.1.2 - MySQL - Workbench

O MySQL Workbench é um software utilizado para gerenciamento de banco de dados, é uma ferramenta que foi criada para desenvolvedores, arquitetos e administradores de banco de dados. Este software é uma solução visual que fornece funcionalidades para modelagem de dados, desenvolvimento SQL, ferramentas administrativas, configuração de servidores, administração de usuários, *backup* entre outras. O Software permite que um usuário projete, modele, gereencie o banco de dados de forma visual. Esta ferramenta é compatível com os sistemas operacionais Windows, Linux e Mac OS X (MySQL, 2024).

3.1.3 - Visual Studio Code

Para o desenvolvimento utilizou-se o editor de código fonte Visual Studio Code . O Visual Studio é gratuito e possui compatibilidade com as linguagens utilizadas para o desenvolvimento e é um software leve para desenvolvimento (MICROSOFT, 2024).

3.1.4 - XAMPP

Para simular um servidor local foi utilizado o software XAMPP (X para qualquer dos diferentes sistemas operativos, Apache, MariaDB, PHP, Perl), ele possui um pacote com os principais servidores do mercado incluindo Apache, MySQL, Perl. Trata-se de um software livre de fácil utilização e disponível para a maioria dos sistemas operacionais (XAMPP, 2024).

3.1.5 - Slim Framework

Slim Framework Slim é um *micro framework* Php que auxilia no desenvolvimento de aplicações web e APIs Rest (*Representational State Transfer*) simples e poderosas. O *framework* Slim é uma ferramenta que utiliza requisições e respostas HTTP (*Hypertext Transfer Protocol*)(SLIM, 2024).

3.1.6 - Postman

Postman é um software utilizado para realizar requisições Http. Esta ferramenta oferece uma interface simples e funcional que agiliza o desenvolvimento e envio de requisições para APIs, rotas e sites (POSTMAN, 2024).

3.2 - Linguagens, Metodologias e Conceitos

Nesta seção do trabalho serão apresentadas as tecnologias, linguagens de programação, metodologias e alguns conceitos pertinentes utilizados para o desenvolvimento do sistema.

3.2.1 - Front-end

Quando se fala sobre desenvolvimento, é essencial falar sobre *front-end*. Este termo é utilizado para denominar a interface gráfica que o usuário tem acesso e interage. Podemos citar como exemplos botões, tabelas, textos, caixas de seleção. É uma etapa muito importante do desenvolvimento, haja visto que é através dele que o usuário terá acesso às funcionalidades do sistema, ferramentas, ou site. Se um sistema ou site apresenta erros constantes e falhas, isso diminui a sua utilização ou torna impossível a navegação, nesta etapa do desenvolvimento é essencial preocupar-se com a usabilidade, responsividade, intuitividade (TOTVS, 2024).

3.2.2 - Back-end

O termo *back-end* é utilizado para denominar a parte do sistema responsável pelas funcionalidades não visuais do sistema ou site. Podemos citar como parte do backend banco de dados, servidores, cálculos, lógicas e regras de negócio. Nesta etapa onde entra a segurança, acessos, estruturas e gerenciamento de dados (TOTVS, 2024).

3.2.3 - PHP

O PHP (um acrônimo para *Hypertext Preprocessor*) é uma linguagem de programação *open source* de uso geral, utilizada no servidor. É uma linguagem bastante utilizada especialmente adequada para o desenvolvimento *Web* e que pode ser mesclada com HTML (*HyperText Markup Language*). As páginas web que utilizam essa linguagem possuem HTML e PHP misturados, e os trechos de PHP executam algumas funções. O código executado no servidor gera o HTML e apresenta para o cliente no navegador.

Esta linguagem de programação possui diversas funcionalidades, como citado, gerar HTML, gerar imagens, gerar arquivos PDF (Portable Document Format), gerar textos utilizando Json (*JavaScript Object Notation*) e/ou XML (*Extensible Markup Language*), criptografar dados, enviar email entre outras. Uma das suas principais características é a compatibilidade com diversos modelos de banco de dados, por exemplo MySQL, ou usando uma camada de abstração como o PDO (*Php Data Objects*), que é um módulo utilizando o paradigma de orientação de objetos ou conectar a qualquer banco de dados que suporte a extensão ODBC (*Open Database Connection*) (PHP, 2024).

3.2.4 - Javascript

JavaScript é um linguagem de programação padrão mantida pela entidade W3C (*World Wide Web Consortium*), essa entidade desenvolve padrões e diretrizes para ajudar todos a construir uma *Web* baseada nos princípios de acessibilidade, internacionalização, privacidade e segurança. Essa linguagem de programação é executada diretamente no navegador e é responsável por executar a parte lógica da página e oferecer dinamismo aos componentes apresentados.

O JavaScript é responsável por implementar recursos à página web, é uma das três tecnologias mantidas pela W3C, as outras duas são HTML e CSS. Podemos citar como algumas funcionalidades do JavaScript exibição de atualizações de conteúdo, regras de formatação em formulários, mapas interativos, gráficos animados 2D/3D, jukeboxes de vídeo com rolagem. A linguagem possui diversas APIs para trabalhar com diversos tipos de dados, além do ambiente *Web* também é utilizada para desenvolvimento de aplicativos móveis, sistema operacional, processadores de texto e etc (Mozilla, 2024).

3.2.5 - HTML

Traduzido literalmente como Linguagem de Marcação de Hipertexto, é uma linguagem de marcação que é padronizada pela W3C, é utilizada para estruturar e dar significado ao nosso conteúdos da *web*, por exemplo, definindo parágrafos, títulos e tabelas de dados, ou incorporando imagens e vídeos na página. É um componente base da *web*, atualmente está na sua versão 5, que foi produzida em 2014 (Mozilla, 2024).

3.2.6 - CSS (*Cascading Style Sheets*)

Literalmente traduzido para folha de estilo em cascata, *Css* é uma linguagem de regras de estilo, é um dos 3 pilares tecnológicos mantidos pela W3C, utiliza-se para estilizar os conteúdos exibidos no HTML, podendo estilizar todos os elementos como fonte, parágrafos, tabelas, botões. Pode-se no *Css* criar regras como mudar de cor quando passa o mouse, exibir e ocultar itens conforme o tipo de dispositivo que está acessando (Mozilla, 2024).

3.2.7 - MVC

O padrão MVC (*Model-View-Controller*) é uma arquitetura de desenvolvimento que determina a separação do código fonte em três camadas. Mencionado acima o *Model* refere-se à entidades que representam os dados da aplicação, podemos citar como exemplo classes, banco de dados. A *View* refere-se às interfaces disponíveis no sistema, estas são responsáveis pela apresentação dos dados e capturar eventos da página. O *Controller* traduzido literalmente para controlador, é a camada responsável pela comunicação entre o *Model* e a *View*, nele estão presentes as lógicas e tratamentos de dados que serão exibidos na *View*, como também a execução de tratamentos e eventos (Barros, 2007).

3.2.8 - Protocolo HTTP

O HTTP é um protocolo que permite a comunicação e obtenção de recursos entre cliente e servidor, pode-se citar como exemplo a obtenção de documentos HTML, vídeos, imagens, como diversos outros tipos de dados. Pode-se dizer que na maioria dos casos o cliente é o navegador *web*, este protocolo é um padrão utilizado para troca de dados da *web*. O cliente e os servidores se comunicam através de mensagens individuais. Essas mensagens são denominadas requisições (*request*) enviadas normalmente por um navegador e resposta (*response*) enviada pelo servidor. Pode-se usar também para atualizar apenas algumas partes de documentos HTML utilizando requisições assíncronas com *ajax*. Para o acesso de uma página *web* são feitas diversas

requisições, haja visto que alguns elementos podem estar inclusive em outros servidores.

Um requisição HTTP possui cinco elementos: método, caminho, versão do protocolo, o cabeçalho da solicitação e o corpo da requisição. Os métodos mais utilizados são GET (pegar), POST (enviar), DELETE (deletar), PUT (Atualizar). O caminho denominado PATCH, é a *url* onde a solicitação será feita. A versão do protocolo, atualmente utiliza-se 2 versões HTTP 1.1 e HTTP 2.0. Os outros elementos são o cabeçalho da solicitação que pode ter informações adicionais e o corpo do documento, que em alguns métodos carregam dados para o servidor, podemos utilizar o método POST como exemplo. O cabeçalho da solicitação de autorização HTTP pode ser usado para fornecer credenciais que autenticam um agente de usuário com um servidor, permitindo acesso a recursos protegidos.

As respostas do servidor também apresentam um padrão de elementos, o primeiro elemento de uma resposta é a versão do HTTP, seguida do código de status e a mensagem de status. Em seguida encontra-se o cabeçalho e o corpo do documento. Os códigos mais comuns são código 200 com a mensagem “Ok” que significa que foi encontrado e 404 “*Not Found*” que significa não encontrado (Mozilla, 2024).

3.2.9 - API Rest

Representational State Transfer (Rest) Transferência de Estado Representacional é um estilo de arquitetura baseado em um conjunto de princípios que descrevem como os recursos em rede são definidos e endereçados. A arquitetura Rest é considerada uma API e se mostrou uma alternativa popular para o desenvolvimento de *web services*. Ela funciona utilizando o protocolo HTTP, o cliente faz uma solicitação ao servidor utilizando os métodos HTTP disponíveis GET, PUT entre outros conforme a implementação, e o servidor retorna uma resposta conforme pré estabelecidas, neste caso o cliente não necessita conhecer os detalhes da implementação da resposta, ele apenas solicita e recebe a resposta conforme o método e rota solicitada (Mozilla, 2024).

3.2.10 - Banco de dados relacional

Trata-se de um modelo de banco de dados que armazena e fornece acesso a pontos de dados relacionados entre si. Bancos de dados relacionais são baseados no modelo relacional, uma maneira intuitiva e direta de representar dados em tabelas. Em um banco de dados relacional, cada linha na tabela é um registro com uma ID exclusiva chamada chave primária. As colunas da tabela contém os atributos dos dados e cada registro geralmente tem um valor para cada atributo, facilitando o estabelecimento das relações entre os pontos de dados e tabelas (Oracle, 2024).

3.2.11- Composer

Composer é uma ferramenta para gerenciamento de dependências em PHP. Ele permite que você declare as bibliotecas das quais seu projeto depende e ele as gerenciará para você (Composer, 2024).

3.2.12 - DAO

O DAO (*Data Object Access*) trata-se de um padrão de projeto para persistência de dados utilizado na programação orientada a objetos, seu surgimento se deve a necessidade de fazer uma separação de lógica e de funções de banco de dados. Através dele pode-se fazer alterações nas implementações de negócios sem precisar fazer alterações nas classes DAO, e vice e versa, como também melhora a visualização das classes e do código. Classes DAO são responsáveis por trocar informações com o SGBD (Sistema Gerenciador de Banco de Dados) e fornecer as operações de consultas, edição, inserção de registros e remoção. Essa classe fica responsável por receber os objetos ou dados e transformar em uma query SQL que será enviada para o Banco de dados, também é responsável por transformar a resposta em SQL em um objeto para a parte lógica do sistema (DevMedia, 2024).

3.2.13 - Bootstrap

Bootstrap é uma biblioteca para utilização no front-end que possui diversos recursos, um deles é a responsividade de páginas e componentes. Essa biblioteca atua na parte de visualização utilizando CSS para estilizar e JavaScript para fazer alterações de componentes do HTML (Bootstrap, 2024).

3.2.14 - Ajax

É uma metodologia utilizada na página *web* que permite fazer requisições HTTP em APIs. Ajax é o acrônimo para JavaScript assíncrono + Xml. Trata-se de de um termo utilizado para descrever um método de utilizar juntas tecnologias já existentes, podemos citar como exemplo HTML, XHTML, JavaScript, Xml entre outras. Apesar do termo mencionar Xml, atualmente o formato Json é o mais utilizado devido a ser um formato nativo do JavaScript e fornecer algumas vantagens para as aplicações, uma delas é por ser mais leve que Xml. Uma das vantagens do ajax é permitir que se façam atualizações assíncronas nas páginas *web* sem necessariamente atualizar a página inteira (Ajax, 2024).

3.2.15 - JQuery

JQuery é uma biblioteca de JavaScript, através dela é possível manipular elementos HTML de forma mais prática, manipulação de eventos, animações e utilização do ajax. É uma biblioteca fácil de usar e possui muitos recursos (jQuery, 2024).

3.2.16 - Chart.js

Trata-se de uma biblioteca para criação de gráficos muito popular em desenvolvimento *Web*. Esta biblioteca possui integração com diversos *frameworks* e bibliotecas de JavaScript, pode-se citar React, Angular, Vue.js, Svelte, Next.js. Essa biblioteca possui animações como padrão, e também possui uma configuração padrão para os gráficos, ou seja, mesmo sem utilizar muitas configurações o gráfico terá uma boa aparência. Essa biblioteca fornece um grande conjunto de tipos de gráficos e *plugins* (Chart.js, 2024).

3.2.17 - Mapas - API Google Maps

A API Maps JavaScript é uma API da *Web* do lado do cliente que permite a criação de mapas para exibição de locais em todo o mundo, com ela pode-se adicionar gráficos, marcadores, animações e recursos interativos. É uma API paga, mas fornece uma quantidade de créditos gratuitos para testes. A API possibilita a criação de mapas 2D e 3D, permite adicionar marcadores personalizáveis para indicar locais no mapa, criar mapas personalizados, adicionar camadas e dados personalizados, como também adicionar gráficos personalizados ao mapa (Google Maps Platform, 2024).

4. Desenvolvimento

O KR7 Prospect foi desenvolvido para a plataforma *Web* com responsividade para dispositivos móveis, desenvolvido utilizando PHP, HTML e JavaScript, levando em consideração que será utilizado majoritariamente em ambientes externos. No contexto em que o software está inserido é indispensável o uso de computador para a visualização das interfaces mais complexas e com maior volume de dados, neste caso de administrador. Para o representante o importante é que seja útil e fácil de usar em qualquer dispositivo.

As primeiras etapas do desenvolvimento foram os requisitos e funcionalidades do software, para isso foi utilizado um diagrama de casos de uso para contextualizar as funcionalidades e atores. Após definidas as funcionalidades começou a etapa de testes de funcionalidades, o sistema tem como pré - requisito a utilização de georeferenciamento para apresentar de forma visual em um mapa áreas com maior número de clientes, prospecções ou *leads*. Para a coleta de localização foi utilizada uma função nativa do JavaScript “*navigator.geolocation.getCurrentPosition*”, para dispositivos móveis ele utiliza o GPS (*Global Positioning System*). O GPS do celular pega a localização, latitude e longitude com precisão, não necessitando neste caso o estudo de outras bibliotecas ou funcionalidades. Para a apresentação dos dados foi utilizado a API Maps do Google, a mesma possui diversas funcionalidades como por exemplo clusterização de marcadores e criação de marcadores personalizados.

O Sistema foi implementado utilizando o padrão de projeto MVC (*Model View Controller*), a criação de páginas utilizou-se HTML com estilizações de CSS e a parte lógica do sistema utilizou-se JavaScript, no caso desta aplicação é responsável pelas requisições, funções e métodos. O sistema em um primeiro momento utilizou apenas PHP para comunicação com o banco de dados utilizando a arquitetura DAO (*Data object Access*), para algumas funcionalidades das páginas *web* e o lado servidor da aplicação. Posteriormente optou-se pelo uso de uma API Rest para implementação de funcionalidades adicionais. Para o desenvolvimento da API Rest utilizou-se o Slim Framework na sua versão 4 (*Slim Framework*, 2024).

4.1. Modelagem

Nesta sessão serão apresentados alguns artefatos de software utilizados para a modelagem do sistema.

4.1.1 Análise e Classificação dos Requisitos

Segundo Sommerville (2018), a análise e classificação de requisitos é uma etapa muito importante no desenvolvimento de software, através dele pode-se obter informações importantes sobre as funcionalidades do sistema e efetuar validações. Segundo ele, frequentemente os requisitos são classificados em requisitos funcionais (RF), requisitos não funcionais (RNF). Requisitos funcionais se referem a serviços e funcionalidades do sistema, como deve se comportar em determinadas situações e entradas específicas. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer. Requisitos não funcionais podem ser restrições a serviços e funcionalidades do sistema, restrições impostas por normas e legislação.

Os requisitos não funcionais em muitos casos podem ser aplicados ao sistema com um todo. Apesar de parecerem simples as definições Somerville (2018) aponta que nem sempre essa especificação pode ser aplicada aos requisitos, haja visto que conforme o nível de detalhamento dos requisitos, requisitos não funcionais podem gerar outros requisitos funcionais. Os RFs exprimem funcionalidades reais de um sistema. Eles resultam do tipo de software que será desenvolvido, do tipo de usuário e a forma como foram abordados no processo de desenvolvimento.

Para o desenvolvimento do KR7 Prospect utilizou-se a metodologia ágil, onde alguns requisitos sofreram pequenas alterações no processo de desenvolvimento como também receberam um nível de detalhamento maior. Abaixo serão elencadas explicações sobre os requisitos e sobre o processo de implementação do requisito.

RF1- *Login* de sistema: a autenticação é uma etapa básica de qualquer sistema de gerenciamento. No caso deste sistema todos os usuários são a mesma entidade, porém com atribuições distintas, o *login* de sistema além de efetuar uma autenticação, inicializa uma sessão no navegador, e conforme o status do usuário, terá acessos distintos. O representante tem acesso à interface de representante com as informações pertinentes à sua função, o administrador tem acesso a todas as funcionalidades do sistema.

RF2 - Gerenciamento de Usuários: somente o administrador tem acesso a funções de gerenciamento de usuário, cadastrar, trocar senha, ativar e inativar os representantes. Por regra de negócios os representantes não devem ser excluídos pois há clientes atrelados a eles. O intuito é que o administrador altere esses clientes para outro representante, pois o intuito do sistema é que por conta da rotatividade, que não se percam clientes em potencial ou a empresa perca os dados de contatos de cliente porque o representante ou vendedor saiu da empresa.

RF3 - Criar *leads* com localização: os registros de clientes devem conter obrigatoriamente a localização, latitude e longitude, é uma das funcionalidades críticas do sistema, sem ela não será possível analisar no mapa os pontos de maior concentração de *leads*. Por este motivo este campo é preenchido automaticamente e não há e nem deve conter a opção de criar registro sem a localização. A localização atual é utilizada

como referência, mas pode haver casos em que o endereço do cliente é diferente, por este motivo o usuário pode preencher os campos direcionados ao endereço.

RF04 - Apresentar registros no mapa: para a apresentação dos registros no mapa utiliza-se georreferenciamento. Este requisito depende da implementação do requisito anterior. Além de apresentar no mapa, o sistema deve agrupar os registros conforme o zoom utilizado no mapa para uma melhor visualização. O sistema deve fornecer uma opção de visitar e editar clientes que estão cadastrados, através da interface de visualização do mapa. A funcionalidade de visitar e editar foram acrescentadas na interface mapa no processo de desenvolvimento.

RF05 - Apresentar relatórios: o sistema deve apresentar de forma clara uma página com relatórios para visualização de resultados com opção de filtros mensais, esta página somente deve estar disponível para administradores.

RF06 - Botão de visitar / *check in*: deve haver a opção de criar um registro de localizações que o usuário visitou. Este registro possui três finalidades; registrar visitas a clientes; mapeamento de rotas e ser uma métrica para comparativo na aba de relatórios.

RF07 - Apresentar *leads* em tabela: este requisito depende do RF1, devem ser apresentados em tabelas os registros efetuados no sistema, entretanto, os representantes só deverão ter acesso a registros efetuados por ele, e os administradores devem ter acesso de todos os registros.

RF08 - Apresentação dos resultados: os dados devem ser apresentados em uma interface com gráficos, apenas os administradores devem ter acesso.

RF09 - Apresentar todos os registros: este requisito é de uso exclusivo do administrador e difere-se do RF04 devido às restrições de usuário. O representante deve ver somente seus registros, e os administradores devem ver todos os registros.

RF10 - Edição / visualização de registros: no mapa deve conter uma opção de edição de dados do cliente, tanto para o usuário convencional quanto para o administrador.

4.1.2 Casos de Uso

Para uma melhor visualização das funcionalidades necessárias do sistema foi criado um Diagrama de Casos de Uso utilizando o software Lucidchart (2024), conforme descrito em Sommerville (2018). Para o sistema proposto foram criados apenas dois atores, o administrador e o representante. O administrador estende as funcionalidades de um representante. O representante ilustrado na Figura 5 em relação ao sistema pode cadastrar-se, fazer *login*, editar perfil, cadastrar e editar registros de *leads*, visualizar os registros no mapa, registrar visitas de clientes e efetuar *check in*.

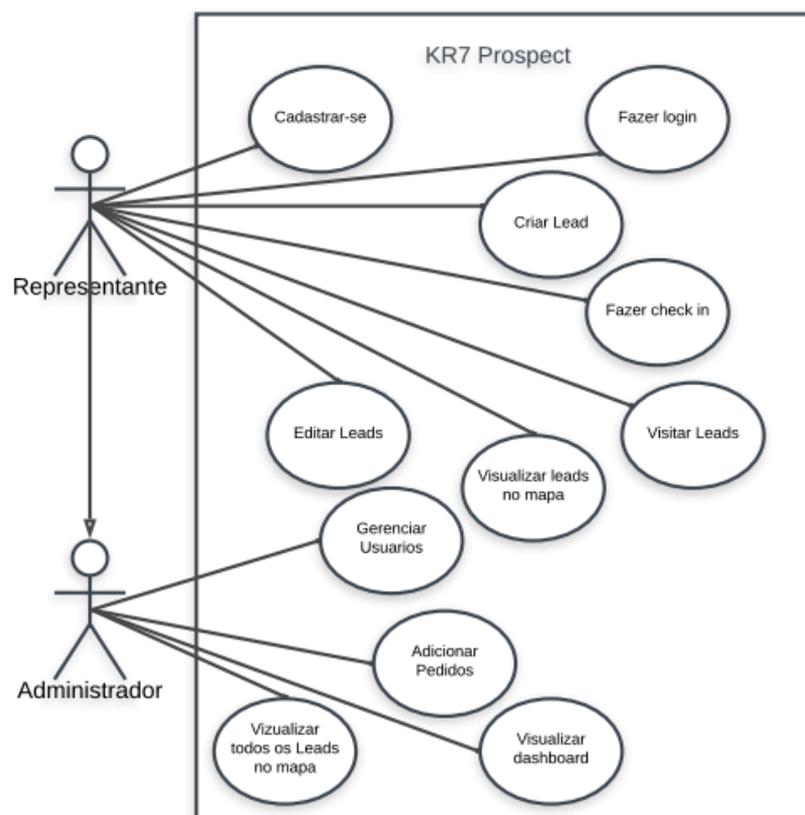


Figura 5: Diagrama de Casos de Uso do sistema proposto
Fonte: Autoria Própria.

Em relação aos registros de clientes em potencial, eles não aparecem como atores pois não irão utilizar o sistema. O sistema é para uso interno e controle da empresa que irá utilizar. O administrador pode além de todas as funcionalidades de representante, ativar ou inativar representantes, cadastrar novos representantes, visualizar todos os registros do sistema no mapa, visualizar a interface de dashboard.

4.1.3 Entidade e Relacionamento

O sistema possui quatro entidades conforme apresentado na Figura 6, são elas: representante, cliente, pedidos, dailyvisit. A entidade representante possui os registros dos usuários do sistema e informações pessoais. Esta entidade possui o campo status, este campo é reservado para indicar o tipo de usuário, status 2 para administrador, status 0 para usuário inativo e 1 para usuário ativo.

A entidade cliente possui os campos com informações de contato, coordenadas, endereço e observações que pode ser preenchida com informações adicionais pertinentes à prospecção. Esta entidade é responsável por armazenar os registros efetuados pelos representantes de clientes que ficaram interessados no produto ou serviço.

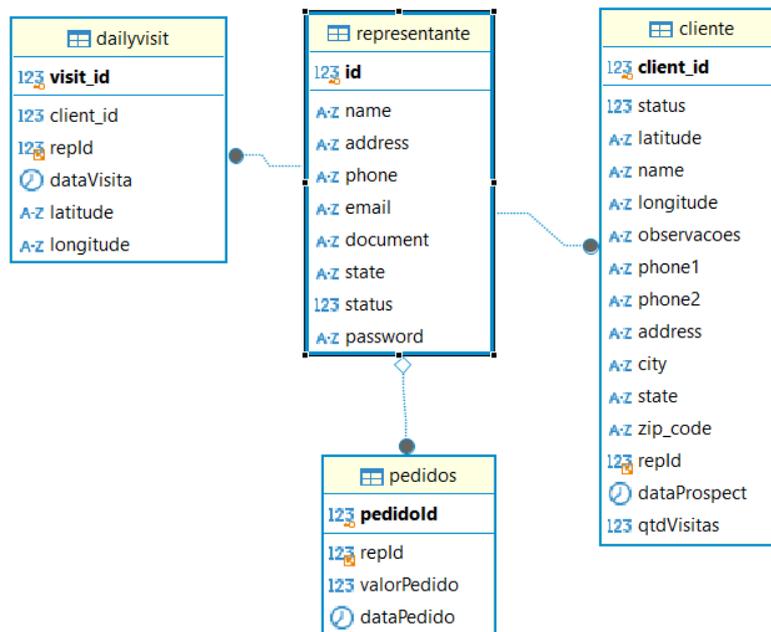


Figura 6. Diagrama ER da base de dados da aplicação.
Fonte: Autoria própria, gerado pela ferramenta DBeaver, 2024.

A entidade pedidos é utilizada para armazenar os registros de pedidos efetuados e finalizados. Esta entidade só é acessível para administradores e é utilizada para apresentar uma métrica de resultados comparativos entre os representantes do sistema. Todas as entidades são relacionadas pelo ID do representante. A entidade dailyvisit armazena os registros de visitas a *leads* e *check in* efetuados, pode ser utilizada para mapeamento de rotas e registros de localização.

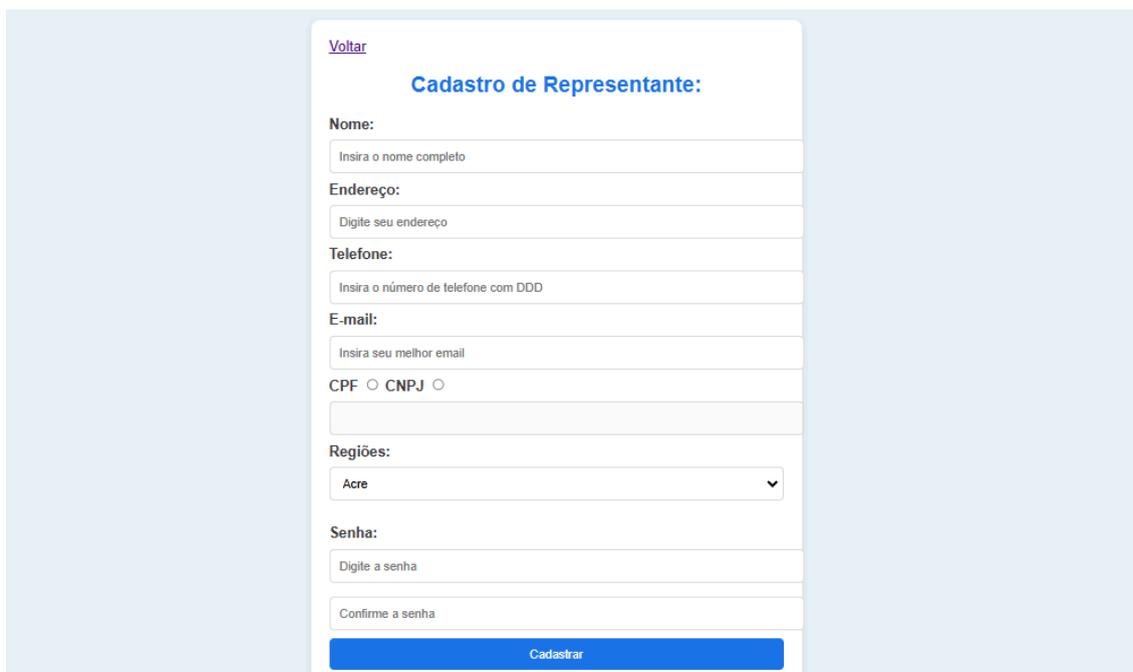
4.1.4 Protótipo

Serão apresentados aqui algumas interfaces bem como algumas explicações sobre o funcionamento. O KR7 Prospect assemelha-se a uma agenda de clientes, com funcionalidades de geolocalização integradas, apresentação de métricas de produtividade e resultados. Além de apresentar no mapa os registros de clientes em potencial ou clientes recorrentes para o caso de representantes comerciais.

4.1.4.1- Interface de cadastro de representante

Para cadastro de novos representantes, basta enviar o link do sistema e o mesmo pode se cadastrar, sem a necessidade de solicitar todas as informações e o administrador efetuar o cadastro, fica a responsabilidade do administrador apenas ativar o usuário. A interface de cadastro possui um *design* simples e responsivo, as informações exigidas são: nome, endereço, telefone, email, documento CPF para pessoa física ou CNPJ para pessoa jurídica, região e senha. Em relação a senha é necessário apenas que tenha no mínimo 5 caracteres. É muito importante ressaltar que a senha utilizada não aparece para nenhum usuário. A senha digitada pelo usuário utiliza a função nativa de criptografia do php

“password_hash”, onde a função cria um novo hash de senha usando um algoritmo de hash unidirecional forte. A senha é criptografada e salva no banco de dados e apenas o usuário que fez o cadastro tem conhecimento da senha utilizada. O KR7 Prospect não permite repetir telefone, email e documento de representantes. Esses dados podem ser utilizados apenas uma vez.



[Voltar](#)

Cadastro de Representante:

Nome:
Insira o nome completo

Endereço:
Digite seu endereço

Telefone:
Insira o número de telefone com DDD

E-mail:
Insira seu melhor email

CPF CNPJ

Regiões:
Acre

Senha:
Digite a senha
Confirme a senha

Cadastrar

Figura 7. Interface de Cadastro Representante
Fonte: Autoria própria.

4.1.4.2- Interface Inicial para o Administrador

Esta interface é apresentada após o *login*, como apresenta a Figura 8, após o login e o usuário seja administrador ele será direcionado para a interface *home Admin* onde no padrão aparecerá os representantes cadastrados no sistema. Caso o administrador queira ativar os usuários inativos ou recém cadastrados basta clicar no botão da coluna Status e alterar o status. Apenas o administrador pode editar as informações de outros representantes e visualizar os representantes cadastrados no sistema. Nesta interface caso clique no botão flutuante com o símbolo “+” ele poderá adicionar um novo representante ao sistema.

Olá Rep Douglas, seja bem-vindo à área administrativa!



Representantes Cientes

Meu perfil Ver no Mapa DASHBOARD Pedidos Sair

Nome	Email	Telefone	Estado	Status	Editar
Rep Douglas	repdouglas@hotmail.com	(54) 98432-0653	RS	Adm	Editar
Rep Joao	reppaula@hotmail.com	(54) 99263-1099	RS	Ativo	Editar
Rep Fernanda	repfernanda@hotmail.com	(54) 99118-5195	RS	Ativo	Editar
Rep Michael	repmichael@hotmail.com	(54) 99133-3249	RS	Ativo	Editar
Rep Paulo	reppaulo@hotmail.com	(54) 99960-5119	RS	Inativo	Editar

Figura 8. Interface Administrador
Fonte: Autoria própria.

4.1.4.2.1- Botões e funções Administrativas

Será apresentado um resumo das funcionalidades dos botões da interface administrativa apresentada na Figura 9. Botão Representantes / Clientes: este botão tem como intuito fazer uma alternância da tabela de visualização, como padrão aparecem representantes mas se clicar em clientes a tabela passa a apresentar os clientes cadastrados, com opção de edição. Botão Meu perfil: O representante pode alterar as informações do seu perfil. Botão Ver no mapa: nesta interface pode-se visualizar os todos os *leads* cadastrados no sistema, caso o usuário seja administrador pode visualizar os registros de todos os representantes, caso seja usuário padrão visualiza apenas os clientes cadastrados por ele. Botão Dashboard: Será redirecionado para uma interface que apresenta gráficos e resultados comparativos de todos os representantes da equipe. Botão Pedidos: essa interface apenas fica disponível para administradores, que após confirmação podem cadastrar os pedidos efetivados/finalizados, de modo a apresentar um comparativo entre os representantes. Botão Sair: faz *logout* do sistema.



Representantes Cientes

Meu perfil Ver no Mapa DASHBOARD Pedidos Sair

+

Figura 9. Botões Interface Administrador
Fonte: Autoria própria.

4.1.4.3 - Interface Inicial para o Representante

Nesta interface o usuário tem acesso às funções básicas disponíveis para ele, como visualizar os clientes que ele cadastrou no sistema, editar os clientes como também botões para outras interfaces.



KR7 TELECOM PROSPECT

Olá Douglas Kramer, seja bem vindo!!

Lista de clientes

+ Cadastrar Cliente Meu perfil Ver no Mapa Sair

Pesquisar

Nome	Telefone	Observações	Cidade	Status	Editar
denaira kramer	(54) 9913-3245	cliente vai falar com o cachorro, esta no nome do cachorro	Caxias do Sul	1	Editar
Tatiane abreu	(54) 99234-5467	tatiane vai falar com a mãe	Caxias do Sul	1	Editar
joatan da silva	(54) 99133-3240	possui fidelidade até dezembro	Caxias do Sul	1	Editar

Figura 10. Interface inicial Representante

Fonte: Autoria própria.

4.1.4.3.1- Botões e funções Representante

Serão exibidos os botões e funcionalidades da interface representante. Botão Cadastrar cliente: será direcionado para uma interface com um formulário para registro de cliente. Botão Meu Perfil: Permite fazer alterações dos dados e alterar a senha do usuário logado. Botão Ver no mapa: muda para a interface de mapa e mostra os clientes próximos a ele. Esta interface tem a funcionalidade de iniciar o mapa com a localização atual do usuário como centro do mapa. Botão Editar: vai para uma interface de edição de cliente para adicionar mais informações como também visualizar todas as informações, a tabela apresenta uma visualização básica dos dados do cliente. Botão Sair: faz *logout* do sistema.

4.1.4.4 - Interface para o cadastro de clientes

Esta interface possui todas as informações pertinentes a um cliente conforme a especificação como também possui a função de buscar a localização automaticamente, basta preencher o restante das informações.

[← Voltar](#)

Cadastro de Cliente:

Nome

Latitude

Longitude

Telefone 1

Telefone 2

Observações

CEP

Endereço

Cidade

Estado

[Cadastrar](#)

Figura 11. Interface Cadastro Cliente
Fonte: Autoria própria.

4.1.4.5 - Interface para o georeferenciamento

A interface para georeferenciamento apresenta no mapa os marcadores de registros conforme latitude e longitude como apresentado na Figura 12. Pode-se visualizar os marcadores, os mesmos são agrupados conforme o zoom do mapa. O mapa sempre inicia com a localização atual como ponto central, isso facilita a exibição dos clientes próximos ao usuário. A interface mapa é igual para Administrador e usuários, porém utiliza uma lógica para exibir apenas os registros do representante logado. Caso seja administrador exibe os registros de todos os representantes.

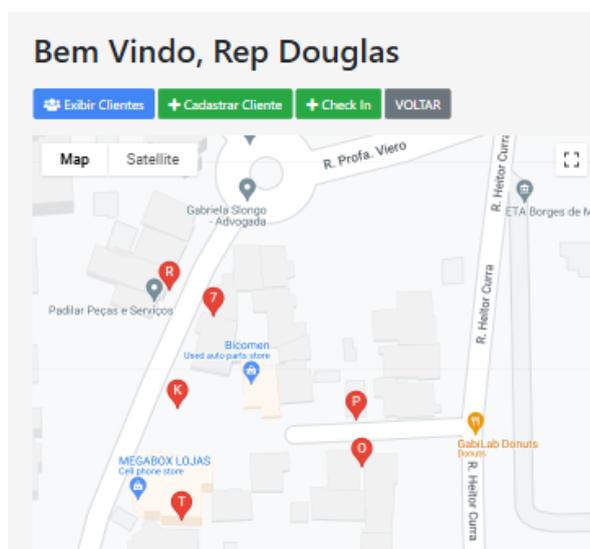


Figura 12. Interface Mapa Representante
Fonte: Autoria própria.

4.1.4.5.1 - Botões Interface Mapas

Serão exibidas as funcionalidades da interface Maps. Botão Exibir clientes: inicializa o mapa com os clientes conforme a localização, latitude e longitude. Botão Cadastrar cliente: abre uma nova interface para efetuar o registro de um novo *lead*. Botão Check in: cria um log no sistema com a localização atual do representante, essa função pode posteriormente ser utilizada para mapeamento de rota que cada representante fez. Botão Voltar: redireciona para a interface inicial.

4.1.4.5.2 - Clicando em um marcador

Após clicar em um marcador, como apresenta a Figura 13, o mesmo exibirá algumas informações básicas onde pode-se efetuar duas ações: visitar que criará um log no sistema com os dados da visita, Id do representante, Id do cliente, data e localização atual. A outra ação é editar, onde pode-se completar o cadastro, adicionar uma nova observação ou apenas visualizar o restante das informações do cliente. A interface de cadastro, edição e visualização de cliente são semelhantes, se encontra na seção 4.1.4.4 .



Figura 13. Janela de Informações Marcador

Fonte: Autoria própria.

4.1.4.6 - Interface de pedidos

Esta é uma interface reservada também apenas para administradores do sistema. Na interface de pedidos é possível carregar os pedidos de um determinado mês e adicionar um novo pedido. As informações de pedidos serão utilizados na interface *dashboard*.

Painel de Pedidos

Selecione o mês:

Setembro

Carregar Pedidos

Adicionar Pedido Voltar para a Home

ID do Pedido	ID do Representante	Valor do Pedido	Data do Pedido
11	4	R\$ 79.90	2023-09-21
12	6	R\$ 89.90	2023-09-16
13	2	R\$ 89.90	2023-09-11
14	4	R\$ 109.90	2023-09-19
15	6	R\$ 79.90	2023-09-03
16	2	R\$ 79.90	2023-09-15

Figura 14. Interface para cadastro de Pedidos

Fonte: Autoria própria.

4.1.4.7 - Interface de Relatórios e Dashboard

A Figura 15 apresenta a interface dashboard, nesta interface pode-se selecionar um mês para carregar os dados do mês selecionado. O sistema cria 4 gráficos com as informações: Novos *leads*, receita total dos pedidos efetivados, quantidade total de visitas a clientes, quantidade de pedidos efetivados. Na parte inferior da Figura 15 pode-se observar uma tabela com algumas métricas aferidas no sistema.

Dashboard

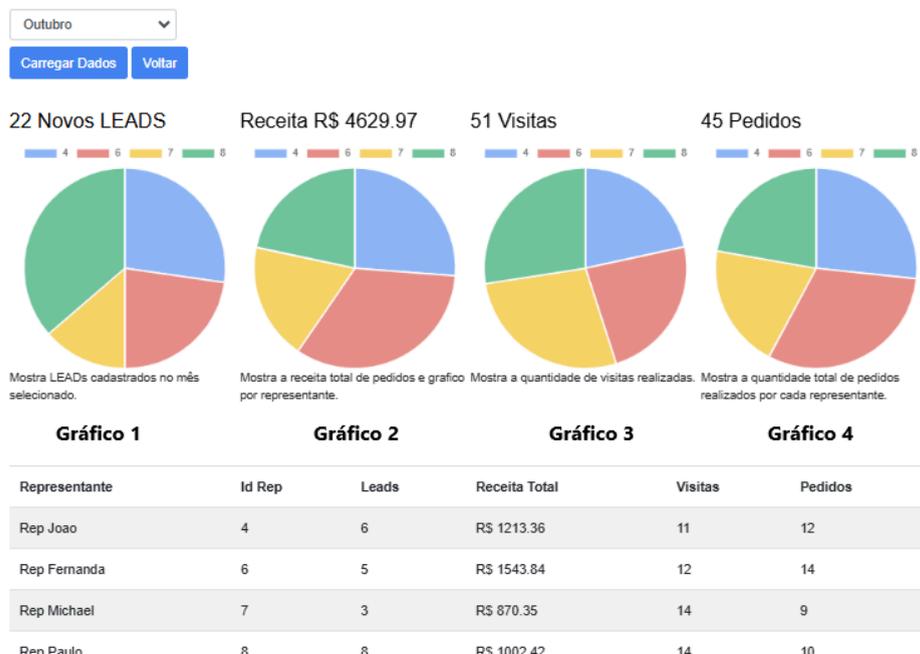


Figura 15. Interface Dashboard

Fonte: Autoria própria.

4.2 Implementação

Nesta seção serão apresentados alguns trechos de código, estudos e funções que foram considerados peças chave no desenvolvimento do projeto. No desenvolvimento do trabalho houveram algumas etapas de estudo que demandaram um pouco mais de atenção, a primeira destas etapas foi o estudo e teste de funções com localização, e apresentação dos dados em mapas.

4.2.1 - Localização do dispositivo

Para coletar e acessar a localização atual do dispositivo foi utilizado a função nativa do JavaScript “*navigator.geolocation.getCurrentPosition*”. Segue abaixo na Figura 16 o trecho do código fonte responsável pela coleta da latitude e longitude.

1	<code>if(navigator.geolocation){</code>
2	<code> navigator.geolocation.getCurrentPosition(function(position){</code>
3	<code> latAtual=position.coords.latitude;</code>
4	<code> longAtual=position.coords.longitude;</code>
5	<code> document.getElementById("latitude").value = latAtual;</code>
6	<code> document.getElementById("longitude").value = longAtual;</code>
7	<code> })</code>
8	<code>}</code>

Figura 16. Trecho do código-fonte que pega a localização automaticamente.
Fonte: Autoria própria.

Essa é uma função que executa quando a interface de cadastro de cliente é aberta, esse trecho de código faz uma chamada a função *navigator.geolocation.getCurrentPosition* na linha 2, atribui os valores às variáveis *latAtual* e *longAtual* nas linhas 3 e 4, nas linhas 5 e 6 os valores da variáveis são atribuídos aos campos latitude e longitude da página HTML.

4.2.2 - Inicialização do Mapa

A documentação da API fornece todas as informações e passo a passo para utilização do google *maps*, em termos simples é necessário criar uma chave da API. Para isto deve criar um projeto dentro da plataforma Google Cloud. A plataforma fornece uma quantidade de créditos mensais para testes. A segunda etapa é criar uma página HTML, CSS e JavaScript. Após a criação da página basta adicionar um elemento mapa. Existem diversas maneiras de adicionar um mapa, mas para fins de exemplo apenas com HTML

já é possível exibir um mapa na página em uma localização pré definida, com exibição de alguns marcadores estáticos. Para o projeto foram utilizadas algumas outras funcionalidades conforme os requisitos do sistema. A interface do mapa deve ser inicializada com o ponto central da localização atual do usuário. Para isso, primeiramente a localização atual é coletada com a função “*navigator.geolocation*” e após isso o mapa é inicializado, conforme apresenta a Figura 17.

1	<code>const map = new google.maps.Map(document.getElementById("map"), {</code>
2	<code>zoom: 16,</code>
3	<code>center: { lat: latAtual, lng: longAtual },</code>
4	<code>mapId: "DEMO_MAP_ID",</code>
5	<code>});</code>

Figura 17. Trecho do código responsável pela centralização e zoom do mapa.

Fonte: Autoria própria.

4.2.3 - Adicionar Marcadores

Os marcadores exibidos no mapa possuem algumas variáveis e podem ser personalizados de diversas maneiras. Pode-se citar como variáveis a localização do marcador (latitude e longitude), o tipo de marcador, eles podem ser personalizados com outras imagens, por exemplo, no KR7 Prospect foi utilizado um marcador padrão, e com letras para diferenciar um do outro. Um dos atributos de um marcador é a janela de informações “*infoWindow*”. Esta janela fornece informações sobre o marcador e é possível apresentar elementos HTML.

Primeiramente deve-se pegar do banco de dados os registros de clientes, para isso foi criada a API Rest, a interface do mapa é a mesma para administradores e usuários comuns, o que difere a exibição dos marcadores no mapa é a url da API, se o usuário é um administrador, ele chama uma url que retorna todos os dados do sistema, se é um usuário comum chama apenas os clientes atribuídos a ele. Como as chamadas do sistema são assíncronas, foi utilizado um botão para exibir o mapa e o mapa só é inicializado após a página estar carregada, assim é possível garantir que o usuário está autenticado, também é possível testar o tipo de usuário administrador ou usuário convencional, como também já estar com a localização atual carregada. Segue o trecho de código que faz essa diferenciação de requisição.

1	<code>var url = "http://localhost:8080/markers";</code>
2	<code>const url2 = "http://localhost:8080/repsid/";</code>
3	<code>function botaoBuscar(idRepresentante, repStatus) {</code>
4	<code> if (repStatus == 2) {</code>
5	<code> buscarclientes();</code>
6	<code> } else {</code>
7	<code> url = url2 + idRepresentante;</code>
8	<code> buscarclientes(); }</code>
9	<code>}</code>

Figura 18. Trecho do código-fonte responsável pela lógica de busca dos clientes na API

Fonte: Autoria própria.

Após buscar os clientes no banco de dados através do método *fetch* (buscar) na url da API, a lista de clientes é dividida em duas listas distintas: *locations* com as localizações dos marcadores “lat” e “lng”, padronizando os dados conforme especificação da API Maps JavaScript, “clientes” armazena os dados do cliente para a exibição na janela de informações dos marcadores. Segue abaixo o código que atribui os dados retornados do banco de dados às listas *locations* e *clientes*.

1	<code>clientes = data.map(function (elemento) {</code>
2	<code> return {</code>
3	<code> Id: elemento.client_id, nome: elemento.name, telefone1:</code>
4	<code> elemento.phone1, telefone2: elemento.phone2,</code>
5	<code> repId: elemento.repId, latitude: elemento.latitude,</code>
6	<code> longitude: elemento.longitude</code>
7	<code> }</code>
8	<code>});</code>
9	<code>locations = data.map(function (elemento) {</code>
10	<code> return { lat: Number(elemento.latitude.replace(",", ".")),</code>
11	<code> lng: Number(elemento.longitude.replace(",", ".")) }</code>
12	<code>});</code>

Figura 19. Trecho do código-fonte que armazena as coordenadas dos registros.

Fonte: Autoria própria.

Após carregados os dados nos Arrays basta exibir os marcadores no mapa, além de exibi-los é utilizada a função “*markerClusterer.MarkerClusterer*”, que agrupa marcadores próximos permitindo uma melhor visualização dos marcadores no mapa. A Figura 20 apresenta o trecho do código fonte responsável pela exibição dos marcadores.

```
1 // CRIA MARCADORES COM ICONE DE LETRAS
2 const labels = "KR7PROSPECT";
3 // ADICIONA MARCADORES NO MAPA
4 const markers = locations.map((position, i) => {
5   const label = labels[i % labels.length];
6   const pinGlyph = new google.maps.marker.PinElement({
7     glyph: label,
8     glyphColor: "white",   });
9   const marker = new google.maps.marker.AdvancedMarkerElement({
10    position,
11    content: pinGlyph.element,   });
12   var cliente = clientes[i];
13   console.log(cliente);
14   const contentString = String("Html da janela de informações");
15   marker.addListener("click", () => {
16     infoWindow.setContent(contentString);
17     infoWindow.open(map, marker);
18   });
19   return marker;
20 });
21 new markerClusterer.MarkerClusterer({ markers, map });
22 // Função para agrupar os marcadores .
23 }
```

Figura 20. Trecho responsável pela criação de marcadores, criação de janela de informações e agrupamento de marcadores.

Fonte: Autoria própria.

4.2.4 - Resumo da API Rest

A utilização da API Rest melhorou bastante a usabilidade do sistema, as páginas carregam mais rapidamente, visto que a parte lógica das páginas não fica responsável por buscar os dados no banco. As páginas ficam responsáveis por realizar uma requisição na url e recebem os dados esperados. Uma das vantagens também é a reutilização da API para desenvolvimento de um aplicativo ou outras páginas. Todas as requisições são feitas a partir do endereço “<https://api.kr7telecom.com.br>”, que é um endereço local. Segue abaixo um pequeno resumo das funcionalidades da API. Os campos com chaves, são requisições que esperam uma ou mais variáveis.

4.2.4.1 -Rotas de Clientes

1. GET /markers

- Descrição: Retorna todos os clientes.
- Saída: Lista de clientes.
- 2. GET /clients/{id}
 - Descrição: Retorna informações de um cliente específico.
 - Entrada: {id} (ID do cliente).
 - Saída: Detalhes do cliente.
- 3. POST /clients
 - Descrição: Adiciona um novo cliente.
 - Entrada: Dados do cliente em JSON.
 - Saída: Confirmação da inserção.
- 4. PUT /clients/{id}
 - Descrição: Atualiza informações de um cliente.
 - Entrada: {id} (ID do cliente) e dados atualizados em JSON.
 - Saída: Confirmação da atualização.
- 5. DELETE /clients/{id}
 - Descrição: Remove um cliente específico.
 - Entrada: {id} (ID do cliente).
 - Saída: Confirmação de exclusão.

4.2.4.2 -Rotas de representante

- 1. GET /reps
 - Descrição: Lista todos os representantes.
 - Saída: Lista de representantes.
- 2. GET /reps/{id}
 - Descrição: Busca informações de um representante pelo ID.
 - Entrada: {id} (ID do representante).
 - Saída: Dados do representante.
- 3. POST /representantes
 - Descrição: Adiciona um novo representante.
 - Entrada: Dados do representante em JSON.
 - Saída: Confirmação da inserção.
- 4. PUT /representantes/{id}
 - Descrição: Atualiza informações de um representante.
 - Entrada: {id} (ID do representante) e dados em JSON.
 - Saída: Confirmação da atualização.
- 5. DELETE /representantes/{id}
 - Descrição: Remove um representante pelo ID.
 - Entrada: {id} (ID do representante).
 - Saída: Confirmação de exclusão.
- 6. POST /representantes/login
 - Descrição: Realiza *login* do representante.
 - Entrada: E-mail e senha em JSON.
 - Saída: Token de autenticação (ou mensagem de erro).

4.2.4.3 -Rotas de Visitas

1. GET /visitas/{repId}/{mes}
 - Descrição: Busca visitas realizadas por um representante em um mês específico.
 - Entrada: {repId} (ID do representante), {mês} (mês).
 - Saída: Lista de visitas.
2. POST /visitar
 - Descrição: Adiciona uma nova visita.
 - Entrada: Dados da visita em JSON.
 - Saída: Confirmação da inserção.

4.2.4.4 -Rotas de Dashboards e Estatísticas

1. GET /comparativo/{mês}
 - Descrição: Retorna uma análise comparativa de representantes para um mês específico.
 - Entrada: {mês} (número do mês).
 - Saída: Dados de comparação (clientes, receitas, visitas, pedidos).
2. GET /dash1/{mes}
 - Descrição: Busca clientes prospectados no mês especificado.
 - Entrada: {mês} (número do mês).
 - Saída: Lista de clientes prospectados.
3. GET /dash2/{id}/{mês}
 - Descrição: Busca informações de clientes prospectados por um representante em um mês.
 - Entrada: {id} (ID do representante), {mês} (mês).
 - Saída: Lista de dados.

4.2.4.5 -Rotas de Pedidos

1. GET /pedidos/{mes}
 - Descrição: Retorna todos os pedidos feitos em um mês específico.
 - Entrada: {mês} (número do mês).
 - Saída: Lista de pedidos.
2. GET /pedidos/rep/{repId}/{mês}
 - Descrição: Busca pedidos feitos por um representante em um determinado mês.
 - Entrada: {repId} (ID do representante), {mês}.
 - Saída: Lista de pedidos.
3. POST /pedidos
 - Descrição: Adiciona um novo pedido.
 - Entrada: Dados do pedido em JSON.
 - Saída: Confirmação da inserção.
4. PUT /pedidos/{id}

- Descrição: Atualiza informações de um pedido.
 - Entrada: {id} (ID do pedido) e dados em JSON.
 - Saída: Confirmação da atualização.
5. DELETE /pedidos/{id}
- Descrição: Remove um pedido específico.
 - Entrada: {id} (ID do pedido).
 - Saída: Confirmação de exclusão.

5. Resultados

Os resultados apresentados nesta seção são dados fictícios utilizados somente para demonstrar as funcionalidades do KR7 Prospect, o software está atualmente em uma versão inicial, não foi utilizado na prática. A Figura 21 apresenta 4 gráficos distintos com informações coletadas do banco de dados do sistema. O gráfico 1 da Figura 21 faz uma requisição na API através da rota “/dash1/{mês}”, esta rota retorna uma lista de todos os registros efetuados no mês selecionado. Após receber os dados, é criado o gráfico de pizza com as quantidades de registros separadas por representante. O gráfico 3 apresenta de forma quantitativa os dados de visitas realizadas, o título apresenta a quantidade total de visitas, o gráfico divide as visitas por representante.

Os gráficos 2 e 4 são criados através dos dados recebidos pela rota “/pedidos/{mês}” que retorna todos os pedidos efetuados no mês selecionado. Após o recebimento dos dados de pedidos são efetuadas as somas por representante e soma total de todos os pedidos, em seguida os dados são apresentados no gráfico 2, e a receita total do mês pode ser visualizada no título do gráfico. O gráfico 4 apresenta de forma quantitativa os mesmos dados, fazendo uma contagem da quantidade de pedidos por representante. O título do gráfico apresenta a quantidade total e as divisões apresentam a soma total de pedidos por representante.



Figura 21. Gráficos com Resultados

Fonte: Autoria própria.

A Figura 22 apresenta uma tabela comparativa de todos os representantes cadastrados no sistema. Com o intuito de diminuir a quantidade de dados enviados através da requisição e diminuir a quantidade de requisições, estes dados são todos

obtidos na mesma requisição através da rota “/comparativo/{mês}”. A requisição recebe como parâmetro o mês selecionado no início da página, faz uma contagem na tabela clientes e agrupa o resultado por representante, e após isso faz uma soma dos valores dos pedidos por representante na tabela de pedidos agrupa por representante. A próxima etapa é realizar uma contagem da quantidade de visitas e quantidade de pedidos e agrupar todos os dados pelo id do representante. O resultado desta requisição é um resumo de todos os dados do banco de dados do mês selecionado.

Tabela 1. Comparativa de Resultados dos representantes

Representante	Id Rep	Leads	Receita Total	Visitas	Pedidos
Rep Joao	4	4	R\$ 758.30	10	8
Rep Fernanda	6	4	R\$ 658.40	9	7
Rep Michael	7	2	R\$ 489.50	11	5
Rep Paulo	8	6	R\$ 428.60	10	5

A interface mapa apresentada na Figura 22 apresenta os registros da tabela cliente do banco de dados no mapa. A função de clusterização de marcadores, agrupa os marcadores apresentados no mapa conforme o zoom do dispositivo.

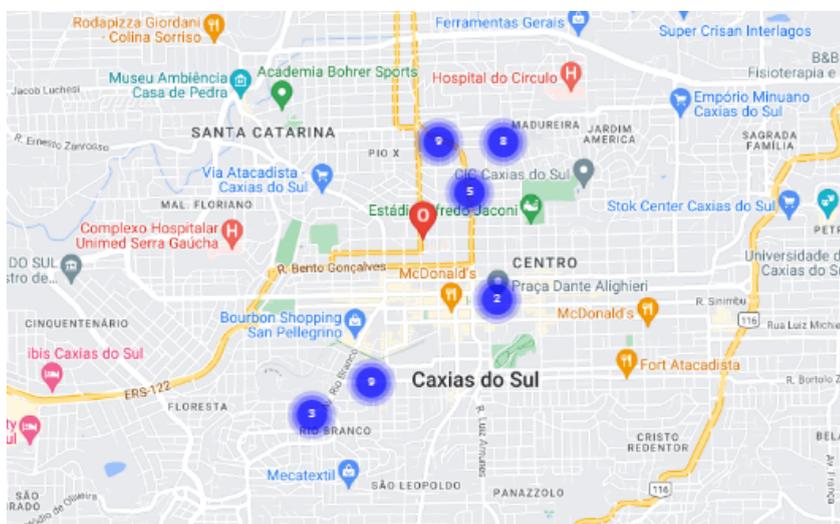


Figura 22. Mapa com Agrupamento de Marcadores
Fonte: Autoria própria.

6. Considerações Finais

O estudo e implementação das tecnologias utilizadas para o desenvolvimento do KR7 Prospect atendeu as expectativas e forneceu um software funcional conforme as especificações. Ele pode ser utilizado para registros de prospecções de clientes com georreferenciamento, registro de visitas, acompanhamento de resultados para empresas

que utilizam o canal de venda direta ou possuem representantes comerciais. Todos os requisitos propostos foram implementados e as interfaces do sistema apresentam uma visualização clara dos dados coletados. A interface de *dashboard* apresenta quatro gráficos com a soma de todas as métricas e também cria uma tabela comparativa com os pedidos efetuados, novos *leads*, visitas e quantidade de pedidos dos representantes cadastrados no sistema. Os dados apresentados na tabela são dados fictícios simulando uma situação real, através da análise dos dados apresentados pode-se calcular métricas e parâmetros para acompanhar a produtividade dos colaboradores. A interface mapa apresenta de forma clara e objetiva os registros no mapa, e faz o agrupamento dos marcadores para uma melhor análise visual dos registros cadastrados no sistema, através da visualização pode-se fazer análises de rotas e mapeamento de bairros, ruas ou cidades com maior quantidade de potenciais clientes.

Neste sentido conclui-se que o software atendeu a todos os requisitos e especificações desejadas e seria útil para empresas desse ramo de atuação, vendas diretas, haja visto que apresenta uma interface simples e objetiva com dados e métricas consideradas importantes para este setor. Conforme apresentado o Brasil possui uma alta rotatividade de empregos e uma baixa produtividade dos colaboradores, então fornecer ferramentas adequadas e utilizar as tecnologias pode ser um diferencial competitivo para as empresas. No setor de vendas diretas quanto maior o número de representantes ou vendedores, maior a dificuldade de acompanhamento de desempenho pelos gestores. Através da coleta dos dados e informações dos clientes que têm interesse, pode-se aplicar outras metodologias de vendas, ofertar promoções e outras abordagens, com o intuito do fechamento do pedido ou contratação de serviços.

O desenvolvimento do KR7 Prospect exigiu uma grande revisão na literatura e disciplinas do curso de Tecnologia em Análise e Desenvolvimento de Sistemas. Os principais desafios foram encontrados na etapa de desenvolvimento, exigindo análises em trechos de código fonte, testes contínuos de funcionalidades e funções, estudo de documentações de diversas bibliotecas e revisão e estudo de linguagens de programação já abordadas no curso. A etapa mais complexa do desenvolvimento foi a etapa de utilização de mapa, apesar de possuir diversos exemplos, nem sempre os exemplos apresentados têm as funções semelhantes ao que era exigido.

Para as próximas versões sugere-se utilizar para coletar dados reais e avaliar se houve melhora de resultados. Utilizar um sistema de segurança mais robusto é essencial se tratando de dados reais de possíveis clientes. Pode-se fazer a implementação de um aplicativo nativo ou utilizar outra metodologia por exemplo de interface única para a exibição das páginas. Uma melhoria necessária é utilizar outros *frameworks* de mapa para agrupamento e exibição de marcadores, a API do Google não é gratuita e para um maior volume de dados deve-se avaliar o custo benefício da API ou utilizar uma plataforma de mapas gratuita. Conforme o porte da empresa que irá utilizar o KR7 Prospect, deve-se pensar em termos de escalabilidade outros tipos de usuários como também equipes distintas, assim o gestor de uma equipe terá acesso aos registros da sua equipe, na versão atual ele apenas tem acesso aos registros efetuados por ele.

7. Referências

- ABEVD. Associação Brasileira de Empresas de Vendas Diretas. Dados e informações. Disponível em: <<https://www.abevd.org.br/dados-e-informacoes/>>. Acesso em: 11 nov. 2024.
- BARROS, T; SILVA, M; ESPÍNOLA, E. State MVC: Estendendo o padrão MVC para uso no desenvolvimento de aplicações para dispositivos móveis. Centro de Estudos e Sistemas Avançados do Recife, 2007. Disponível em:<<https://www.tiagobarros.org/docs/SMVC.pdf>>. Acesso em 28 nov. 2024.
- BOOTSTRAP. Site Oficial Bootstrap. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 10 out. 2024.
- CAMPOS, André G. et al. Instituições trabalhistas e produtividade do trabalho: uma análise do caso brasileiro. In: NEGRI, João Alberto De; ARAÚJO, Bruno César; BACELETTE, Ricardo (Org.). Desafios da Nação: artigos de apoio. Brasília: Instituto de Pesquisa Econômica Aplicada (Ipea), 2018. cap. 16, p. 13-44. v. 2.
- CHARTS.JS. Site Oficial Charts.JS. Disponível em: <<https://www.chartjs.org/docs/latest/getting-started/>>. Acesso em: 10 out. 2024.
- COMPOSER. Site Oficial Composer. Disponível em: <<https://getcomposer.org/>>. Acesso em: 5 out. 2023.
- CORRÊA, Henrique Luiz; CORREA, Carlos Alberto. Administração de produção e operações: manufatura e serviços: uma abordagem estratégica. São Paulo: Atlas, 2006. 689 p.
- DBAVER.Site Oficial DBEaver Community. Disponível em; <<https://dbeaver.io/>>. Acesso em 27 nov. 2024.
- DEVMEDIA. DAO Pattern: Persistência de Dados utilizando o padrão DAO. Disponível em: <<https://www.devmedia.com.br/dao-pattern-persistencia-de-dados-utilizando-o-padrao-dao/30999>>. Acesso em: 28 nov. 2024.
- FERREIRA, Natália Trigo. Rotatividade no Mercado de Trabalho e Incentivos do Arcabouço Institucional Brasileiro. Pontifícia Universidade Católica Rio de Janeiro. Rio de Janeiro, 2018.
- GOMES, Gustavo. O que é Funil de Vendas. Disponível em: <<https://www.agendor.com.br/blog/o-que-e-funil-de-vendas/>>. Acesso em: 7 nov. 2024.
- GOOGLE MAPS PLATFORM. Maps JavaScript API. Disponível em: <<https://developers.google.com/maps/documentation/javascript?hl=pt-br>>. Acesso em: 10 ago. 2024.
- GONZAGA, Gustavo; PINTO, Rafael Cayres. Rotatividade do Trabalho e Incentivos da Legislação Trabalhista. In: BONELLI, Regis; VELOSO, Fernando (Org.). Panorama do mercado de trabalho no Brasil. 1. ed. Rio de Janeiro: FGV, 2014. p. 181-199.

GRASSI, Vinicius. Estratégias de Localização de uma Rede de Supermercados: o geomarketing aplicado à Companhia Zaffari em Porto Alegre. Trabalho de Conclusão de Curso – Universidade Federal do Rio Grande do Sul. Porto Alegre, 2010.

JQUERY. Site Oficial JQuery. Disponível em: <<https://jquery.com/>>. Acesso em: 10 out. 2024.

KOTLER, Philip; KELLER, Kevin Lane; CHERNEV, Alexander. Administração de marketing. 16. ed. Porto Alegre, RS: Bookman, 2024. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 12 dez. 2024.

LUCIDCHART. Site oficial Lucidchart. Disponível em: <<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>>. Acesso em: 7 nov. 2024.

MDN, Mozilla Web Docs. Aprenda a estilizar HTML utilizando CSS. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Learn/CSS>>. Acesso em: 22 nov. 2024.

MDN, Mozilla Web Docs. Tecnologia Web para desenvolvedores HTML. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Learn/HTML>>. Acesso em: 22 nov. 2024.

MDN, Mozilla Web Docs. Tecnologia Web para desenvolvedores HTTP. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP>>. Acesso em: 22 nov. 2024.

MICROSOFT VISUAL STUDIO. Site Oficial Microsoft. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 26 nov. 2024.

MYSQL. Why MySQL?. Disponível em: <<https://www.mysql.com/why-mysql/>>. Acesso em: 20 nov. 2024.

ORACLE. O que é um banco de dados relacional (RDBMS)?. Disponível em: <<https://www.oracle.com/br/database/what-is-a-relational-database/>>. Acesso em: 15 nov. 2024.

PHP. A popular general-purpose scripting language that is especially suited to web development. Disponível em: <<https://www.php.net/>>. Acesso em: 20 nov. 2024.

PIPEDRIVE. 10 dicas para empresas pequenas usando vendas diretas. Disponível em: <<https://www.pipedrive.com/pt/blog/tecnica-venda-vendas-diretas>>. Acesso em: 11 nov. 2024.

PEREIRA. André Luis Soares. A história do comércio e suas curiosidades. Disponível em: <<https://www.soarespereira.com.br/blog/a-historia-do-comercio/>>. Acesso em 30 nov. 2024.

POSTMAN. Site oficial Postman. Disponível em: <<https://www.postman.com/>>. Acesso em: 17 out. 2024: 25 nov. 2024.

SBC – SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Modelo para publicação de artigos. Disponível em: <<https://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros>>. Acesso em: 25 nov. 2024.

SEBRAE. “Decifrando os canais de venda: Conceitos e aplicações.” Disponível em: <<https://sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/MS/Anexos/Decifrando-os-canais-de-venda.pdf>>. Acesso em: 7 nov. 2024.

SEBRAE. “As vantagens do geomarketing” Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/artigos/as-vantagens-do-geomarketing,5bfe16d8d6f28810VgnVCM1000001b00320aRCRD#:~:text=O%20geomarketing%20utiliza%20a%20intelig%C3%Aancia,campanhas%20de%20forma%20mais%20assertiva.>>>. Acesso em: 17 dez. 2024.

SLIM FRAMEWORK. Site oficial do projeto Slim Framework. Disponível em: <<https://www.slimframework.com/>>. Acesso em: 17 out. 2024.

SOMMERVILLE, Ian. Software Engineering. 10. ed. São Paulo, SP, Pearson Education Brazil, Inc., 2018. 49-104 p. Disponível em: <<https://archive.org/details/sommerville-engenharia-de-software-10e>>. Acesso em: 30 nov. 2024.

SYDLE. Indicadores de Produtividade: o que são e como medi-los? Disponível em: <<https://www.sydle.com/br/blog/indicadores-de-produtividade-60c3708b688db6117f3d5c09>>. Acesso em: 9 nov. 2024.

TOTVS. Front end: O que é, como funciona e qual a importância. Disponível em: <<https://www.totvs.com/blog/developers/front-end/>>. Acesso em: 15 nov. 2024.

XAMPP, Apache + MariaDB + PHP + Perl. Site oficial do software. Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em 14 nov. 2024.