

Use of ChatGPT in Low-Code Development: An Exploratory Study in the Salesforce Platform

Odimar Falcão Filho¹, Brenno Buarque¹, Camilo Almendra²

¹Campus Quixadá – Universidade Federal do Ceará (UFC)
Quixadá, Brazil

²Departamento de Computação – Universidade Federal do Ceará (UFC)
Fortaleza, Brazil

falcaof14@alu.ufc.br, brennobuarque@ufc.br, camilo.almendra@ufc.br

Abstract. Research Context: Large Language Models (LLMs) have emerged as promising resources to support software development. Beyond code generation, these tools can serve as technical and educational support, assisting developers in learning and applying concepts in complex low-code development platforms (LCDP). **Scientific and/or Practical Problem:** While LCDPs lower entry barriers, organizations still struggle to adopt them. As a first step, we investigate whether LLMs can effectively assist LCDP's newcomers in learning, producing and deploying quality enterprise applications. **Proposed Solution and/or Analysis:** This paper presents an exploratory study on the use of ChatGPT to support the development of a Salesforce application. We focus on a zero-shot learning approach to resemble the profile of newcomers. The study aims to identify the suitability of LLMs to assist beginner low-code developer. **Related IS Theory:** ChatGPT is seen as part of a socio-technical network that transforms low-code development. Information systems are complex and require interoperability; in this context, generative AI expands accessibility and innovation. **Research Method:** Development of a low-code application through prompts, using ChatGPT as support. Evaluation of the application based on quantitative and qualitative metrics. **Summary of Results:** Simple requirements were often solved with a single prompt, whereas the most complex required iterations and significant debugging, revealing the model's limits. Expert reviewers found the AI-generated solutions correct and aligned with Salesforce best practices, but noted that the code lacked structural best practices. **Contributions and Impact to IS area:** Contributes to SI by demonstrating that ChatGPT is a co-creative, educational partner that can boost development productivity and democratize development by lowering technical barriers. However, human expertise remains critical. This work demonstrates how strategic AI integration can enhance system delivery while supporting developer learning.

1. Introdução

A crescente demanda por soluções digitais tem impulsionado a adoção das **Low-Code Development Platforms** (LCDPs) como uma abordagem alternativa para acelerar o desenvolvimento de soluções digitais. Segundo [Matvitsky et al. 2024], até 2028, 60% das organizações de software utilizarão LCDPs, como sua principal plataforma interna de desenvolvimento, ante 10% em 2024. Nesse contexto, surge um novo paradigma chamado

Low-code Systems Development (LCSD), uma abordagem de programação visual auxiliada por interfaces gráficas de usuário e designs orientados a modelo que permite o desenvolvimento de software com pouca ou nenhuma codificação [Al Alamin et al. 2021]. O principal objetivo das plataformas low-code é aumentar a produtividade no desenvolvimento de software, reduzindo a necessidade de conhecimento técnico profundo e permitindo que profissionais de diferentes áreas contribuam para a construção de soluções [Vincent et al. 2020]. O que torna as LCDPs atrativas para profissionais iniciantes ou com pouca experiência técnica, pois reduz barreiras tradicionais de entrada no desenvolvimento de software.

Além da dimensão tecnológica, o fenômeno envolve diretamente pessoas e processos. De acordo com [Gartner 2023], até 2026, 80% dos usuários dessas ferramentas estarão fora das áreas tradicionais de TI, um aumento expressivo em relação aos 60% registrados em 2021. Essa adesão pode mudar o cenário de desenvolvimento de software, corroborando com o conceito de quarta geração de linguagens de programação de [Moskal 2021]. Reforçando o papel do *low-code* na democratização do desenvolvimento de software, permitindo que profissionais de diversas áreas participem ativamente na construção de soluções tecnológicas.

Como destacam [Bruhin et al. 2024], as LCDPs surgiram como uma solução crucial, permitindo ciclos de desenvolvimento mais rápidos e tornando a criação de software mais acessível a diferentes perfis de profissionais. Essa tendência reflete a capacidade dessas plataformas de atender a um público mais amplo e diversificado, como destacado por [Moskal 2021], ao afirmar que LCDPs podem ser usadas não apenas por programadores para acelerar o processo de desenvolvimento de software, mas também por desenvolvedores cidadãos, analistas de negócios, consultores ou *designers* de soluções para criar aplicativos de negócios sem uma linha de código, resolvendo o problema da escassez de programadores qualificados no mercado.

Nesse cenário, o tripé **Pessoas–Processos–Tecnologias** se manifesta de forma clara: pessoas com pouca experiência em programação passam a atuar em processos críticos de negócio, utilizando tecnologias como as LCDPs para construir soluções que impactam diretamente a dinâmica organizacional. Para auxiliar nessa transição, uma alternativa promissora é o uso de IA generativa baseada em *Large Language Models* (LLMs), como o ChatGPT. Essa ferramenta tem se destacado por sua capacidade de oferecer soluções versáteis, robustas e altamente personalizáveis, respondendo a perguntas, criando conteúdo, reescrevendo texto, desenvolvendo e depurando código, realizando testes, modificando dados, explicando e tutorando [Martins et al. 2023].

A capacitação de desenvolvedores é um fator decisivo para o sucesso da adoção de plataformas *low-code*. Profissionais precisam aprender não apenas os aspectos técnicos das LCDPs, mas também a conduzir processos de mudança organizacional. De acordo com [Martins et al. 2023] o ChatGPT pode desempenhar papel estratégico ao fornecer orientação durante o desenvolvimento, sugerindo abordagens para tarefas específicas e apoiando a tomada de decisão, permitindo assim ciclos de desenvolvimento mais rápidos e eficientes. Desse modo a IA pode reduzir barreiras de entrada para novos desenvolvedores e favorecer um processo de aprendizagem mais dinâmico e acessível.

Nesse contexto, a Salesforce se destaca como uma das líderes no mercado de pla-

taformas *low-code*, integrando recursos de inteligência artificial, automação de processos e conectores nativos que facilitam a construção de soluções empresariais em larga escala [Gartner 2025]. O Salesforce tem ganhado ainda mais relevância por ser uma das plataformas de gerenciamento de relacionamento com o cliente mais populares, amplamente utilizada por empresas para automatizar e otimizar seus processos de atendimento ao cliente. A plataforma combina robustez técnica e ecossistema global, mas também apresenta desafios ligados à complexidade de integração e à concentração de soluções em domínios de CRM, fatores que reforçam a importância de investigar como o uso do ChatGPT pode potencializar seu aproveitamento em cenários organizacionais.

Embora as plataformas *low-code* diminuam as barreiras iniciais de programação, organizações ainda enfrentam diversos desafios práticos para sua adoção plena. Em um estudo empírico, [Al Alamin et al. 2021] analisaram cerca de 5 mil postagens no Stack-Overflow, identificando os principais tópicos de dificuldades enfrentadas por desenvolvedores: personalização, que representa 40% das questões analisadas, especialmente na adaptação de interfaces e na lógica de negócios; adoção da plataforma, que envolve questões de configuração e suporte a funcionalidades; integração com sistemas externos, que inclui a conexão com APIs e serviços externos; e o gerenciamento de banco de dados, com desafios na manipulação e importação/exportação de dados.

Persiste o desafio de qualificação profissional, uma vez que desenvolvedores precisam adquirir conhecimento tanto sobre as ferramentas técnicas das LCDPs quanto sobre as práticas de negócio subjacentes, o que demanda tempo e investimento em treinamento. Nesse contexto, o problema central consiste em como capacitar rapidamente novos desenvolvedores nessas plataformas e, ao mesmo tempo, garantir que produzam aplicações de qualidade alinhadas às regras de negócio, em ambientes marcados por complexidade e necessidade de interoperabilidade, características essenciais dos Sistemas de Informação [Boscarioli et al. 2017].

Do ponto de vista científico, ainda há uma lacuna em compreender até que ponto ferramentas de IA podem mitigar esses desafios práticos, uma vez que, apesar do potencial de assistentes como o ChatGPT em fornecer apoio contextual, não se sabe quão efetivamente eles podem auxiliar iniciantes em *low-code* a aprender a plataforma e a desenvolver aplicações corporativas de qualidade. Assim, este trabalho investiga se LLMs podem atuar como copilotos de desenvolvimento para iniciantes em LCDPs, apoiando a aprendizagem de conceitos da plataforma, a produção de implementações corretas e a implantação de soluções alinhadas ao negócio, em consonância com a Visão Sociotécnica do GranDSI-BR [Boscarioli et al. 2017]. A compreensão desses limites e possibilidades é fundamental para o avanço da área de Sistemas de Informação, que entende a computação como um meio para resolver problemas no mundo real, na sociedade e nas organizações, e cujo caráter aplicado e multidisciplinar exige uma reflexão aprofundada sobre as tecnologias e seu desdobramento na prática [Boscarioli et al. 2017].

Para abordar o problema acima, este artigo apresenta uma investigação exploratória sobre o uso do ChatGPT como assistente no desenvolvimento *low-code* na plataforma Salesforce. Especificamente, conduziu-se um estudo de caso no qual o modelo de LLM ChatGPT, na versão GPT-4o foi utilizado como suporte durante a implementação de uma aplicação realista.. O estudo foi conduzido pelo próprio autor, que possui experiência prévia com a plataforma Salesforce, porém cuja formação técnica foi baseada

predominantemente em materiais de treinamento oficial, em especial os cursos e trilhas disponibilizados no ambiente Trailhead. Assim, embora não se trate de um usuário completamente leigo, o perfil considerado aproxima-se do de um desenvolvedor em processo de capacitação formal na plataforma.

Adotou-se uma estratégia de interação *zero-shot*, isto é, sem fornecer exemplos ou modelos prévios ao ChatGPT, simulando o perfil de um desenvolvedor iniciante que depende apenas do próprio enunciado dos requisitos e do conhecimento padrão da plataforma.

O caso de uso envolve a construção de um sistema de gestão para um pequeno frigorífico – denominado Boi Forte – abrangendo funcionalidades típicas como controle de estoque, registro de entradas/saídas, gestão de vendas, emissão de notas fiscais e relatórios gerenciais, com o desenvolvimento concluído, a aplicação passa por avaliação. No total, dez requisitos funcionais distintos (RF01–RF10) foram definidos para cobrir esse escopo operacional, e o desenvolvimento de cada um deles foi assistido pelo ChatGPT.

Essa abordagem visa contribuir para uma melhor compreensão do papel que o ChatGPT pode desempenhar no desenvolvimento de software *low-code*, destacando suas vantagens, desafios e melhores práticas no contexto específico da plataforma Salesforce. No contexto do Salesforce, o ChatGPT oferece um novo paradigma de colaboração e expertise. Não é uma mera ferramenta ou repositório, mas um parceiro ativo, envolvente e inestimável na navegação das complexidades do Salesforce. Seu vasto conhecimento, *recall* e interação dinâmica o tornam um ativo essencial para qualquer um que trabalhe no ecossistema do Salesforce, promovendo eficiência, inovação e excelência [Forbes et al. 2023].

2. Trabalhos Relacionados

[Martins et al. 2023] investigam a integração do ChatGPT em plataformas *low-code* como forma de ampliar eficiência e acessibilidade no desenvolvimento, propondo um modelo híbrido em que a IA apoia todas etapas do ciclo de desenvolvimento de software, desde prototipação e design inicial até personalização de fluxos, integrações com sistemas externos e atividades de teste. O estudo é estruturado sob *Design Science Research*, partindo do problema de escassez de desenvolvedores qualificados e da necessidade de reduzir barreiras para implementação e evolução de soluções.

A avaliação é conduzida por meio de grupos focais com profissionais experientes, em cenários simulados próximos à prática. Os achados apontam que o ChatGPT tende a ser útil em tarefas operacionais recorrentes (como geração de código, integração de APIs e construção de interfaces), mas que a validação e o refinamento humano seguem necessários para garantir qualidade e precisão, sobretudo quando aumentam a complexidade e as dependências do contexto.

[Monteiro et al. 2025] apresentam um relato empírico de construção ponta a ponta de software com apoio exclusivo do ChatGPT, buscando evidências mais concretas sobre o desempenho de LLMs em um desenvolvimento completo (do entendimento do requisito à implementação e correção). O estudo compara uma aplicação de referência desenvolvida manualmente com a replicação feita por participantes com níveis distintos de experiência, analisando o processo

Os resultados indicam que a conclusão do sistema é viável para parte dos participantes, mas depende fortemente da estratégia de interação e da capacidade técnica do usuário para depuração e integração. Os autores observam diferenças relevantes entre abordagens *top-down* e *bottom-up*, e mostram que dificuldades em integração entre componentes (por exemplo, entre *front-end* e *back-end*) podem inviabilizar a finalização quando o participante tem menor experiência, reforçando limitações práticas do uso de LLMs como suporte primário.

[Denny et al. 2023] discutem implicações do avanço de LLMs para o ensino de programação e argumentam que a habilidade de formular bons *prompts* precisa ser tratada como competência explícita, dado o risco de dependência da IA sem compreensão adequada dos conceitos. Para isso, propõem os *Prompt Problems*, em que aprendizes elaboram e refinam instruções em linguagem natural para orientar a geração de soluções, operacionalizados pela ferramenta Promptly com avaliação automática e iteração até a resposta correta.

A avaliação empírica sugere que mesmo participantes experientes enfrentam dificuldades iniciais na formulação de *prompts* claros, e que o processo de tentativa, ajuste e validação promove uma forma específica de raciocínio computacional centrada em compreender requisitos, traduzi-los em instruções e criticar a saída do modelo. Esses achados reforçam que o uso efetivo de IA geradora de código envolve interação qualificada, não apenas consumo passivo das respostas.

2.1. Análise Comparativa

Em conjunto, os estudos indicam que LLMs podem reduzir esforço em tarefas recorrentes, mas que resultados funcionais dependem do *prompting* e da iteração.[Martins et al. 2023] defendem essa integração em LCDPs por meio de um modelo híbrido avaliado com especialistas, enquanto [Monteiro et al. 2025] mostram, em um cenário *end-to-end*, que o sucesso é sensível à estratégia de interação e à experiência do usuário.[Denny et al. 2023] complementam ao tratar o *prompting* como habilidade-chave para obter saídas mais adequadas e confiáveis.

Assim, os trabalhos analisados complementam a proposta deste trabalho ao fornecerem fundamentos teóricos e exemplos práticos relevantes para o uso de tecnologias *low-code* e inteligência artificial generativa. No entanto, minha pesquisa se destaca ao unir esses elementos em um contexto específico, explorando como o ChatGPT pode capacitar desenvolvedores e oferecer soluções práticas e inovadoras no ambiente Salesforce. Ao adotar uma abordagem experimental, voltada para a criação e avaliação de *prompts* em cenários reais, o estudo oferece evidências empíricas para compreender o potencial de serviços de IA generativa como suporte ao desenvolvimento *low-code*, ampliando o debate sobre suas aplicações práticas e educativas.

3. Teoria de SI Relacionada

Actor–Network Theory (ANT) constitui uma abordagem crítica relevante para os estudos em Sistemas de Informação, ao enfatizar que revelar as redes sociotécnicas já é, por si só, um ato de crítica [Doolin and Lowe 2002]. Essa perspectiva é particularmente pertinente ao presente estudo, uma vez que o uso do ChatGPT no desenvolvimento em LCDPs não pode ser entendido apenas como uma ferramenta técnica, mas como um ator inserido

em uma rede mais ampla de desenvolvedores, plataformas, práticas de engenharia de *prompts* e métricas de avaliação. Ao adotar essa visão, torna-se possível compreender o ChatGPT como elemento que não apenas apoia, mas também transforma a dinâmica do desenvolvimento, expondo negociações, dependências e novas formas de estabilização no processo de criação de sistema de informação.

A ampliação do público atendido pelas LCDPs, conforme destacado por [Moskal 2021], não apenas diversifica os perfis profissionais envolvidos, mas também potencializa o alcance dessas plataformas ao articular diferentes perspectivas e necessidades organizacionais. Nessa linha, [Rokis and Kirikova 2023] ressaltam que, ao possibilitar essa democratização no desenvolvimento de software, o LCSD promove a inovação digital ao fortalecer aspectos culturais como aprendizado e experimentação, criatividade, inovação e entrega rápida ao mercado.

A combinação entre a acessibilidade técnica das LCDPs e sua capacidade de transformar práticas organizacionais evidencia o caráter sociotécnico dessas ferramentas, conectando atores heterogêneos em torno de objetivos compartilhados de inovação. [Boscarioli et al. 2017] resalta que os SI devem ser entendidos como sistemas sociotécnicos, formados pela integração de pessoas, processos e tecnologias, e que só podem ser projetados, desenvolvidos e pesquisados a partir de abordagens que considerem essa complexidade.

A área de Sistemas de Informação reconhece que tais plataformas e tecnologias não podem ser analisadas apenas sob a ótica do desenvolvimento de software. [Boscarioli et al. 2017] apontam que os sistemas de informação vêm se tornando cada vez mais complexos, dada a quantidade e diversidade de elementos e interações que os compõem, muitas vezes distribuídos em organizações geograficamente dispersas. [KPMG 2024] destaca que o desenvolvimento *low-code* abrange inúmeras áreas de negócios e processos. Especificamente, aplicativos críticos de negócios, ERPs, CRM e SCM, 59% das empresas pesquisadas usam *low-code* para seu desenvolvimento.

A complexidade crescente dos Sistemas de Informação reforça ainda mais essa necessidade de abordagens críticas. De acordo com o GranDSI-BR, os SI atuais e futuros são constituídos por uma diversidade de componentes, muitas vezes distribuídos em diferentes plataformas e organizações, o que demanda interoperabilidade como requisito essencial para garantir eficiência e eficácia [Boscarioli et al. 2017]. Nesse sentido, a incorporação de recursos de IA generativa junto às LCDPs surge como alternativa para enfrentar tais desafios, ao facilitar a integração entre sistemas heterogêneos e apoiar a criação de soluções que atendam a demandas de mobilidade, colaboração e escalabilidade. Essa articulação entre complexidade técnica e diversidade organizacional reforça o argumento de que a adoção dessas tecnologias deve ser analisada sob uma ótica sociotécnica, em consonância com os grandes desafios de pesquisa em Sistemas de Informação no Brasil.

O caráter multidisciplinar da área de Sistemas de Informação amplia o escopo de análise do desenvolvimento *low-code* assistido por IA. Como destaca [Boscarioli et al. 2017], a computação deve ser compreendida como um meio para a resolução de problemas reais da sociedade e das organizações, exigindo que pesquisas em SI considerem não apenas as tecnologias construídas, mas também seus impactos so-

ciais, organizacionais e culturais. Essa visão converge com a perspectiva da ANT, ao enfatizar que os sistemas não se restringem a linhas de código ou a usuários individuais, mas emergem de redes compostas por pessoas, processos, práticas institucionais e artefatos tecnológicos. Dessa forma, o desenvolvimento em LCDPs e o uso de IA generativa devem ser entendidos como parte de uma transformação sociotécnica mais ampla, que reorganiza papéis, práticas e valores no ecossistema da inovação digital.

4. Metodologia

Esta pesquisa combinou uma abordagem exploratória e prática, envolvendo a compreensão da plataforma, o planejamento de uma aplicação de exemplo, o desenvolvimento assistido pela IA e a avaliação dos resultados. A Figura 1 apresenta um fluxograma com as etapas gerais do método. Os artefatos produzidos durante a pesquisa estão disponíveis neste link:

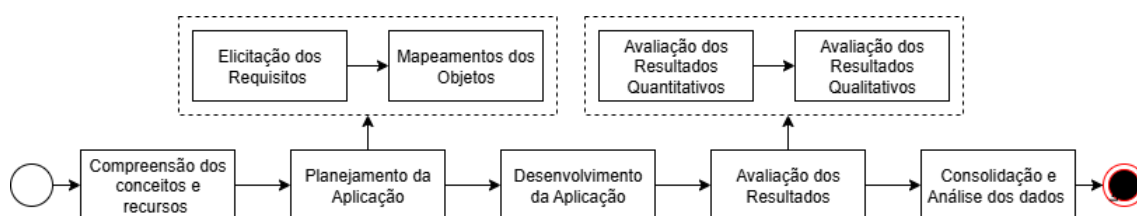


Figura 1. Fluxograma da metodologia.

4.1. Compreensão da plataforma

Antes da implementação, foi necessário conhecer os recursos oferecidos pela Salesforce. A plataforma oferece ferramentas visuais e componentes declarativos que possibilitam a criação de aplicações sem código tradicional, tais como:

- *App Builder*, utilizado para construir páginas e interfaces de forma visual;
- *Flow Builder*, que permite criar fluxos de trabalho baseados em regras de negócio e interações com o usuário;
- *Object Manager* e *Schema Builder*, responsáveis pela modelagem e visualização da estrutura de dados;
- *Experience Builder*, voltado à construção de portais e sites; e
- recursos de código como *Apex* e *Lightning Web Components* (LWC), utilizados quando funcionalidades específicas exigem programação.

Para aprofundar esse conhecimento foram utilizados módulos do *Trailhead* sobre modelagem de dados, noções de *Flow Builder* e desenvolvimento em *Apex*. Essa ambientação inicial permitiu interpretar e validar as respostas geradas pelo ChatGPT com maior criticidade.

4.2. Planejamento da aplicação

Compreendida a plataforma, definiu-se um cenário de negócio realista a fim de testar o ChatGPT como copiloto de desenvolvimento. O estudo escolheu como caso de uso a criação de um sistema de gestão para um frigorífico de pequeno porte (denominado *Boi Forte*), envolvendo operações como controle de estoque, registro de entradas e saídas, gestão de vendas, relatórios e emissão de notas fiscais.

Foi elaborado um documento de requisitos contendo dez requisitos funcionais (RF01–RF10) que representavam as demandas operacionais do frigorífico (disponível em [Falcão Filho et al. 2026]). Esses requisitos serviram como guia para toda a implementação e permitiram avaliar a capacidade da IA em lidar com diferentes níveis de complexidade. O documento descrevia as funcionalidades do sistema e as regras de negócio correspondentes.

4.2.1. Mapeamento de objetos e planejamento da implementação

Em seguida, os requisitos foram fornecidos ao modelo GPT-4o em uma conversa única (*prompts* disponíveis em [Falcão Filho et al. 2026]). O ChatGPT recebeu a orientação de atualizar sua memória com esse conteúdo para compreender plenamente o escopo do projeto. O modelo foi instruído a mapear os objetos Salesforce necessários e a sugerir um plano de implementação detalhado. A estratégia metodológica priorizou o uso de objetos padrão para representar clientes, produtos, vendas e pedidos, e a criação de objetos personalizados apenas quando o domínio exigia relações não contempladas nativamente. O mapeamento e o plano de implementação foram utilizados como referência nas etapas seguintes.

4.3. Desenvolvimento da aplicação

A construção da aplicação *Boi Forte* seguiu uma abordagem incremental baseada nos requisitos definidos. Cada funcionalidade foi implementada isoladamente, permitindo acompanhar com precisão a performance do ChatGPT em cada etapa. A IA atuou como agente de consulta, geradora de código e esclarecedora de dúvidas técnicas. Para manter o controle metodológico, cada requisito foi iniciado com um *prompt* neutro contendo apenas a descrição do requisito. Essa escolha eliminou direcionamentos sobre a solução, permitindo que o modelo interpretasse o problema de forma autônoma.

Essa decisão metodológica reflete a proposta central deste trabalho: avaliar o desempenho da IA não apenas como uma ferramenta auxiliar de codificação, mas como um agente ativo no processo de desenvolvimento de software *low-code*. Ao confiar ao modelo a liberdade para definir a abordagem técnica mais adequada para cada caso, tornou-se possível observar com maior nitidez sua capacidade de articular soluções coerentes, explorar os recursos disponíveis na plataforma e estruturar as funcionalidades.

4.3.1. Sequência de implementação e intervenções técnicas

Os requisitos foram implementados sequencialmente; cada nova tarefa começava apenas após a conclusão da anterior. O pesquisador permaneceu responsável por testar as implementações, ajustar nomenclaturas ou erros de sintaxe e garantir o funcionamento correto. Algumas correções foram pontuais (e.g., nomes de objetos ou campos) e não exigiram novos *prompts*. Em outros casos, falhas de lógica, erros de *deploy* ou inconsistências no fluxo exigiram enviar novos *prompts* ao ChatGPT; nesses casos, os *prompts* utilizados continham o contexto sobre o erro para que o modelo propusesse soluções. Após cada implementação, a funcionalidade era validada no ambiente Salesforce, verificando conformidade com as regras de negócio e corrigindo erros identifica-

dos. O diagrama de entidades e relacionamentos sugeridos pelo modelo de linguagem, assim como o projeto completo exportado da plataforma Salesforce está disponível em [Falcão Filho et al. 2026].

4.4. Avaliação

O desempenho do ChatGPT como ferramenta de apoio foi analisado de maneira quantitativa e qualitativa, permitindo medir a eficácia e a percepção dos profissionais sobre as soluções geradas.

4.4.1. Avaliação quantitativa

A avaliação quantitativa baseou-se no modelo Goal–Question–Metric (GQM). Esse modelo estrutura a análise em três níveis: metas, perguntas e métricas. Foram definidas três métricas principais:

- **Número de reformulações (M1)** – contabiliza quantas vezes foi necessário reformular um *prompt* para obter uma resposta satisfatória; mede a complexidade da interação com a IA.
- **Número de erros detectados nos testes (M2)** – registra falhas de execução, lógica ou inconsistências funcionais nos artefatos gerados; avalia o impacto da IA na qualidade técnica da solução.
- **Tempo estimado por etapa (M3)** – estima o esforço (em horas) empregado na implementação e validação de cada requisito.

Durante o desenvolvimento, os *prompts* e respostas do ChatGPT foram classificadas segundo sua utilidade prática. A resposta era considerada **correta** quando permitia avançar no desenvolvimento e atendia à funcionalidade principal do requisito, como no RF01, em que a solução proposta gerou a entrada do produto no estoque. A resposta era classificada como **parcial** quando a funcionalidade apresentava erros ou lacunas que exigiam ajustes, como no RF09, em que a emissão da nota fiscal ocorreu, mas alguns campos não foram retornados adequadamente. Por fim, a resposta era considerada **errada** quando a solução resultava em erro de compilação ou execução, inviabilizando sua aplicação, como nos casos de objetos personalizados definidos via XML que não compilavam na plataforma Salesforce. A rotulação das saídas seguindo essa categorização foi realizada pelos autores.

4.4.2. Avaliação qualitativa

Na avaliação qualitativa, a pesquisa buscou compreender a efetividade das soluções do ponto de vista de profissionais experientes na plataforma Salesforce.

Foram realizadas entrevistas com três profissionais especialistas em Salesforce (ver Tabela 1), todos com mais de cinco anos de experiência em desenvolvimento de sistemas *low-code* e nível de senioridade pleno, sendo dois arquitetos de soluções e um analista funcional. Conduzimos entrevistas semiestruturadas de aproximadamente 30 minutos, realizadas via Google Meet. Os participantes analisaram três requisitos (RF01, RF02 e RF08) que abrangiam diferentes camadas do sistema – utilização de objetos padrão,

criação de objetos personalizados, automações com Flow/Trigger e componentes LWC. As entrevistas investigaram a clareza das implementações, aderência às boas práticas, robustez técnica e aplicabilidade no contexto profissional. Após as entrevistas, cada participante respondeu a um questionário para registrar percepções sobre usabilidade, robustez técnica, conformidade com padrões da plataforma e aplicabilidade das soluções. Esse instrumento complementou as entrevistas e permitiu quantificar a percepção dos especialistas.

Tabela 1. Perfil dos Entrevistados

Entrevistado	Cargo/Função	Experiência	Senioridade
Entrevistado 1	Arquiteto de Soluções	5 anos	Pleno
Entrevistado 2	Arquiteto de Soluções	6 anos	Pleno
Entrevistado 3	Analista Funcional	7 anos	Pleno

4.4.3. Análise de dados

Os dados quantitativos foram coletados manualmente durante a implementação de cada requisito e organizados em planilhas de acordo com as métricas GQM. Para a métrica M1, cada reformulação de *prompt* foi contabilizada, ou seja quando a funcionalidade do requisito não era atendida e eram necessários novos comandos para um mesmo erro. Para a métrica M2, registraram-se falhas encontradas durante os testes, no RF09 por exemplo erro; campos em branco no retorno de um Nota fiscal. Para a métrica M3, o pesquisador estimou o tempo gasto ao final de cada ciclo de desenvolvimento. As respostas às entrevistas foram transcritas e categorizadas por temas, enquanto os resultados do questionário foram tabulados automaticamente pelo Google Forms. Na análise final, as evidências qualitativas e quantitativas foram consolidadas para responder às perguntas de pesquisa e identificar tanto os benefícios quanto as limitações do ChatGPT.

5. Resultados

5.1. Avaliação Quantitativa

Com base no processo de coleta descrito na Seção 4.4, foi possível organizar os dados obtidos em uma matriz comparativa, apresentada na Tabela 2. Nela estão consolidados os principais indicadores extraídos para cada um dos dez requisitos funcionais desenvolvidos. Os requisitos foram classificados de acordo com o grau de complexidade envolvido na implementação, que foi determinado com base nas características técnicas de cada funcionalidade. A categorização foi feita da seguinte maneira:

- **Simple:** Requisitos que exigem pouca interação entre os objetos e apresentam lógicas de negócios diretas, sem a necessidade de personalizações complexas. No RF01 e RF02 que são condições simples de cadastro
- **Médio:** Requisitos que envolvem uma interação maior entre os objetos e lógicas intermediárias, exigindo um nível de personalização mais detalhado sendo necessário flow e código Apex.
- **Complexo:** Requisitos com múltiplas interações entre objetos e lógicas de negócios detalhadas, que exigem integração entre componentes padrão e personalizados da plataforma exigência de mais de um tipo de flow, desenvolvimento de código Apex .

Tabela 2. Requisitos do Sistema

Requisito	Complexidade	Descrição	# de Reformulações (M1)	Falhas (M2)	Tempo (min) (M3)
RF01	Simples	Entrada de Produtos	6	0	90
RF02	Médio	Estoque em Tempo Real	2	2	30
RF03	Médio	Atualização Automática	10	3	50
RF04	Simples	Gestão de Vendas	0	0	30
RF05	Simples	Histórico de Movimentações	0	0	5
RF06	Simples	Compras por Cliente	0	0	20
RF07	Simples	Relatórios Operacionais	1	1	15
RF08	Complexo	Compra Recorrente	11	6	120
RF09	Complexo	Emissão de Nota Fiscal	25	13	145
RF10	Médio	Alertas de Estoque Baixo	5	3	50

A primeira métrica avaliada foi a quantidade de reformulações de *prompt* necessárias durante o desenvolvimento de cada requisito funcional. Observando o número de reformulações de *prompt* por requisito da m1 2, percebe-se uma variação significativa conforme a complexidade de cada funcionalidade. Requisitos de lógica mais simples ou cuja implementação se beneficiou do reaproveitamento de estruturas previamente configuradas, como RF04 (Gestão de Vendas) e RF05 (Histórico de Movimentações), não demandaram reformulações de *prompt* (M1 = 0). Nesses casos, o modelo foi capaz de oferecer soluções coerentes já na primeira interação, não sendo necessário reformular a pergunta original para corrigir falhas ou inconsistências.

No caso do RF04, a clareza da demanda e a aderência da tarefa a padrões recorrentes do Salesforce permitiram que o ChatGPT oferecesse uma solução técnica coesa já na primeira interação, evidenciando seu potencial como agente facilitador no desenvolvimento de funcionalidades simples e bem definidas.

Em contraste com os requisitos mais simples, o **RF09 (Emissão de Nota Fiscal)** apresentou o maior número de reformulações de *prompt* (25), reflexo de sua complexidade técnica. A solução envolvia múltiplas etapas interdependentes: criação do objeto personalizado *Nota Fiscal*, implementação de *triggers* em *Opportunity* e *Order*, simulação da emissão por meio de uma classe *Apex* e construção de uma página *Visualforce* para exibição da nota.

Esses casos reforçam que, em tarefas com lógica linear e dependência reduzida entre objetos, o modelo é capaz de propor soluções eficazes com poucas iterações, tornando o processo de desenvolvimento mais fluido e direto. Conforme demonstrado por [Bucaioni et al. 2024], ChatGPT obteve taxa de sucesso de 90% já na primeira tentativa para a maioria dos problemas, atingindo 100% com apenas duas interações em tarefas fáceis e médias.

A segunda métrica analisada refere-se à quantidade de erros identificados durante os testes práticos das funcionalidades desenvolvidas. O bom desempenho nas funcionalidades simples pode ser atribuído ao uso de recursos nativos do Salesforce e à lógica direta envolvida. Por exemplo, no RF04, o ChatGPT associou automaticamente os preços corretos de acordo com o tipo de venda, utilizando um objeto padrão do Salesforce *PriceBook*. Já no RF06, o modelo sugeriu o uso de relatórios e *dashboards* nativos, o que viabilizou uma solução funcional para análise de compras. Esses exemplos ilustram a capacidade do ChatGPT de propor respostas coerentes com os padrões da plataforma quando os requisitos envolvem cenários mais lineares e pouco integrados.

Em contraste, o RF08 (Compra Recorrente), foram registrados seis erros durante os testes, reflexo das dificuldades em automatizar um processo com múltiplas dependências. O modelo falhou inicialmente ao não prever que a criação de um Order só seria possível mediante a existência de um Contrato ativo. Esse equívoco resultou em falhas na execução do fluxo proposto. Além disso, houve dificuldade na associação correta dos OrderItems ao PricebookEntry etapa fundamental para que a precificação dos produtos ocorresse corretamente.

A ausência dessas associações revelou que o modelo não foi capaz de antecipar completamente as interdependências necessárias entre os objetos envolvidos. Esse tipo de erro indica que, em requisitos mais elaborados, o ChatGPT tende a apresentar soluções incompletas quando não há uma orientação precisa por parte do desenvolvedor. Esse comportamento também foi observado por [Yan et al. 2023], que identificaram a necessidade de um processo iterativo para que o modelo possa refinar suas respostas.

A terceira métrica analisada refere-se ao tempo estimado para implementação de cada requisito funcional, considerando desde a elaboração dos *prompts* até a validação da solução final. Os requisitos RF02, RF04, RF05 e RF06 apresentaram tempos de desenvolvimento significativamente mais baixos em comparação aos demais. Esse desempenho está relacionado tanto à clareza das tarefas quanto à aderência das soluções sugeridas aos recursos nativos da plataforma Salesforce. No RF02 (Estoque em Tempo Real), por exemplo, o ChatGPT produziu um componente LWC funcional logo na primeira tentativa, exigindo apenas ajustes pontuais por parte do desenvolvedor para a solução ser totalmente contemplada.

O RF05 (Histórico de Movimentações) foi finalizado em apenas cinco minutos, pois sua lógica já havia sido parcialmente implementada durante o desenvolvimento dos requisitos RF01 e RF03. O registro de movimentações foi contemplado no flow de entrada e na *trigger* de saída de produtos, antecipando de forma não intencional a maior parte da lógica necessária para atender ao RF05. Esse desempenho ilustra o papel duplo do ChatGPT, que é capaz de ir além da simples geração de código para atuar como um assistente integrado, apoiando diferentes atividades do ciclo de vida do desenvolvimento e promovendo a otimização dos processos [Li et al. 2025].

Os requisitos RF08 e RF09 demandaram o maior tempo de desenvolvimento, reflexo direto da complexidade envolvida e das limitações do ChatGPT ao lidar com regras de negócio mais sofisticadas. Em ambos os casos, o modelo foi capaz de iniciar a construção da solução, mas não previu interdependências cruciais entre objetos nem os fluxos de dados exigidos pela plataforma Salesforce. Isso gerou a necessidade de diversos ajustes manuais, principalmente em pontos como contratos ativos, associações com PricebookEntry e renderização de páginas Visualforce.

Embora o ChatGPT tenha sido útil na construção das primeiras versões, o número de detalhes e ajustes posteriores demandados expôs seus limites como ferramenta autônoma de desenvolvimento. Essa limitação é consistente com as descobertas de [Liu et al. 2024], que demonstram que LLMs, ao receberem requisitos gerais, frequentemente geram apenas estruturas de código básicas que servem como ponto de partida, mas não cumprem todos os requisitos funcionais de alto nível.

5.2. Avaliação Qualitativa

De modo geral, as soluções propostas pelo ChatGPT foram consideradas satisfatórias para o escopo do sistema, com boa aderência funcional aos requisitos e alinhamento com práticas típicas da plataforma Salesforce. Os avaliadores destacaram positivamente o uso de abordagens *low-code*, com ênfase no emprego de Flows automatizados e na estruturação de objetos personalizados.

Um entrevistado ressaltou que a abordagem de *low-code* proposta pelo ChatGPT resultou em soluções simples, robustas e escaláveis, evitando o uso de recursos desnecessários. Ele comentou que *“a qualidade geral das soluções é... uma solução simples, que pode ser escalável, sem fazer utilização de recursos desnecessários”*. Outro avaliador elogiou a implementação do componente LWC: *“o código está bem simples... está com o layout ajustadinho... está bom de ser utilizado; tem muito desenvolvedor que não consegue fazer isso”*.

A avaliação sobre a capacidade do ChatGPT em compreender e atender aos requisitos do sistema foi positiva. Os especialistas indicaram que, mesmo com um contexto limitado, o modelo foi capaz de propor soluções que contemplaram os objetivos centrais de cada funcionalidade.

Entretanto, outro entrevistado reconheceu que, embora a solução atenda aos requisitos centrais, certas especificidades do domínio ainda poderiam ser melhor contempladas, como o controle de peso, considerado recorrente na rotina de frigoríficos. Segundo ele, *“só acho que faltou o peso, o quilo, o quilograma... é mais recorrente nesse nicho de frigorífico”*.

Do ponto de vista técnico, os entrevistados avaliaram de forma positiva a escolha dos recursos utilizados na construção da aplicação, como Apex, Flow e componentes LWC. Os Flows se destacaram por sua clareza, baixo acoplamento e facilidade de manutenção. Um deles afirmou que usar Flow *“é bem robusto... simples de dar manutenção... e segue boas práticas low-code ao utilizar flow o máximo possível”*, e outro considerou que essa abordagem torna futuras modificações *“bem mais simples do que seria se fosse via código”*.

Por outro lado, embora as classes Apex tenham cumprido suas funções básicas, foram apontadas limitações estruturais relevantes, ausência de organização em camadas (como separação entre lógica de negócio e consultas) e falta de tratamento de exceções. Um especialista observou que *“em um ambiente real de produção... jamais seria feito [uma trigger sem handler]; tem que trabalhar com Map... e usar with security enforced”*.

Além de sua atuação como ferramenta de apoio direto à codificação, o ChatGPT foi percebido pelos avaliadores como um recurso com potencial educativo significativo. As soluções geradas pelo modelo, mesmo que simples em termos de estrutura, foram consideradas eficazes como ponto de partida para compreensão de fluxos e componentes dentro da plataforma Salesforce. Um dos especialistas comparou os códigos produzidos com os exemplos disponíveis no Trailhead, plataforma oficial de aprendizado da Salesforce, sugerindo que o ChatGPT pode assumir um papel complementar nesse processo de formação, ao oferecer exemplos práticos contextualizados conforme os desafios propostos pelo usuário.

Essa característica torna a IA particularmente útil para profissionais em fase de aprendizado ou que desejam experimentar abordagens funcionais sem depender exclusivamente de documentação formal. Um entrevistado comentou que, seguindo as respostas “*de cabo a rabo*”, o modelo conseguiu demonstrar funcionalidades e interações, e que poderia ser uma ferramenta de apoio semelhante ao Trailhead. Outro avaliador reforçou que “*com certeza*” o ChatGPT pode servir como recurso complementar de aprendizado. Ele também comentou que, “*se o desenvolvedor pedisse boas práticas de separação de classes, o ChatGPT provavelmente sugeriria a estrutura correta*”, evidenciando seu potencial como ferramenta de ensino interativa.

5.3. Discussão

Considerando a métrica M1 (número de reformulações), observou-se que a complexidade técnica dos requisitos influenciou diretamente a quantidade de interações necessárias com o modelo. Essa relação entre complexidade e número de reformulações também foi apontada por [Monteiro et al. 2025], ao descreverem situações em que os participantes de seu estudo precisaram de dezenas de interações para ajustar funcionalidades mais sofisticadas e resolver erros persistentes no código gerado pela IA.

Por outro lado, funcionalidades mais simples exigiram menos ajustes e apresentaram uma execução mais direta. [Monteiro et al. 2025] relata que tarefas com menor grau de complexidade demandam menos *prompts*, e que o detalhamento das instruções iniciais tem papel fundamental para o sucesso das primeiras respostas. Segundo os autores, *prompts* mais detalhados levam a resultados mais precisos pelo modelo GPT.

Considerando a métrica M2 (número de erros detectados), observou-se que funcionalidades com interdependência entre objetos apresentaram maior propensão a falhas. [Ryan et al. 2024] apontam que LLMs geram casos de teste apenas para caminhos mais comuns, deixando de cobrir ramificações profundas e tratamentos de erro. [Ryan et al. 2024] também destaca, apesar de captarem restrições contextuais, nem todas as relações internas e lógicas entre os elementos do código são plenamente representadas nas saídas do modelo.

A métrica M3 (tempo de desenvolvimento) mostrou-se diretamente proporcional à complexidade dos requisitos. Este achado é corroborado por [Monteiro et al. 2025], que observaram em seu estudo de caso que a correção de erros de integração entre múltiplos componentes foi a atividade que mais consumiu tempo e esforço cognitivo dos desenvolvedores.

Na avaliação da qualidade geral, os especialistas consideraram as soluções satisfatórias, mas apontaram que a simplicidade das classes Apex, embora funcionais, poderia comprometer a escalabilidade. Essa dualidade, onde a IA gera código funcional que ainda carece de maturidade arquitetônica, é relatada em outros estudos. Como o de [Liu et al. 2024] analisou uso de LLMs em sistemas críticos e concluíram que, apesar de atenderem aos requisitos funcionais, os modelos muitas vezes não seguem padrões técnicos e de segurança sem orientação clara, o que reforça a importância da supervisão.

Na avaliação da aderência aos requisitos, a análise qualitativa indicou que o ChatGPT compreendeu os objetivos centrais, mas falhou em capturar detalhes específicos do domínio. Essa limitação é discutida por [White et al. 2023], que afirmam que a qualidade da resposta de um LLM depende diretamente da qualidade do *prompt*. A dificuldade

do modelo em entender regras implícitas, como o controle de peso, valida a tese de que a IA, sem um *prompt* bem detalhado, opera com uma compreensão superficial do problema

Do ponto de vista técnico, a necessidade de refatoração do código Apex e a adoção de padrões como Trigger Handlers foram pontos críticos levantados pelos especialistas para garantir a manutenibilidade. Essa preocupação com a qualidade técnica é alinhada às observações de [Martins et al. 2023], que embora a IA possa gerar código e automatizar tarefas, a validação e o refinamento humano são indispensáveis para assegurar a qualidade e precisão do produto final.

Na avaliação do ChatGPT como suporte técnico e educacional, a análise qualitativa revelou que os especialistas o perceberam como um recurso com grande potencial educativo, atuando como um complemento à plataforma Trailhead. Esta visão é fortemente corroborada por [Denny et al. 2023], que defendem o ensino da habilidade de induzir a geração de código por meio de instruções precisas, em vez de focar apenas na escrita manual. Para os autores, essa prática posiciona a interação com a IA como uma nova competência fundamental para desenvolvedores em formação.

De forma geral, os achados indicam que, embora o ChatGPT apresente limitações em tarefas com lógica complexa ou que exigem conhecimento contextual aprofundado, sua aplicação no desenvolvimento *low-code* demonstra potencial significativo. A relação observada entre complexidade dos requisitos, desempenho da IA e necessidade de refinamento reforça seu papel como ferramenta de apoio técnico. Além disso, os especialistas destacaram seu valor educativo, para iniciantes, ao oferecer soluções compreensíveis e funcionais que facilitam a exploração da plataforma Salesforce.

6. Conclusões

A crescente adoção da inteligência artificial tem transformado a forma como soluções digitais são desenvolvidas. Nesse cenário, o uso do ChatGPT como apoio em LCDPs, como a Salesforce, mostra-se uma alternativa promissora para ampliar a produtividade, reduzir barreiras técnicas e democratizar o desenvolvimento de software. Este trabalho avaliou, por meio de uma aplicação prática, como a IA pode contribuir para a criação de sistemas funcionais e servir como recurso de apoio à aprendizagem, auxiliando profissionais em formação a entenderem melhor os processos e ferramentas envolvidos no desenvolvimento *low-code*.

Os resultados indicaram que, embora o uso do ChatGPT proporcione ganhos relevantes em termos de agilidade e acessibilidade, ainda exige cautela em tarefas mais complexas. A superficialidade das soluções geradas em cenários que demandam maior profundidade, a necessidade de validação humana e a dependência de instruções bem definidas evidenciam que a ferramenta, por si só, não substitui o conhecimento técnico do desenvolvedor. Nesse sentido, [Ehsani et al. 2026] destacam que o ChatGPT é empregado principalmente para a geração de ideias e rascunhos, sendo o código produzido raramente aproveitado sem modificações significativas. Além disso, os autores observaram desempenho inferior da ferramenta em problemas complexos que exigem compreensão aprofundada e específica do projeto.

Essas limitações reforçam a necessidade de um uso consciente e estratégico da IA, no qual o desenvolvedor atua como mediador ativo do processo, garantindo que

as soluções estejam alinhadas às demandas do sistema e às boas práticas da engenharia de software. Como apontam [Yan et al. 2023], o ChatGPT depende de um processo iterativo para refinar suas respostas, sendo capaz de corrigir soluções apenas quando recebe informações adicionais. Assim, o valor da ferramenta reside menos na substituição de competências técnicas e mais em sua capacidade de acelerar etapas iniciais e apoiar o raciocínio criativo, desde que inserida em um ciclo contínuo de interação e validação humana. Esta necessidade de supervisão é consistente com as descobertas de [Li et al. 2025], que alertam para o risco de erros conceituais e lógicos no código gerado pela IA, tornando a validação humana uma salvaguarda essencial.

A experiência documentada neste trabalho amplia a compreensão sobre o uso de LLMs em ambientes de desenvolvimento *low-code*, ao evidenciar aspectos técnicos, operacionais e pedagógicos que influenciam sua eficácia. Com base nas decisões adotadas e nos resultados observados, a pesquisa oferece subsídios concretos para refletir sobre a incorporação estratégica da IA nos fluxos de desenvolvimento. O estudo propõe uma base metodológica que pode ser adaptada e expandida em futuras experimentações, contribuindo para o avanço do conhecimento na área e para a construção de práticas mais conscientes e eficazes no uso de ferramentas baseadas em IA.

Assim como a qualidade dos *prompts* influencia diretamente o desempenho do ChatGPT, a área de Sistemas de Informação também ressalta a necessidade de capacitar profissionais para lidar com a complexidade sociotécnica que caracteriza esses sistemas. Nesse contexto, a resolução de problemas em SI de maneira eficaz exige o desenvolvimento de competências em pesquisa, educação e atuação profissional, de modo a consolidar uma visão sociotécnica e aplicar métodos interdisciplinares na solução de problemas reais [Boscarioli et al. 2017]. Essa perspectiva evidencia que a adoção de modelos de linguagem em plataformas *low-code* só alcançará seu potencial pleno quando acompanhada de esforços de formação e capacitação, permitindo que profissionais de TI atuem como mediadores ativos entre pessoas, processos e tecnologias.

A pesquisa reforça e aprofunda o debate, no campo de SI, sobre o papel da IA generativa no desenvolvimento de soluções. Os resultados auxiliam na compreensão de que ferramentas de LLMs podem atuar como facilitadores na engenharia de software, porém não substituem inteiramente o conhecimento humano. Identificamos que, apesar dos ganhos de agilidade e acessibilidade proporcionados, a atuação do ChatGPT requer uma utilização consciente e estratégica: em tarefas complexas, o desenvolvedor precisa intervir ativamente, validando e orientando a IA para assegurar que as soluções estejam alinhadas às demandas do sistema e às boas práticas de SI. Esse achado tem implicações diretas para a gestão de TI nas organizações reforça-se que a adoção de assistentes de IA deve vir acompanhada de treinamento em engenharia de *prompts* e de uma redefinição do papel do desenvolvedor humano como mediador e curador das soluções propostas pela máquina.

Agradecimentos

Em conformidade com o Código de Conduta da SBC, declaramos que utilizamos IA generativa - especificamente o ChatGPT (OpenAI) - de forma assistiva na escrita e revisão deste manuscrito, incluindo sugestões de estrutura e fluidez textual, padronização de termos em tabelas/quadros e rascunhos descritivos de figuras. Todo material gerado

ou sugerido pela IA foi revisado criticamente, checado contra fontes primárias e submetido a verificação de originalidade; não houve criação automática de dados, resultados ou citações não verificadas.

Referências

- Al Alamin, M. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. B., and Iqbal, A. (2021). An empirical study of developer discussions on low-code software development challenges. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, pages 46–57.
- Boscarioli, C., Araujo, R. M., and Maciel, R. S. P., editors (2017). *I GranDSI-BR: Grand Research Challenges in Information Systems in Brazil 2016–2026*. Sociedade Brasileira de Computação (SBC), Porto Alegre. Disponível em : <https://booksol.sbc.org.br/index.php/sbc/catalog/view/28/99/249>.
- Bruhin, O., Ebel, P., and Elshan, E. (2024). Exploring the paradoxical tensions in low code environments. In *ICIS 2024 Proceedings*. Disponível em : <https://aisel.aisnet.org/icis2024/isdesign/isdesign/5>. Acesso em : 22 ago. 2025.
- Bucaioni, A., Ekedahl, H., Helander, V., and Nguyen, P. T. (2024). Programming with chatgpt: How far can we go? *Machine Learning with Applications*, 15:100526.
- Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., and Reeves, B. N. (2023). Promptly: Using prompt problems to teach learners how to effectively utilize ai code generators.
- Doolin, B. and Lowe, A. (2002). To reveal is to critique: Actor–network theory and critical information systems research. *Journal of Information Technology*, 17(2):69–78.
- Ehsani, R., Pathak, S., Parra, E., Haiduc, S., and Chatterjee, P. (2026). What characteristics make chatgpt effective for software issue resolution? an empirical study of task, project, and conversational signals in github issues. *Empirical software engineering*, 31(1):22.
- Falcão Filho, O., Buarque, B., and Almendra, C. (2026). Material suplementar do artigo "use of chatgpt in low-code development: An exploratory study in the salesforce platform". Disponível em: <https://doi.org/10.5281/zenodo.18544945>.
- Forbes, A., Safir, P., Kubon, J., and Fálder, F. (2023). *ChatGPT for Accelerating Salesforce Development*. Packt Publishing.
- Gartner (2023). Gartner forecasts worldwide low-code development technologies market to grow 20% in 2023. Disponível em : <https://www.gartner.com/doc/reprints?id=1-2J42UZEC&ct=241017&st=sb>. Acesso em : 2 set. 2025.
- Gartner (2025). Magic quadrant for enterprise low-code application platforms. Disponível em : <https://www.gartner.com/doc/reprints?id=1-2LJIU4E6&ct=250728&st=sb>. Acesso em : 5 set. 2025.
- KPMG (2024). Plataformas low-code moldam a transformação digital. Disponível em : <https://kpmg.com/pt/pt/home/insights/2024/02/plataformas-low-code-moldam-transformacao-digital.html>. Acesso em : 10 ago. 2025.

- Li, R., Liang, P., Wang, Y., Cai, Y., Sun, W., and Li, Z. (2025). Unveiling the role of chatgpt in software development: Insights from developer–chatgpt interactions on github. *ACM Transactions on Software Engineering and Methodology*, pages 1–25.
- Liu, M., Wang, J., Lin, T., Ma, Q., Fang, Z., and Wu, Y. (2024). An empirical study of the code generation of safety-critical software using llms. *Applied Sciences*, 14:1046.
- Martins, J., Branco, F., and Mamede, H. (2023). Combining low-code development with chatgpt to novel no-code approaches: A focus-group study. *Intelligent Systems with Applications*, 20:200289.
- Matvitsky, O., Davis, K., and Jain, A. (2024). Magic quadrant for enterprise low-code application platforms. Disponível em : <https://www.gartner.com/doc/reprints?id=1-2J42UZEct=241017st=sb>.
- Monteiro, M., Branco, B. C., Silvestre, S., Avelino, G., and Valente, M. T. (2025). Nocodept: A no-code interface for building web apps with language models.
- Moskal, M. (2021). No-code application development on the example of logotec app studio platform. In *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, pages 54–57.
- Rokis, K. and Kirikova, M. (2023). Exploring low-code development: A comprehensive literature review. *Complex Systems Informatics and Modeling Quarterly (CSIMQ)*, (36):68–86.
- Ryan, G., Jain, S., Shang, M., Wang, S., Ma, X., Ramanathan, M. K., and Ray, B. (2024). Code-aware prompting: A study of coverage-guided test generation in regression setting using llm. *Proceedings of the ACM on Software Engineering*, 1(Issue FSE):21.
- Vincent, P., Natis, Y., Iijima, K., Wong, J., Ray, S., Jain, A., and Leow, A. (2020). Magic quadrant for enterprise low-code application platforms. Technical report, Gartner.
- White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt.
- Yan, D., Gao, Z., and Liu, Z. (2023). A closer look at code generation abilities of chatgpt. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1887–1897. IEEE.