SIGEP: Sistema Web Multiempresa para Gestão de Pedidos de Representantes Comerciais

Rômulo Jordão Bogoni[1], **Rafael Henrique Dalegrave Zottesso**[1], **Evanise Araujo Caldas Ruiz**[1]

[1]Curso de Bacharelado em Engenharia de Software
Instituto Federal do Paraná (IFPR) – Campus Paranavaí
Paranavaí – PR – Brasil

20230006183@estudantes.ifpr.edu.br

{rafael.zottesso,evanise.ruiz}@ifpr.edu.br

Abstract. *This paper presents SIGEP, a responsive multi-company web application developed to replace the informal order recording (photographed handwritten forms sent through messaging apps) observed in a wholesale company that works with sales representatives. Following Design Science Research, the system was built with Python/Django and PostgreSQL, deployed in the cloud and verified against its specification through manual functional tests performed by the developer. Of the 15 functional requirements, 10 were met, 1 partially and 4 remain pending; of the 9 business rules, 5 were met, 1 partially and 3 remain pending. Performance goals were not measured and no user validation was carried out. The objective was partially achieved: order recording with multi-company isolation and transactional stock control is implemented and verified, while the order lifecycle and e-mail notifications remain pending.*

Resumo. *Este artigo apresenta o SIGEP, aplicação web responsiva e multiempresa desenvolvida para substituir o registro informal de pedidos (formulários manuscritos fotografados e enviados por aplicativos de mensagens) observado em uma empresa atacadista que opera com representantes comerciais. Seguindo a Design Science Research, o sistema foi construído com Python/Django e PostgreSQL, implantado em nuvem e verificado em relação à especificação por testes funcionais manuais executados pelo desenvolvedor. Dos 15 requisitos funcionais, 10 foram atendidos, 1 parcialmente e 4 permanecem pendentes; das 9 regras de negócio, 5 foram atendidas, 1 parcialmente e 3 permanecem pendentes. As metas de desempenho não foram aferidas e não houve validação com usuários. O objetivo foi parcialmente alcançado: o registro do pedido, com isolamento multiempresa e controle transacional de estoque, está implementado e verificado, enquanto o ciclo de vida do pedido e as notificações por e-mail permanecem pendentes.*

Palavras-chave: *Gestão de pedidos; Representantes comerciais; Sistema web.*

1. Introdução

A transformação digital tem levado as organizações a substituir rotinas manuais por soluções informatizadas [Verhoef et al. 2021], e a tecnologia passou a integrar a própria

estratégia dos negócios [Bharadwaj et al. 2013]. Entre os processos apoiados por sistemas, a gestão de pedidos de vendas destaca-se por conectar vendas, estoque e atendimento ao cliente [Gali 2025].

Este trabalho foi motivado pela rotina de uma empresa atacadista que opera com representantes comerciais e na qual o primeiro autor atua como auxiliar administrativo e financeiro. Nela, os representantes preenchiam os pedidos à mão e enviavam fotos dos formulários a um grupo de aplicativo de mensagens, o estoque era confirmado por texto e os valores eram conferidos e, quando necessário, recalculados pelo setor administrativo (Figura 1). Nos registros desse grupo há, por exemplo, pedidos reenviados depois de já terem sido encaminhados e marcações manuais para indicar quais pedidos já haviam sido impressos. Os problemas efetivamente observados foram a falta de padronização no registro, a conferência manual de cálculos e de estoque e a dificuldade de saber em que etapa cada pedido se encontrava; a frequência desses episódios não foi contabilizada.

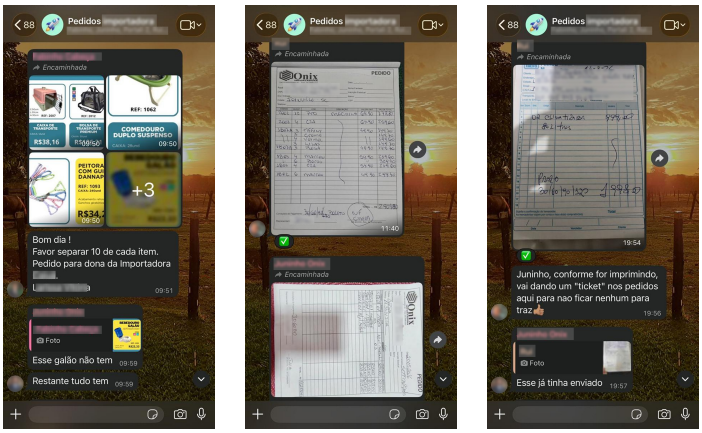


Figura 1. Registro informal de pedidos em aplicativo de mensagens (dados pessoais e comerciais ocultados). Fonte: empresa parceira, com autorização de uso.

Em organizações semelhantes, a literatura indica que a fragmentação das informações em ferramentas desconectadas tende a gerar retrabalho e inconsistências [Davenport 1998, Gali 2025]; esses efeitos são, portanto, esperados, mas não foram medidos neste trabalho. Outras três empresas do segmento manifestaram interesse informal no sistema, o que sugere demanda, mas não constitui evidência de eficácia nem de aplicabilidade a outros contextos.

Esse problema costuma ser tratado por ERPs (*Enterprise Resource Planning*) e ferramentas comerciais oferecidos por assinatura, com código e regras de negócio controlados pelo fornecedor [Winsite 2026, Omie 2026, Mercos 2026], ou por sistemas desenvolvidos sob medida para uma única empresa [Amorim 2023]. Diante disso, este trabalho teve como objetivo projetar, desenvolver e verificar funcionalmente o SIGEP, sistema web responsivo e multiempresa para a gestão de pedidos entre representantes e empresas, com a finalidade de apoiar a organização, a rastreabilidade e a confiabilidade do processo. Como contribuições, apresentam-se: (i) a implementação de um escopo multiempresa em esquema compartilhado, aplicado de forma transversal em um sistema Django; (ii) a criação transacional de pedidos, com validação integral dos itens e bloqueio de linha no estoque; e (iii) uma verificação requisito a requisito, que explicita o que foi atendido,

parcialmente atendido, pendente ou não aferido.

2. Fundamentação Teórica

Gestão de pedidos. Compreende as atividades que vão da solicitação do cliente até a entrega do produto, incluindo registro, validação, controle de disponibilidade e acompanhamento. Os sistemas de gestão de pedidos (*Order Management Systems* – OMS) integram vendas, estoque e logística e oferecem funções como cálculo automático de valores, verificação de estoque e controle dos estados do pedido [Gali 2025].

Arquitetura MTV. O padrão MVC separa dados, controle e apresentação [Krasner and Pope 1988]; o Django adota uma variação denominada MTV (*Model-Template-View*), também referida como MVT, na qual o *Model* é mapeado para o banco pelo ORM, a *View* processa as requisições e aplica as regras de controle e o *Template* gera o HTML. O *framework* oferece ainda autenticação, autorização e proteção contra CSRF [Django Software Foundation 2026].

Multilocação e consistência. Multilocação (*multi-tenancy*) é a capacidade de uma instância de software atender a vários clientes, os locatários, com dados logicamente isolados [Bezemer and Zaidman 2010]; no esquema compartilhado, cada registro é associado ao seu locatário nas mesmas tabelas, o que exige filtrar todas as consultas [Chong et al. 2006]. A consistência é garantida por transações ACID e por bloqueios de linha [Elmasri and Navathe 2016]; no PostgreSQL, `SELECT ... FOR UPDATE` bloqueia as linhas lidas até o fim da transação [The PostgreSQL Global Development Group 2026].

3. Trabalhos Relacionados

Segundo as informações divulgadas pelos fornecedores, o Winsite Gestão [Winsite 2026] é um ERP online para pequenas e médias empresas que integra vendas, financeiro, estoque e notas fiscais; o Omie ERP [Omie 2026] abrange CRM, vendas, NF-e, finanças, estoque e produção, contratado por planos de assinatura; e o Mercos [Mercos 2026], mais próximo deste trabalho, é um gerenciador de pedidos para representantes, com operação *offline*, comercializado por assinatura e sem acesso ao código-fonte. Como não foram feitos levantamento de preços nem avaliação de uso dessas ferramentas, a comparação limita-se ao escopo e ao modelo de comercialização divulgados. No meio acadêmico, um sistema de controle de pedidos para uma empresa varejista foi desenvolvido em Django, Bootstrap e SQLite, implantado em rede local e avaliado por seis colaboradores, com percepção positiva de usabilidade [Amorim 2023]. A Tabela 1 sintetiza a comparação.

As soluções comerciais envolvem assinatura, escopo mais amplo que o necessário ou ausência de controle sobre o código, e o trabalho acadêmico restringe-se a um único locatário e à implantação local. O SIGEP propõe-se a combinar o foco no fluxo de pedidos com arquitetura multiempresa em nuvem, controle transacional de estoque e regras de negócio adaptáveis.

4. Metodologia

A pesquisa é aplicada, qualitativa e exploratória e descritiva [Gil 2017], e segue a *Design Science Research* (DSR), na qual se constrói e avalia um artefato para resolver um

Tabela 1. Comparação entre os trabalhos relacionados e o presente trabalho.

Trabalho	Objetivo e abordagem	Tecnologia e contexto	Resultados e limitações
Winsite Gestão	Gestão empresarial (PMEs); ERP comercial	SaaS proprietário; vendas, estoque, NF-e	Escopo além do registro de pedidos
Omie ERP	Gestão empresarial; ERP modular	SaaS proprietário; CRM, NF-e, produção	Contratação por planos de assinatura
Mercos	Pedidos de representantes; ferramenta especializada	SaaS proprietário; operação *offline*	Escopo próximo; assinatura; código fechado
[Amorim 2023]	Pedidos de uma empresa varejista; sob medida	Django, SQLite; servidor local	Validado com 6 usuários; empresa única
Este trabalho	Pedidos entre representantes e empresas; DSR	Django, PostgreSQL; nuvem; multiempresa	Registro verificado; ciclo de vida pendente; sem validação com usuários

problema relevante [Hevner et al. 2004]. Seguindo essas diretrizes, a **relevância do problema** foi estabelecida pela observação da rotina descrita na Seção 1; o **artefato** — o SIGEP, com seus modelos, código-fonte e instância implantada — foi construído sobre padrões consolidados (MTV, transações ACID, multilocação), o que corresponde ao **rigor**; e a **avaliação** foi feita por verificação funcional requisito a requisito. A validação com usuários, também prevista pela DSR, não foi realizada. O desenvolvimento seguiu oito etapas: definição do problema e levantamento de requisitos (identificação do problema); modelagem UML, arquitetura, implementação, integração com o banco e implantação em nuvem (construção); e testes e verificação (avaliação).

4.1. Preparação

O SIGEP opera em regime multiempresa com três perfis: o administrador *master*, que cadastra empresas e usuários; o gerente, que administra produtos, clientes e representantes de sua empresa e aprova pedidos; e o representante, que registra os pedidos. Ficaram fora do escopo a integração com ERPs, a emissão de NF-e, os pagamentos e a logística.

Os requisitos foram levantados por observação direta da rotina, complementada por conversas informais. Entre março de 2025 e abril de 2026, o primeiro autor, que acompanha diariamente o recebimento, a conferência e a impressão dos pedidos, observou as etapas descritas na Seção 1. Embora inspirado na etnografia [Sommerville 2018], o procedimento não constituiu um estudo etnográfico formal, pois não houve protocolo de observação, diário de campo sistemático nem codificação dos dados: trata-se de observação da rotina profissional por um participante do processo. As situações foram **registradas** em anotações do autor, com data e descrição da dificuldade, esclarecidas em conversas com a gerência e com representantes e complementadas pelas mensagens do grupo; depois, foram **organizadas** por perfil de usuário e por etapa do fluxo (registro, conferência de estoque, cálculo e acompanhamento); por fim, cada dificuldade recorrente foi **transformada** em requisito funcional (RF), não funcional (RNF) ou regra de negócio (RN). Por exemplo, a confirmação de estoque por mensagem originou a RN04 e o RF09; a correção manual de valores, o RF10 e as RN06 e RN07; e a dificuldade de saber se um pedido já havia sido atendido, os estados do pedido (RN05, RN08 e RN09) e a aprovação pelo gerente (RF14 e RN03). O resultado foram 15 RF, 9 RNF e 9 RN.

Como o observador participa do processo e também desenvolveu o sistema, há

risco de viés: os problemas podem refletir sobretudo a perspectiva do setor administrativo, e não a dos representantes. Para reduzi-lo, as anotações foram confrontadas com as mensagens e com as conversas; ainda assim, a especificação não foi validada formalmente com os usuários.

Na especificação, o ciclo de vida do pedido usa três termos, adotados de forma padronizada neste artigo: **criação**, o registro inicial pelo representante, no estado "Em Digitação"; **confirmação**, o envio pelo representante, que leva o pedido a "Pendente" e deve acionar a baixa de estoque (RN08); e **aprovação** ou rejeição, a decisão do gerente sobre o pedido pendente (RN03). O cancelamento de um pedido aprovado deve estornar o estoque (RN09).

Cada RF, RNF e RN foi classificado como **Atendido**, quando o comportamento observado corresponde integralmente à especificação; **Parcialmente atendido**, quando apenas parte foi implementada; **Pendente**, quando não há implementação; ou **Não aferido**, quando o requisito depende de medição que não foi realizada. Essas categorias expressam o resultado da **verificação**, que avalia se o software corresponde à especificação, e não da **validação**, que avalia se ele atende às necessidades dos usuários [Sommerville 2018]. Assim, distinguem-se funcionalidades **implementadas** (há código), **verificadas** (conferidas em teste contra a especificação) e **validadas** (confirmadas por usuários reais); neste trabalho, nenhuma foi validada.

4.2. Execução

Foram utilizados Python 3.12, Django 5.2.11, django-braces 1.17, django-crispy-forms 2.4 com crispy-bootstrap5, django-filter 25.1, Bootstrap 5 e JavaScript; PostgreSQL na plataforma Neon, via psycopg2 2.9.10; e implantação no Google App Engine com Gunicorn e escalonamento de zero a duas instâncias. A modelagem resultou no diagrama de casos de uso (Figura 2) e no diagrama de classes (Figura 3), que representam o escopo especificado; a situação de implementação de cada funcionalidade é apresentada na Seção 5.

No *back-end*, os modelos herdam de uma classe abstrata que registra datas de criação e atualização e o usuário responsável; o CRUD usa *views* baseadas em classes; e o controle de acesso tem três camadas: autenticação obrigatória, autorização por grupo (gerente) e um *mixin* de escopo multiempresa, que filtra cada consulta pela empresa ativa na sessão e associa a ela os novos registros. A criação de pedidos é uma transação atômica: cada item é validado antes de qualquer gravação (produto da empresa ativa, quantidade positiva, estoque suficiente e desconto não superior ao subtotal), a baixa de estoque usa bloqueio de linha (`select_for_update`) e, se algum item for inválido, toda a operação é revertida. No *front-end*, um *script* monta o pedido dinamicamente e replica o cálculo de valores como retorno imediato, sem dispensar a validação no *back-end*.

Os testes foram funcionais, manuais e exploratórios, executados pelo próprio desenvolvedor em uma instância com dados de teste, contendo duas empresas (A e B) com seus clientes, produtos e pedidos, um gerente e dois representantes da empresa A. Cada caso de teste registrou o requisito, o cenário, o resultado esperado, o resultado observado e a situação final. Não foram executados testes de concorrência nem medições de tempo de resposta.

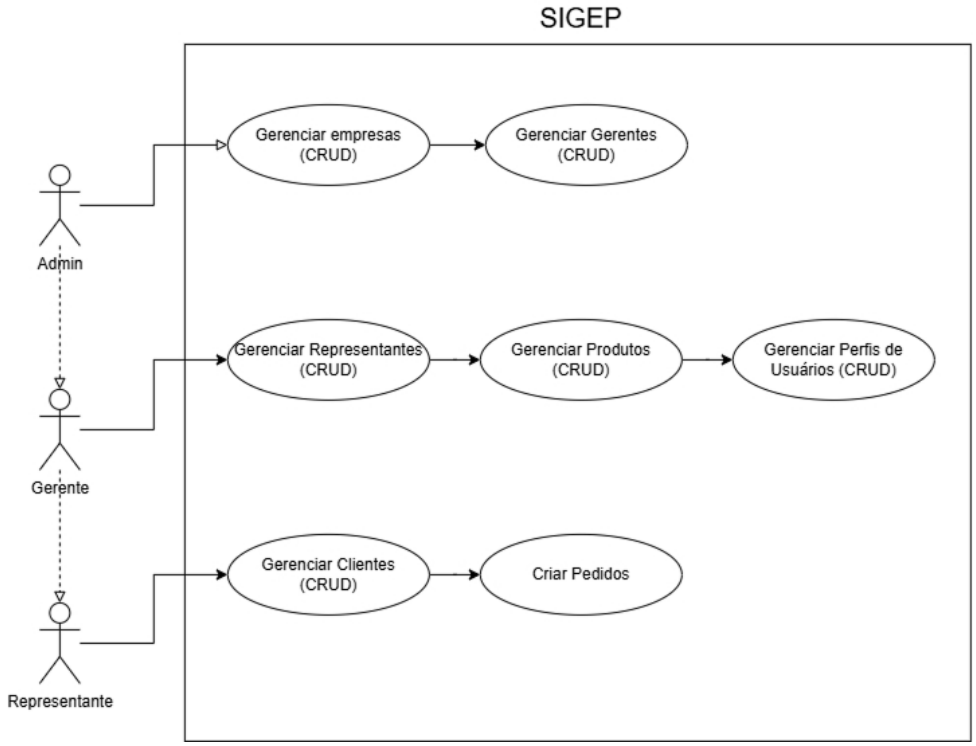


Figura 2. Diagrama de casos de uso do SIGEP.

4.3. Análise dos Dados

Os resultados foram consolidados em uma matriz de rastreabilidade [Gotel and Finkelstein 1994], que relaciona cada requisito aos casos de teste e às evidências no código; quando o teste e a inspeção do código divergiam, prevaleceu a situação mais restritiva. Considerou-se o objetivo alcançado na medida em que o núcleo do registro de pedidos (criação, cálculo e controle de estoque) estivesse implementado e verificado e as pendências pudessem, segundo a análise da arquitetura, ser implementadas sem alterar a estrutura do sistema.

5. Resultados

A versão implantada em nuvem oferece autenticação, controle de acesso por perfil, isolamento multiempresa, cadastros, criação transacional de pedidos com controle de estoque e um painel gerencial com indicadores (Figura 4). A Tabela 2 apresenta a situação de cada requisito e regra de negócio após a verificação.

Dos 15 RF, 10 foram atendidos, 1 parcialmente e 4 permanecem pendentes. O RF11 é parcial porque a edição altera apenas os dados gerais do pedido, sem alterar os itens nem restringir a edição ao estado “Em Digitação”. As pendências referem-se ao ciclo de vida do pedido (RF12 e RF14) e ao envio de e-mails (RF05 e RF07). Das 9 RN, 5 foram atendidas, 1 parcialmente e 3 permanecem pendentes; dos 9 RNF, 4 foram atendidos, 3 parcialmente e 2 não foram aferidos. A Tabela 3 apresenta os casos de teste referentes ao acesso por empresa, ao acesso por representante e ao controle de estoque.

Quanto ao ciclo de vida, a especificação prevê que a criação não movimente o estoque, que a confirmação faça a baixa (RN08), que o gerente aprove ou rejeite o pedido

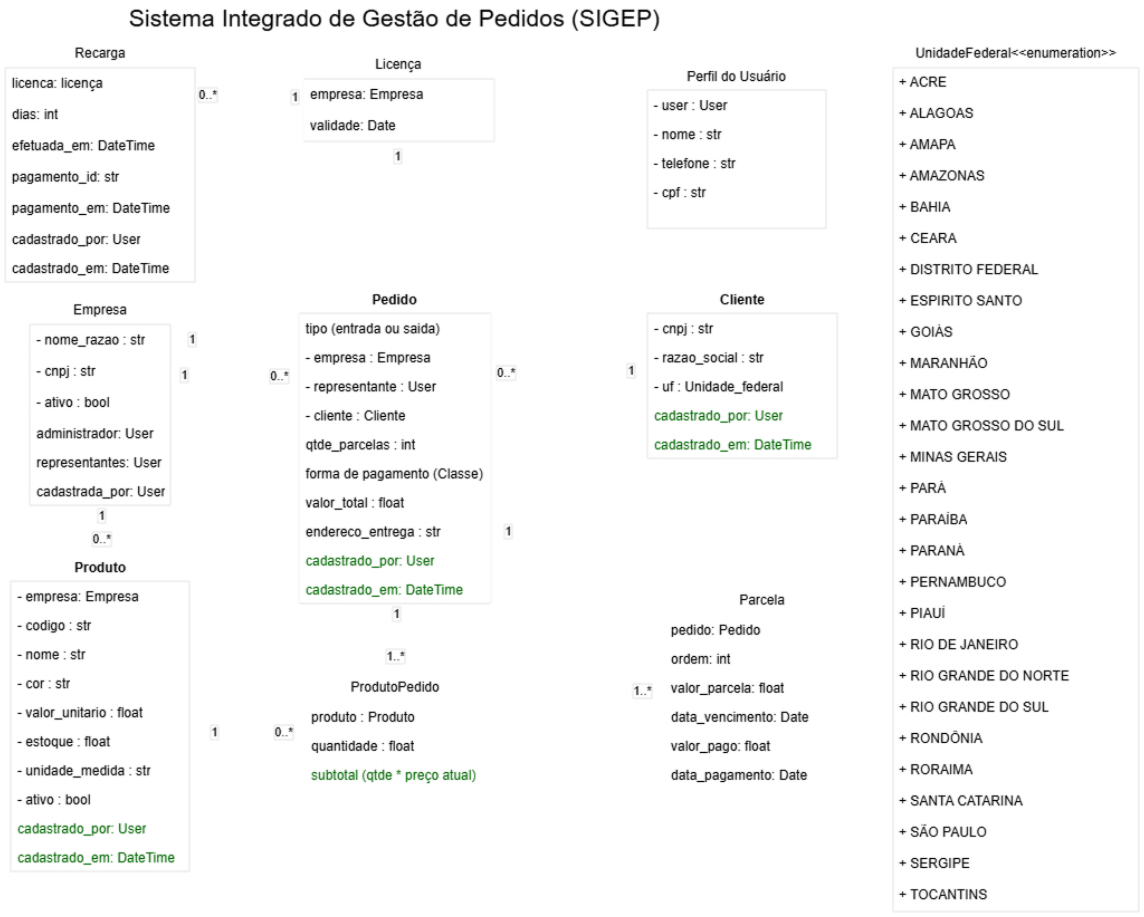


Figura 3. Diagrama de classes do SIGEP.

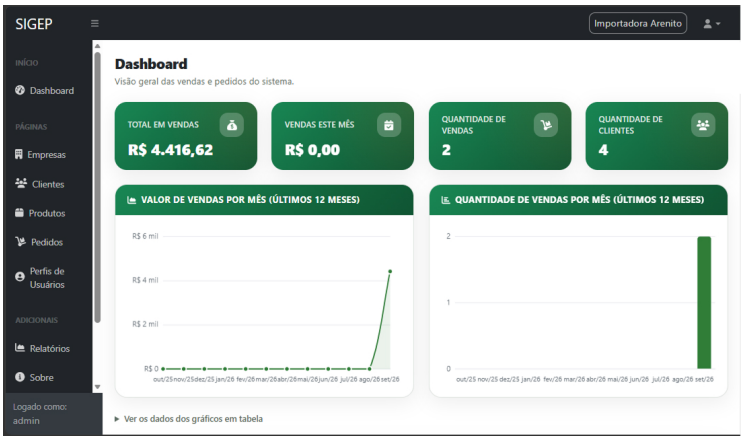


Figura 4. Painel gerencial do SIGEP com dados de teste.

Tabela 2. Situação dos requisitos e regras de negócio (A: atendido; PA: parcialmente atendido; P: pendente; NA: não aferido).

ID	Requisito funcional	Sit.	ID	Regra de negócio / requisito não funcional	Sit.
RF01	Autenticação por login e senha	A	RN01	Isolamento por empresa ativa	A
RF02	Cadastro de empresas (administrador)	A	RN02	Visibilidade de pedidos por perfil	A
RF03	Cadastro de gerentes e representantes	A	RN03	Aprovação exclusiva do gerente	P
RF04	Gerente cadastra representantes	A	RN04	Bloqueio por estoque insuficiente	A
RF05	Redefinição de senha por e-mail	P	RN05	"Em Digitação" e bloqueio de edição	P
RF06	Cadastro de produtos	A	RN06	Cálculo de subtotal e total	A
RF07	Notificação por e-mail ao gerente	P	RN07	Desconto até o subtotal do item	A
RF08	Criação de pedidos	A	RN08	Baixa de estoque na confirmação	PA
RF09	Inclusão de produtos e quantidades	A	RN09	Estorno no cancelamento	P
RF10	Cálculo de subtotal e total	A	RNF01	Senhas armazenadas por meio de *hash*	A
RF11	Edição do pedido	PA	RNF02	Uso de HTTPS	PA
RF12	Duplicação de pedido	P	RNF03	Acesso por perfil	A
RF13	Gerente vê os pedidos da empresa	A	RNF04	Resposta em até 3 s	NA
RF14	Aprovação ou rejeição de pedidos	P	RNF05	Cálculo do pedido em menos de 2 s	NA
RF15	Representante vê só os seus pedidos	A	RNF06	Interface responsiva	A
			RNF07	Mensagens de erro claras	PA
			RNF08	Integridade dos dados	A
			RNF09	Consistência do estoque	PA

Tabela 3. Casos de teste de acesso por empresa, acesso por representante e controle de estoque.

CT	Requisito	Cenário	Resultado esperado	Resultado observado	Situação
CT01	RN01	Usuário da empresa A, com A ativa, lista registros e abre pela URL um pedido da empresa B	Apenas registros de A; acesso ao pedido de B negado	Apenas registros de A; pedido de B retornou HTTP 404	Atendido
CT02	RF13, RF15, RN02	Representantes R1 e R2 com pedidos próprios; R1 e o gerente abrem a lista de pedidos	R1 vê só os seus; o gerente vê os de R1 e R2	Conforme o esperado	Atendido
CT03	RN04, RNF07	Produto com estoque 10; pedido de 12 unidades	Rejeição com mensagem; nada gravado; estoque inalterado	"Estoque insuficiente: disponível 10, solicitado 12"; estoque em 10	Atendido
CT04	RNF08	Pedido com um item válido e outro acima do estoque	Rejeição integral, sem gravação parcial	Nenhum item gravado; estoque inalterado	Atendido
CT05	RN08	Produto com estoque 10; criação de pedido com 4 unidades	Baixa apenas na confirmação	Estoque passou a 6 já na criação	Parcialmente atendido
CT06	RN09, RNF09	Exclusão do pedido do CT05	Estorno das 4 unidades	Estoque permaneceu em 6; não há cancelamento	Pendente (RN09); parcial (RNF09)

(RN03) e que o cancelamento de um pedido aprovado faça o estorno (RN09). Como o modelo de Pedido ainda não possui campo de situação, criação e confirmação coincidem na versão atual: ao ser criado, o pedido é validado e o estoque é baixado imediatamente (CT05). Por isso, a RN08 é parcial: a baixa automática existe e foi verificada, mas ocorre na criação. Não há aprovação, rejeição nem cancelamento, e a exclusão não estorna o estoque (CT06); por esse motivo, RN03, RN05 e RN09 estão pendentes.

Entre os RNF, a segurança apoia-se em recursos nativos do Django, como o armazenamento de senhas por meio de *hash*. O RNF02 é parcial porque o redirecionamento obrigatório para HTTPS não foi configurado; o RNF04 e o RNF05 não foram aferidos por falta de medição; o RNF07 é parcial porque a clareza das mensagens (CT03) depende de validação com usuários; e o RNF09 é parcial porque o bloqueio de linha não foi testado sob concorrência e a exclusão não reconcilia o estoque.

6. Discussão

Na etapa de registro, o SIGEP implementa mecanismos voltados aos problemas observados: os pedidos são estruturados, calculados automaticamente e validados contra o estoque em uma única transação, o que, no ambiente verificado, impede registros com valores inconsistentes ou acima do estoque. Se isso reduz erros e retrabalho na empresa é uma hipótese que depende de validação com usuários. A rastreabilidade do ciclo de vida e a comunicação entre representantes e gestores, que dependem de estados e notificações, ainda não foram implementadas.

Esse quadro decorre da prioridade dada à camada de acesso e escopo multiempresa, transversal a quase todas as entidades, e da ausência de um campo de situação no Pedido, que responde por RF11 (parcial), RF14, RN03, RN05, RN08 (parcial) e RN09. Observou-se também que a consistência do estoque depende de todos os caminhos que alteram pedidos: a criação é transacional, mas a exclusão não reconcilia o estoque, o que indica a conveniência de centralizar essa lógica em um único ponto da aplicação.

Em comparação com o sistema acadêmico analisado [Amorim 2023], o SIGEP trata o isolamento multiempresa, usa banco cliente-servidor com bloqueio de linha e é implantado em nuvem; em contrapartida, aquele trabalho já dispunha de fluxo de estados e foi validado com usuários. O risco de vazamento entre locatários quando uma consulta deixa de ser filtrada [Bezemer and Zaidman 2010] justifica centralizar o filtro por empresa em um único *mixin*. Em relação às soluções comerciais, o SIGEP permite adaptar o código, mas ainda não alcança sua maturidade operacional. Como as pendências exigem novos campos, migrações e *views* sobre a base existente, há indicação preliminar de que a arquitetura comporta essas extensões, o que só poderá ser confirmado com sua implementação.

7. Conclusão

O objetivo de projetar, desenvolver e verificar funcionalmente o SIGEP foi parcialmente alcançado: a aplicação, implantada em nuvem, atende a 10 dos 15 RF e a 1 parcialmente, e a 5 das 9 RN e a 1 parcialmente. A verificação requisito a requisito explicita o grau de maturidade do sistema e orienta sua continuidade.

Como limitações, a verificação foi manual, feita pelo próprio desenvolvedor, em ambiente controlado e em execução sequencial, sem testes automatizados nem de con-

corrência; os requisitos foram levantados por um participante do processo, sujeito a viés, e não foram validados formalmente; e não houve validação do sistema com usuários, de modo que não se pode afirmar que ele reduz erros ou retrabalho, apenas que corresponde à especificação nos itens atendidos.

Como trabalhos futuros, propõem-se: implementar os estados do pedido, com confirmação, aprovação, rejeição, bloqueio de edição, baixa na confirmação e estorno no cancelamento, centralizando a lógica de estoque; implementar a duplicação de pedidos e o envio de e-mails; configurar o redirecionamento para HTTPS; desenvolver testes automatizados de unidade, integração e concorrência; medir as metas de desempenho; realizar uma avaliação exploratória com perfis sintéticos e cenários simulados; e conduzir a validação com usuários das empresas interessadas.

Referências

Amorim, T. H. (2023). Desenvolvimento de um sistema web para controle de pedidos de balões personalizados de uma empresa de Sombrio, Santa Catarina. Trabalho de Conclusão de Curso (Bacharelado em Tecnologias da Informação e Comunicação), Universidade Federal de Santa Catarina, Araranguá, SC. Disponível em: https://repositorio.ufsc.br/handle/123456789/253695. Acesso em: 17 abr. 2026.

Bezemer, C.-P. and Zaidman, A. (2010). Multi-tenant SaaS applications: maintenance dream or nightmare? In *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE)*, pages 88–92, Antwerp, Belgium. ACM. DOI: https://doi.org/10.1145/1862372.1862393.

Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., and Venkatraman, N. (2013). Digital business strategy: toward a next generation of insights. *MIS Quarterly*, 37(2):471–482.

Chong, F., Carraro, G., and Wolter, R. (2006). Multi-tenant data architecture. Technical article (MSDN), Microsoft Corporation. Disponível em: https://docs.citusdata.com/en/v10.2/_static/mt-data-arch.pdf. Acesso em: 19 abr. 2026.

Davenport, T. H. (1998). Putting the enterprise into the enterprise system. *Harvard Business Review*, 76(4):121–131.

Django Software Foundation (2026). Django documentation. https://docs.djangoproject.com/en/5.2/. Acesso em: 22 abr. 2026.

Elmasri, R. and Navathe, S. B. (2016). *Fundamentals of Database Systems*. Pearson, Hoboken, 7 edition.

Gali, H. R. (2025). Understanding order management systems in retail and E-Commerce. *Journal of Information Systems Engineering and Management*, 10(59s):1066–1077. DOI: https://doi.org/10.52783/jisem.v10i59s.12997.

Gil, A. C. (2017). *Como elaborar projetos de pesquisa*. Atlas, São Paulo, 6 edition.

Gotel, O. C. Z. and Finkelstein, A. C. W. (1994). An analysis of the requirements traceability problem. In *Proceedings of the IEEE International Conference on Requirements Engineering (ICRE)*, pages 94–101, Colorado Springs, USA. IEEE.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Krasner, G. E. and Pope, S. T. (1988). A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3):26–49.

Mercos (2026). Sistema para representantes comerciais. `https://mercos.com/representantes-comerciais/`. Acesso em: 22 abr. 2026.

Omie (2026). Omie ERP: sistema de gestão empresarial. `https://www.omie.com.br`. Acesso em: 26 abr. 2026.

Sommerville, I. (2018). *Engenharia de Software*. Pearson, São Paulo, 10 edition.

The PostgreSQL Global Development Group (2026). PostgreSQL documentation: SELECT (the locking clause). `https://www.postgresql.org/docs/current/sql-select.html`. Acesso em: 28 abr. 2026.

Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Dong, J. Q., Fabian, N., and Haenlein, M. (2021). Digital transformation: a multidisciplinary reflection and research agenda. *Journal of Business Research*, 122:889–901.

Winsite (2026). Winsite Gestão. `https://winsitegestao.com.br/sobre`. Acesso em: 15 abr. 2026.