

ENTRE PROMPTS E POST-ITS:

O NOVO JEITO DE FAZER
REQUISITOS DE SOFTWARE

Guia com Estudo de Caso
e 20 Prompts Aplicados



CÍCERO GONÇALVES DOS SANTOS

*Entre Prompts e Post-Its: O Novo Jeito de Fazer Requisitos de Software
Guia com Estudo de Caso e 20 Prompts Aplicados*

***Entre Prompts e Post-Its: O Novo Jeito de Fazer Requisitos de Software
Guia com Estudo de Caso e 20 Prompts Aplicados***

Autor:

Cícero Gonçalves dos Santos

Edição: 1ª Edição

Ano: 2025

Local: Aracaju – Sergipe, Brasil

Formato: Guia Técnico-Acadêmico

Palavras-chave:

Engenharia de Requisitos, Inteligência Artificial, Métodos Ágeis, Scrum, Kanban, GPTs, Elicitação, Documentação Técnica

Este guia integra a linha de publicações de apoio à formação em Engenharia de Software e Métodos Ágeis, com foco na aplicação de Inteligência Artificial no ciclo de requisitos. O conteúdo foi desenvolvido como material didático e referência técnica para estudantes, docentes e profissionais da área de tecnologia.

Sobre o Autor

Cícero Gonçalves dos Santos




Analista de Requisitos Sênior no Sebrae Sergipe, com sólida experiência na área de análise e engenharia de requisitos, atuando desde 2018 em times ágeis com foco em melhoria contínua, qualidade e entrega de valor ao cliente. Mestre em Ciência da Computação, graduado em Sistemas de Informação e com três especializações: em Governança de TI, Gestão Pública e Teste de software Ágil. Graduando em Inteligência Artificial, escritor e roteirista de Almanques para Popularização de Ciência da Computação - SBC. Com ampla vivência acadêmica, atuou como docente e gestor

universitário, premiado, na área de gestão e tecnologia.

Conecte-se

 **LinkedIn:** www.linkedin.com/in/cicergoncalves

 **Instagram:** www.instagram.com/prof.cicergoncalves

 **E-mail:** cic.goncalves.santos@gmail.com

Apresentação

Vivemos uma nova era na Engenharia de Software — uma era em que a inteligência artificial deixou de ser apenas ferramenta de apoio e passou a ser **parceira ativa no pensamento analítico, criativo e técnico**. Este guia, intitulado “**Entre Prompts e Post-Its: O Novo Jeito de Fazer Requisitos de Software**”, nasce desse encontro entre a **metodologia ágil** e o **poder dos modelos de linguagem**, transformando o modo como documentamos, validamos e comunicamos requisitos.






Ao longo das próximas páginas, o leitor é convidado a percorrer um percurso que une teoria e prática. São **vinete prompts inteligentes**, cuidadosamente elaborados, que demonstram como a IA pode apoiar o analista em cada fase do ciclo de requisitos — desde a **elicitação inicial** até a **validação com stakeholders**, passando por **modelagem, prototipagem, backlog e testes**. Tudo é contextualizado por meio de um **estudo de caso aplicado**: o desenvolvimento do aplicativo de delivery “**Muita Fome**”, um projeto que espelha a realidade dos times ágeis e a necessidade de equilibrar valor de negócio, qualidade e inovação.

O guia foi estruturado conforme as normas **IEEE 830-1998** e **ISO/IEC/IEEE 29148-2018**, adotando a mesma seriedade documental da Engenharia de Requisitos clássica, mas traduzida para um formato mais dinâmico, prático e acessível. Assim, os “post-its” das metodologias ágeis são reinventados na forma de **prompts estratégicos**, capazes de gerar artefatos técnicos, histórias de usuário e documentação automatizada.

Mais do que um material didático, este guia é uma proposta de futuro: um convite para que estudantes, profissionais e pesquisadores **repensem a engenharia de software como uma prática colaborativa entre humanos e inteligências artificiais**, mantendo o rigor técnico, mas incorporando criatividade, empatia e agilidade.

Cícero Gonçalves dos Santos

Sumário

	1. Introdução	2
	2. Estudo de Caso: Aplicativo “Muita Fome”	2
	2.1 Contexto	2
	2.2 Objetivos do Sistema	2
	2.3 Metodologia	2
	3. Como Funciona um Prompt Eficaz	3
	4. Aplicação Prática – 20 Prompts Inteligentes para o Projeto “Muita Fome”	3
	♦ Prompt 1 — Levantamento Inicial de Requisitos	3
	♦ Prompt 2 — Mapeamento de Stakeholders	4
	♦ Prompt 3 — Identificação de Épicas	5
	♦ Prompt 4 — Histórias de Usuário (HU)	6
	♦ Prompt 5 — Critérios de Aceite (CA)	6
	♦ Prompt 6 — Requisitos Funcionais (RF)	7
	♦ Prompt 7 — Requisitos Não Funcionais (RNF)	8
	♦ Prompt 8 — Regras de Negócio (RN)	8
	♦ Prompt 9 — Diagrama de Caso de Uso (PlantUML)	9
	♦ Prompt 10 — Fluxo de Processo BPMN	10
	♦ Prompt 11 — Protótipo de Interface (Wireframe Textual)	10
	♦ Prompt 12 — Backlog do Produto (Product Backlog)	11
	♦ Prompt 13 — Planejamento de Sprint (Sprint Planning)	11
	♦ Prompt 14 — Quadro Kanban	12
	♦ Prompt 15 — Plano de Testes de Aceitação	13
	♦ Prompt 16 — Documento de Especificação IEEE 830 (ERS)	13
	♦ Prompt 17 — Validação com Stakeholders	14
	♦ Prompt 18 — Métricas de Satisfação e Desempenho	14
	♦ Prompt 19 — Refinamento de Backlog (Backlog Refinement)	15
	♦ Prompt 20 — Geração Automática de Artefatos Integrados	16
	6. Referências Bibliográficas	16

1. Introdução

A Engenharia de Requisitos é uma disciplina essencial para o sucesso de projetos de software, pois define as bases que orientam todo o ciclo de vida do desenvolvimento (PRESSMAN; MAXIM, 2016). Tradicionalmente, esse processo depende de entrevistas, reuniões e artefatos estáticos, que muitas vezes se tornam obsoletos ou ambíguos.

Com o avanço da **Inteligência Artificial generativa (IAg)**, ferramentas como ChatGPT, Copilot e Gemini permitem transformar a coleta e a documentação de requisitos em um processo dinâmico, colaborativo e automatizado (SOMMERVILLE, 2019).

Neste guia, você aprenderá como utilizar **20 prompts inteligentes** aplicados ao estudo de caso do aplicativo de delivery “**Muita Fome**”, combinando **boas práticas de engenharia de software (IEEE 830; ISO/IEC/IEEE 29148)** com **metodologias ágeis (Scrum e Kanban)**. Cada prompt é apresentado com **contexto, objetivo, modelo de uso e exemplo de resposta esperada**, formando um guia prático e robusto para analistas, Product Owners e desenvolvedores.

2. Estudo de Caso: Aplicativo “Muita Fome”

2.1 Contexto

O aplicativo **Muita Fome** é uma plataforma de delivery voltada para pequenos restaurantes e empreendedores locais. Seu objetivo é conectar clientes a estabelecimentos próximos, oferecendo **pedidos rápidos, entregas eficientes e suporte ao microempreendedorismo gastronômico**.

2.2 Objetivos do Sistema

- Permitir que o usuário **cadastre restaurantes e cardápios**;
- Facilitar **pedidos e pagamentos online**;
- Gerar **relatórios e análises de vendas** para o comerciante;
- Integrar-se a serviços externos (Google Maps, Pix, WhatsApp Business).

2.3 Metodologia

O projeto será desenvolvido com **Scrum**:

- **Product Owner**: coordenador do produto;
- **Scrum Master**: analista de processos;
- **Time**: 6 desenvolvedores e 1 designer;

- **Artefatos:** Product Backlog, Sprint Backlog, Incremento, Burndown e Retrospectiva.

O controle das tarefas seguirá o **Kanban**: *To Do* → *Doing* → *Done*.

3. Como Funciona um Prompt Eficaz

Cada prompt deve conter:

1. **Papel** – quem o modelo deve representar (ex: analista de requisitos).
2. **Contexto** – descrição do cenário e do objetivo.
3. **Ação esperada** – o que o modelo deve produzir.
4. **Formato de saída** – como o resultado deve ser entregue (tabela, HU, lista, etc.).

Exemplo genérico:

“Atue como analista de requisitos sênior. Gere 5 requisitos funcionais e 3 não funcionais para um aplicativo de delivery, apresentando em formato de tabela IEEE 830.”

4. Aplicação Prática – 20 Prompts Inteligentes para o Projeto “Muita Fome”

Cada prompt a seguir traz **contexto**, **texto do prompt**, e **exemplo de resposta esperada**.

Estudo de Caso Base: Aplicativo “Muita Fome”

O aplicativo **Muita Fome** é uma plataforma de delivery local destinada a conectar pequenos restaurantes a clientes em busca de refeições rápidas e acessíveis. O sistema inclui módulos de cadastro de estabelecimentos, exibição de cardápios, gestão de pedidos, pagamentos online e integração com serviços externos como Google Maps e Pix.

A equipe de desenvolvimento adota a **metodologia Scrum**, com **ciclos iterativos de duas semanas**, e usa **ChatGPT** como ferramenta auxiliar para geração e revisão de requisitos, documentação e validação. O processo segue as boas práticas da **norma IEEE Std 830-1998** (Especificação de Requisitos de Software) e da **ISO/IEC/IEEE 29148:2018**, que define diretrizes de engenharia de requisitos e rastreabilidade.

◆ Prompt 1 — Levantamento Inicial de Requisitos

Norma / Prática aplicada:

Baseado na *IEEE 830:1998 (Seção 5 – Requisitos Funcionais)* e na *ISO/IEC/IEEE 29148:2018*

(Cláusula 6.1 – Elicitação de Requisitos), que orientam a identificação clara das necessidades dos stakeholders antes de formalizar qualquer especificação.

Objetivo:

Permitir que a IA atue como analista facilitador, conduzindo uma elicitação estrutural e empatia para compreender o propósito, escopo e restrições do sistema “Muita Fome”.

Construção do Prompt:

O prompt deve descrever o papel (“analista de requisitos”), o contexto (“primeira reunião com o cliente”) e a ação desejada (“gerar perguntas abertas e técnicas para entender o problema”).

Isso guia a IA a adotar uma postura de descoberta, evitando assumir respostas prontas.

Prompt:

“Atue como analista de requisitos sênior. Gere 10 perguntas de levantamento inicial para entender os objetivos, público-alvo, principais dores, diferenciais e limitações do aplicativo ‘Muita Fome’.”

Como a IA deve trabalhar:

Ao responder, a IA deve gerar perguntas que contemplem dimensões de negócio, técnicas e operacionais — como performance, integração, experiência do usuário e segurança. Essas perguntas devem fomentar conversas exploratórias entre o Product Owner e os stakeholders, servindo de base para a criação do Documento de Visão (IEEE 830).

♦ **Prompt 2 — Mapeamento de Stakeholders**

Norma / Prática aplicada:

Corresponde à *ISO/IEC/IEEE 29148:2018 – Seção 6.2 (Análise de Stakeholders)*, que determina que todos os participantes do sistema devem ser identificados e classificados quanto à sua influência e responsabilidade.

Objetivo:

Gerar uma matriz de stakeholders que permita identificar papéis, expectativas e prioridades. A IA, neste caso, auxilia o analista a pensar de forma sistêmica sobre os atores envolvidos.

Construção do Prompt:

A instrução deve especificar que se deseja uma **tabela** com colunas padronizadas (Nome,

Papel, Influência, Expectativa). O papel da IA é simular uma análise organizacional e contextual.

Prompt:

“Liste e categorize os principais stakeholders do projeto ‘Muita Fome’, indicando papel, interesse, influência e nível de poder na decisão. Apresente o resultado em formato de tabela.”

Como a IA deve trabalhar:

A IA deve classificar cada stakeholder (cliente, restaurante, entregador, equipe técnica, suporte, etc.) de acordo com seu impacto nas decisões e no sucesso do produto. Essa análise contribui diretamente para priorizar requisitos de alto valor agregado — um princípio do *Agile Manifesto* e da norma *IEEE 830*, que recomenda foco em requisitos críticos.

♦ **Prompt 3 — Identificação de Épicos**

Norma / Prática aplicada:

Relacionado à *IEEE 830:1998 (Seção 5.2 – Agrupamento de Requisitos)* e à prática de decomposição ágil segundo *Scrum Guide (SCHWABER; SUTHERLAND, 2020)*.

Os épicos representam grandes funcionalidades do produto, posteriormente decompostas em histórias de usuário.

Objetivo:

Usar a IA para organizar a visão macro do sistema em blocos lógicos e priorizados, antes da fragmentação detalhada em requisitos.

Construção do Prompt:

O prompt deve mencionar o contexto do sistema e pedir à IA que agrupe funcionalidades correlatas em épicos. É essencial citar o domínio (“delivery”) para que o modelo utilize conhecimento contextual.

Prompt:

“Com base na descrição do aplicativo ‘Muita Fome’, crie 5 épicos que representam grandes áreas funcionais (ex: cadastro, pedidos, pagamentos, entregas e relatórios).”

Como a IA deve trabalhar:

A IA deve gerar respostas categorizadas, mostrando que compreende o domínio. Espera-se

que ela reconheça áreas de negócio e de suporte, respeitando boas práticas de rastreabilidade recomendadas na *ISO/IEC/IEEE 29148*, que enfatiza a decomposição hierárquica e a ligação entre requisitos de alto e baixo nível.

♦ Prompt 4 — Histórias de Usuário (HU)

Norma / Prática aplicada:

Fundamentado na *IEEE Std 830:1998* (Seção 5.3 – Descrição de Funções do Usuário) e nas boas práticas ágeis descritas no *Scrum Guide (2020)*.

As HU são essenciais para comunicar requisitos em linguagem acessível, representando necessidades reais do usuário.

Objetivo:

Gerar histórias de usuário claras, simples e rastreáveis, que servirão de base para critérios de aceite, priorização e planejamento de sprint.

Construção do Prompt:

O comando deve informar explicitamente o formato clássico de HU (*Como [usuário], quero [função], para [benefício]*) e o contexto do sistema “Muita Fome”.

É importante indicar que se deseja **diversidade de papéis** (cliente, entregador, restaurante, administrador).

Prompt:

“Atue como analista de requisitos ágil. Crie 10 histórias de usuário para o aplicativo de delivery ‘Muita Fome’, seguindo o formato: *Como [usuário], quero [ação], para [benefício]*. Inclua diferentes perfis de usuários (cliente, entregador, restaurante, administrador).”

Como a IA deve trabalhar:

A IA deve empregar a técnica de *user-centered design*, descrevendo o valor de negócio por trás de cada história. As HU devem ser **independentes, negociáveis, valiosas, estimáveis, pequenas e testáveis (INVEST)** — conforme recomendação de *Cohn (2004)* e da *ISO/IEC/IEEE 29148:2018* (Cláusula 6.3 – Requisitos de Usuário).

♦ Prompt 5 — Critérios de Aceite (CA)

Norma / Prática aplicada:

Relaciona-se diretamente à *ISO/IEC/IEEE 29148:2018 (Cláusula 6.4 – Verificação e Validação de Requisitos)* e à metodologia *Behavior-Driven Development (BDD)*, que define critérios de aceite em formato narrativo (Dado/Quando/Então).

Objetivo:

Gerar critérios de aceite objetivos, testáveis e vinculados às histórias de usuário, garantindo rastreabilidade entre HU e testes de aceitação.

Construção do Prompt:

O comando deve instruir a IA a criar os critérios em formato BDD e mencionar que devem estar vinculados a HU específicas.

Prompt:

“Para cada história de usuário do aplicativo ‘Muita Fome’, gere critérios de aceite em formato BDD (*Dado/Quando/Então*), garantindo que cada requisito seja verificável e rastreável.”

Como a IA deve trabalhar:

A IA deve identificar cenários de sucesso e exceção, traduzindo o comportamento esperado em casos de teste interpretáveis.

Isso assegura conformidade com a *IEEE 830*, que recomenda que os requisitos sejam **testáveis e verificáveis**.

Os resultados podem alimentar diretamente o backlog de testes no Jira ou Trello.

♦ **Prompt 6 — Requisitos Funcionais (RF)**

Norma / Prática aplicada:

Segue o modelo da *IEEE 830:1998 (Seção 5.2 – Requisitos Funcionais)* e o princípio da *ISO/IEC/IEEE 29148:2018 (Cláusula 6.5 – Documentação de Requisitos)*.

Objetivo:

Gerar uma lista de RF que descreva o que o sistema deve realizar, de forma completa, sem ambiguidades e com rastreabilidade.

Construção do Prompt:

O comando deve explicitar que se deseja um **formato tabular** com ID, nome, descrição,

prioridade e HU relacionada.

Mencionar a norma orienta a IA a responder de maneira estruturada e formal.

Prompt:

“Liste 10 Requisitos Funcionais para o aplicativo ‘Muita Fome’ no formato IEEE 830, contendo: ID, Nome do Requisito, Descrição, Prioridade e História de Usuário associada.”

Como a IA deve trabalhar:

A IA deve estruturar a resposta com linguagem técnica, evitando redundâncias.

Ela deve seguir os atributos de qualidade da *IEEE 830*: **correção, completude, consistência, verificabilidade e rastreabilidade**.

Esses RF podem alimentar automaticamente um modelo de ERS (*Software Requirements Specification*).

♦ **Prompt 7 — Requisitos Não Funcionais (RNF)**

Norma / Prática aplicada:

Baseado na *ISO/IEC/IEEE 25010:2011* (Qualidade de Produto de Software) e *IEEE 830:1998*.

Objetivo:

Descrever as características de qualidade do sistema (segurança, desempenho, usabilidade, disponibilidade, escalabilidade).

Construção do Prompt:

O comando deve especificar que se deseja RNF organizados por categoria e aplicáveis ao contexto de delivery (tempo de resposta, disponibilidade do servidor, experiência do usuário).

Prompt:

“Gere 8 Requisitos Não Funcionais para o app ‘Muita Fome’, distribuídos nas categorias: desempenho, segurança, usabilidade, disponibilidade, acessibilidade, portabilidade, interoperabilidade e manutenibilidade.”

Como a IA deve trabalhar:

A IA deve interpretar o domínio e associar cada requisito à categoria de qualidade correspondente.

Cícero Gonçalves dos Santos

[linkedin.com/in/ciceroconcalves](https://www.linkedin.com/in/ciceroconcalves)

Esses RNF garantem conformidade com o *ISO/IEC 25010* e podem servir como base para metas de SLA.

O resultado orienta testes de desempenho e critérios de escalabilidade.

♦ **Prompt 8 — Regras de Negócio (RN)**

Norma / Prática aplicada:

Com base na *ISO/IEC/IEEE 29148:2018 (Seção 6.7 – Regras de Negócio)* e no princípio de *Business Rules Analysis* de BABOK v3 (IIBA, 2015).

Objetivo:

Formalizar as políticas e restrições que governam o funcionamento do aplicativo — por exemplo, cálculo de taxas, prazos de cancelamento, regras de entrega mínima, etc.

Construção do Prompt:

O comando deve pedir que a IA identifique e descreva **regras de negócio em linguagem natural padronizada**, distinguindo-as de requisitos técnicos.

Prompt:

“Liste e descreva 5 Regras de Negócio aplicáveis ao aplicativo ‘Muita Fome’, especificando o motivo da regra, impacto no sistema e consequência de violação.”

Como a IA deve trabalhar:

A IA deve gerar enunciados normativos claros, começando por “O sistema deve...”, “O restaurante deverá...”, “Não é permitido...”.

Essas regras garantem coerência e previsibilidade, sendo essenciais para auditorias e conformidade regulatória.

♦ **Prompt 9 — Diagrama de Caso de Uso (PlantUML)**

Norma / Prática aplicada:

Relaciona-se à *IEEE 830 (Apêndice C – Modelos Gráficos)* e à *UML 2.5 Specification (OMG, 2017)*.

Objetivo:

Visualizar interações entre atores e funcionalidades do sistema, documentando os principais fluxos de uso.

Construção do Prompt:

Deve instruir a IA a gerar o código em **PlantUML**, com atores, casos de uso e relacionamentos.

Prompt:

“Gere o código PlantUML para o diagrama de casos de uso do app ‘Muita Fome’, incluindo os atores: Cliente, Restaurante, Entregador e Sistema de Pagamento.”

Como a IA deve trabalhar:

A IA deve representar graficamente as interações de maneira hierárquica e consistente com as HU. O diagrama resultante deve ser validado com o Product Owner, atendendo ao requisito de **verificabilidade e rastreabilidade** da *ISO/IEC/IEEE 29148*.

♦ **Prompt 10 — Fluxo de Processo BPMN**

Norma / Prática aplicada:

Baseado na *BPMN 2.0 Specification (OMG, 2011)* e na *ISO/IEC/IEEE 29148:2018 (Cláusula 7 – Modelagem de Processos)*.

Objetivo:

Representar o fluxo de pedido e entrega no app “Muita Fome”, mapeando atividades, eventos e gateways de decisão.

Construção do Prompt:

Deve solicitar que a IA descreva o fluxo BPMN em texto estruturado, usando eventos de início/fim, atividades e gateways.

Prompt:

“Descreva o fluxo BPMN do processo de pedido e entrega do aplicativo ‘Muita Fome’, identificando eventos de início, tarefas, gateways e eventos de término, em linguagem textual estruturada.”

Como a IA deve trabalhar:

A IA deve simular a visão do analista de processos, traduzindo o fluxo em uma estrutura de

Cícero Gonçalves dos Santos
[linkedin.com/in/ciceroconcalves](https://www.linkedin.com/in/ciceroconcalves)

etapas lógica e compreensível. O resultado pode ser importado em ferramentas como Bizagi, Lucidchart ou [Draw.io](https://draw.io). Esse artefato reforça o **entendimento interfuncional**, conforme as práticas de modelagem da *ISO/IEC/IEEE 29148*.

♦ Prompt 11 — Protótipo de Interface (Wireframe Textual)

Norma / Prática aplicada:

Baseado na *ISO/IEC/IEEE 29148:2018* (Cláusula 6.8 – Requisitos de Interface do Usuário) e nas práticas de *Design Thinking* e *UX Design Guidelines* (Nielsen, 2012).

Objetivo:

Permitir que a IA simule um esboço textual da interface do aplicativo “Muita Fome”, auxiliando na visualização das interações e organização da informação antes da prototipagem visual.

Construção do Prompt:

O comando deve pedir que a IA crie uma **descrição textual estruturada**, simulando o layout de tela e os elementos de interação.

Prompt:

“Crie um wireframe textual para a tela principal do aplicativo ‘Muita Fome’, descrevendo seções, botões, menus e elementos de navegação. Utilize uma estrutura hierárquica e indique as interações principais.”

Como a IA deve trabalhar:

A IA deve gerar um modelo descritivo, segmentado por áreas (cabeçalho, corpo, rodapé, menus, botões de ação). Essa prática está alinhada ao princípio de *prototipação iterativa*, previsto na *ISO/IEC 9241-210:2010* (Design centrado no usuário). A resposta serve de insumo para ferramentas de UX (Figma, Adobe XD, Miro).

♦ Prompt 12 — Backlog do Produto (Product Backlog)

Norma / Prática aplicada:

Conforme o *Scrum Guide* (SCHWABER; SUTHERLAND, 2020) e a *ISO/IEC/IEEE 29148:2018* (Cláusula 6.9 – Priorização de Requisitos).

Objetivo:

Gerar um backlog completo com histórias de usuário, valor de negócio, prioridade e estimativas de esforço.

Construção do Prompt:

O comando deve pedir que a IA liste o backlog em **formato tabular**, contendo HU, prioridade (MoSCoW ou Fibonacci) e esforço estimado.

Prompt:

“Monte o Product Backlog inicial do aplicativo ‘Muita Fome’, com colunas: ID da História, Descrição, Prioridade, Esforço (Story Points) e Status.”

Como a IA deve trabalhar:

A IA deve adotar a mentalidade de *Product Owner*, priorizando valor de negócio sobre complexidade técnica. Essa estrutura reflete boas práticas de rastreabilidade e planejamento ágil. O resultado pode ser migrado para ferramentas como Jira, Trello ou Notion.

♦ **Prompt 13 — Planejamento de Sprint (Sprint Planning)**

Norma / Prática aplicada:

Baseado no *Scrum Guide (2020)* e no conceito de *Iteration Planning* do *Scaled Agile Framework (SAFe 5.0)*.

Objetivo:

Definir o escopo e o plano de entrega da Sprint, identificando tarefas e metas de incremento do produto.

Construção do Prompt:

O comando deve solicitar à IA que gere uma **tabela de Sprint**, incluindo HU selecionadas, responsáveis, dependências e critérios de aceite.

Prompt:

“Elabore o plano de Sprint 1 do aplicativo ‘Muita Fome’, com duração de 2 semanas. Liste as histórias de usuário selecionadas, responsáveis, dependências e metas da Sprint.”

Como a IA deve trabalhar:

A IA deve agir como facilitadora da reunião de *Sprint Planning*, ajudando o time a decompor

Cícero Gonçalves dos Santos
[linkedin.com/in/cicerogoncalves](https://www.linkedin.com/in/cicerogoncalves)

histórias e planejar incrementos tangíveis.

O resultado deve respeitar o *time-box* da sprint e estar alinhado à capacidade da equipe.

♦ Prompt 14 — Quadro Kanban

Norma / Prática aplicada:

Inspirado nas práticas de fluxo contínuo do *Kanban Method* (ANDERSON, 2010) e na *ISO/IEC/IEEE 12207:2017* (Processos do ciclo de vida de software).

Objetivo:

Visualizar o progresso das tarefas em um quadro textual Kanban, permitindo o acompanhamento do fluxo de trabalho em tempo real.

Construção do Prompt:

O comando deve especificar as colunas padrão (*To Do*, *Doing*, *Done*) e permitir que a IA liste tarefas com responsáveis e status.

Prompt:

“Monte um quadro Kanban textual para o projeto ‘Muita Fome’, com colunas To Do, Doing e Done, incluindo as tarefas derivadas das HU e os responsáveis por cada uma.”

Como a IA deve trabalhar:

A IA deve reproduzir a lógica do fluxo contínuo, identificando gargalos e equilibrando a carga de trabalho. Essa estrutura é essencial para controle visual e melhoria contínua (Kaizen), de acordo com as práticas do *Kanban Maturity Model* (2019).

♦ Prompt 15 — Plano de Testes de Aceitação

Norma / Prática aplicada:

Relaciona-se à *IEEE Std 829:2008* (*Software Test Documentation*) e à *ISO/IEC/IEEE 29119-3:2013* (*Test Documentation*).

Objetivo:

Gerar um plano de testes para validar requisitos funcionais e critérios de aceite.

Construção do Prompt:

O comando deve pedir à IA que gere casos de teste baseados nos critérios BDD, com campos: ID, Descrição, Pré-condição, Passos e Resultado Esperado.

Prompt:

“Gere casos de teste de aceitação para o módulo de pedidos do app ‘Muita Fome’, baseados nos critérios de aceite BDD, incluindo ID, Descrição, Pré-condições, Passos e Resultados Esperados.”

Como a IA deve trabalhar:

A IA deve traduzir requisitos em cenários testáveis, promovendo **verificabilidade e rastreabilidade** — princípios essenciais da *IEEE 830* e da *29148*. O produto resultante pode ser usado em ferramentas como TestLink, Zephyr ou Azure Test Plans.

◆ **Prompt 16 — Documento de Especificação IEEE 830 (ERS)**

Norma / Prática aplicada:

Diretamente vinculado à *IEEE Std 830-1998* (Recommended Practice for Software Requirements Specification).

Objetivo:

Gerar uma seção formal de especificação de requisitos para o módulo de pagamentos.

Construção do Prompt:

O comando deve pedir à IA que estruture o texto no formato da norma IEEE, com subtítulos: Introdução, Propósito, Escopo, Requisitos Funcionais, RNF e Restrições.

Prompt:

“Monte uma seção de Especificação de Requisitos de Software (ERS) para o módulo de pagamentos do app ‘Muita Fome’, seguindo o formato IEEE 830, com Introdução, Escopo, RF, RNF e Restrições.”

Como a IA deve trabalhar:

A IA deve produzir uma estrutura formal, com terminologia técnica e clareza documental. Essa prática garante consistência e serve como base para homologação e auditoria de software.

♦ **Prompt 17 — Validação com Stakeholders**

Norma / Prática aplicada:

Baseada na *ISO/IEC/IEEE 29148:2018* (Seção 7.3 – Validação de Requisitos) e no *BABOK v3* (IIBA, 2015).

Objetivo:

Gerar um roteiro de validação para reuniões de revisão de sprint, coletando feedbacks estruturados dos stakeholders.

Construção do Prompt:

O comando deve solicitar perguntas e critérios de análise baseados em HU e RF concluídos.

Prompt:

“Crie um roteiro de validação de requisitos para a reunião de revisão do app ‘Muita Fome’, incluindo perguntas, critérios e métricas de aceitação baseados nos RF implementados.”

Como a IA deve trabalhar:

A IA deve auxiliar na formulação de perguntas de validação, promovendo o diálogo entre usuários e desenvolvedores. O objetivo é assegurar que o incremento entregue atenda ao propósito original (valor de negócio).

♦ **Prompt 18 — Métricas de Satisfação e Desempenho**

Norma / Prática aplicada:

Baseado na *ISO/IEC 25010:2011* (Qualidade de Produto) e na *ISO/IEC 25023:2016* (Medições de Qualidade).

Objetivo:

Estabelecer indicadores quantitativos e qualitativos para avaliar a performance e a satisfação do usuário.

Construção do Prompt:

O comando deve solicitar métricas ligadas a tempo de resposta, taxa de sucesso em pedidos, NPS e feedback dos restaurantes.

Prompt:

Cícero Gonçalves dos Santos

[linkedin.com/in/ciceroconcalves](https://www.linkedin.com/in/ciceroconcalves)

“Crie indicadores de desempenho e satisfação do usuário para o app ‘Muita Fome’, considerando tempo de entrega, taxa de erro, NPS e tempo médio de atendimento.”

Como a IA deve trabalhar:

A IA deve sugerir métricas SMART (Específicas, Mensuráveis, Atingíveis, Relevantes e Temporais), fundamentadas em dados de uso. Isso permite criar dashboards BI e relatórios de qualidade.

♦ **Prompt 19 — Refinamento de Backlog (Backlog Refinement)**

Norma / Prática aplicada:

Conforme o *Scrum Guide (2020)* e a *ISO/IEC/IEEE 29148:2018 (Seção 6.9 – Revisão e Atualização de Requisitos)*.

Objetivo:

Auxiliar na revisão e reescrita de histórias de usuário com base em feedbacks de sprint e novos aprendizados.

Construção do Prompt:

O comando deve pedir que a IA simule uma reunião de refinamento, reescrevendo HU e ajustando critérios de aceite conforme retorno dos stakeholders.

Prompt:

“Simule uma reunião de Backlog Refinement para o app ‘Muita Fome’. Reescreva as histórias de usuário e critérios de aceite com base nos feedbacks recebidos durante a Sprint Review.”

Como a IA deve trabalhar:

A IA deve agir como um *facilitador de revisão*, garantindo consistência, rastreabilidade e clareza nas HU revisadas. Essa prática mantém o backlog atualizado e alinhado ao valor do negócio.

♦ **Prompt 20 — Geração Automática de Artefatos Integrados**

Norma / Prática aplicada:

Baseado na *ISO/IEC/IEEE 29148:2018* (Cláusula 8 – Gestão de Requisitos) e nas boas práticas de automação de documentação com IA.

Objetivo:

Criar um comando mestre capaz de gerar automaticamente todos os artefatos essenciais de requisitos (HU, RF, RN, RNF, CA, ERS).

Construção do Prompt:

O comando deve instruir a IA a atuar como “gerador de documentação técnica integrada”, com formato padronizado IEEE.

Prompt:

“Atue como engenheiro de requisitos automatizado. Gere um documento consolidado de requisitos do app ‘Muita Fome’ contendo: HU, RF, RN, RNF, CA e uma seção ERS no padrão IEEE 830.”

Como a IA deve trabalhar:

A IA deve consolidar os artefatos em formato padronizado, criando um documento de requisitos completo, rastreável e verificável. Essa automação agiliza a entrega de documentação técnica e reduz erros humanos, mantendo conformidade normativa.



6. Referências Bibliográficas

PRESSMAN, Roger S.; MAXIM, Bruce R. *Engenharia de Software: uma abordagem profissional*. 9. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.

IEEE Std 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*. New York: IEEE, 1998.

ISO/IEC/IEEE 29148:2018. *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*. Geneva: ISO, 2018.

SCHWABER, Ken; SUTHERLAND, Jeff. *The Scrum Guide*. Scrum.org, 2020.