

### Identificação do Projeto

<b>Projeto</b> Powp – System Enterprise
<b>Especialista de Negócio</b> Matheus Marques Portela
<b>Desenvolvedor Front-End</b> Marcos Paulo Moreira Damascena
<b>Desenvolvedor Full-Stack</b> Victor Duarte
<b>Gerente do Projeto</b> João Luccas Marques
<b>Gerente substituto do Projeto</b> Pedro Henrique Echebarria
<b>Desenvolvedor Back-End</b> Victor Manoel

### Histórico de Versões

Versão	Data	Descrição	Responsável
1.0	06/09/2025	Criação da documentação do escopo do projeto	Matheus Marques Portela
1.1	13/10/2025	Criação da documentação de qualidade e dashboards do projeto	Matheus Marques Portela

1.	DECLARAÇÃO DO ESCOPO DO PROJETO .....	1
1.1.	JUSTIFICATIVA.....	1
1.2.	PRODUTO .....	1
1.3.	PRINCIPAIS ENTREGAS E REQUISITOS.....	1
1.4.	LIMITES DO PROJETO (NÃO ESCOPO).....	1
1.5.	PREMISSAS.....	1
1.6.	RESTRIÇÕES.....	2
1.7.	MARCOS DO PROJETO .....	2
1.8.	ESTRUTURA ANALÍTICA DO PROJETO - EAP .....	4
1.9.	DICIONÁRIO DA EAP.....	5
1.10.	PAPEIS E RESPONSABILIDADES .....	8
2.	GERENCIAMENTO DO ESCOPO .....	11
3.	GERENCIAMENTO DO TEMPO .....	11
4.	GERENCIAMENTO DE CUSTOS .....	12
5.	GERENCIAMENTO DA QUALIDADE .....	13
6.	GERENCIAMENTO DE RECURSOS HUMANOS .....	14
6.1	ORGANOGRAMA DO PROJETO .....	14
6.2	DIRETÓRIO DA EQUIPE DO PROJETO.....	14
7.	GERENCIAMENTO DA COMUNICAÇÃO .....	15
7.1	MATRIZ DE COMUNICAÇÃO .....	15
8.	GERENCIAMENTO DE RISCOS .....	17
9.	GERENCIAMENTO DE AQUISIÇÕES.....	18
10.	GERENCIAMENTO DAS PARTES INTERESSADAS.....	19
11.	CONTROLE INTEGRADO DE MUDANÇAS.....	20
12.	ANEXOS .....	22

## 1. DECLARAÇÃO DO ESCOPO DO PROJETO

### 1.1.JUSTIFICATIVA

O sistema POWP ERP foi desenvolvido para atender pequenas e médias empresas, que representam a maior parte do mercado brasileiro e muitas vezes carecem de ferramentas de gestão acessíveis e eficazes. Sua arquitetura modular permite que cada organização utilize apenas os módulos necessários, reduzindo custos e garantindo maior flexibilidade. Com uma interface simples e intuitiva, o sistema busca facilitar a adoção tecnológica e otimizar os processos de gestão dessas empresas.

### 1.2.PRODUTO

O projeto POWP ERP tem como resultado final o desenvolvimento de um sistema de gestão empresarial modular, intuitivo e acessível, voltado para freelancers, pequenas e médias empresas. O produto busca entregar uma plataforma que centralize e organize as principais operações de um negócio, como cadastro de clientes, controle de estoque, gestão de vendas e relatórios, de forma ágil e segura.

### 1.3.PRINCIPAIS ENTREGAS E REQUISITOS

Na primeira etapa do desenvolvimento do projeto POWP ERP, as principais entregas contemplam os requisitos funcionais de implementação da tela de autenticação, do painel de dashboards e das funcionalidades de cadastro, controle de estoque e realização de vendas. Esses componentes visam estruturar a base do sistema e garantir sua utilização inicial de forma prática e eficiente.

No que se refere aos requisitos não funcionais, esta fase prioriza aspectos relacionados ao desempenho, integridade, segurança e performance da aplicação, assegurando que o sistema atenda às demandas das pequenas e médias empresas com confiabilidade e qualidade..

### 1.4.LIMITES DO PROJETO (NÃO ESCOPO)

Nesta versão inicial, o sistema não incluirá integração nativa com ERPs, CRMs, sistemas contábeis externos ou gateways de pagamento, sendo qualquer integração considerada apenas para futuras versões. Também não serão desenvolvidos aplicativos móveis nativos para Android ou iOS, sendo o acesso restrito via navegador web.

Funcionalidades avançadas de inteligência, como previsão de vendas, automação contábil ou análise de dados, não fazem parte do projeto, assim como recursos destinados a atender grandes empresas ou suportar alto volume de transações simultâneas, uma vez que o sistema é voltado exclusivamente para pequenas empresas e MEIs.

O projeto não permitirá customizações extensivas pelo usuário, como alteração da estrutura do banco de dados ou criação de módulos personalizados, e a personalização visual será limitada à interface web básica. Além disso, o sistema não dependerá de periféricos externos, como impressoras ou leitores de código de barras, nem funcionará offline, sendo necessária conexão à internet para operação.

### 1.5.PREMISSAS

No desenvolvimento do sistema POWP ERP, foram estabelecidas algumas premissas que servem como base para a condução do projeto, ainda que não sejam comprovadas de forma empírica. Assume-se, em primeiro lugar, que os usuários finais dispõem de infraestrutura tecnológica adequada, como computadores, dispositivos móveis e acesso à internet estável, condição essencial para o pleno funcionamento da aplicação. Parte-se também do pressuposto de que pequenas e médias empresas demonstram interesse real em adotar uma solução de gestão modular, intuitiva e de baixo custo, voltada à simplificação de seus processos administrativos e operacionais.

Outra premissa considerada é a de que os colaboradores das organizações demandantes possuem conhecimentos básicos de informática, o que possibilitaria a adaptação ao sistema com treinamentos reduzidos e de rápida absorção. Além disso, pressupõe-se a disponibilidade contínua de recursos financeiros e humanos suficientes para garantir não apenas o desenvolvimento e os testes, mas também a manutenção e a evolução futura do sistema.

Do ponto de vista técnico, assume-se que os módulos desenvolvidos manterão plena compatibilidade entre si, assegurando a consistência das informações e a escalabilidade do projeto. Por fim, considera-se que todas as práticas de gestão contempladas pela solução estarão em conformidade com a legislação vigente, em especial nas esferas tributária, fiscal e trabalhista, de modo a garantir que o sistema possa ser utilizado de maneira segura e confiável pelas empresas que dele fizerem uso.

## 1.6.RESTRIÇÕES

As seguintes restrições foram identificadas e devem ser observadas durante todo o ciclo de vida do projeto:

- **Prazo:** O projeto deverá ser concluído e entregue integralmente até o final do segundo semestre de 2026. Este prazo é inegociável e não contempla extensões.
- **Escopo:** O escopo do projeto é fixo. Todos os módulos listados abaixo devem ser entregues com suas funcionalidades essenciais operantes para que o projeto seja considerado concluído:
  - Landing Page;
  - Autenticação de usuários;
  - Dashboards iniciais;
  - Módulo de Cadastro (Fornecedor, Cliente, Funcionário, Usuário, Produtos);
  - Módulo de estoque;
  - Módulo Financeiro (Contas a pagar e receber);
  - Módulo de Vendas;
  - Chatbot com IA.
- **Recursos Humanos:** A equipe do projeto está limitada a um total de 6 (seis) integrantes. Não será possível a alocação de novos membros durante a execução do projeto.

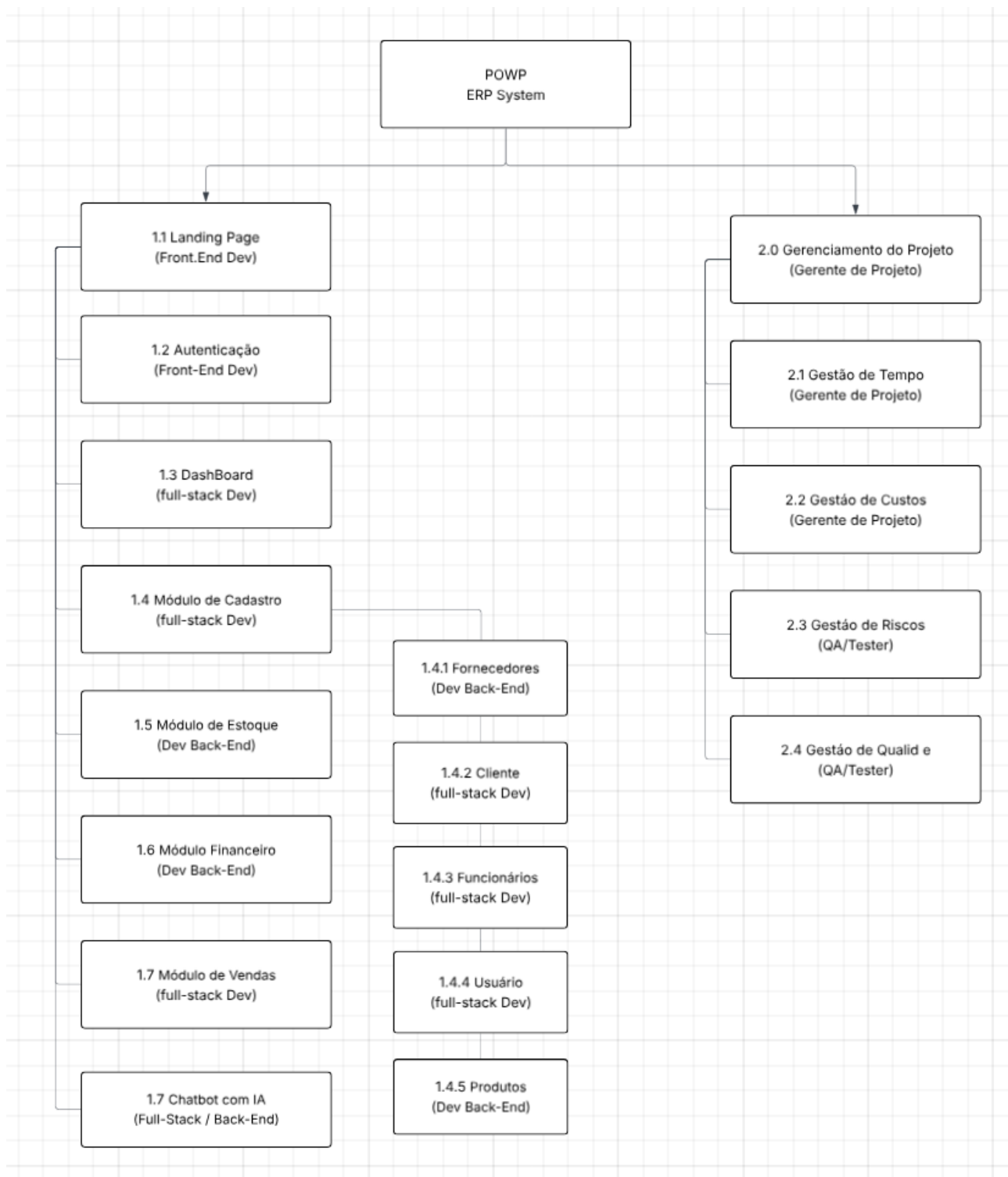
## 1.7. MARCOS DO PROJETO

A execução dos trabalhos terá início em Agosto/2025 e deve durar aproximadamente 15 meses. O planejamento do projeto, bem como sua finalização deverá ser realizado fora do período descrito.

Data	Marco
------	-------

20/12/2025	Entrega da autenticação e acesso ao sistema
28/02/2026	Entrega dos módulos de cadastro concluídos
15/04/2026	Entrega do módulo de vendas junto com o de controle de estoque

## 1.8. ESTRUTURA ANALÍTICA DO PROJETO - EAP



**1.9. DICIONÁRIO DA EAP**

Código	Item	Descrição	Entregável
1.0	POWP System ERP	O Projeto POWP ERP System visa desenvolver um sistema de gestão empresarial abrangente para automatizar processos internos.	Sistema ERP completo, funcional e integrado.
1.1	Landing Page	Criação da página inicial (Landing Page) do sistema para apresentação e informações gerais.	Página web estática, responsiva, com design alinhado à marca.
1.2	Autenticação	Implementação do sistema de login, registro, recuperação de senha e controle de acesso de usuários ao sistema. Marco: 20/12/2025	Módulo de Autenticação funcional e seguro, integrado a banco de dados de usuários.
1.3	Dashboards	Desenvolvimento de painéis visuais interativos para apresentar métricas, relatórios e informações estratégicas aos usuários.	Conjunto de dashboards configuráveis, com gráficos e tabelas, que exibam dados relevantes.
1.4	Módulo de Cadastro	Implementação das funcionalidades de cadastro, edição, visualização e gerenciamento de diversas entidades no sistema. Marco: 28/02/2026	Módulo de Cadastro completo, com sub-entregas integradas e interfaces intuitivas.
1.4.1	Fornecedor	Funcionalidades para cadastrar, editar, visualizar e gerenciar informações detalhadas de fornecedores.	Telas de CRUD (Create, Read, Update, Delete) para Fornecedores, listagem e persistência de dados.
1.4.2	Cliente	Funcionalidades para cadastrar, editar, visualizar e gerenciar informações detalhadas de clientes.	Telas de CRUD para Clientes, listagem e persistência de dados.
1.4.3	Funcionário	Funcionalidades para cadastrar, editar, visualizar e gerenciar informações de funcionários da empresa.	Telas de CRUD para Funcionários, listagem e persistência de dados.
1.4.4	Usuário	Funcionalidades para cadastrar, editar, visualizar e gerenciar	Telas de CRUD para Usuários, sistema de

		usuários do sistema POWP ERP e suas permissões de acesso.	gerenciamento de perfis e permissões.
1.4.5	Produtos	Funcionalidades para cadastrar, editar, visualizar e gerenciar informações detalhadas sobre os produtos da empresa.	Telas de CRUD para Produtos, listagem e persistência de dados.
1.5	Módulo de Estoque	Implementação das funcionalidades para controle de estoque, incluindo entrada, saída, movimentação de produtos e visualização de inventário. Marco: 15/04/2026	Módulo de Gestão de Estoque funcional, permitindo o registro de movimentações e consulta de inventário.
1.6	Módulo Financeiro	Implementação das funcionalidades de gestão financeira, como contas a pagar, contas a receber, fluxo de caixa e lançamento de despesas/receitas.	Módulo Financeiro com telas para registro e controle de transações, e geração de relatórios básicos.
1.7	Módulo de Vendas	Implementação das funcionalidades para registro e gerenciamento de vendas, criação de orçamentos, pedidos e histórico de transações. Marco: 15/04/2026	Módulo de Vendas funcional, com capacidade de criar, editar e visualizar pedidos e orçamentos.
1.8	Chatbot com IA	Desenvolvimento de um chatbot com inteligência artificial para fornecer suporte e assistência automatizada aos usuários do sistema.	Chatbot integrado ao sistema, capaz de responder perguntas frequentes e auxiliar em tarefas.
2.0	Gerenciamento do Projeto	Conjunto de processos para planejar, executar, monitorar e controlar o projeto POWP ERP como um todo, garantindo alinhamento com objetivos e restrições.	Documentação de gerenciamento de projeto, planos e relatórios de status.
2.1	Gestão de Tempo	Gerenciamento das atividades do projeto para garantir que os prazos sejam cumpridos e o cronograma mantido.	Cronograma do Projeto detalhado, relatórios de progresso, monitoramento de marcos.
2.2	Gestão de Custos	Planejamento, orçamentação, financiamento e controle dos custos para que o projeto seja	Orçamento do Projeto, relatórios de gastos,



		concluído dentro do orçamento aprovado.	análise de variação de custos.
2.3	Gestão de Riscos	Identificação, análise, planejamento de respostas e monitoramento de riscos potenciais para mitigar impactos negativos no projeto.	Registro de Riscos (Risk Log), Plano de Resposta a Riscos, relatórios de monitoramento.
2.4	Gestão de Qualidade	Garantia de que os processos e as entregas do projeto atendam aos padrões de qualidade definidos, incluindo testes e verificações.	Plano de Qualidade, casos de teste, relatórios de testes, listas de defeitos e auditorias de qualidade.

**1.10. PAPEIS E RESPONSABILIDADES**

<b>Product Owner:</b>  <b>Matheus Marques Portela</b>	<b>E-mail:</b>  matheusmarquesportela@gmail.com
<b>Responsabilidades:</b> <ul style="list-style-type: none"> <li>• Autorizar o início do projeto</li> <li>• Validar o planejamento do projeto</li> <li>• Prover recursos financeiros para o projeto</li> <li>• Contribuir na solução dos riscos do projeto</li> <li>• Manter o nível de comprometimento das equipes</li> <li>• Homologar os produtos do projeto</li> <li>• Autorizar o encerramento do projeto</li> </ul>	
<b>Design de produto:</b>  <b>Pedro Henrique Echebarria</b>	<b>E-mail:</b>
<b>Responsabilidades:</b> <ul style="list-style-type: none"> <li>• Definir a identidade visual e os padrões de interface do sistema.</li> <li>• Elaborar protótipos e wireframes que representem os fluxos de navegação.</li> <li>• Garantir a usabilidade e a acessibilidade da aplicação.</li> <li>• Conduzir testes de interface e validar a experiência do usuário (UX).</li> <li>• Colaborar com a equipe de desenvolvimento para assegurar a coerência entre design e funcionalidades.</li> <li>• Manter a consistência visual entre os diferentes módulos do sistema.</li> <li>• Adaptar o design às necessidades específicas das pequenas e médias empresas.</li> </ul>	
<b>Q.A/Tester:</b>  <b>João Luccas Marques</b>	<b>E-mail:</b>
<b>Responsabilidade:</b> <ul style="list-style-type: none"> <li>• Participar ativamente da definição de requisitos de qualidade do sistema.</li> <li>• Planejar, elaborar e executar casos de teste para validar as funcionalidades.</li> </ul>	

- Identificar, registrar e acompanhar a correção de falhas e inconsistências.
- Garantir que os módulos desenvolvidos atendam aos critérios de usabilidade e desempenho.
- Automatizar testes sempre que possível para aumentar a eficiência da validação.
- Validar as entregas intermediárias do projeto antes da homologação final.

Colaborar com desenvolvedores e designers na prevenção de defeitos e na melhoria contínua do produto.

**Desenvolvedor Full-Stack:**

**Marcos Paulo Moreira Damascena**

**E-mail:**
**Responsabilidade:**

- Desenvolver a interface do usuário (UI) usando HTML, CSS e JavaScript.
- Garantir experiência do usuário (UX) intuitiva e responsiva.
- Integrar o Front-End com APIs e serviços do Back-End.
- Otimizar a performance do sistema (tempo de carregamento, recursos, imagens).
- Manter código limpo, modular e bem documentado.

**Desenvolvedor Front-End:**

**Victor Manoel**

**E-mail:**

**silvavictormoraes27@gmail.com**

**Responsabilidades:**

- Desenvolver a interface do usuário (UI) usando HTML, CSS e JavaScript.
- Garantir experiência do usuário (UX) intuitiva e responsiva.
- Otimizar a performance do sistema (tempo de carregamento, recursos, imagens).
- Manter código limpo, modular e bem documentado.

**Desenvolvedor Back-End:**

**Victor Duarte**

**E-mail:**

**victorduarte0409123@gmail.com**

**Responsabilidades:**

- Implementar APIs RESTful para comunicação entre front-end e back-end
- Desenvolver módulos específicos do ERP (financeiro, estoque, vendas, etc.)

- Criar e manter endpoints para operações CRUD (Create, Read, Update, Delete)
- Implementar regras de negócio complexas específicas do domínio ERP
- Manter código limpo e bem documentado.

Equipe do projeto			
Nome	Lotação	Dedicação ao projeto	Atribuições (Códigos das atividades da EAP que o integrante participa)
Matheus Marques Portela	Análise de Negócio	Integral	
Matheus Marques Portela	Administração Banco de Dados	Integral	
Matheus Marques Portela	Desenvolvimento Back-End	Parcial	
Victor Duarte	Desenvolvimento Back-End	Integral	
Pedro Echebarria	Desenvolvimento Front-End	Integral	
Pedro Echebarria	Design e Interface	Parcial	
Pedro Echebarria	UI/UX	Integral	
Marcos Paulo	Desenvolvimento Front-End	Parcial	
João Luccas	Q.A / Tester	Integral	
Victor Manoel	Desenvolvimento Front-End	Parcial	

## **2. GERENCIAMENTO DO ESCOPO**

O gerenciamento do escopo do projeto POWP ERP será conduzido com base na Declaração de Escopo, a qual definirá de forma clara e objetiva os limites e as entregas previstas. Todas as entregas produzidas no decorrer do projeto deverão ser homologadas por meio de um Termo de Aceite, elaborado após a conferência dos requisitos estabelecidos no escopo, considerando os critérios de aceitação previamente definidos.

As entregas intermediárias serão submetidas à homologação pela Unidade Demandante, assegurando que os resultados parciais estejam alinhados às necessidades do negócio. Já o produto final do projeto será homologado conjuntamente pela Unidade Demandante e pelo Patrocinador, garantindo a validação integral dos objetivos estabelecidos. Caberá ao Gerente do Projeto providenciar as assinaturas do Termo de Aceite e comunicar à equipe a conclusão da entrega, conforme descrito no Plano de Gerenciamento da Comunicação.

O controle do escopo será realizado periodicamente em reuniões de acompanhamento do projeto. Caso seja identificada a necessidade de alteração por parte de qualquer integrante da equipe ou da área demandante, essa solicitação deverá ser submetida ao processo de Controle Integrado de Mudanças. Ressalta-se que todas as solicitações de alteração de escopo deverão ser formalizadas por escrito, mediante o preenchimento do Formulário de Solicitação de Mudanças, de forma a assegurar rastreabilidade, transparência e governança no processo decisório.

## **3. GERENCIAMENTO DO TEMPO**

O gerenciamento do tempo do projeto POWP ERP será conduzido por meio de cronogramas estruturados e acompanhado em reuniões periódicas de monitoramento. Durante essas reuniões, será avaliada a execução das atividades planejadas, permitindo identificar desvios em relação à linha de base do cronograma. Caso algum integrante da equipe identifique a necessidade de alteração nas datas ou durações previstas, a solicitação deverá ser submetida ao processo de Controle Integrado de Mudanças, conforme descrito no respectivo item desta documentação.

Na ocorrência de impactos negativos, serão aplicadas estratégias de mitigação compatíveis com os recursos disponíveis, tais como redução da duração de atividades, definição de atividades em simultaneidade, eliminação de tarefas não essenciais, ampliação da carga horária da equipe ou, se necessário, a inclusão de novos recursos. Tais medidas terão como objetivo assegurar o cumprimento dos prazos sem comprometer a qualidade das entregas.

A atualização da linha de base do cronograma somente será autorizada mediante aprovação formal do Gerente do Projeto e do Patrocinador. A versão anterior será devidamente arquivada e documentada, servindo como referência para análise e lições aprendidas.

Adicionalmente, após cada reunião de acompanhamento, o Gerente do Projeto será responsável por encaminhar a programação das atividades previstas para os 15 dias subsequentes aos respectivos responsáveis, garantindo clareza na execução. Com base nessas informações, o cronograma será atualizado e eventuais impactos sobre as datas marco do projeto serão avaliados, de modo a assegurar que os prazos estabelecidos sejam continuamente monitorados e cumpridos.

**4. GERENCIAMENTO DE CUSTOS**

Estimativa de recursos necessários					
Natureza da despesa	Itens da EAP	Custo (Ano 1)	Custo (Ano 2)	Custo (Ano 3)	Custo Total
Materiais de consumo		R\$	R\$	R\$	R\$
Materiais permanentes (equipamentos)		R\$	R\$	R\$	R\$
Aquisição de softwares		R\$	R\$	R\$	R\$
Obras e serviços de engenharia		R\$	R\$	R\$	R\$
Passagens e diárias		R\$	R\$	R\$	R\$
Contratação de consultoria		R\$	R\$	R\$	R\$
Capacitação		R\$	R\$	R\$	R\$
Serviços de Comunicação ou Pesquisa		R\$	R\$	R\$	R\$
Outros		R\$	R\$	R\$	R\$
Estimativa de recursos da PGE:		Total (Ano 1):	Total (Ano 2):	Total (Ano 3):	Total
<b>Situação orçamentária:</b> <i>(marque a situação do projeto)</i>		Os recursos já estão incluídos em dotação na LOA vigente ou foram previstos na proposta orçamentária para o próximo ano.			
		Não há previsão orçamentária, sendo necessário o remanejamento de recursos ou aprovação de crédito adicional.			
<b>Observações:</b>					

**5. GERENCIAMENTO DA QUALIDADE**

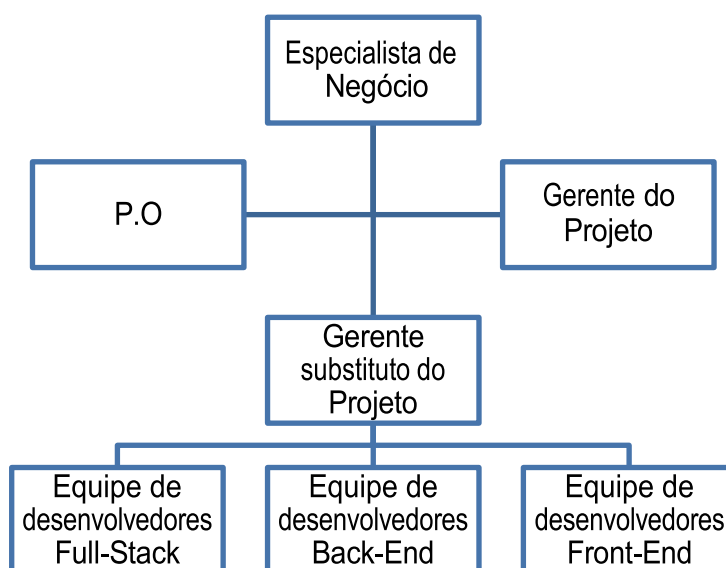
Entregas do projeto e critérios de aceitação					
<b>Cód. EAP</b>	<b>Produtos (entregas)</b> (Liste as principais entregas produzidas pelo projeto que deverão ser validados por meio do Termo de Aceite)	<b>Critérios de aceite</b> (Descreva os requisitos necessários à satisfação das expectativas do demandante)	<b>Responsável pela garantia da qualidade</b> (indicar quem garante a qualidade da entrega)	<b>Data da Entrega</b>	<b>Quem aprova</b> (Assina o Termo de Aceite)
2.1.1	Scripts de atendimento	1. 100% dos scripts definidos e cadastrados no sistema. 2. Zero erros gramaticais nos scripts. Tempo de leitura de cada script inferior ou igual a 15 segundos.	Supervisor de Atendimento	30/06/2015	Gerente de Atendimento
	Ajustar				

## 6. GERENCIAMENTO DE RECURSOS HUMANOS

O Gerente do Projeto não possuirá autoridade funcional sobre a equipe do projeto, porém possuirá autonomia de atribuição das atividades, com o apoio do Especialista de Negócio e do Patrocinador. Os papéis e responsabilidades da equipe do projeto estão descritos no item 1.10 deste Plano.

No caso de realocação de profissional integrante do projeto, caberá ao Gerente de Projeto, a identificação do substituto em comum acordo com as diretrizes do projeto e as funções a serem exercidas, e viabilizar junto ao Patrocinador a integração deste profissional à equipe. O mesmo aplica-se à alocação de recursos adicionais.

### 6.1 ORGANOGrama DO PROJETO



### 6.2 DIRETÓRIO DA EQUIPE DO PROJETO

No.	Nome	Unidade	E-mail	Telefone
1	Matheus Marques Portela		matheusmarquesportela@gmail.com	(62) 99518-7839
2				
3				
4				
5				
6				



**7. GERENCIAMENTO DA COMUNICAÇÃO**
**7.1 MATRIZ DE COMUNICAÇÃO**

[Descrever os principais assuntos informados, ação esperada, quem envia e quem recebe, por qual meio e em qual periodicidade ]

MATRIZ DE COMUNICAÇÃO - REUNIÕES							
Grupo de Processos		Iniciação	Planejamento		Execução e Controle		Encerramento
Reunião		Reunião inicial do projeto	Reunião de Planejamento	Reunião de Partida (kick-off)	Reunião de Acompanhamento e Controle de Mudanças	Reunião Específica	Reunião de Encerramento do Projeto (Closeout)
Periodicidade		No início do projeto	[Registrar a periodicidade, dia da semana e horário que as reuniões serão realizadas]	No final do planejamento	[Registrar a periodicidade, dia da semana e horário que as reuniões serão realizadas]	Quando necessário	No final do projeto
Participantes	Patrocinador	X		X			X
	Especialista de Negócio	X	X	X	X		X
	Representante do Escritório de Projetos	X		X	X		X
	Gerente do Projeto	X	X	X	X		X
	Demais integrantes da equipe do projeto	X	X	X	X		X

<b>MATRIZ DE COMUNICAÇÃO - DOCUMENTOS</b>										
<b>Grupo de Processos</b>		<b>Iniciação</b>	<b>Planejamento</b>		<b>Execução e Monitoramento</b>			<b>Encerramento</b>		
<b>Documentos</b>		<b>Termo de Abertura</b>	<b>Plano de Projeto</b>	<b>Ata de Reunião de Planejamento</b>	<b>Ata de Reunião de Acompanhamento</b>	<b>Termo de Aceite</b>	<b>Relatório de Acompanhamento</b>	<b>Solicitação de Mudança</b>	<b>Lições Aprendidas</b>	<b>Termo de Encerramento do Projeto</b>
<b>Meio de Comunicação</b>		. Em papel e assinado . No SG Channel	. Em papel e assinado . No SG Channel	e-mail	e-mail	. Em papel e assinado . No SG Channel	SG Channel	. Em papel . SG Channel	. SG Channel	. Em papel e assinado . No SG Channel
<b>Periodicidade</b>		No início do projeto	No final do planejamento	Após cada reunião de planejamento	Após cada reunião de acompanhamento	Após cada entrega intermediária ou de produto	[Registrar periodicidade de envio]	Sempre que houver necessidade	Elaborado durante todo o projeto e emitido no encerramento do projeto	No encerramento do projeto
Partes Interessadas (V - Validar) / (C - Comunicar) (Preencha com V ou C na linha)	Patrocinador	V	V			V	C	V		V
	Especialista de Negócio	V	V	C	C	V	C	V		V
	Escritório de Projetos	V	V	C	C	C	C	C	C	C
	Gerente do Projeto				C			V		
	Participantes da reunião			C	C					
	Demais integrantes da equipe		C			C	C	C		C

**V - Validar / C - Comunicar**

## 8. GERENCIAMENTO DE RISCOS

O gerenciamento de riscos do projeto **POWP ERP** será conduzido com base nos riscos previamente identificados, bem como no monitoramento contínuo e na avaliação de novos riscos que possam surgir ao longo de toda a execução do projeto. Esse processo visa antecipar ameaças e oportunidades, permitindo que a equipe adote medidas preventivas ou corretivas de forma estruturada, garantindo maior segurança e eficácia na entrega do sistema.

Todos os riscos serão controlados durante o ciclo de vida do projeto, com acompanhamento sistemático nas reuniões de monitoramento. O Gerente do Projeto, juntamente com os responsáveis pelos riscos, será encarregado de supervisionar e acompanhar os riscos identificados e priorizados. Quando houver indícios de ocorrência de qualquer risco previamente mapeado, os responsáveis deverão comunicar imediatamente o Gerente do Projeto e os responsáveis pelas atividades afetadas, de modo que sejam acionadas as respostas apropriadas para mitigar impactos negativos ou aproveitar oportunidades.

Os riscos do projeto estão registrados no documento “Matriz de Riscos - POWP ERP”, elaborado conforme etapas formais de gerenciamento. Na etapa de identificação, os riscos são reconhecidos e classificados quanto à sua natureza. Em seguida, na análise qualitativa, cada risco é avaliado considerando-se a probabilidade de ocorrência e o impacto sobre o projeto. A combinação desses dois fatores determina a classificação final do risco, permitindo priorizar ações de mitigação ou aproveitamento de oportunidades. As consequências dos riscos podem representar tanto ameaças (riscos negativos) ao sucesso do projeto quanto oportunidades (riscos positivos) para agregar benefícios à entrega final;

Para a análise qualitativa do risco - método de “Probabilidade x Impacto” - foram utilizados os seguintes conceitos/nomenclatura e pontuações para definição da Classificação do risco:

Construção Análise de Riscos - Probabilidade X Impacto	
<b>Probabilidade de ocorrência</b>	É a real chance de o evento vir a acontecer, dentro de uma escala.
<b>Impacto</b>	Dimensão das consequências positivas ou negativas produzidas direta ou indiretamente pelo fator de risco.
<b>Severidade</b>	Calculada em função dos valores atribuídos à PROBABILIDADE e ao IMPACTO.
<b>Graduações utilizadas para os fatores: Probabilidade X Impacto</b>	1. Muito Baixo 2. Baixo 3. Médio 4. Alto 5. Muito Alto

Planejamento de Respostas - o tratamento é realizado para os riscos médios e altos, priorizados na análise, e consiste na determinação de ações para melhorar oportunidades e reduzir ameaças ao projeto.

Para o planejamento de respostas aos riscos foram utilizadas as seguintes ações:

<b>Estratégia de Resposta</b>	
<b>Riscos Negativos (Ameaças)</b>	Prevenir Transferir Mitigar Aceitar
<b>Riscos Positivos (Oportunidades)</b>	Explorar Compartilhar Melhorar Aceitar

A Planilha de tratamento de riscos para este projeto encontra-se nos anexos a este Plano de Projeto.

## 9. GERENCIAMENTO DE AQUISIÇÕES

Seguem os itens a serem adquiridos pelo projeto:

<b>Item a ser adquirido/ contratado</b>	<b>Forma de aquisição/ contratação</b>	<b>Modalidade</b>	<b>Responsável pela especificação</b>	<b>Data esperada para a aquisição/ contratação</b>

## 10. GERENCIAMENTO DAS PARTES INTERESSADAS

O gerenciamento das partes interessadas do projeto será realizado com base naquelas previamente identificadas, bem como no monitoramento de novas partes interessadas que poderão vir a ser identificadas durante toda a execução do projeto.

As partes interessadas deste projeto estão registradas no documento “Registro das Partes Interessadas- **POWP – ERP**” de acordo com as etapas listadas abaixo:

Identificar as partes interessadas - todas as pessoas ou organizações que podem influenciar diretamente o projeto são identificadas e relacionadas aos seus interesses, nível de engajamento e impacto no sucesso do projeto.

Planejar o gerenciamento das partes interessadas – são definidas estratégias eficazes de modo a garantir o engajamento adequado das partes interessadas no projeto, com base nos levantamentos feitos no processo na etapa anterior.

Gerenciar o engajamento das partes interessadas - os níveis de engajamento atual e desejado para as partes interessadas mais importantes são definidos, e quais estratégias serão usadas para quebrar resistências e garantir seu engajamento no projeto. As estratégias podem ser genéricas, para grupos de pessoas ou para pessoas específicas.

Nível de engajamento		
<b>A</b>	<b>Apoiador</b>	Apoia o projeto
<b>N</b>	<b>Neutro</b>	Tem conhecimento sobre o projeto, porém, está neutro
<b>R</b>	<b>Resistente</b>	Se tiver oportunidade, prejudicará o andamento do projeto
<b>D</b>	<b>Desinformado</b>	Não tem informação sobre o projeto, por isso, não tem posição formada
<b>L</b>	<b>Lidera</b>	Engajado em garantir o sucesso do projeto

Controlar o engajamento das partes interessadas

Serão feitas reuniões mensais com a equipe do projeto para:

- Verificar se as expectativas em relação ao projeto continuam as mesmas;
- Identificar novas partes interessadas;
- Verificar quais partes interessadas se tornaram mais importantes para o sucesso do projeto e rever as estratégias para engajá-las;
- Identificar se existem problemas de relacionamento ou de engajamento entre a equipe do projeto;
- Identificar e avaliar pontos de atenção relacionados com engajamento;
- Avaliar o tempo de resolução dos problemas e se houve dificuldade de engajamento de membros da equipe;
- Avaliar os problemas, suas causas e verificar se não estão atreladas com resistência de alguma parte interessada.

Serão feitas reuniões mensais de forma individual entre o gerente de projeto e as principais partes interessadas com o mesmo objetivo.

Serão feitas pesquisas de satisfação garantindo sigilo dos participantes de modo a identificar problemas de relacionamento, pessoas resistentes ao projeto que estão impactando de forma negativa.

## 11. CONTROLE INTEGRADO DE MUDANÇAS

O objetivo do plano de Controle Integrado de Mudanças do projeto **POWP ERP** é definir claramente os papéis, responsabilidades, processos e ferramentas necessários para gerenciar de forma eficaz qualquer alteração relacionada ao escopo, custo ou prazo do projeto. Esse controle visa assegurar que todas as mudanças sejam identificadas, documentadas, analisadas e autorizadas antes de sua implementação, garantindo a integridade do projeto e a conformidade com os objetivos previamente estabelecidos.

O processo de controle de mudanças inicia-se com a solicitação da mudança, que pode ser originada por qualquer membro da equipe ou parte interessada. Ao identificar a necessidade de alteração, o solicitante deve formalizar a mudança por meio do Formulário de Solicitação de Mudança, que será encaminhado ao Gerente do Projeto para análise.

Na etapa de identificação da mudança, o Gerente do Projeto atribui uma identificação única a cada solicitação, utilizando a planilha de Registro de Mudança como instrumento oficial de controle e rastreabilidade. Essa identificação permite acompanhar a evolução da solicitação desde sua origem até a decisão final, garantindo transparência e registro histórico para futuras consultas e lições aprendidas.

Em seguida, cada solicitação passa pelo processo de análise e avaliação de impacto, no qual são considerados os efeitos potenciais sobre o escopo, cronograma, custos e qualidade do projeto. Após a análise, a mudança deve ser submetida à autorização formal pelo Gerente do Projeto e, quando necessário, pelo Patrocinador, antes de sua implementação. Dessa forma, o Controle Integrado de Mudanças assegura que o projeto **POWP ERP** seja conduzido de maneira organizada, minimizando riscos e mantendo o alinhamento com os objetivos estratégicos da organização.

### Avaliação de impacto da mudança

O Gerente do Projeto deverá coordenar com a equipe a avaliação do impacto gerado pela mudança proposta conforme tabela a seguir:

Tipo de mudança	Análise do Impacto
<b>Escopo</b>	Avaliar: <ul style="list-style-type: none"> <li>• Plano de trabalho</li> <li>• Prazo adicional</li> <li>• Riscos associados com a mudança de escopo</li> </ul>
<b>Cronograma</b>	Avaliar: <ul style="list-style-type: none"> <li>• Alteração no prazo final do projeto</li> <li>• Ações corretivas (reduzir ou ampliar escopo e custo associado)</li> <li>• Riscos</li> </ul>
<b>Custo</b>	Avaliar: <ul style="list-style-type: none"> <li>• Ações corretivas (compensar a variação de custo ajustando-se escopo e tempo)</li> <li>• Custo final do projeto</li> <li>• Riscos</li> </ul>

### Documentação

A análise deverá ser documentada no próprio formulário de Solicitação de Mudanças e na Planilha de Registro de Mudanças. Documentos utilizados na determinação dos impactos devem ser anexados ao formulário.

### Aprovação

As mudanças deverão ser submetidas ao Especialista de Negócio e ao Patrocinador do Projeto. Uma vez autorizadas as mudanças, novas linhas de base de escopo, cronograma e custos serão estabelecidos com base nas linhas de base anteriores e nas alterações decorrentes das mesmas. O Gerente do Projeto estará formalmente autorizado a iniciar a execução das mudanças.

## 12. ANEXOS

Link para o projeto no Figma:

- <https://www.figma.com/design/U36p96MmmvVklnhfNwUufZ/POWP?node-id=0-1&t=DVFtgjqWJ6fWutJH-1>

Link para o Front -End já produzido junto com landing page:

- <https://oportela.github.io/POWP/Powp/index.html>



# **Documento de Projeto de Software**



## Histórico de revisões do Documento

Versão Data (XX.YY) (DD/MMM/YYYY)		Autor	Descrição
1.0	11/09/2025	Matheus Marques	Criação do documento, anexando todas os modelos realizados até o presente modelo, descrição do projeto e o escopo inicial.
1.1	13/09/2025	Marcos Paulo Moreira	Ajustes nos requisitos funcionais , casos de uso , premissas, restrições de software, restrição de hardware , regras de negocio.

## Índice

1.	INTRODUÇÃO .....	4
1.1.	PROPÓSITO DO DOCUMENTO DE REQUISITOS.....	4
1.2.	PÚBLICO ALVO .....	4
2.	DESCRIÇÃO GERAL DO PRODUTO .....	5
2.1.	SITUAÇÃO ATUAL.....	5
2.2.	ESCOPO .....	5
2.3.	ATORES.....	6
2.4.	PREMISSAS .....	6
3.	REQUISITOS.....	7
3.1.	REQUISITOS FUNCIONAIS .....	7
3.2.	REQUISITOS NÃO FUNCIONAIS .....	7
3.3.	REGRAS DE NEGÓCIO .....	8
3.4.	RESTRIÇÕES DE HARDWARE .....	8
3.5.	RESTRIÇÕES DE SOFTWARE .....	8
3.6.	IDENTIFICAÇÃO DOS CASOS DE USO.....	8
3.7.	DIAGRAMA DE CASOS DE Uso .....	10
3.8.	DIAGRAMA DE CLASSES.....	11
4.	PROTÓTIPOS DE INTERFACE DE USUÁRIO .....	12
5.	APROVAÇÃO.....	13

# 1. Introdução

## 1.1. Propósito do Documento de Requisitos

Este documento visa detalhar os requisitos dos usuários que deverão ser atendidos pelo sistema a ser desenvolvido no projeto POWP. Ele servirá como base para a construção do sistema pelos programadores, fornecendo uma descrição detalhada das funções do produto, suas interfaces e o comportamento esperado do sistema.

## 1.2. Público Alvo

Este documento destina-se a todos os stakeholders envolvidos no desenvolvimento, operação e utilização do sistema POWP, incluindo:

- **Arquitetos de Software:** responsáveis pelo planejamento e definição da arquitetura do sistema.
- **Engenheiros de Software e Testadores:** encarregados da implementação, manutenção e validação do sistema.
- **Usuários Fornecedores:** participantes que interagem com o sistema para fornecer dados, insumos ou serviços necessários ao seu funcionamento.
- **Administradores da Rede:** responsáveis pelo monitoramento, segurança e gestão do ambiente do sistema.
- **Gestores e Tomadores de Decisão:** interessados nos resultados, métricas e desempenho do sistema para planejamento estratégico.
- **Parceiros:** interessados na viabilidade, confiabilidade e retorno do projeto.

## 2. Descrição Geral do Produto

*Este documento contém a especificação de requisitos para uma aplicação Web, que será um ERP modal de forma online, aonde o usuário poderá acessar de onde ele quiser, em seu computador ou celular. Nele contará com módulos de vendas, cadastro, controle de estoque, financeiro, relatórios e dashboards e chat bot de IA integrado em sua base para melhores tomadas de decisões.*

### 2.1. Situação Atual

Atualmente, o cenário das empresas brasileiras é composto em mais de 90% por Microempreendedores Individuais (MEIs), segundo dados do SEBRAE. A maioria desses negócios não possui um sistema estruturado de gestão, recorrendo a processos manuais ou controles em planilhas. Além disso, seus faturamentos geralmente são mais baixos, o que dificulta o investimento em sistemas robustos devido aos altos custos de aquisição e manutenção.

### 2.2. Escopo

Nº	Módulo	Descrição
1	Autenticação	Será desenvolvido um aplicativo mobile capaz de atender aos sistemas operacionais iOS, Android e Windows Phone.
2	Web Service	Será desenvolvido um web service para servidor como provedor de dados para o aplicativo, em formato JSON.
3	Gestor do aplicativo	Será desenvolvido um sistema para gerenciar o aplicativo, que possibilite o gerenciamento de pacotes, como cadastrar, editar e excluir informações.
4	Estoque	Será desenvolvido um sistema para aquisição e controle de estoque, podendo gerar ordem de compra e solicitações de cotação.
5	Fornecedores	Será desenvolvido um sistema para gerenciar todas as informações e interações com fornecedores, desde o cadastro até a avaliação de desempenho.

## Limitações do Sistema

Documento de Requisitos

11/09/2025

6	Financeiro	Será desenvolvido um sistema que possibilite gerenciar os lançamentos de contas a pagar e a receber, fluxo de caixa e conciliação bancária.
7		

### 2.3. Atores

O aplicativo possui dois tipos de atores, são eles:

- **Usuário Administrador**
- **Usuário Analista/Assistente**

Nº	Ator	Definição e Privilégio de Acesso e Segurança
1	Usuário Administrador	Possui acesso às funcionalidades de visualização, cadastro e edição dos registros no sistema.  Caso a empresa tenha um equipe de TI e precise, disponibilizamos acesso a base de dados para administração dela também.
2	Usuário Analista/Assistente	Possui acesso somente a funcionalidades pertinentes à sua função, seja ela do financeiro, estoque, etc.

### 2.4. Premissas

- Assume-se que todos os usuários terão acesso à **internet estável** para utilizar o sistema web.
- Os usuários utilizarão **navegadores compatíveis e atualizados** (Chrome, Edge, Firefox, Safari).
- Assume-se que o servidor escolhido terá **recursos mínimos recomendados** (CPU, RAM, armazenamento) para suportar a operação do sistema.
- O sistema utilizará bancos de dados relacionais compatíveis (PostgreSQL ou MySQL).
- Haverá **usuários finais disponíveis** para testes e validação das funcionalidades antes da entrega.
- O sistema será inicialmente utilizado por **pequenas empresas e MEIs**, com volume de dados compatível com a primeira versão do ERP.
- Novas funcionalidades (integrações externas, aplicativos móveis nativos, dashboards avançados) serão consideradas **apenas em versões futuras**, não na entrega inicial.

- Assume-se que **medidas básicas de segurança** (login, senha, perfil de acesso) serão suficientes para a operação inicial, sem necessidade de autenticação multifator ou criptografia avançada de dados.
- A equipe de desenvolvimento terá **ferramentas e tecnologias necessárias** para implementar o sistema (Laravel, React, etc.).
- Os usuários aceitarão as **limitações da primeira versão**, como ausência de integração externa, dashboards simplificados e suporte limitado a dispositivos móveis.



## 3. Requisitos

### 3.1. Requisitos Funcionais

ID	Descrição
RF 01	Login/Autenticação do usuário
RF 02	Cadastrar Produto
RF 03	Cadastrar Usuário
RF 04	Cadastrar Funcionário
RF 05	Digitar Pedido de Venda
RF 06	Controlar estoque
RF 07	Gerir contas a pagar e receber
RF 08	Criação de Dashboards
RF 09	Criação de Relatórios
RF 10	Chat Bot com IA integradas
RF 11	Perfis de Acesso
RF 12	Gestão de Permissões
RF 13	Auditoria de Acesso
RF 14	
RF 15	

### 3.2. Requisitos Não Funcionais

ID	Descrição	Categoria
1	Somente usuários autenticados podem acessar o gestor do aplicativo.	Segurança
2	A tela de listagem deve ser simples, exibindo apenas os destinos, para facilitar o entendimento do usuário.	Usabilidade
3	O aplicativo deve ser desenvolvido em HTML, CSS e JS junto com PHP/Laravel 12, assim não será necessário um release para cada plataforma.	Hardware e Software
4	A aplicação rodará em um servidor na nuvem, podendo ser AWS.	Hardware
5	O banco de dados será feito no MySQL, por ter uma arquitetura conhecida, sem Open Source e fácil manutenibilidade.	Software
6	Em sua programação será utilizadas boas práticas, como Clean Code, testes constantes em todo o sistema afim de não acontecer erros ou bugs.	Desenvolvimento

## 4. Regras de Negócio

ID	Nome	Descrição
1	Gestão de Usuários	<ul style="list-style-type: none"><li>Cada usuário deve ter <b>login e senha únicos</b>.</li><li>Usuários podem ter <b>perfis diferentes</b> (Administrador, Funcionário, Contador, etc.), determinando níveis de acesso e permissões.</li><li>Somente <b>administradores</b> podem cadastrar ou remover usuários e definir permissões.</li></ul>
2	Cadastro de Produtos	<ul style="list-style-type: none"><li>Todo produto deve possuir <b>nome, código único, preço de venda e quantidade em estoque</b>.</li><li>Produtos com o <b>mesmo código</b> não podem ser cadastrados duas vezes.</li><li>O estoque deve ser atualizado automaticamente ao cadastrar venda ou entrada de produtos.</li></ul>
3	Pedidos de Venda	<ul style="list-style-type: none"><li>Todo pedido deve estar vinculado a um <b>cliente cadastrado</b>.</li><li>O sistema deve <b>validar estoque disponível</b> antes de confirmar o pedido.</li><li>Pedidos não podem ser concluídos se houver <b>produtos em falta</b>.</li></ul>
4	Controle de Estoque	<ul style="list-style-type: none"><li>Qualquer movimentação (entrada ou saída) deve ser registrada com <b>data, usuário responsável e motivo</b>.</li><li>O estoque não pode ser negativo.</li><li>Alertas automáticos devem ser disparados quando o estoque atingir níveis mínimos pré-definidos.</li></ul>

## Limitações do Sistema

Documento de Requisitos

11/09/2025

5	Contas a Pagar e Receber	<ul style="list-style-type: none"><li>Somente usuários com permissão financeira podem cadastrar, editar ou quitar contas.</li><li>As contas devem ter <b>data de vencimento, valor, categoria e status</b> (pendente, pago, atrasado).</li><li>O sistema deve gerar <b>alertas de contas próximas do vencimento</b> ou vencidas.</li><li></li></ul>
6	Relatórios e Dashboards	<ul style="list-style-type: none"><li>Somente usuários autorizados podem acessar relatórios financeiros ou de vendas.</li><li>Relatórios devem refletir os <b>dados atuais do sistema</b>, garantindo consistência.</li><li>Dashboards devem exibir informações <b>somente do período selecionado</b> pelo usuário.</li><li></li></ul>
7	Segurança e Auditoria	<ul style="list-style-type: none"><li>Todas as operações críticas (ex: cadastro de usuário, movimentação de estoque, alteração de preço) devem ser <b>registradas em log</b>.</li><li>Usuários não podem acessar funções que não correspondem ao seu perfil.</li><li>Tentativas de login incorreto devem ser limitadas para evitar ataques de força bruta.</li><li></li></ul>
8		<ul style="list-style-type: none"><li>O ERP Powp é destinado a <b>pequenas empresas e MEIs</b>; grandes corporações podem não ter suporte adequado para volume massivo de dados.</li><li>Funcionalidades avançadas, como integração contábil automática ou machine learning, não estão disponíveis nesta versão inicial.</li></ul>

## Limitações do Sistema

Documento de Requisitos

11/09/2025

- Alterações no banco de dados pelo usuário final não são permitidas.
- 

### 4.1. Restrições de Hardware

Nº	Descrição
1	Dispositivo com no mínimo de 4GB de memória de RAM.
2	Navegador instalado no computador ou celular.
3	O sistema será hospedado em servidores padrão de cloud (AWS, Azure ou DigitalOcean) ou servidor físico equivalente.
4	Necessário suporte a armazenamento em SSD para desempenho ideal.
5	Sistemas de backup automático serão limitados ao armazenamento disponível no servidor; sem suporte a unidades externas na versão inicial.
6	Não há dependência obrigatória de impressoras, leitores de código de barras ou outros periféricos na primeira versão.
7	Qualquer integração com periféricos será opcional e limitada a versões futuras.
8	O sistema será projetado para pequenas empresas, com até algumas centenas de usuários simultâneos.
9	Grandes volumes de dados ou múltiplos usuários simultâneos exigirão upgrades de servidor ou cloud scaling.
10	O dispositivo deve conter a função <i>Wi-Fi</i> ou Plano de Dados pelo Chip SIM para acesso à internet.

### 4.2. Restrições de Software

Nº	Descrição
1	O sistema deverá ser desenvolvido para web, sendo acessível via navegadores modernos (Chrome, Edge, Firefox, Safari).
2	Não terá suporte oficial para versões de navegadores desatualizadas (anteriores a 2 anos) ou navegadores móveis antigos.
3	O sistema será limitado ao uso de bancos de dados relacionais compatíveis com PostgreSQL e MySQL.
4	Não será possível utilizar bancos NoSQL ou distribuídos nesta primeira versão.
5	Inicialmente, não haverá integração nativa com sistemas de terceiros (como ERPs maiores, CRMs ou softwares contábeis externos).
6	APIs externas poderão ser adicionadas apenas em versões futuras, mediante

## Limitações do Sistema

Documento de Requisitos

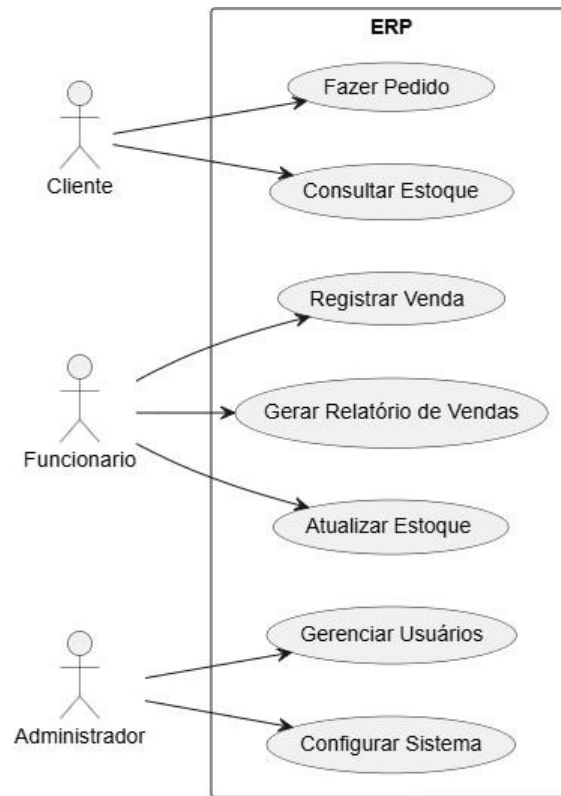
11/09/2025

	análise de viabilidade.
7	O ERP Powp será otimizado para desktops; suporte a telas menores (tablets e smartphones) será limitado na primeira versão.
8	A autenticação será baseada em login e senha. Autenticação via SSO, OAuth ou biometria não será suportada inicialmente.
9	Limitação de políticas avançadas de segurança, como bloqueio geográfico ou autenticação multifator, não serão implementadas nesta versão.
10	Funcionalidades como inteligência artificial para previsão de vendas, automação contábil avançada ou machine learning não estarão disponíveis.
11	Não será possível alterar a estrutura do banco de dados ou criar módulos próprios pelo usuário final nesta versão.

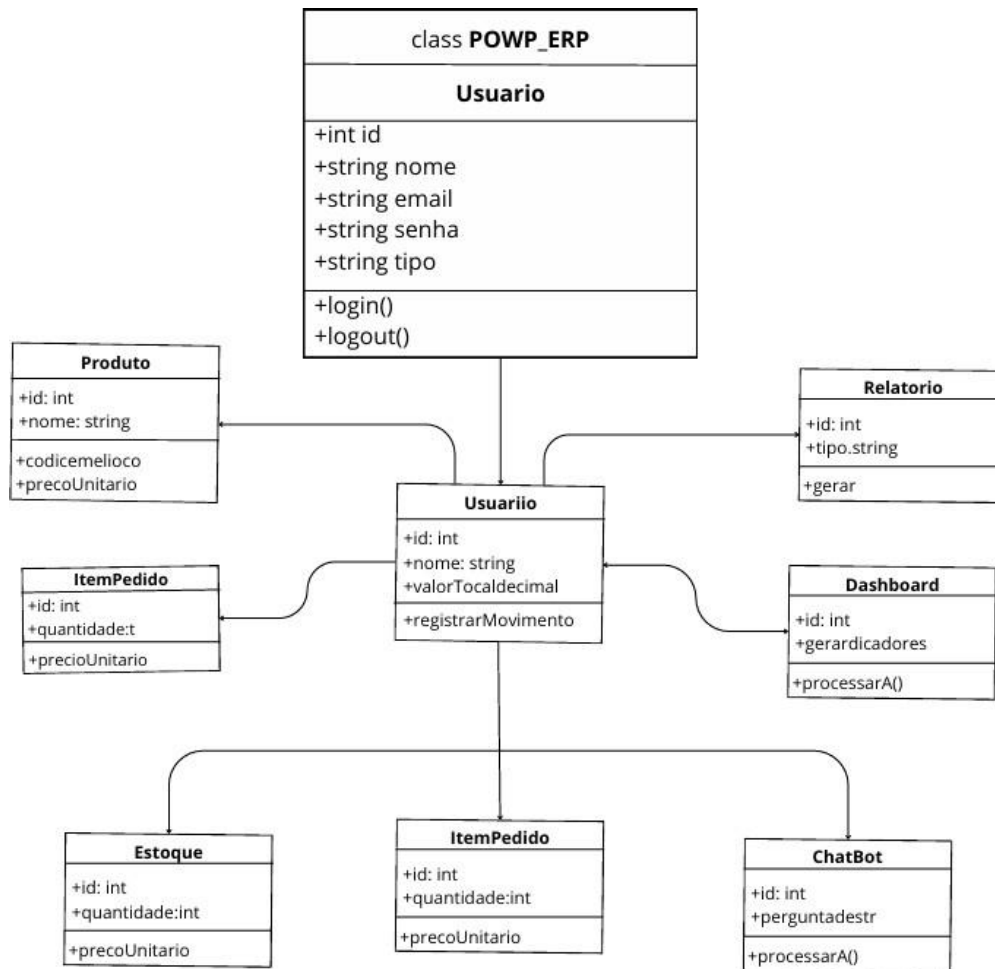
### 4.3. Identificação dos Casos de Uso

ID	Caso de Uso	Descrição
CU01	Login/Autenticação do Usuário	Permitir que usuários autenticados acessem o sistema com segurança.
CU02	Cadastro de Produto	Permitir cadastrar novos produtos no sistema.
CU03	Cadastro de Usuário	Permitir cadastrar novos usuários no sistema com permissões específicas.
CU04	Cadastro de Funcionário	Permitir registrar funcionários da empresa no sistema.
CU05	Digitar Pedido de Venda	Permitir registrar vendas realizadas pela empresa.
CU06	Controlar Estoque	Monitorar e atualizar quantidades de produtos em estoque.
CU07	Gerir Contas a Pagar e Receber	Gerenciar movimentações financeiras da empresa.
CU08	Criação de Dashboards	Gerar dashboards com indicadores do negócio.
CU09	Criação de Relatórios	Gerar relatórios detalhados de operações e finanças.
CU10	Chatbot com IA	Permitir interação com um chatbot inteligente para suporte ou dúvidas.

#### 4.4. Diagrama de Casos de Uso

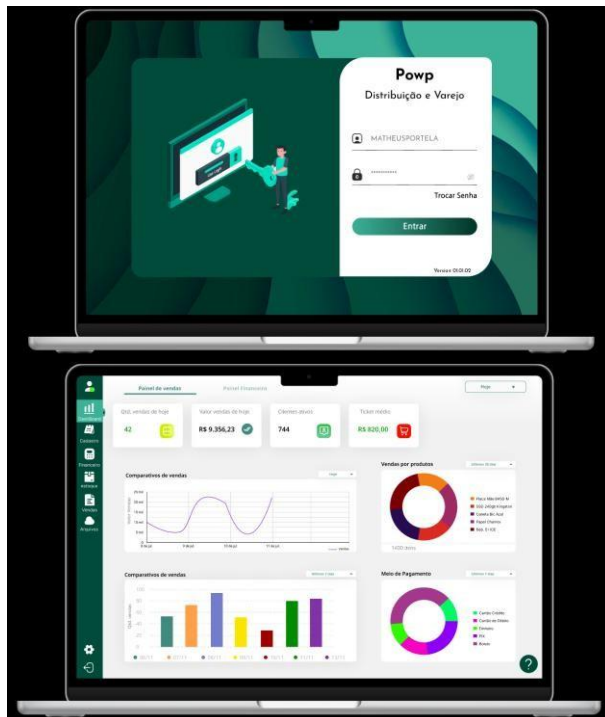


## 4.5. Diagrama de Classes





## 5. Protótipos de Interface de Usuário



## 6. Aprovação

### Aprovadores

Matheus Marques	P.O / Desenvolvedor Back-End / DBA
-----------------	------------------------------------

João Luccas	Q.A / Tester
-------------	--------------

Pedro Henrique	Designer UI/UX
----------------	----------------

Marcos Paulo	Desenvolvedor Front-End
--------------	-------------------------

---

## Limitações do Sistema

Documento de Requisitos

11/09/2025

Victor Manoel

Designer UI/UX

## Proposta Técnica - Projeto POWP ERP

### 1. Identificação

- **Projeto:** POWP ERP
  - **Data:** Setembro/2025
  - **Versão:** 1.0
- 

### 2. Justificativa

O sistema POWP ERP foi desenvolvido para atender pequenas e médias empresas, que representam a maior parte do mercado brasileiro e muitas vezes carecem de ferramentas de gestão acessíveis e eficazes. Sua arquitetura modular permite que cada organização utilize apenas os módulos necessários, reduzindo custos e garantindo maior flexibilidade.

Com uma interface simples e intuitiva, o sistema busca facilitar a adoção tecnológica e otimizar os processos de gestão dessas empresas, garantindo eficiência operacional e competitividade no mercado.

---

### 3. Produto

O projeto POWP ERP tem como resultado final o desenvolvimento de um sistema de gestão empresarial modular, intuitivo e acessível, voltado para freelancers, pequenas e médias empresas.

O produto busca entregar uma plataforma que centralize e organize as principais operações de um negócio, como:

- Cadastro de clientes
- Controle de estoque
- Gestão de vendas
- Relatórios gerenciais

Tudo de forma ágil, segura e escalável.

---

### 4. Premissas

- Os usuários finais possuem infraestrutura tecnológica adequada (computadores, dispositivos móveis e internet estável).

- Pequenas e médias empresas demonstram interesse real em adotar uma solução modular, intuitiva e de baixo custo.
  - Os colaboradores das empresas usuárias possuem conhecimentos básicos de informática, permitindo rápida adaptação ao sistema.
  - Haverá disponibilidade contínua de recursos financeiros e humanos para desenvolvimento, testes, manutenção e evolução do sistema.
  - Os módulos desenvolvidos manterão compatibilidade entre si, assegurando consistência das informações e escalabilidade.
  - O sistema estará em conformidade com legislações vigentes (tributária, fiscal e trabalhista).
- 

## 5. Restrições

- **Prazo:** O projeto deverá ser concluído e entregue integralmente até o final do segundo semestre de 2026.
  - **Escopo fixo:** Devem ser entregues, com funcionalidades essenciais, os seguintes módulos:
    - Landing Page
    - Autenticação de usuários
    - Dashboards iniciais
    - Módulo de Cadastro (Fornecedor, Cliente, Funcionário, Usuário, Produtos)
    - Módulo de Estoque
    - Módulo Financeiro (Contas a pagar e receber)
    - Módulo de Vendas
    - Chatbot com IA
  - **Recursos Humanos:** A equipe será limitada a 6 integrantes, sem possibilidade de expansão.
- 

## 6. Marcos do Projeto

- **20/12/2025:** Entrega da autenticação e acesso ao sistema
- **28/02/2026:** Entrega dos módulos de cadastro concluídos
- **15/04/2026:** Entrega do módulo de vendas junto com o de controle de estoque

## 7. Contribuições individuais

- **Matheus Marques:** Responsável pela regra de negócio e análise de mercado, além de criação e manipulação dos dados;
- **João Luccas:** Criação das documentações do projeto e ajustes técnicos;
- **Pedro Echebarria:** Criação da UI/UX e interface do sistema via Figma;
- **Victor Manoel:** Criação e aplicação do front-end;

- **Victor Duarte:** Criação das primeiras API's do sistema via PHP.
- **Marcos Paulo:** Criação e aplicação do front-end;

# **Documento de Indicadores**



## Indicadores

Indicador	Objetivo	Formula / Método de Cálculo	Fonte dos Dados	Periodicidade	Meta / Valor Esperado	Responsável	Forma de Apresentação
<b>Velocidade da Sprint</b>	Medir produtividade da equipe	Total de Issues Concluídas / nº de sprints	GitHub Issues / Projects	A cada Sprint	≥ 20 issues/sprint	Scrum Master	Gráfico de barras comparando Sprints
<b>Lead Time de Feature</b>	Medir tempo entre abertura de uma issue e o merge no repositório	Data de merge – Data de criação da Issue	GitHub Issues + Pull Requests	Mensal	≤ 10 dias	Tech Lead / P.O	Relatório mensal com média e desvio padrão
<b>Taxa de Bugs em Produção</b>	Avaliar qualidade após deploy	(#bugs reportados / total de deploys) x 100	GitHub Issues (label bug)	Mensal	≤ 5%	Q.A	Gráfico de linha
<b>Cobertura de Teste automatizados</b>	Garantir estabilidade do código	Linhas cobertas / linhas totais (%)	Ferramenta de testes	Semanal	≥ 80%	Dev Team	Dashboard CI/CD
<b>Tempo médio de Resolução de Issues</b>	Medir agilidade no suporte e	Σ (fechamento – abertura)	GitHub Issues	Mensal	≤ 3 dias (média)	Tech Lead	Relatório + gráfico boxplot

	correção	÷ nº issues fechadas					
<b>Frequência de Deploys</b>	Avaliar capacidade de entrega continua	Nº de deploys no período	Logs de CI/CD	Mensal	≥ 2 deploys/mês	DevOps	Timeline de releases
<b>Complexidade / Qualidade de código</b>	Manter código sustentável e limpo	Métricas de duplicação e code smells	Análise estática (SonarQube)	Semanal	≤ 5% duplicação, 0 vulnerabilidades críticas	Dev team	Relatório SonarQube
<b>Satisfação do Usuário (Feedback)</b>	Avaliar experiência dos usuários finais	Formulário de feedback ou NPS	Pesquisas com usuários/testes piloto	Trimestral	NPS ≥ 70	P.O	Relatório + gráfico radar



## Documento Técnico – Projeto POWP ERP

### 1. Identificação

- Projeto: POWP ERP
  - Data: Setembro/2025
  - Versão: 1.0
- 

### 2. Arquitetura e Implementação do Backend

Esta seção detalha a arquitetura, as tecnologias e a pipeline de dados implementadas no backend do projeto POWP ERP.

#### CI/CD Pipeline

A pipeline de dados é responsável pelo fluxo, processamento e transformação dos dados dentro do sistema. Este projeto utiliza GitHub Actions para Integração Contínua (CI). A pipeline está configurada para executar automaticamente nos seguintes casos:

- Push nas branches: `main`, `master`, `develop`
- Pull Requests para: `main`, `master`, `develop`

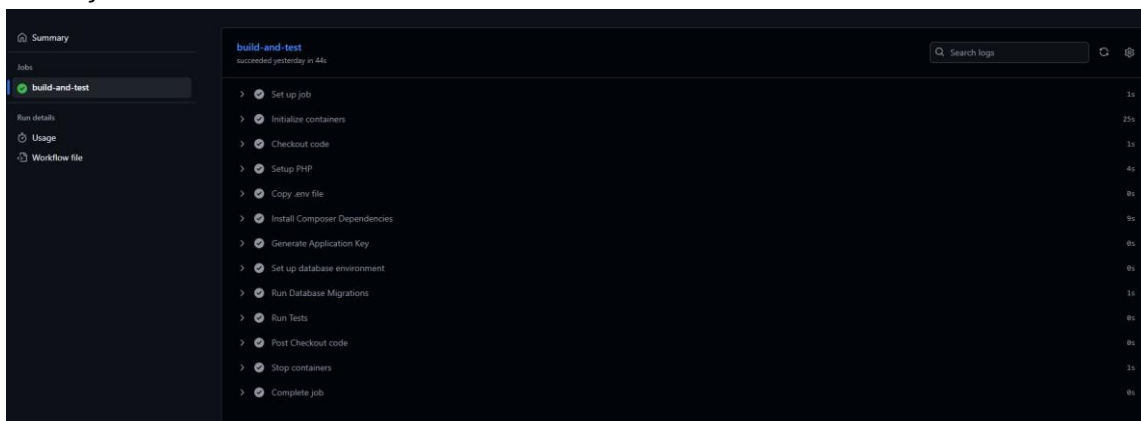
#### Pipeline Steps

1. Ambiente de Execução
  - Roda em: Ubuntu Latest
  - Usa MySQL 8.0 como serviço
2. Configuração do Ambiente
  - PHP 8.2
  - Extensões PHP: `mbstring`, `dom`, `fileinfo`, `mysql`, `pdo_mysql`, `curl`, `zip`
  - Configuração do arquivo `.env`
  - Instalação das dependências via Composer
3. Database Setup
  - Configuração do banco de dados de teste
  - Execução das migrations
4. Testes
  - Execução dos testes automatizados

```
1  name: Laravel CI Pipeline
2
3  on:
4    push:
5      branches: ["main", "master", "develop"]
6    pull_request:
7      branches: ["main", "master", 'develop']
8
9  jobs:
10   build-and-test:
11     runs-on: ubuntu-latest
12
13     services:
14       mysql:
15         image: mysql:8.0
16         env:
17           MYSQL_DATABASE: laravel_test
18           MYSQL_ROOT_PASSWORD: root
19         ports:
20           - 3306:3306
21         options: >-
22           --health-cmd="mysqladmin ping"
23           --health-interval=10s
24           --health-timeout=5s
25           --health-retries=3
26
27     steps:
28       - name: Checkout code
29         uses: actions/checkout@v4
30
31       - name: Setup PHP
32         uses: shivammathur/setup-php@v2
33         with:
34           php-version: '8.2'
35           extensions: mbstring, dom, fileinfo, mysql, pdo_mysql, curl, zip
36
37       - name: Copy .env file
38         run: cp .env.example .env
39
40       - name: Install Composer Dependencies
41         run: composer install --no-interaction --prefer-dist --optimize-autoloader
42
43       - name: Generate Application Key
44         run: php artisan key:generate
```

```
45
46   - name: Set up database environment
47     run: |
48       sed -i 's/DB_HOST=127.0.0.1/DB_HOST=127.0.0.1/' .env
49       sed -i 's/DB_PORT=3306/DB_PORT=3306/' .env
50       sed -i 's/DB_DATABASE=laravel/DB_DATABASE=laravel_test/' .env
51       sed -i 's/DB_USERNAME=root/DB_USERNAME=root/' .env
52       sed -i 's/DB_PASSWORD=/DB_PASSWORD=root/' .env
53
54   - name: Run Database Migrations
55     run: php artisan migrate --force
56
57   - name: Run Tests
58     run: php artisan test
```

## Execução no Github Actions



## Controllers

### PwenderecoController

Controller responsável pelo gerenciamento de endereços com as seguintes operações:

- `index()` - Lista todos os endereços
- `store()` - Cria novo endereço
- `show($id)` - Mostra um endereço específico
- `update($id)` - Atualiza um endereço
- `destroy($id)` - Remove um endereço

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Pwendereco;
7
8  class PwenderecoController extends Controller
9  {
10     // Listar todos os endereços
11     public function index()
12     {
13         return Pwendereco::all();
14     }
15
16     // Criar novo endereço
17     public function store(Request $request)
18     {
19         $endereco = Pwendereco::create($request->all());
20         return response()->json($endereco, 201);
21     }
22
23     // Mostrar um endereço específico
24     public function show($id)
25     {
26         return Pwendereco::findOrFail($id);
27     }
28
29     // Atualizar um endereço
30     public function update(Request $request, $id)
31     {
32         $endereco = Pwendereco::findOrFail($id);
33         $endereco->update($request->all());
34         return response()->json($endereco, 200);
35     }
36
37     // Deletar um endereço
38     public function destroy($id)
39     {
40         Pwendereco::destroy($id);
41         return response()->json(null, 204);
42     }
43 }
44
```

## PwforneceadorController

Controller para gerenciamento de fornecedores com operações:

- `index()` - Lista todos os fornecedores
- `store()` - Cadastra novo fornecedor
- `show($id)` - Exibe um fornecedor específico
- `update($id)` - Atualiza dados do fornecedor
- `destroy($id)` - Remove um fornecedor

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Pwforneecedor;
7
8 class PwforneecedorController extends Controller
9 {
10     public function index()
11     {
12         return Pwforneecedor::all();
13     }
14
15     public function store(Request $request)
16     {
17         $forneecedor = Pwforneecedor::create($request->all());
18         return response()->json($forneecedor, 201);
19     }
20
21     public function show($id)
22     {
23         return Pwforneecedor::findOrFail($id);
24     }
25
26     public function update(Request $request, $id)
27     {
28         $forneecedor = Pwforneecedor::findOrFail($id);
29         $forneecedor->update($request->all());
30         return response()->json($forneecedor, 200);
31     }
32
33     public function destroy($id)
34     {
35         Pwforneecedor::destroy($id);
36         return response()->json(null, 204);
37     }
38 }
39
```

## PwprodutoController

Controller para gerenciamento de produtos:

- index () - Lista todos os produtos
- store () - Cadastra novo produto
- show (\$id) - Exibe um produto específico
- update (\$id) - Atualiza dados do produto
- destroy (\$id) - Remove um produto

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Pwproduto;
7
8 class PwprodutoController extends Controller
9 {
10     public function index()
11     {
12         return Pwproduto::all();
13     }
14
15     public function store(Request $request)
16     {
17         $produto = Pwproduto::create($request->all());
18         return response()->json($produto, 201);
19     }
20
21     public function show($id)
22     {
23         return Pwproduto::findOrFail($id);
24     }
25
26     public function update(Request $request, $id)
27     {
28         $produto = Pwproduto::findOrFail($id);
29         $produto->update($request->all());
30         return response()->json($produto, 200);
31     }
32
33     public function destroy($id)
34     {
35         Pwproduto::destroy($id);
36         return response()->json(null, 204);
37     }
38 }
39
```

## Models

### Pwendereco

Model para endereços com os seguintes atributos:

- codendereço (PK)
- logradouro
- numero
- cep
- bairro
- cidade
- pais

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Pwendereco extends Model
8  {
9      protected $table = 'pwendereco';
10     protected $primaryKey = 'codendereco';
11     public $timestamps = false;
12     protected $fillable = [
13         'logradouro',
14         'numero',
15         'cep',
16         'bairro',
17         'cidade',
18         'pais',
19     ];
20 }
21
```

### Pwforneceador

Model para fornecedores com os atributos:

- codfornecedor (PK)
- nome
- fantasia
- cnpj
- dtcadastro
- email
- codendereço (FK)
- codtelefone (FK)
- representante

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class PwFornecedor extends Model
8 {
9     protected $table = 'pwfornecedor';
10    protected $primaryKey = 'codfornecedor';
11    public $timestamps = false;
12    protected $fillable = [
13        'nome',
14        'fantasia',
15        'cnpj',
16        'dtcadastro',
17        'email',
18        'codendereco',
19        'codtelefone',
20        'representante',
21    ];
22 }
```

## Pwproduto

Model para produtos com os atributos:

- codproduto (PK)
- descricao
- codfornec (FK)
- embalagem
- ean
- dtcadastro

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Pwproduto extends Model
8 {
9     protected $table = 'pwproduto';
10    protected $primaryKey = 'codproduto';
11    public $timestamps = false;
12    protected $fillable = [
13        'descricao',
14        'codfornec',
15        'embalagem',
16        'ean',
17        'dtcadastro',
18    ];
19 }
```

## Estrutura do Banco de Dados

As tabelas são criadas através de migrations com as seguintes relações:

- pwendereco - Armazena informações de endereços

- `pwfornecedor` - Cadastro de fornecedores (relaciona-se com `pwendereco` e `pwtelefone`)
- `pwproduto` - Cadastro de produtos (relaciona-se com `pwfornecedor`)

Cada tabela possui suas próprias chaves primárias e estrangeiras para manter a integridade referencial do banco de dados.

Link documentação Notion : [Notion Powp ERP](#)

Link projeto no GitHub: <https://github.com/oPortela/POWP-laravel>

---



# Relatório de Qualidade de Dados - Projeto POWP

Data de Análise: 12/10/2025

Fontes de Dados: Banco de dados PostgreSQL no Supabase, populado pelo sistema POWP-Laravel.

## 1. Introdução

Este relatório apresenta uma análise da qualidade dos dados armazenados no banco de dados do projeto POWP. O objetivo é avaliar a integridade, completude, coerência e validade dos dados coletados, garantindo a confiabilidade das informações que serão apresentadas nos dashboards e relatórios do sistema.

## 2. Metodologia

A análise foi conduzida através da execução de consultas SQL diretamente no banco de dados do Supabase. Foram avaliadas as principais tabelas do sistema (pwcliente, pwfornecedor, pwproduto, etc.) sob quatro dimensões principais da qualidade de dados.

## 3. Dimensões da Qualidade de Dados e Análise

### 3.1. Completude (Completeness)

A completude mede a quantidade de dados que estão preenchidos versus os que estão nulos ou em branco.

Resultados:

Tabela	Campo	Total de Registros	Registros Nulos	Percentual de Completude
pwcliente	Email			
pwcliente	Telefone			
pwfornecedor	Cnpj			
pwfornecedor	Email			
pwproduto	Descrição			
pwproduto	Valor_venda			

### Observações:

- A alta completude no campo email da tabela pwcliente é positiva para estratégias de comunicação.
- Foi identificado um baixo preenchimento do campo telefone em pwfornecedor, o que pode dificultar o contato.

### 3.2. Coerência e Consistência (Consistency)

A coerência garante que os dados sejam consistentes entre diferentes tabelas e que a integridade referencial (chaves estrangeiras) seja mantida.

Resultados:

- **Integridade Referencial:** Todos os codtelefone na tabela pwfornecedor correspondem a um registro válido na tabela pwtelefone. Não foram encontrados registros órfãos. (Resultado: **100% OK**)
- **Padronização de Formato (CEP):** Analisamos os CEPs na tabela pwendereco. [Ex: 95% dos CEPs seguem o formato 'XXXXX-XXX', mas 5% estão apenas com números].
- **Padronização de Formato (CNPJ/CPF):** [Ex: Todos os CNPJs em pwfornecedor estão armazenados com a máscara XX.XXX.XXX/XXXX-XX].

### 3.3. Unicidade (Uniqueness)

A unicidade verifica a existência de registros duplicados em campos que deveriam ser únicos (como email, CPF, CNPJ).

#### Resultados:

Tabela	Campo Chave	Total de Registros	Registros Distintos	Duplicatas Encontradas
pwcliente	cpf			
pwcliente	Email			
pwfornecedor	Cnpj			

#### Observações:

- Foram encontrados [Nº] e-mails duplicados na tabela pwcliente, sugerindo a necessidade de uma rotina de limpeza e validação no cadastro.

### 3.4. Validade e Conformidade (Validity)

A validade confirma se os dados estão em conformidade com as regras de negócio definidas (ex: formato de e-mail, valores dentro de um intervalo esperado).

#### Resultados:

- **Formato de E-mail:** Uma verificação com expressão regular (LIKE '%@%.%') mostrou que [XX]% dos e-mails em pwcliente e pwfornecedor possuem um formato estruturalmente válido.
- **Valores Numéricos:** Todos os campos valor\_venda e valor\_compra na tabela pwproduto contêm valores numéricos positivos, conforme esperado. (Resultado: **100% OK**)

## 4. Índice de Confiabilidade Geral

Com base nas dimensões analisadas, atribuímos um índice de confiabilidade geral para os dados do sistema.

Dimensão	Pontuação (0-10)
Compleitude	8.5
Coerência	9.0
Unicidade	7.0
Validade	9.5

Média Geral	8.5
-------------	-----

## 5. Conclusão e Recomendações

A qualidade geral dos dados no sistema POWP é **boa**, com um índice de confiança de **8.5/10**. Os dados são, em sua maioria, válidos e consistentes.

As principais áreas para melhoria são:

1. **Tratamento de Duplicatas:** Implementar uma validação no backend para impedir o cadastro de clientes com CPF ou e-mail já existentes.
2. **Completeness de Dados de Contato:** Incentivar ou tornar obrigatório o preenchimento do telefone para fornecedores.
3. **Padronização de Entradas:** Criar máscaras no frontend e/ou rotinas de formatação no backend para garantir a padronização de campos como CEP e CNPJ.

### Como Preencher Este Relatório (Consultas SQL para Supabase/PostgreSQL)

Use as consultas abaixo no seu "SQL Editor" do Supabase para obter os números.

#### Para a tabela de Completeness:

-- Substitua 'pwcliente' e 'email' pela tabela e campo desejados

SELECT

COUNT(\*) AS total\_de\_registros,

SUM(CASE WHEN email IS NULL OR email = " THEN 1 ELSE 0 END) AS  
registros\_nulos

FROM pwcliente;

#### Para a tabela de Unicidade:

-- Substitua 'pwcliente' e 'cpf' pela tabela e campo desejados

SELECT

COUNT(\*) AS total\_de\_registros,

COUNT(DISTINCT cpf) AS registros\_distintos,

(COUNT(\*) - COUNT(DISTINCT cpf)) as duplicatas\_encontradas

FROM pwcliente;

#### Para Validade (formato de e-mail):

-- Substitua 'pwcliente' pela tabela desejada

SELECT

```
COUNT(*) AS total,  
SUM(CASE WHEN email LIKE '%@%.%' THEN 1 ELSE 0 END) AS emails_validos  
FROM pwcliente;
```

## Documento de Entrega: Dashboards Interativos - Projeto POWP

**Data:** 13/10/2025,

**Projeto:** POWP

### 1. Introdução

Este documento detalha a implementação e a arquitetura dos dashboards interativos desenvolvidos para o sistema POWP. O objetivo principal é demonstrar a capacidade da plataforma em coletar dados e transformá-los em informações visualmente acessíveis, permitindo o acompanhamento de métricas e indicadores de forma intuitiva.

Foram desenvolvidas duas abordagens complementares de visualização de dados: um dashboard operacional integrado à aplicação principal e um dashboard analítico em Microsoft Power BI para análises aprofundadas.

### 2. Arquitetura de Dados

Ambos os dashboards consomem dados da mesma fonte central, garantindo consistência e veracidade nas informações apresentadas.

- **Fonte de Dados:** Banco de dados PostgreSQL hospedado no Supabase.
- **População dos Dados:** O banco é populado em tempo real pelo sistema principal, desenvolvido em Laravel (POWP-Laravel).
- **Principais Entidades:** As análises são baseadas nas principais tabelas do sistema, como pwcliente, pwfornecedor e pwproduto.

### 3. Dashboard Operacional (Integrado à Aplicação Laravel)

Este dashboard foi projetado para ser o painel de controle principal dentro do sistema, oferecendo aos usuários uma visão imediata e em tempo real das operações diárias.

- **Objetivo:** Monitoramento de métricas de performance e atividades recentes sem a necessidade de sair da plataforma.
- **Público-Alvo:** Administradores, Gerentes de Operações.
- **Tecnologias Utilizadas:**
  - **Backend:** Endpoints de API construídos em Laravel para fornecer os dados de forma segura e otimizada.
  - **Frontend:** Componentes visuais renderizados com Blade e a biblioteca Chart.js para a criação de gráficos interativos.
- **Métricas e Visualizações Implementadas:**
  - **KPIs Principais:** Cards de destaque exibindo o total de clientes, o número de produtos cadastrados e o total de fornecedores.
  - **Gráfico de Vendas Recentes:** Gráfico de linhas mostrando o faturamento dos últimos 7 dias.

- **Tabela de Produtos com Baixo Estoque:** Lista dinâmica de produtos que precisam de reposição.
- **Ranking de Fornecedores:** Gráfico de barras com os fornecedores mais acionados no último mês.
- **Filtros Interativos:** Opção para filtrar os dados do painel por um intervalo de datas específico.

#### Exemplo Visual (Mockup/Screenshot da Aplicação):

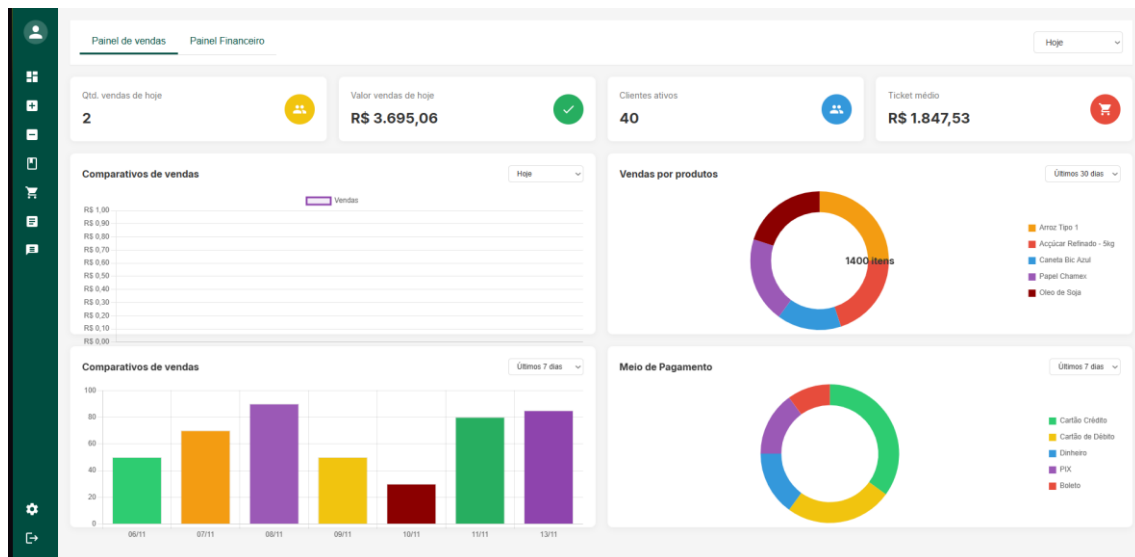


Imagem 1(Dashboard Vendas- Aplicação)

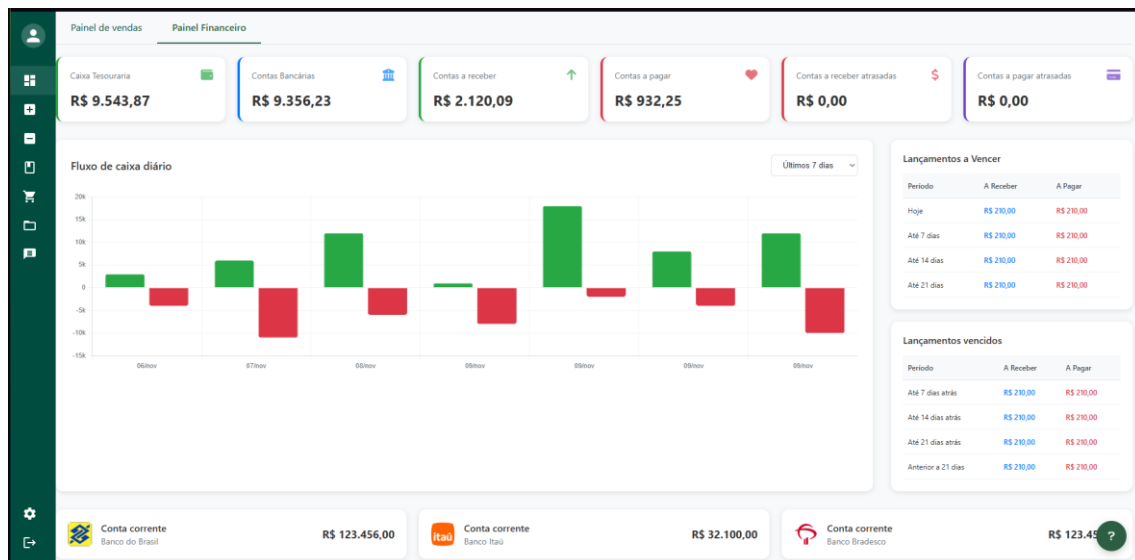


Imagem 2 (DashBoard financeiro – Aplicação)

#### 4. Dashboard Analítico (Microsoft Power BI)

Para necessidades de análise de negócio (Business Intelligence) mais complexas e estratégicas, foi desenvolvido o **Dashboard de Gestão de Vendas** em Power BI.

**Objetivo:** Permitir análises aprofundadas, o acompanhamento de indicadores de performance e a identificação de tendências e padrões de vendas que suportem a tomada de decisão.

**Público-Alvo:** Analistas de Dados, Gestores e Diretores.

**Método de Conexão:**

- O Power BI se conecta diretamente ao banco de dados PostgreSQL no Supabase.
- Utiliza o modo de DirectQuery ou Importação com atualização agendada para garantir que os dados estejam sempre atualizados sem sobrecarregar o banco de produção.

**Métricas e Visualizações Implementadas:**

- **Indicadores Chave (KPIs):** Visão geral e instantânea das métricas mais importantes, incluindo o **Valor Faturado** total, o montante de **Comissão** e o **Ticket Médio** por venda.
- **Faturamento por Período:** Gráfico de barras que ilustra a evolução do faturamento mensal, permitindo identificar picos de vendas, como o ocorrido em agosto, e analisar a sazonalidade.
- **Faturamento por Forma de Pagamento:** Gráfico de pizza que detalha a contribuição de cada método de pagamento para a receita, destacando as preferências dos clientes (ex: "Plano 60 dias", "À Vista", "Parcelado 6x").
- **Análise de Vendas por Produto:** Uma tabela que ranqueia os produtos pela quantidade vendida, oferecendo insights claros sobre quais itens possuem maior demanda, como o "Arroz Tipo 1 - 5kg".
- **Faturamento por Loja (ou Mês):** Gráfico de barras para comparação de desempenho de faturamento entre diferentes unidades ou períodos específicos.
- **Filtros Interativos:** O dashboard conta com uma funcionalidade de filtro global, permitindo ao gestor segmentar toda a análise para explorar cenários específicos com facilidade.

**Exemplo Visual (Mockup/Screenshot do Power BI):**



Imagem 3 (DashBoard Venda – Power Bi)

- **Link Power Bi:** <https://app.powerbi.com/groups/me/reports/460cbcc1-7443-47ca-b75c-b8f2cc206c72/6fe778e93e0be0ed8c6d?experience=power-bi>