



Ferramentas CI/CD para E-commerce: Uma Análise Comparativa

Yuri Alexandre Okida Sato Islene Calciolari Garcia

Relatório Técnico - IC-PFG-23-15
Projeto Final de Graduação
2023 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Ferramentas CI/CD para E-commerce: Uma Análise Comparativa

Yuri Alexandre Okida Sato, Islene Calciolari Garcia *

Resumo

Este documento visa expor os conceitos e metodologias relacionados aos tópicos de integração, entrega e implantação contínuos — CI/CD — e aos princípios e atualidades do e-commerce, de modo a argumentar sobre a importância particular e fundamental desse conjunto de práticas para este setor da indústria. Os aspectos de complexidade, dinamicidade e demanda, bem como os desafios enfrentados por aqueles que atuam nessa indústria, também são explicitados e discutidos neste relatório.

Além disso, este texto também tem como finalidade experimental a apresentação dos aspectos e particularidades de oito diferentes ferramentas CI/CD do mercado atual, de modo a elaborar uma análise comparativa entre elas. Essa análise é baseada em critérios que descrevem os requisitos e recursos desejáveis para o desenvolvimento e manutenção não apenas de um software, mas também de uma plataforma de comércio eletrônico de sucesso.

1 Introdução

O crescente avanço da tecnologia tem impulsionado uma revolução no setor de comércio eletrônico, proporcionando um ambiente competitivo e dinâmico. Nesse contexto, a adoção de metodologias ágeis e eficientes de desenvolvimento de software tornou-se essencial para a viabilidade das empresas do setor, permitindo aos empreendedores e desenvolvedores de software automatizarem e agilizarem o processo de compra e entrega de produtos e serviços online.

Este artigo apresenta ao leitor os conceitos do e-commerce e os pilares do popular conjunto de metodologias e práticas de desenvolvimento, o “CI/CD”, bem como uma exposição e comparação entre ferramentas populares para automatização de seus processos.

A leitura deste documento visa, portanto, estimular a reflexão sobre os aspectos principais da indústria do comércio eletrônico e a forte dependência deste setor no que se refere à responsividade e eficiência dos processos de desenvolvimento e manutenção de software. Adicionalmente, este artigo pretende fornecer valiosos *insights* e orientações para

*Instituto de Computação, Universidade Estadual de Campinas, SP, Brasil.

empreendedores e desenvolvedores de software interessados em aprimorar seus processos de desenvolvimento e entrega de produtos e serviços no ambiente do e-commerce.

2 CI/CD: Conceitos e Metodologia

“Integração, entrega e implantação contínuos”: um conjunto de práticas para desenvolvimento de software popularmente conhecido pelo acrônimo “CI/CD” (“*Continuous Integration, Continuous Delivery, Continuous Deployment*”).

Sua função como metodologia é permitir que a equipe de desenvolvimento de um determinado software seja capaz de automatizar a integração contínua, testes e implantações de código em ambientes de produção, de modo a entregar novas funcionalidades e correções aos usuários finais de maneira confiável, rápida, repetível e de baixo risco. Essa metodologia tem como estrutura três conceitos fundamentais principais: integração contínua, entrega contínua e implantação contínua.

2.1 Integração Contínua

Cada um dos três pilares do CI/CD aborda um aspecto principal dessa metodologia. O primeiro deles, a integração contínua, é um processo que envolve a adição frequente e automatizada do código-fonte de vários desenvolvedores em um repositório compartilhado.

Cada nova alteração de código submetida pelos desenvolvedores o torna presumivelmente quebrado — ou não confiável — até que alguém prove que ele funciona, o que geralmente é feito durante as etapas de teste [1]. É nesse ponto que o segundo pilar de CI/CD, considerado por muitos uma extensão do primeiro, entra em ação: a entrega contínua.

2.2 Entrega Contínua

O conceito de entrega contínua é primordialmente descrito como um processo que vai além da simples integração do código-fonte e envolve a automação do processo de implantação de software em ambientes de teste ou pré-produção. Nesse conjunto de etapas, todo código que passa pela integração contínua é submetido a testes adicionais automatizados que podem ser de integração, desempenho, segurança e outros tipos de testes relevantes para a aplicação em questão.

Com os testes sendo executados a cada nova modificação aplicada sobre o código-fonte, bugs e inconsistências nessas alterações de código podem ser detectados mais cedo — quando ainda são fáceis e menos custosos de corrigir — e com mais frequência, garantindo uma entrega de funcionalidades eficiente e em conformidade com os requisitos especificados [1].

2.3 Implantação Contínua

O terceiro pilar das práticas CI/CD é a implantação contínua. Esse último e abrangente conceito fundamental visa implantar modificações no software em um ambiente de homologação ou produção de maneira automatizada. Isso significa tornar automáticos e consecutivos os processos de construção, implantação, teste e lançamento da aplicação em questão, colocando nas mãos dos seus usuários funcionalidades e correções vindas diretamente do controle de versão de maneira rápida e de baixo risco [2].

O objetivo dessa automatização é que a execução dessa complexa cadeia de processos seja disparada com a compilação (ou “build”) especificada com o simples apertar de um botão ou mesmo com o mesclar de uma ramificação. Assim, implantar builds em qualquer ambiente se torna um processo monitorável para os desenvolvedores e de fácil execução para qualquer divisão do time de desenvolvimento — como equipes de teste ou operações, por exemplo —, bem como possibilita que qualquer problema ou gargalo encontrado em uma dada etapa da cadeia processos seja identificado e corrigido rapidamente. É o que J. Humble e D. Farley definem como o “pull system” [2].

Automatizado, de fácil implantação e programado para testar com frequência o software, um fluxo de trabalho que opera com a implantação contínua pode ser considerado um processo de desenvolvimento não só mais rápido como também mais seguro [2]. Entretanto, para tornar automático todo esse fluxo — primordialmente manual — de processos que compõem o caminho de uma alteração no código-fonte do controle de versão até o usuário, é necessário elaborar o chamado “fluxo de implantação” (ou “deployment pipeline”).

2.4 Fluxo de Implantação

Como já discutido nos parágrafos anteriores, um fluxo de implantação nada mais é que a manifestação automatizada da série de etapas e processos executados para a implantação de software em diferentes ambientes, desde o desenvolvimento até a produção [2]. Esse fluxo (ou “pipeline”) comumente abrange a compilação, testes, empacotamento, implantação em ambientes de teste, pré-produção e, finalmente, a implantação em ambiente de produção.

Um fluxo de implantação, portanto, deve ser construído a partir do mapeamento dessa série de etapas e processos, antes realizados manualmente, resultando em um “mapa de fluxo de valor”. Esse mapa deve descrever cada detalhe que o time de desenvolvimento achar relevante para o processo de implantação — descrever o funcionamento e definir responsáveis para determinadas etapas, descrever condições excepcionais, verificar quais recursos são necessários para determinada etapa, e assim por diante [2].

O fluxo de implantação construído a partir do mapa de fluxo de valor para automatizar a série de processos original, entretanto, deve conter etapas comuns a todos os projetos, como descrevem Humble e Farley. Essas etapas (ou quaisquer etapas que atuem de maneira similar no fluxo de implantação de um determinado software) garantem a qualidade,

a funcionalidade e a usabilidade do sistema antes de ser entregue aos usuários finais, bem como garantem a rapidez, repetibilidade e confiabilidade do processo de implantação como um todo [2]. Essas etapas padrão são:

- **Estágio de Commit:** Essa etapa é responsável por garantir que o sistema funciona em nível técnico. É realizada a compilação e testagem unitária do código [2].
- **Estágios de Teste de Aceitação Automatizado:** Essa etapa é responsável por verificar se o sistema funciona em nível funcional e não funcional, isto é, se seu comportamento atende às necessidades dos usuários e às especificações requisitadas. São realizados testes automatizados de aceitação, que podem incluir testes de desempenho, segurança e outros [2].
- **Estágios de Teste Manual:** Essa etapa é sobre detectar quaisquer defeitos que não tenham sido capturados pelos testes automatizados e verificar se o sistema é utilizável e cumpre seus requisitos. Esse tipo de etapa também pode incluir ambientes de testes, integração e de aceitação do usuário (UAT, ou “User Acceptance Testing”) para que sejam corretamente validados, a depender do plano de trabalho elaborado pelo time de desenvolvimento [2].
- **Estágio de Lançamento:** Por fim, essa etapa é sobre a entrega do sistema aos usuários com a build especificada, isto é, com as funcionalidades e correções requisitadas agora testadas, validadas e implementadas. Seja como software empacotado ou por meio de implantação em um ambiente de produção ou ambiente de pré-produção [2].

2.5 Feedback

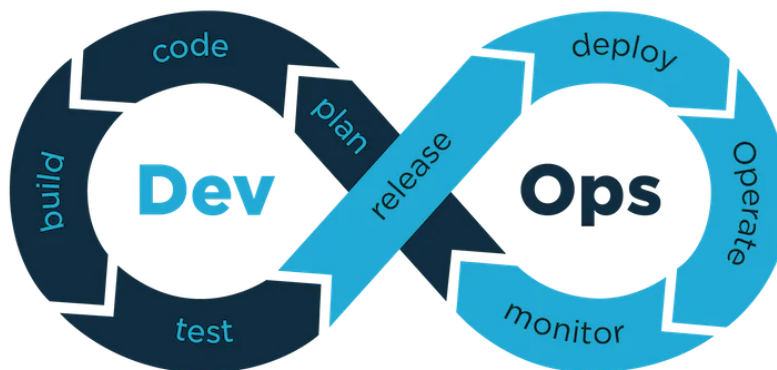
Mecanismos CI/CD seriam dificilmente viáveis se não tivessem como uma de suas mais importantes prioridades a busca por meios otimizados de fornecer feedback: uma entidade, que chamaremos aqui de “fornecedor”, cria — ou executa processos sobre — o software e o entrega para uma segunda entidade, que podemos chamar de “cliente”, a qual emitirá uma avaliação sobre o que foi recebido.

Esteja no papel de cliente um time de QA — *quality assurance* — ou mesmo os próprios usuários finais, um retorno avaliativo emitido por este para quem quer que esteja no papel de fornecedor é essencial para permitir a identificação de falhas e gaps de qualidade enquanto promove melhorias contínuas no software, garantindo assim a satisfação dos usuários e a entrega de um produto de alta qualidade.

Como já deve ter ficado claro em tópicos anteriores, os fluxos de implantação estão repletos de mecanismos fornecedor-cliente, justamente para que cada tarefa só avance para a próxima etapa do fluxo mediante à validação desse “cliente”. Caso não cumpra com os requisitos necessários e precise ser refatorada, a tarefa volta para o fornecedor, formando um ciclo que nunca se quebra enquanto essa validação não estiver de acordo com o esperado.

O que as práticas de CI/CD tentam fazer é acelerar esses ciclos de retroalimentação presentes no fluxo de desenvolvimento de tal forma a gerar rapidamente valiosos relatórios de bugs e resultados de testes (para o caso de feedback interno) ou mesmo para avaliar a qualidade, usabilidade e adequação do software entregue (para o caso de feedback emitido pelos usuários finais, por exemplo).

A imagem a seguir representa, de modo geral, o ciclo de desenvolvimento de acordo com a abordagem “DevOps Architecture”, um conjunto de práticas que inclui o CI/CD e que visa facilitar a implementação e a adoção de metodologias DevOps — isto é, núcleos de desenvolvimento (Dev) e infraestrutura (Ops) trabalhando de maneira integrada e colaborativa — em um ambiente de desenvolvimento de software.



Note que a etapa do ciclo responsável por fazer com que ele volte à etapa de planejamento é “Monitor” (ou “Monitorar”), que justamente refere-se a monitorar os feedbacks de cada tarefa em andamento e planejar/atuar de acordo: essa é a retroalimentação que liga o fim de um ciclo ao início do próximo, seja qual for o contexto fonecedor-cliente.

3 Indústria do E-commerce: Conceitos e Atualidades

Com o advento da internet comercial na década de 90 e sua popularização crescente nos anos seguintes, o chamado comércio eletrônico começou a ganhar impulso e se tornar o que mudaria para sempre o mercado empresarial. “E-commerce”, abreviação popular do nome dessa indústria, refere-se à compra e venda de produtos e serviços por meio da internet. É um modelo de negócio que permite que empresas e consumidores realizem transações comerciais online, de qualquer parte do mundo e sem a necessidade de interações presenciais.

Analogamente próximo do conceito de uma loja física, em plataformas de e-commerce os consumidores — sejam eles organizações ou indivíduos — podem navegar por catálogos de produtos ou serviços, adicioná-los a um “carrinho” ou “sacola”, fazer pedidos, efetuar pagamentos e receber essas mercadorias ou serviços por meio de entregas físicas ou digitais. Em suma, o conceito de e-commerce tem como núcleo fundamental a troca de valor entre quaisquer entidades participantes que assumam o papel de clientes ou comerciantes [3].

Atualmente — devido principalmente aos esforços empresariais, tecnológicos e sociais das últimas décadas —, a indústria do e-commerce tem alcançado proporções extraordinárias e continua a expandir-se de forma crescente a cada ano, tornando-se uma parte cada vez mais importante do cenário do comércio global.

3.1 Tipos de E-commerce

Assim como já eram categorizadas em diferentes arquétipos antes da grande ascensão do setor comercial eletrônico, as transações comerciais são descritas a partir de uma nomenclatura simples, a qual convenientemente chamaremos neste relatório de “tipos de e-commerce”. Ela tem como função descrever a natureza do negócio, a troca de valor entre duas entidades.

No e-commerce, esses arquétipos ainda são plenamente válidos e caracterizam o conceito base do negócio, explicitando quem são as entidades participantes e o tipo de transações envolvidas.

A seguir estão listados os principais tipos de e-commerce presentes na indústria, e que também são os mais relevantes para o conteúdo deste relatório:

- **B2C (Business-to-Consumer)**: É o tipo mais comum de e-commerce, em que as empresas vendem diretamente para os consumidores individuais finais [3]. Exemplos incluem compras em lojas online de varejo, como Amazon, Americanas e Magazine Luiza.
- **B2B (Business-to-Business)**: Nesse arquétipo, as transações ocorrem entre empresas. Empresas compram produtos, serviços ou informações de outras empresas para uso em suas operações [3]. Um exemplo seria um fabricante que vende seus produtos para um varejista ou mesmo uma empresa que disponibiliza à outra empresa uma chave única para executar chamadas às APIs de seus microsserviços.
- **C2C (Consumer-to-Consumer)**: É definido pela venda direta de produtos ou serviços entre consumidores com o auxílio da plataforma, muitas vezes atuando como uma entidade mediadora das transações [3]. Plataformas como eBay e Mercado Livre, por exemplo, são exemplos desse tipo de e-commerce, onde os consumidores podem vender produtos usados e realizar transações entre si.

3.2 Categorias de Infraestrutura

No dinâmico mundo do comércio eletrônico, a escolha da infraestrutura desempenha um papel crucial no planejamento, criação, manutenção e, sobretudo, funcionamento das plataformas de e-commerce. Diferentes abordagens e tecnologias são utilizadas na construção dessas plataformas, oferecendo opções personalizadas para atender às necessidades e requisitos específicos de cada empresa.

Cada negócio específico pode ser mais compatível com uma categoria de infraestrutura do que outra, dado o contexto e a complexidade desse negócio, bem como as necessidades e objetivos do empreendedor. A seguir estão listados dois dos modelos de infraestrutura popularmente utilizados pela indústria do comércio eletrônico.

3.2.1 Plataformas Genéricas

Nessa abordagem, as plataformas *white label* — ou “rótulo branco” — são desenvolvidas como aplicações genéricas prontas para uso, flexíveis e customizáveis, geralmente sendo fornecidas como serviços ao cliente empreendedor. Elas fornecem uma estrutura básica em branco, como o nome sugere, a qual os próprios clientes podem personalizar e adaptar de acordo com suas necessidades e preferências.

Esse tipo de aplicação geralmente oferece recursos básicos de comércio eletrônico, como gerenciamento de produtos, carrinho de compras e processamento de pagamentos, por exemplo. A intenção é que os clientes empreendedores construam sua própria identidade de marca e experiência de compra por conta própria. Mercado Shops e Shopify são dois exemplos de aplicação *white label*.

Em contraponto, por esse tipo de abordagem não envolver uma equipe de desenvolvimento propriamente alocada para atender as necessidades de cada cliente — já que trata-se de uma aplicação genérica —, ela pode não permitir modificações e personalizações mais profundas, o que, a depender da natureza e complexidade do negócio, pode acabar se tornando um problema.

3.2.2 Plataformas de Desenvolvimento Personalizado

Oposta ao caso em que a plataforma de e-commerce é construída pelo próprio empreendedor a partir de uma aplicação genérica, as plataformas dessa categoria são construídas do zero por uma equipe de desenvolvimento de software alocada para esse cliente. Essas equipes desenvolvem uma solução personalizada e sob medida para as necessidades específicas do empreendedor.

A plataforma pode ser projetada para lidar com a criação e alteração de recursos e funcionalidades específicas. Integrações complexas com aplicações legado e serviços externos, implementação de processos de back-end personalizados ou mesmo requisitos exclusivos de negócios são alguns dos exemplos mais comuns.

Essas plataformas geralmente requerem mais tempo e exigem mais recursos para serem desenvolvidas, mas oferecem maior controle e flexibilidade de customização, uma vez que cada requisição efetuada pelo cliente significa uma alteração de código ou parâmetro que a equipe de desenvolvimento alocada pode aplicar. Dado o escopo de estudo deste relatório, é sobre esse modelo de infraestrutura que discutiremos em tópicos e seções posteriores.

Adicionalmente, é importante mencionar que essas aplicações de desenvolvimento personalizado podem ser subcategorizadas em dois tipos comuns de abordagem infraestrutural: aplicações que executam chamadas externas e aplicações monolíticas. Uma aplicação de comércio eletrônico, bem como qualquer outro tipo de aplicação web, entretanto, pode conter aspectos de ambas as abordagens.

O primeiro tipo de abordagem diz respeito a aplicações que contenham funcionalidades baseadas em chamadas a APIs ou microsserviços externos, públicos ou privados. Com essa estratégia, aproveita-se de recursos pré-existentes e serviços especializados de tal forma que os desenvolvedores se concentrem em aspectos específicos da plataforma enquanto aproveitam soluções externas para tarefas comuns ou complexas. Isso reduz o esforço de desenvolvimento e acelera o tempo de lançamento, mas também torna a aplicação mais dependente de terceiros.

Já o segundo tipo de abordagem de infraestrutura, aplicações monolíticas, pode ser descrito como um sistema único em que todos os componentes estão interligados, dependentes uns dos outros e internos à aplicação, sem necessidade de efetuar chamadas externas. Nesse caso, apesar da maior independência em relação à estratégia do parágrafo anterior, alterações ou adições de recursos normalmente exigem modificações no sistema como um todo, o que pode fazer com que alterações de código sejam mais complexas do que seriam em uma aplicação não monolítica.

3.3 Desafios Enfrentados

Como já deve ter ficado evidente pelas informações explicitadas nos tópicos anteriores, a indústria de comércio eletrônico apresenta uma natureza complexa de negócio, de estado volátil, e que demanda respostas rápidas por parte do empreendedor para manter sua plataforma atualizada e viável competitivamente.

Os empreendedores devem estar atentos para garantir que sua plataforma atenda às demandas e expectativas dos consumidores, enquanto os desenvolvedores devem responder rapidamente às necessidades de manutenção, melhorias e implementação de novos recursos. Essa dinâmica acelerada só reitera como a agilidade é um fator crítico.

Além dessas, há uma série de outras necessidades essenciais para uma plataforma de sucesso que precisam ser levadas em conta pelo empreendedor e pela equipe de desenvolvimento. Os grandes desafios do e-commerce residem justamente em manter o negócio em dia perante essas necessidades, e é sobre algumas delas em destaque que vamos discutir neste tópico.

3.3.1 Experiência do Cliente

Investir em uma boa experiência do cliente é uma necessidade essencial para o sucesso no setor de comércio eletrônico. À medida que essa indústria continua a crescer e se tornar cada vez mais competitiva, os consumidores estão se tornando mais exigentes em relação à sua experiência de compra online.

Os consumidores são atraídos, engajados e retidos em uma plataforma cuja a experiência de compra é bem construída, baseada em boas práticas de UX e tendências atuais de mercado. Seguem alguns dos fatores importantes para o design de uma plataforma bem-sucedida.

- **Funcionalidade:** Páginas que sejam funcionais, carregem o mais rápido possível e que direcionem o consumidor para os produtos ofertados [3].
- **Informatividade:** Presença de links que permitam que o consumidor descubra mais sobre a empresa e seus produtos/serviços [3].
- **Facilidade de Uso:** A navegação ser intuitiva para qualquer consumidor que acesse a plataforma [3].
- **Redundância de Navegação:** Navegações alternativas que levem o consumidor ao mesmo conteúdo [3].
- **Facilidade de Compra:** Compras que exigem poucos cliques para serem efetuadas [3].
- **Ubiquidade:** Funcionamento da plataforma em diferentes navegadores e dispositivos [3].
- **Minimalismo Gráfico:** Ausência de recursos visuais e sonoros os quais não estejam sobre o controle do consumidor e possam atrapalhá-lo ou distraí-lo durante o fluxo de compra [3].
- **Legibilidade:** Presença de uma fonte altamente legível para os textos da plataforma bem como a ausência de sobreposição de textos e imagens que não sejam contrastantes, e portanto, menos legíveis [3].

3.3.2 Segurança e Privacidade

A segurança e privacidade são aspectos essenciais no comércio eletrônico, e sua presença é fundamental para construir confiança nos clientes e garantir uma experiência segura e tranquila. Nesse contexto, os e-commerces devem priorizar a implementação de medidas robustas de segurança e privacidade para garantir a proteção dos dados dos clientes [5].

A principal razão para a necessidade de segurança e privacidade no e-commerce é a proteção dos dados do cliente. Ao fazer uma compra online, os clientes fornecem informações pessoais e financeiras sensíveis, como números de cartão de crédito e endereços

de entrega. É fundamental que esses dados sejam protegidos de forma adequada para evitar o acesso não autorizado ou o uso indevido [5].

É no documento popularmente conhecido como “política de privacidade”, oferecido pelas plataformas de comércio eletrônico, que são informados quais dos dados dos consumidores serão usados e onde, quando, como e por quem (ou pelo quê) serão usados. Esse documento também deve informar sobre o uso de cookies, tecnologias de segurança e compartilhamento de dados [5].

Além disso, seja por parte do próprio empreendedor ou da equipe desenvolvimento responsável, deve-se sempre estar atento à presença de ameaças ou brechas de segurança. Vulnerabilidades encontradas em funcionalidades da plataforma, por exemplo, bem como aquelas encontradas em plugins e bibliotecas que as constituem a nível de código, devem ser detectadas e corrigidas o quanto antes. Daí mais um aspecto desta indústria que colabora para a exigência de ações ágeis.

3.3.3 Logística e Prazo

A logística é um elemento fundamental no comércio eletrônico e desempenha um papel crucial na satisfação dos clientes. Em um mundo cada vez mais conectado e com expectativas de entregas rastreáveis e rápidas, os e-commerces precisam se destacar na gestão eficiente da logística e no cumprimento dos prazos de entrega [6, 7].

No contexto atual, inclusive, mudanças de hábitos de consumo ocasionadas pela pandemia contribuíram para uma massa consumidores mais exigente e ansiosa por ter seus produtos entregues o mais rápido possível. A necessidade de uma logística eficiente, portanto, torna-se um dos principais fatores para que um negócio se destaque dentre os seus concorrentes [6, 7]. A plataforma Mercado Livre, por exemplo, já é capaz de entregar produtos no mesmo dia ou no dia seguinte à criação do pedido.

Dito isso, com e-commerces promocionando prazos de entrega cada vez mais curtos, é seguro afirmar que os participantes desta indústria, para proporcionar tal efetividade logística, têm de lidar com complexas lógicas de negócio — e terão cada vez mais, com o passar do tempo e exigências cada vez mais rígidas por parte da massa de consumidores — bem como precisam prontamente atuar para a implementação destas em seus empreendimentos. Isso para manterem-se atualizados e competitivos no mercado [6, 7].

3.3.4 A Importância das Tendências

Toda empresa que participa ativamente do setor de comércio eletrônico precisa estar atenta às tendências dessa indústria para melhorar suas chances de se destacar. Para o contexto de mercado no qual estamos inseridos hoje, algumas das principais tendências que merecem destaque são:

- **M-Commerce:** O Mobile Commerce, também conhecido como “M-Commerce”, é uma tendência crescente no comércio eletrônico que se refere às transações comerciais realizadas através de dispositivos móveis, como celulares e tablets. Essa forma de comércio eletrônico tem se tornado cada vez mais popular devido ao aumento do uso de dispositivos móveis e à conveniência e acessibilidade que esse método de compra oferece aos consumidores, permitindo que eles façam pedidos a qualquer momento e em qualquer lugar, desde que tenham acesso à internet [8].
- **Social Commerce:** A tendência do “Social Commerce” refere-se à integração do comércio eletrônico com as redes sociais. O Social Commerce envolve o uso de plataformas de mídia social, como Facebook, Instagram e outras, para promover e vender produtos diretamente aos consumidores [8].
- **Checkout sem necessidade de login:** Essa tendência se refere à prática de permitir que os clientes concluam suas compras online sem a necessidade de criar uma conta ou fazer login na plataforma. Com tal abordagem, o processo de finalização do pedido é simplificado e agilizado, eliminando a etapa de registro obrigatório e otimizando a experiência de compra, principalmente para novos clientes [8].
- **Pagamentos instantâneos:** A tendência de pagamentos instantâneos para plataformas e-commerce refere-se à adoção de sistemas de pagamento que permitem a transferência de fundos de forma rápida e imediata, oferecendo uma experiência de pagamento ágil e conveniente para os consumidores. Essa tendência visa eliminar a necessidade de processos demorados e tediosos, como a espera por confirmações de pagamento ou mesmo o cadastro de cartões de crédito. Como exemplo, é possível citar o PIX, o sistema brasileiro de pagamentos instantâneos [8].
- **Cashback:** A tendência do “Cashback” no comércio eletrônico se refere a um programa ou modelo de recompensa em que os clientes recebem um retorno financeiro em forma de dinheiro ou crédito por suas compras. Essa estratégia tem como objetivo atrair novos clientes ao passo que incentiva a fidelidade e estimula a repetição de compras [8].
- **Realidade aumentada:** A tendência de realidade aumentada tem como premissa a integração de elementos virtuais interativos para melhorar a experiência de compra online. A Realidade Aumentada permite que os consumidores visualizem produtos em um ambiente virtual, utilizando tecnologias como câmeras de smartphones, óculos de realidade aumentada ou até mesmo navegadores da web. Como exemplo, pode-se citar plataformas com provedores virtuais, os quais permitem que os consumidores coloquem suas medidas nos modelos e sejam capazes de visualizar como as peças de roupa os vestirão [8].

É importante mencionar que novas tendências do comércio eletrônico surgem continuamente, o que exige, se fizer sentido para o dado modelo de negócio, uma resposta rápida por parte das plataformas para as implementarem o quanto antes. Manter-se atualizado no que diz respeito às tendências e necessidades, em suma, é o que compõe o desafio fundamental desta indústria.

4 Ferramentas CI/CD: Uma Visão Geral

Nesta seção serão apresentadas algumas das ferramentas de CI/CD mais populares do mercado atual. Ferramentas de propósito geral — isto é, pensadas para funcionar em qualquer projeto de desenvolvimento — mas de características diversas.

Além disso, essas ferramentas irão compor o espectro de opções disponíveis para a discussão principal deste relatório nas seções posteriores. As características chave e considerações associadas a cada uma foram baseadas em [9] e nas páginas oficiais das ferramentas, bem como nas avaliações mais recentes dos usuários presentes nas plataformas g2.com, trustradius.com, softwareadvice.com, getapp.com e producthunt.com. Dito isso, seguem abaixo breves visões gerais acerca de cada uma delas:

4.1 Jenkins

Jenkins é uma ferramenta de CI/CD que oferece centenas de plugins para dar suporte à construção e implantação de projetos. É um servidor de automação de código aberto e uma solução auto-hospedada, o que significa que não possui uma opção de SaaS — “*Software as a Service*” [9, 11].

O Jenkins é um programa baseado em Java, autocontido, que pode ser executado imediatamente após a instalação. A instalação e a manutenção são relativamente fáceis, embora ainda exijam algum conhecimento do usuário. Suportado em Windows, Linux, macOS, FreeBSD, Solaris e outros [9, 11].

Características-chave:

- Extensibilidade através da arquitetura de plugins, oferecendo possibilidades infinitas de personalização e integração com outras ferramentas e sistemas.
- Uma ferramenta de código aberto apoiada por uma comunidade forte e ativa, o que significa que é possível encontrar suporte e recursos adicionais facilmente.
- Capacidade de distribuir cargas de trabalho em várias máquinas, permitindo escalabilidade e melhorando o desempenho geral.
- Criação de pipelines bem flexível e customizável.
- É um software de código aberto disponível gratuitamente.

Considerações:

- Não é fácil de instalar e configurar, principalmente para usuários iniciantes.
- É dependente demais de plugins no que diz respeito a algumas funcionalidades básicas que já deveriam estar na ferramenta.
- Como é uma solução auto-hospedada, uma infraestrutura adequada é exigida para executar e manter a ferramenta.

4.2 GitHub Actions

O GitHub Actions é a ferramenta de integração e entrega contínuos de classe mundial fornecida pelo GitHub. Essencialmente, é possível construir, testar e implantar código diretamente do GitHub. Seja para implantar um serviço web, criar um contêiner ou automatizar as boas-vindas a novos usuários em projetos de código aberto, o GitHub Actions possui uma ação específica para cada caso [9, 12].

Além disso, a ferramenta é capaz de simplificar a gestão de pacotes a partir do recurso Packages para incluir atualizações de versões, resolução de dependências e até mesmo distribuição rápida via CDN global do GitHub. Atua como SaaS mas permite executores auto-hospedados. Gratuito apenas para repositórios públicos. Suportado em Windows, Linux, macOS e ARM [9, 12].

Características-chave:

- Ampla gama de ações pré-definidas para automatizar tarefas comuns no desenvolvimento de software.
- Economiza tempo ao utilizar fluxos de trabalho em matriz, *matrix workflows*.
- Os logs ao vivo fornecem informações sobre os fluxos de trabalho em execução em tempo real.
- Possui um repositório de segredos incorporado para armazenamento seguro de informações sensíveis.
- Permite o uso de máquinas virtuais em nuvem ou local com executores auto-hospedados, *self-hosted runners*.
- Possui um plano gratuito básico para indivíduos e organizações, bem como edições para equipes e empresas, oferecendo diversas opções para escolher.
- Integração perfeita com o ecossistema do GitHub, permitindo uma integração contínua suave com repositórios existentes.
- Integração com o recurso Packages do GitHub para gerenciamento simplificado de pacotes.

Considerações [10]:

- Criar fluxos de trabalho que são disparados por eventos externos ao repositório configurado é difícil e pode ser um problema para projetos não centralizados ou que contenham integrações com terceiros.
- Artefatos produzidos pelos fluxos de trabalho não são visíveis e rastreáveis como o código do repositório. É difícil acessá-los de forma nativa no repositório, e não há um histórico de versões para eles, dificultando sua reutilização em contextos e projetos diferentes.

4.3 Azure Pipelines

Trata-se de uma ferramenta CI/CD que é parte do serviço de automatização em nuvem para DevOps fornecido pela Microsoft, o Azure DevOps. Com recursos abrangentes, essa ferramenta oferece a possibilidade de testar e construir seu código de forma regular e consistente, de forma a enviá-lo para qualquer ambiente alvo ao passo em que economiza tempo que seria gasto corrigindo erros [9, 13].

A ferramenta também apresenta recursos de integração interessantes com GitHub, GitLab e BitBucket, de modo a possibilitar de maneira eficiente a construção, teste e implantação de aplicativos armazenados nos repositórios daquela plataforma. Atua como SaaS mas permite executores auto-hospedados. Suportado em Windows, Linux e macOS [9, 13].

Características-chave:

- Facilita a construção de aplicativos escaláveis de alto desempenho.
- Permite a implantação e dimensionamento de contêineres no Kubernetes gerenciado.
- Executa operações de código serverless baseadas em eventos com uma experiência de desenvolvimento completa.
- Integração com o Azure Board que oferece ferramentas interativas e personalizáveis com um conjunto rico de recursos. Inclui suporte nativo para processos Scrum, Agile e Kanban.
- O modelo de preços varia de acordo com as necessidades do usuário.
- Amplas capacidades de CI/CD para construção, teste e implantação de aplicativos.
- Integração com o GitHub, facilitando a colaboração e o controle de versão para usuários daquela plataforma.

Considerações:

- Não é fácil para usuários iniciantes. Alguns recursos avançados podem exigir conhecimentos especializados em tecnologias específicas.
- Pode apresentar problemas de lentidão em determinadas regiões do mundo — Canadá e países na América do Sul são exemplos.
- Documentação da ferramenta poderia ser mais clara e sucinta.

4.4 GitLab CI/CD

Ferramenta de CI/CD de código aberto contida no pacote de DevOps oferecido pela GitLab como um de seus serviços disponíveis. Essa ferramenta fornece recursos de planejamento robustos, bem como funções de análise da qualidade do código-fonte, de modo a garantir que todo código implantado em ambientes de produção estejam em conformidade com os padrões de código definidos para a aplicação do usuário [9, 14].

Chamado de “Auto DevOps”, o conjunto de recursos e integrações pré-configurados dessa ferramenta CI/CD é capaz de construir, implantar, testar e monitorar o software de maneira automatizada. Atua como SaaS mas pode ser auto-hospedado. Suportado em Windows, Linux e macOS [9, 14].

Características-chave:

- Conecta-se ao seu provedor de nuvem por meio do OpenID Connect para obter credenciais temporárias de acesso a serviços.
- O ChatOps dispara tarefas de CI a partir de conversas via chat.
- Permite determinar rapidamente os efeitos de desempenho do navegador e do servidor devido a alterações de código pendentes.
- Seus relatórios de testes unitários identificam falhas nos testes em solicitações de merge.
- Oferece recursos gratuitos para usuários individuais, que estão disponíveis para sempre.
- Possui recursos avançados de segurança e análise de qualidade de código.

Considerações:

- É necessário definir explicitamente quais artefatos devem ser salvos e compartilhados entre as tarefas.
- Interface de usuário não é tão intuitiva e moderna quanto deveria.
- Documentação da ferramenta poderia ser mais clara e sucinta.

4.5 Bitbucket Pipelines

O Bitbucket Pipelines é uma ferramenta CI/CD fornecida pelo Bitbucket e disponibilizada no serviço de colaboração de desenvolvimento Bitbucket Cloud. Assim como as outras da lista, essa ferramenta tem como objetivo permitir que os desenvolvedores automatizem o processo de construção, teste e implantação de seus aplicativos, facilitando a entrega rápida e confiável de software [9, 15].

Seu diferencial, entretanto, reside na integração nativa com os repositórios Bitbucket, de modo a facilitar o planejamento, configuração e monitoramento de pipelines, execução de tarefas personalizadas em contêineres Docker, bem como pode integrar-se com algumas ferramentas de CI/CD externas, como Jenkins, por exemplo. Atua como SaaS mas permite executores auto-hospedados. Suportado em Windows, Linux e macOS [9, 15].

Características-chave:

- Integração perfeita com o Bitbucket, oferecendo uma experiência unificada de desenvolvimento.
- Configuração simples de pipelines de CI/CD diretamente nos repositórios.
- Suporte a várias linguagens de programação e execução de tarefas personalizadas em contêineres Docker.
- Monitoramento em tempo real e rastreamento detalhado de pipelines, com logs detalhados e visualizações de status.
- Possibilidade de integração com outras ferramentas populares de CI/CD.
- Suporta integração nativa com o Jira.
- Integração com outras ferramentas populares de CI/CD.
- Oferece três planos: gratuito, padrão e premium.

Consideração:

- Artefatos não são compartilhados entre pipelines, prejudicando a eficiência de processos de construção de projetos multi-modulares, por exemplo.

4.6 CircleCI

Trata-se de uma ferramenta CI/CD conhecida por oferecer escalabilidade e flexibilidade aos desenvolvedores ao passo que apresenta uma interface de usuário intuitiva e customizável. O CircleCI também possui integrações com várias ferramentas externas, como os serviços de repositório GitHub, Bitbucket, GitLab e até mesmo serviços de hospedagem como AWS, Google Cloud Platform e Heroku [9, 16].

Para os processos de construção e teste, essa ferramenta apresenta eficiência ao executar tarefas em paralelo. Os testes, inclusive, podem ser executados em ambientes isolados — instâncias isoladas do sistema operacional em um contêiner virtualizado —, o que garante a confiabilidade do código antes da implantação e que cada execução de teste seja independente e não seja afetada por outras execuções ou ambientes externos. Atua como SaaS mas permite executores auto-hospedados. Suportado em Windows, Linux e macOS [9, 16].

Características-chave:

- Apresenta uma velocidade líder com o executor de CI mais rápido.
- Oferece a melhor forma de identificar e solucionar problemas rapidamente ao facilitar inspecionar arquivos de log, caminhos de diretórios e processos em execução. Também permite acessar todos os jobs através de SSH.
- Oferece recurso de divisão de testes do CircleCI que permite definir eficientemente onde ocorrem as divisões em um conjunto de testes.
- Suporta armazenamento em cache para tornar os jobs mais rápidos na plataforma, reutilizando dados de operações de busca caras de jobs anteriores.
- Integração fácil com repositórios Git e outros sistemas de controle de versão.
- Configuração simples e rápida para começar a usar.
- Ambiente de teste isolado, permitindo a execução de testes automatizados confiáveis.
- Oferece quatro planos: gratuito, performance, scale e server.

Consideração:

- Número de construções é limitado por créditos.

4.7 TeamCity

Oferecida pela JetBrains, TeamCity é uma ferramenta CI/CD de propósito geral oferece flexibilidade para todos os tipos de fluxos de desenvolvimento em empresas e equipes de todos os tamanhos [9, 17].

A ferramenta também apresenta recursos de monitoramento e controle de testes, bem como envio automático de relatórios de erro em tempo real, avisando a equipe de desenvolvimento sobre os problemas sem precisar esperar que a construção de software seja finalizada. Por apresentar integração facilitada com diversas IDEs, o TeamCity também é capaz de levar o usuário exatamente à linha de código onde esse problema relatado foi detectado. Atua apenas de maneira auto-hospedada. Suportado em Windows, Linux e macOS [9, 17].

Características-chave:

- Interface amigável e de fácil configuração.
- Sem necessidade de plugins.
- Integração fácil com várias IDEs.
- Fornece um painel de controle centralizado para acompanhar o status dos builds, testes e implantações em tempo real. Além disso, oferece relatórios detalhados para análise de métricas e tendências de desempenho.
- É capaz de lidar com projetos de diferentes tamanhos e escalas, permitindo a distribuição de builds em múltiplos agentes para otimizar o desempenho e reduzir o tempo de espera.
- Fornecimento de documentação abrangente.
- Execução remota e commit pré-testado.
- A licença do servidor profissional é gratuita para sempre. A edição empresarial permite usar recursos mais avançados.

Considerações:

- Atualizar a ferramenta é um processo pesado e lento.
- Como é uma solução auto-hospedada, uma infraestrutura adequada é exigida para executar e manter a ferramenta.

4.8 Travis CI

Travis CI é uma ferramenta CI/CD conhecida por ter sido a primeira a oferecer serviços a projetos de código aberto de forma gratuita. A ferramenta é capaz de executar o processo de construção, teste e implantação diretamente sobre aplicações em repositórios GitHub ou Bitbucket [9, 18].

É utilizada através de serviços em nuvem serverless para dimensionamento sob demanda (SaaS), mas também pode ser auto-hospedada em sua versão paga. Além disso, ela oferece recursos avançados, como fluxos de trabalho baseados em pull requests, execução paralela de testes e backup automático de builds, para o caso de uma nova compilação falhar. Suportado em Windows, Linux e macOS [9, 18].

Características-chave:

- Configuração rápida e fácil de usar.
- Suporte para fluxos de trabalho baseados em pull request para facilitar a colaboração.
- Revisões de compilação ao vivo para monitorar o progresso e identificar problemas rapidamente.
- Integração fácil com ambos GitHub e Bitbucket.
- Serviços de banco de dados pré-instalados.
- Máquinas virtuais limpas para cada compilação.
- Fornece um teste gratuito de 30 dias na edição Cloud. Após esse período, um dos planos (Bootstrap, Startup, Small Business, Premium ou Platinum) deve ser escolhido. A edição Enterprise tem um preço fixo mais baixo e é auto-hospedada.

Consideração:

- Considerando o que é oferecido, preços dos planos são altos em relação às ferramentas concorrentes.

5 Critérios de Comparação

Com as ferramentas devidamente apresentadas na seção anterior, serão aqui explicitados os critérios de comparação entre elas. Esses critérios foram elaborados a partir de quais requisitos e recursos de ferramentas CI/CD são comuns, ideais e importantes para qualquer tipo de projeto de software [4].

Entretanto, considerando o tema de estudo deste relatório, é importante lembrar que a criticidade relacionada a um desses critérios (ou a um aspecto dele) pode o tornar mais relevante que outro para escolher a ferramenta e desenvolver o software de e-commerce. Como já exposto na Seção 2, a indústria de e-commerce é repleta de regras de negócio complexas, de estado em constante mudança e que exige respostas rápidas.

Os nuances do negócio, bem como os nuances de desenvolvimento do software e integração da plataforma com outros serviços ou empresas; a estrutura e metodologia de trabalho da equipe responsável pelo desenvolvimento dessa plataforma — incluindo não apenas os desenvolvedores, mas também a equipe de testes e os responsáveis por entender e transmitir as demandas aos desenvolvedores; são todos fatores que podem afetar diretamente a relevância de cada um dos aspectos desses critérios.

Esclarecido o fato de que a relevância desses critérios é mutável, principalmente para o escopo do comércio eletrônico, seguem abaixo suas descrições.

5.1 Funcionalidade

O critério da funcionalidade diz respeito a o quão útil a ferramenta será para os desenvolvedores em termos de oferecimento de recursos e funcionalidades do escopo de integração, entrega e implantação contínuos do projeto. A operacionalidade de recursos básicos como execução de construção — seja ela baseada em sondagem ou eventos -, integração com o sistema de controle de versão utilizado pela equipe (bem como com a plataforma de desenvolvimento colaborativo) e compilação e apresentação do feedback de maneira rápida, transparente, acessível e rastreável são aspectos fundamentais deste critério de comparação [4].

Aspectos relacionados à presença de recursos de segurança, gerenciamento funcional e prático de construções e pipelines e possibilidade de estender as funcionalidades da ferramenta — através de plugins, por exemplo — também devem ser indubitavelmente considerados no que diz respeito ao critério da funcionalidade [4].

5.2 Compatibilidade

A compatibilidade é o critério de comparação que define o quão bem a ferramenta se integra e interage com os componentes do projeto. A suportabilidade da configuração de construção do software, a exigência — ou não exigência — de instalação de softwares para que a ferramenta funcione como o esperado bem como a sinergia dela com a linguagem de

programação e tecnologias escolhidas pela equipe de desenvolvimento são aspectos chave deste critério [4].

5.3 Confiabilidade

A confiabilidade se refere a o quão sólida e polida é a ferramenta, como aplicação, no sentido de entregar os resultados corretos e, no caso de problemas, o quão solucionáveis são esses problemas — tanto via suporte da comunidade quanto via suporte da empresa gestora da ferramenta, se houver. Uma boa reputação, longo período de existência no mercado e uma comunidade e/ou assistência forte — que facilita encontrar soluções e respostas sobre dúvidas acerca do uso da ferramenta — são os aspectos principais da confiabilidade [4].

5.4 Longevidade

O critério da longevidade diz respeito a idade da ferramenta: o quão distante no passado se está da época em que a ferramenta terá deixado de ser utilizada. Apesar do fato de que elaborar uma estimativa dessas não é óbvio para a maioria das ferramentas, faz sentido que ferramentas mantidas pela comunidade — isto é, de código aberto — tenham chance de apresentar uma longevidade maior [4].

Uma ferramenta popular e gerenciada por uma grande e ativa comunidade terá menos chances de ser encerrada do que uma mantida por uma empresa privada, a qual está sujeita a fatores que podem levar ao seu encerramento, como mudanças nas prioridades da empresa, reestruturações, fusões ou até mesmo problemas financeiros [4].

5.5 Usabilidade

A usabilidade descreve o quão fácil é a configuração e uso atrelados à ferramenta. Fatores como tempo para ajustar a configuração inicial, facilidade em integrar a ferramenta com o projeto e o quão intuitivos são os conceitos, documentação e operacionabilidade dela — incluindo praticidade e intuitividade da interface de usuário — são aspectos fundamentais da usabilidade [4].

5.6 Custo

Este critério, como o nome sugere, diz respeito aos custos monetários atrelados à licença e/ou utilização da ferramenta. O fator que descreve se ela é de código aberto — comunitária, gratuita — ou se é atrelada a planos pagos, bem como se há uma versão de teste e se esses planos fornecem e recursos de acordo com o esperado pelos padrões de mercado, dado seu custo, define o aspecto principal deste critério [4].

6 Análise Comparativa entre as Ferramentas

A seguir será apresentada uma tabela que estabelece uma comparação matricial entre as ferramentas CI/CD mencionadas na Seção 3. Nas colunas estão os nomes das ferramentas enquanto nas linhas são apresentados os aspectos baseados nos critérios mencionados na Seção 4.

Cada aspecto nas linhas está vinculado a um dos critérios de comparação. Considere a sequência de 1 a 6 os respectivos índices para os critérios de funcionalidade, compatibilidade, confiabilidade, longevidade, usabilidade e custo. A tabela será preenchida com base no que foi apresentado nas descrições das ferramentas na Seção 3.

	Jenkins	GitHub Actions	Azure Pipelines	GitLab CI/CD	BitBucket Pipelines	CircleCI	TeamCity	Travis CI
Funcionalidades básicas (1)	★★☆☆☆	★★★★★	★★★★★	★★★★★	★★★☆☆	★★★★★	★★★★★	★★★★★
Integração com GitHub, BitBucket e outros (1)	✓	✓ exclusivo GitHub	✓	✓ exclusivo GitLab	✓ exclusivo BitBucket	✓	✓	✓
Feedback rápido, monitorável e informativo (1)	★★★★★	★★★★★	★★★★☆	★★★★☆	★★★★★	★★★★★	★★★★★	★★★★★
Extensibilidade (1)	★★★★★	★★★★★	★★★★☆	★★★☆☆	★★★☆☆	★★★★★	★★★★★	★★☆☆☆
Multi-linguagem (2)	✓	✓	✓	✓	✓	✓	✓	✓
Independe de instalações de software (2)	✗	✓	✓	✓	✓	✓	✗	✓
Polimento (3)	★★★★☆	★★★★★	★★★☆☆	★★★★☆	★★★★★	★★★★☆	★★★★☆	★★★★☆
Performance (3)	depende do host	★★★★☆	★★★☆☆	★★☆☆☆	★★★★☆	★★★★★	depende do host	★★★★☆
Suporte da comunidade ou empresa (3)	★★★★★	★★★★☆	★★★★★	★★★☆☆	★★★☆☆	★★★★★	★★★★★	★★★☆☆
Código aberto (4)	✓	✓	✗	✓	✗	✗	✗	✗
Idade no mercado (4)	19	3	5	7	7	12	17	12
Facilidade de configuração inicial (5)	★★☆☆☆	★★★★★	★★★★★	★★★★★	★★☆☆☆	★★★★★	★★☆☆☆	★★★★★
Qualidade da documentação (5)	★★★☆☆	★★★★★	★★★☆☆	★★★☆☆	★★★★☆	★★★★☆	★★★☆☆	★★★★★
Intuitividade e integridade da UI (5)	★★★☆☆	★★★★★	★★★☆☆	★★★☆☆	★★★★☆	★★★★★	★★★☆☆	★★★☆☆
Gratuidade (6)	Total	Total em repos. públicos	Oferece plano gratuito	Oferece plano gratuito	Oferece plano gratuito	Oferece plano gratuito	Oferece plano gratuito	Nenhuma
Custo vs. recursos oferecidos (6)	★★★★★	★★★★★	★★★☆☆	★★★☆☆	★★★★☆	★★★★☆	★★★★☆	★★☆☆☆

Ao analisar a tabela, podemos inferir que há grande diversidade no que diz respeito aos valores associados a cada um dos diferentes aspectos dos critérios de comparação entre ferramentas. Algumas ferramentas se apresentam mais extensíveis e customizáveis do que outras, enquanto algumas são mais fáceis de se configurar e integrar à aplicação, por exemplo. Entretanto, como já lembrado no início desta seção, a relevância de cada critério em relação aos outros pode variar conforme o contexto da aplicação a ser desenvolvida, e para o escopo de plataformas de e-commerce, isso é particular e fundamentalmente crucial.

Suponha um empreendimento de pequeno ou médio porte que deseja entrar para o ramo de e-commerce pela primeira vez e não deseja, pelo menos por enquanto, arriscar um grande investimento no setor, pretendendo executar, a princípio, uma abordagem mais experimental. Nesse caso, uma ferramenta de baixo custo como Jenkins, por exemplo, poderia ser uma escolha que condiz mais com a situação do cliente, dado que a empresa de desenvolvimento de software contratada esteja disposta a trabalhar com tal ferramenta. O critério de custo, nessa situação, se mostra mais relevante para o cliente empreendedor.

Em outra instância, suponha uma empresa de grande porte que já atue no e-commerce, trabalhe com grandes demandas, esteja atrelada a complexos fluxos de integração com mi-crosserviços externos, sistemas de pagamento, logística, ERPs e outros e ainda se preocupe vigorosamente em acompanhar as tendências de mercado no que diz respeito aos aspectos e funcionalidades de sua plataforma. Essa plataforma deveria funcionar sob a execução de uma ferramenta de CI/CD escalável, confiável e, sobretudo, veloz para suportar a execução de múltiplas tarefas em simultâneo com o mínimo de gargalos possível. O critério de confiabilidade e desempenho, para esse caso, se mostra mais relevante para o negócio.

Cada negócio tem suas exigências, tendências de mudança, urgências, demandas de mercado. Para a elaboração de uma plataforma de e-commerce, uma análise cuidadosa e uma compreensão clara das prioridades, mecanismos e metas do negócio por parte do empreendedor e da equipe de desenvolvimento são essenciais para fazer a escolha mais adequada de ferramenta CI/CD.

7 Conclusão

Neste relatório foram abordados os conceitos de integração, entrega e implantação contínuos, bem como os fundamentos e atualidades de e-commerce, a fim de contextualizar a análise e comparação entre as oito ferramentas de CI/CD escolhidas para o estudo. A finalidade deste relatório, como citado no resumo, foi não só apresentar essas ferramentas e analisar comparativamente seus aspectos, mas também argumentar sobre como a metodologia de CI/CD é vital para a existência da indústria do comércio eletrônico.

Com os avanços tecnológicos acelerando o mundo a cada ano que passa, o mercado do e-commerce tende a ficar cada vez mais exigente e competitivo. Mais do que em uma aplicação comum, as metodologias de CI/CD e a utilização dessas ferramentas para gestão e desenvolvimento das plataformas de vendas são o que as tornam viáveis e competitivas, pois as permitem responder ao já mencionado desafio fundamental da indústria: manter-se atualizado em relação às tendências do mercado e necessidades do negócio.

O que concluímos do que aqui foi apresentado, portanto, é que dada a complexidade e especificidade de cada empreendimento, o estado de mercado em constante fluxo de mudança e a exigência de respostas rápidas por parte do empreendimento e sua natureza, não há algo como a “ferramenta genérica perfeita”.

Por mais que as ferramentas CI/CD tenham como princípio serem esse tipo de aplicação abrangente e válida para qualquer cenário de qualquer magnitude, cada uma delas apresenta diferentes valores e particularidades para os mesmos aspectos, de tal forma que uma ferramenta pode acabar sendo uma escolha melhor do que outra ao considerar-se os critérios priorizados pelo cliente e seu negócio.

Referências

- [1] HUMBLE, Jez. FARLEY, David. Chapter 3, Continuous Integration. In: HUMBLE, FARLEY, **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**. 2010. p. 55-82.
- [2] HUMBLE, Jez. FARLEY, David. Chapter 5, Anatomy of the Deployment Pipeline. In: HUMBLE, FARLEY, **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**. 2010. p. 106-141.
- [3] TAMILARASI, R. ELAMATHI, N. **E-COMMERCE- BUSINESS- TECHNOLOGY- SOCIETY**. International Journal of Engineering Technologies and Management Research, 4(10). 2017. p. 33-41.
- [4] POLKHOVSKIY, Denis. **COMPARISON BETWEEN CONTINUOUS INTEGRATION TOOLS**. Tampere University of Technology. 2016. p. 32-45.
- [5] **Política de privacidade e-commerce: por que você deve criar a sua?** <https://www.agenciaeplus.com.br/politica-de-privacidade-e-commerce-por-que-voce-deve-criar-a-sua>, acessado pela última vez em 07/07/2023.
- [6] **Os desafios e o futuro do e-commerce** <https://www.logweb.com.br/colunas/os-desafios-e-o-futuro-do-e-commerce>, acessado pela última vez em 07/07/2023.
- [7] **Especialistas apontam os desafios do e-commerce para 2023** <https://empreendedor.com.br/noticia/especialistas-apontam-os-desafios-do-e-commerce-para-2023>, acessado pela última vez em 07/07/2023.
- [8] **Tendências de e-commerce 2023: prepare o seu negócio on-line** <https://stoom.com.br/tendencias-do-e-commerce-2023>, acessado pela última vez em 04/07/2023.
- [9] **Best CI/CD Tools for DevOps: A Review of the Top 10** <https://bluelight.co/blog/best-ci-cd-tools>, acessado pela última vez em 14/07/2023.
- [10] **Lessons Learned from Enterprise Usage of GitHub Actions** <https://www.infoq.com/articles/enterprise-github-actions>, acessado pela última vez em 14/07/2023.
- [11] **Jenkins** <https://www.jenkins.io>, acessado pela última vez em 14/07/2023.
- [12] **GitHub Actions** <https://github.com/features/actions>, acessado pela última vez em 14/07/2023.
- [13] **Azure Pipelines** <https://azure.microsoft.com/en-us/products/devops/pipelines>, acessado pela última vez em 14/07/2023.
- [14] **GitLab CI/CD** <https://docs.gitlab.com/ee/ci>, acessado pela última vez em 14/07/2023.

- [15] **BitBucket Pipelines** <https://bitbucket.org/product/features/pipelines>, acessado pela última vez em 14/07/2023.
- [16] **CircleCI** <https://circleci.com>, acessado pela última vez em 14/07/2023.
- [17] **TeamCity** <https://www.jetbrains.com/teamcity>, acessado pela última vez em 14/07/2023.
- [18] **Travis CI** <https://www.travis-ci.com>, acessado pela última vez em 14/07/2023.