



DMI - ESTUDO TÉCNICO PRELIMINAR - AMD 71/2023

Brasília, 01 de abril de 2024.

1. DEFINIÇÃO E ESPECIFICAÇÃO DAS NECESSIDADES E REQUISITOS (ART. 12, INC. I)

Nesta seção, são definidos e especificados as necessidades de negócio e tecnológicas, os requisitos necessários e suficientes à escolha da solução de TI, contendo de forma detalhada, motivada e justificada, inclusive quanto à forma de cálculo, o quantitativo de bens e serviços necessários para a sua composição.

1.1. Descrição geral da necessidade

A Câmara Legislativa do Distrito Federal (CLDF) possui a necessidade de melhorar a produtividade na produção de software e aplicar ferramentas técnicas e de inteligência artificial para apoiar a inovação e modernização do processo legislativo e de fiscalização, conforme especificado no DFD 1555024.

Uma das principais abordagens para alcançar esses objetivos é a técnica de DevOps. Trata-se de uma metodologia na engenharia de software que visa integrar as equipes de desenvolvimento (Dev) e operações (Ops) para otimizar o ciclo de vida do desenvolvimento de software. Esta metodologia promove a colaboração, comunicação e automação entre as equipes, resultando em entregas mais rápidas, confiáveis e de alta qualidade.

A adoção de práticas ágeis e DevOps é fundamental para modernizar o desenvolvimento e a entrega de software no setor público, promovendo agilidade, colaboração e qualidade. A CLDF já possui iniciativas de desenvolvimento interno e através de fábricas de software terceirizadas que adotam essas abordagens. Além disso, no segundo semestre de 2024, uma nova fábrica de software será contratada, o que aumentará o paralelismo de equipes ágeis e o volume de entregas. Para suportar as atividades de planejamento, integração e entrega contínua, é necessária uma plataforma adequada para gerenciar todas essas atividades e suas respectivas etapas.

1.2. Motivação e justificativa

Dado o contexto da necessidade, é possível segmentar as soluções em duas categorias principais: plataformas de DevOps em nuvem e soluções on-premise.

- **Plataformas de DevOps em nuvem:**

Vantagens: Oferecem escalabilidade, flexibilidade e acesso a uma ampla gama de serviços gerenciados, reduzindo a necessidade de gerenciamento de infraestrutura.

Desvantagens: Podem gerar preocupações com soberania de dados, segurança, conformidade e custos a longo prazo.

Desafios: Garantir a segurança dos dados e a conformidade com regulamentações específicas, além de gerenciar os custos contínuos.

- **Soluções on-premise:**

Vantagens: Garantem maior controle sobre dados e infraestrutura, atendendo a requisitos de segurança e conformidade mais rigorosos, além de permitir personalização e integração com sistemas legados.

Desvantagens: Exigem investimentos significativos em hardware, software e equipe especializada.

Desafios: Gerenciar a infraestrutura física, garantir a atualização constante dos sistemas e manter uma equipe técnica qualificada.

Além dessa segmentação, há uma divisão entre a utilização de uma cadeia de ferramentas (toolchain) e uma plataforma integrada:

- **Cadeia de ferramentas (toolchain):**

Vantagens: Oferece flexibilidade na escolha de ferramentas, permitindo que cada uma seja a melhor em sua função específica.

Desvantagens: A integração e o gerenciamento podem ser complexos, gerando possíveis gargalos e ineficiências.

Desafios: Garantir a integração perfeita entre ferramentas diversas e manter a consistência e a comunicação eficaz entre elas.

- **Plataforma integrada:**

Vantagens: Reúne em um único ambiente todas as ferramentas e funcionalidades necessárias para desenvolvimento, teste, implantação e monitoramento de software, simplificando a integração, automatizando processos e melhorando a colaboração e visibilidade.

Desvantagens: Pode ser menos flexível na personalização de ferramentas específicas.

Desafios: Encontrar uma plataforma que atenda a todas as necessidades específicas da organização e garantir a sua adoção e uso eficaz por todas as equipes.

De acordo com o [Gartner](https://www.atlassian.com/gartner/magic-quadrant-devops) (<https://www.atlassian.com/gartner/magic-quadrant-devops>), plataformas DevOps são aquelas que fornecem recursos totalmente integrados para habilitar a entrega contínua de software usando práticas DevOps e Agile. Esses recursos abrangem todo o ciclo de vida do desenvolvimento de software, incluindo planejamento de produto, controle de versão, integração contínua, automação de teste, implantação contínua, orquestração de lançamento, políticas de segurança e conformidade automatizadas, monitoramento e capacidade de observação. Essas plataformas suportam colaboração, desenvolvimento seguro e aferição de métricas para entrega de software, proporcionando uma visão unificada e uma gestão eficaz do ciclo de vida do desenvolvimento.

Nessa linha, observa-se que as plataformas DevOps simplificam a criação, manutenção e gerenciamento dos componentes necessários para a entrega de aplicações de software modernas. A integração pré-construída entre diferentes componentes da plataforma reduz a carga cognitiva e proporciona melhor visibilidade, auditabilidade e rastreabilidade no fluxo de valor do desenvolvimento de software. Essa visão de ponta a ponta incentiva um pensamento sistêmico e acelera os ciclos de feedback.

Organizações utilizam plataformas DevOps para minimizar o atrito causado por cadeias de ferramentas complexas, transferências manuais e falta de visibilidade consistente ao longo do ciclo de vida do desenvolvimento de software. Ao superar essas dificuldades, as equipes de produto conseguem entregar valor ao cliente mais rapidamente, sem comprometer a qualidade. O mercado de plataformas DevOps reflete a consolidação de tecnologias em desenvolvimento, segurança, infraestrutura e operações, otimizando a entrega de software.

Atualmente, a Diretoria de Modernização e Inovação Digital não utiliza uma plataforma DevOps integrada, ao contrário, conta com um conjunto de ferramentas encadeadas para o planejamento, a construção e a entrega de software de forma fracamente acoplada, conforme lista a seguir:

- Redmine para rastreamento de issues e gerenciamento de projetos
- GitLab Enterprise Edition, versão 16.9, para rastreamento de issues, versionamento de código no repositório git e armazenamento de helm charts.
- GitHub onde se hospedou os códigos entregues pela fábrica de software enquanto não se implantou o

gitlab. Em um segundo momento, o github foi utilizado também para espelhar o repositório do gitlab.

- Jenkins, versão 2.401.2 para integração contínua, em que se compila o código, se executam os testes automatizados, se envia o software para análise pelo SonarQube e se registra o artefato na ferramenta apropriada; e para entrega contínua, em que se envia para o ambiente de homologação, teste de qualidade, treinamento ou produção.
- JFrog Artifactory Open Source, versão 7.71.2, para armazenamento das bibliotecas de dependência dos projetos que utilizam Maven
- SonarQube Community Edition, versão 7.9.1, para análise estática, qualidade de código e segurança.
- Sonatype Nexus Repository Manager, versão OSS 3.25.1-04, para armazenagem de bibliotecas de dependência dos projetos que utilizam NPM, bem como para armazenagem de imagens Docker
- Harbor, versão v2.10.2-1a741cb7, para varrer e analisar as imagens Docker em busca de vulnerabilidades.
- ArgoCD, versão v2.10.11+27e49f8, para entrega contínua no ambiente Kubernetes de forma declarativa.

Para superar as dificuldades supracitadas do cenário de cadeia de ferramentas fracamente acopladas e para atender as iniciativas de desenvolvimento interno e com auxílio da nova fábrica de software que adotará metodologias ágeis, se faz necessário o estudo comparativo de soluções de DevOps à luz dos seguintes requisitos:

Requisito		Justificativa
1	Executar pipelines de CI/CD no datacenter da CLDF	Os custos são mais previsíveis ao se utilizarem os recursos de processamento, armazenamento e rede da instituição pública. É possível personalizar o ambiente e as ferramentas de acordo com a necessidade das aplicações construídas e implantadas. Há maior controle sobre ambiente, hardware, software e configurações. Por fim, a execução de pipelines <i>on-premise</i> resulta em menor latência e em melhor desempenho, principalmente porque, neste primeiro momento, as implantações também ocorrerão em ambiente local.
2	Armazenar artefatos, dependências e imagens no <i>storage</i> da CLDF	Os custos são mais previsíveis ao se utilizar recursos de armazenamento e rede da instituição pública. A soberania das informações é garantida com o armazenamento local. Por haver mais controle, pode-se personalizar o ambiente e as ferramentas para alcançar os níveis de controle de acesso, de disponibilidade e de latência necessários.
3	Prover forte acoplamento e integração de informações entre as etapas do ciclo DevOps em uma plataforma integrada	O objetivo é a simplificação da integração, automatização de processos, melhoria da colaboração e da visibilidade, para acelerar o ciclo de vida do desenvolvimento e reduzir o risco de erros.
4	Planejar funcionalidades, defeitos, tarefas, sprints, <i>roadmap</i> , épicos e backlog, de forma produtiva e com boa usabilidade	A metodologia Scrum aplicada ao desenvolvimento de software requer que o planejamento ocorra de forma organizada, com visibilidade do trabalho, colaboração entre equipes e adaptação contínua. O SEASI contratará uma fábrica de software que utilizará Scrum com até quatro equipes simultâneas. Isso exigirá alta produtividade e controle acurado no planejamento das sprints por parte dos <i>product owners</i> , dos fiscais e do gestor do contrato. Também será necessário o rastreamento dos itens de planejamento com as entregas, para aferir a qualidade e completude das entregas simultâneas. Por essa razão, a atividade de planejamento numa plataforma DevOps, com boa usabilidade e alta produtividade, é crítica para o sucesso dos projetos entregues pela futura fábrica de software.

5	Versionar código em repositório Git, com suporte a revisão por pares e merge de <i>branches</i>	Além de ser um padrão de mercado, amplamente suportado pelas ferramentas de desenvolvimento, a utilização do Git permite a criação de <i>feature branches</i> , colaboração iterativa e incremental, versionamento, revisão de novas implementações que serão incorporadas ao repositório principal. Isso é necessário para o desenvolvimento concorrente de diferentes histórias de usuário e aprovação de incorporação de mudanças em repositórios de ambientes de produção ou de pré-produção.
6	Documentar o projeto por meio de wiki	Necessário para edição colaborativa e concorrente de documentação do projeto, com a utilização de uma sintaxe simples e intuitiva e com manutenção de histórico. Isso promove a transparência e a comunicação entre os membros da equipe, o que evita mal-entendidos e retrabalho. O wiki se torna um repositório de conhecimento da equipe, que permite a integração de novos membros ao projeto e a preservação de conhecimento, mesmo com a saída de membros da equipe. Essa resiliência à mudança no time é necessária para o cenário de fábrica de software, em que pode haver rotatividade dos colaboradores da contratada.
7	Registrar artefatos e imagens, com varredura de vulnerabilidade	Necessário para rastrear as mudanças ao longo do tempo, identificar versões e reverter para versões anteriores se necessário, bem como para garantir a consistência e reprodutibilidade de comportamento do software em ambientes de produção e pré-produção. A varredura de dependências e de imagens de container, em busca de vulnerabilidades, é requerida para retroalimentar o ciclo de desenvolvimento com medidas de mitigação do risco de exploração dessa falha e para evitar que novas implantações sejam realizadas com imagens que possuam um nível elevado de risco.
8	Automatizar testes de software, por meio de suporte para execução de testes funcionais e não funcionais, gerenciamento de casos de teste, análise de cobertura de código, testes de desempenho, testes de caos, testes de <i>fuzz</i> , testes de penetração e testes de aceitação automatizados	A equipe de gestão do contrato de fábrica de software necessitará de instrumentos para aferir a qualidade da entrega. Para alcançar tal objetivo, é necessário planejar testes, automatizar grande parte deles, executar manualmente outros, compilar resultados e relatórios de teste, para retroalimentar o ciclo de desenvolvimento com correções e melhorias, ou mesmo para acionar mecanismos contratuais de glosa, notificação ou sanção pelo descumprimento de requisitos de qualidade.
9	Integrar continuamente, com suporte nativo para construção contínua de código, orquestrando funções de verificação e validação (automação de testes, varreduras de segurança e conformidade)	A integração contínua (CI) é necessária para identificar rapidamente erros e conflitos no código, para evitar o acúmulo de problemas, para economizar tempo e recursos com a resolução de dificuldades prematuramente, para fornecer feedback rápido sobre a qualidade do código por meio de testes automatizados, para manter o software em estado funcional e pronto para ser implantado, para realizar mudanças pequenas, frequentes, incrementais e testadas, para incentivar a colaboração do time e evitar conflitos, para aumentar a produtividade dos desenvolvedores com a automação de atividades manuais e repetitivas.

10	Entregar continuamente e orquestrar lançamento	A entrega contínua (CD) é necessária para automatizar o processo de implantação, para entregar novas funcionalidades e correções para os usuários de forma rápida e frequente, para obter o feedback dos usuários de forma iterativa e incremental, para reduzir o risco de falhas na implantação uma vez que ela é automatizada, repetível e testada em ambiente de pré-produção, para promover uma cultura de qualidade que exige que o software sempre esteja pronto para ser implantado, para aumentar a produtividade da equipe de operação de infraestrutura com a automação de atividades manuais e repetitivas.
11	Instalação e configuração	Uma solução <i>on-premise</i> de DevOps pode requerer instalação e configuração complexas. Caso essas atividades fiquem exclusivamente sob responsabilidade da equipe da DMI, podem ocorrer equívocos, retrabalho, dilatação do prazo para início da operação e consumo excessivo de recursos computacionais. É necessário que tal processo seja assistido por um profissional capacitado para realizar a instalação e configuração de forma eficiente e correta.
12	Capacitação para administração e uso da plataforma de DevOps	Após a instalação e a configuração, a plataforma de DevOps será administrada e operada pela equipe da DMI. Rotinas de backup, gestão de usuários, permissionamento, gestão de projetos e repositórios e manutenção de pipelines serão realizadas pelos servidores da CLDF. Para que essas atividades sejam realizadas de forma produtiva, correta e consistente, é necessário que haja um nivelamento de conhecimentos e capacitação na administração e no uso da plataforma.

Embora a adoção de uma solução de DevOps seja essencial para melhorar a produtividade e modernizar o processo de desenvolvimento de software na CLDF, essa medida, por si só, não será suficiente para alcançar o objetivo de inovação no processo legislativo e de fiscalização. A implementação de DevOps e metodologias ágeis impulsiona a eficiência, a colaboração entre equipes e a rapidez nas entregas, mas é necessário um complemento com tecnologias de inteligência artificial (IA) generativa para potencializar os resultados esperados.

Investir em uma solução de IA generativa proporcionará à CLDF ferramentas avançadas para automação de tarefas, análise de grandes volumes de dados e suporte à tomada de decisões. Por exemplo, modelos de IA podem ser usados para criar assistentes virtuais capazes de responder a consultas legislativas, automatizar a redação de documentos legais, prever tendências e resultados legislativos com base em dados históricos, e monitorar o cumprimento de leis e regulamentações de maneira mais eficaz.

Atualmente, a CLDF está testando a capacidade de modelos on-premises, como o LLaMA3 8b. Embora esses modelos ofereçam um certo nível de controle e segurança sobre os dados, a infraestrutura necessária para manter e treinar tais modelos é complexa e custosa. Além disso, o processo de treinamento requer hardware especializado, como GPUs de alto desempenho, e uma equipe técnica altamente qualificada para gerenciar e otimizar os modelos.

Estamos avaliando se a melhor estratégia para integrar serviços de IA às aplicações da CLDF é utilizar serviços de nuvem via API, como os oferecidos pela OpenAI, Google Gemini, ou se manter uma infraestrutura on-premises será mais eficiente, ou até mesmo adotar uma abordagem híbrida. Esta abordagem híbrida poderia utilizar modelos open-source on-premises para tarefas mais básicas e os modelos de ponta providos pela nuvem para tarefas mais complexas. Para não afetar o planejamento de implantação da nova fábrica de software, esse investimento será detalhado em um estudo técnico preliminar e termos de referências separados. Essa abordagem permitirá uma implantação mais estratégica e planejada, garantindo que todas as necessidades tecnológicas da CLDF sejam atendidas de forma

coordenada e eficaz.

Portanto, esse Estudo Técnico Preliminar focará em avaliar a solução de DevOps mais vantajosa para a CLDF a fim de apoiar o seu processo de desenvolvimento e entrega de software na CLDF.

1.3. Alinhamento estratégico

O objeto desta contratação está em consonância com o Plano Diretor de Tecnologia da Informação (PDTI) 2024-2025, conforme tabela abaixo:

OBJ-5 - Prover sustentação computacional			
OBJ-5.1 - Garantir sustentação e funcionamento do complexo computacional			
NEC	Declarante	Necessidade	Função institucional
5.1.21	Diretoria de Modernização e Inovação Digital - DMI	Garantir a sustentação e o pleno funcionamento dos sistemas de informática relativos às funções institucionais de administração, representação, fiscalização e legislação nos níveis adequados às necessidades da Câmara Legislativa do DF.	Representação, Legislação, Fiscalização, Administração (operação chave) (++) cm,tm, Visão: A a H

2. ANÁLISE COMPARATIVA DAS SOLUÇÕES (ART. 12, INC. II)

2.1. Necessidades similares em outros órgãos ou entidades da Administração Pública e as soluções adotadas

A solução Gitlab é adotada nos seguintes órgãos:

1. Ministério Público do Distrito Federal e Territórios
2. Supremo Tribunal Federal
3. Tribunal de Justiça do Estado do Rio Grande do Sul
4. Assembleia Legislativa do Estado de Minas Gerais
5. Defensoria Pública do Estado do Rio Grande do Sul
6. Tribunal de Justiça de Rondônia

A solução Atlassian Jira é adotada nos seguintes órgãos:

1. Tribunal Regional Eleitoral no Espírito Santo
2. Tribunal Superior Eleitoral
3. Tribunal Superior do Trabalho
4. Ministério Público do Estado do Mato Grosso
5. Secretaria de Estado de Administração Penitenciária do Distrito Federal
6. Tribunal Regional do Trabalho da 6ª Região (PE)
7. Tribunal Regional do Trabalho da 17ª Região (ES)
8. Tribunal de Contas do Estado de Roraima
9. Polícia Militar do Estado de Santa Catarina
10. Tribunal de Contas do Estado de Rondônia

A solução Microsoft Azure DevOps é adotada nas seguintes entidades:

1. Conselho Regional de Engenharia e Agronomia de Santa Catarina
2. Agência Nacional de Energia Elétrica
3. Agência Nacional de Cinema
4. Tribunal de Contas do Município de São Paulo
5. Ministério da Infraestrutura
6. Agência Nacional do Petróleo, Gás Natural e Biocombustíveis

7. Empresa Brasileira de Serviços Hospitalares
8. Centro de Tecnologia de Informação e Comunicação do Estado do Rio de Janeiro
9. Ministério da Justiça e Segurança Pública

2.2. Alternativas do mercado

As plataformas de DevOps líderes de mercado, de acordo com o estudo “[Magic Quadrant for DevOps Platforms](#)”, de junho de 2023, elaborado pelo Gartner, são providas por Gitlab, Microsoft e Atlassian. As demais plataformas inspecionadas pela equipe de planejamento da contratação, de fato, não oferecem suporte a todas as atividades de DevOps requeridas pela área demandante ou ainda não estão maduras o suficiente para oferecer um suporte estável, confiável e íntegro.

Magic Quadrant

Figure 1: Magic Quadrant for DevOps Platforms



(Fonte: <https://www.atlassian.com/gartner/magic-quadrant-devops>)

Durante o primeiro semestre, o Setor de Administração de Sistemas realizou testes com as plataformas providas pela Gitlab e pela Microsoft. O objetivo foi confrontar os requisitos de negócio desse setor com as soluções, levando em conta a experiência pretérita dos servidores com outras

ferramentas, inclusive aquelas que vivenciadas nos órgãos de origem.

O teste com a solução Gitlab ocorreu com a versão Ultimate, que ficou disponível por 30 dias. O sistema de planejamento orçamentário utilizou essa plataforma principalmente para planejar, armazenar código e executar pipelines de integração contínua.

O teste com a solução da Microsoft ocorreu com a versão em nuvem, Azure DevOps Services, que permite utilizar todos os serviços com algumas restrições quantitativas ou temporais. Um projeto de desenvolvimento também foi exercitado nessa plataforma, com a execução de pipelines *on-premise*.

Dessa forma, as soluções candidatas são:

- a. Gitlab Premium (gitlab.com)
- b. Gitlab Premium (self-managed)
- c. Gitlab Ultimate (gitlab.com)
- d. Gitlab Ultimate (self-managed)
- e. Microsoft Azure DevOps Services
- f. Microsoft Azure DevOps Server
- g. Atlassian Jira Software Cloud Standard + Atlassian Bitbucket Cloud Standard
- h. Atlassian Jira Software Cloud Premium + Atlassian Bitbucket Cloud Premium
- i. Atlassian Jira Software Data Center + Atlassian Bitbucket Data Center

2.3. Políticas, os modelos e os padrões de governo

2.3.1. Padrões de Interoperabilidade de Governo Eletrônico - ePing

A solução deve ser adequada, no que couber, aos padrões de interoperabilidade do e-Ping, que pode ser acessado em <http://eping.governoeletronico.gov.br/>. A versão vigente do e-Ping traz as seguintes especificações técnicas que são aplicáveis a solução candidata:

Protocolo de transferência de hipertexto	HTTP/1.1
Transferência de dados em redes inseguras	TLS - Transport Layer Security
Algoritmo para transporte de chave criptográfica de conteúdo/sessão	RSA
Formato de intercâmbio de hipertexto	W3C XML versões 1.0 ou 1.1 (adotado) W3C HTML 5W3C HTML versão 4.01 (em transição) W3C XHTML versões 1.0 ou 1.1 (em transição)
Intercâmbio de informações gráficas e imagens estáticas	W3C PNGSVG (recomendado) JPEG File Interchange Format (recomendado)

Por se tratar de padrões consolidados e amplamente adotados pelo mercado, todas as alternativas cumprem esse requisito.

2.3.2. Modelo de Acessibilidade em Governo Eletrônico - eMag

Não aplicável.

2.3.3. Padrões Web em Governo Eletrônico - ePwg

Não aplicável.

2.3.4. Padrões de Design System de governo

Não aplicável.

2.3.5. Infraestrutura de Chaves Públicas Brasileira - ICP-Brasil

Não aplicável.

2.3.6. Modelo de Requisitos para Sistemas Informatizados de Gestão Arquivística de Documentos - e-ARQ Brasil

Não aplicável.

2.4. Necessidades de adequação do ambiente da CLDF para viabilizar a execução contratual

Todas as soluções candidatas oferecem o modelo *Software as a Service (SaaS)* em nuvem e o modelo de instalação no *datacenter* do cliente. Para o primeiro cenário de SaaS, a adequação do ambiente da CLDF é mínimo, uma vez que as soluções utilizam protocolos bem suportados por navegadores da instituição e pelas ferramentas de desenvolvimento. O que precisa se atentar é à possibilidade de implantar agentes de execução de pipelines disparados a partir do serviço em nuvem. Nesse sentido, a solução provida pela Atlassian oferece o Bitbucket Runner para Linux, Windows e MacOS; a solução da Gitlab também oferece Runners para Linux, MacOS, Windows e FreeBSD; por fim, a solução SaaS da Microsoft oferta agentes de execução local para Linux, Windows, MacOS e Docker.

Considerando o segundo cenário, qual seja, instalação da plataforma de DevOps *on-premise*, a adequação do ambiente da CLDF já é mais complexa. A solução provida pela Atlassian consiste de um *cluster* de componentes, cada um em uma máquina virtual Linux dedicada, com um balanceador de carga, três nodos de aplicação, três nodos mesh e um servidor de busca. Por sua vez, a solução Gitlab já está implantada no Kubernetes como um cluster de micro-serviços, o que reduz o esforço e o prazo de adoção, porém mantém internalizado o esforço de sustentação desse ambiente. Por fim, a plataforma de DevOps provida pela Microsoft requer ambiente Windows para ser instalada. Esse requisito pode ser impeditivo, dado o cenário de mudança, planejada para 2025, da arquitetura de servidores do *datacenter* da CLDF, que aumentará o requisito de licenças de Windows Server para cada máquina virtual instanciada nesse novo hardware.

2.5. Diferentes modelos de prestação do serviço

Como discorrido na seção anterior, as soluções candidatas oferecem o modelo de SaaS e *self-managed*. Em soluções de *software as a service*, a gestão do contrato deve ser diligente para instruir renovações do contrato ou para licitar novos fornecedores. Dessa forma, garantir a continuidade da prestação do serviço e a preservação dos dados armazenados em nuvem. Não há diferença entre as soluções candidatas nesse modelo de nuvem.

Quanto ao modelo *self-managed*, em que a plataforma de DevOps é instalada no *datacenter* da CLDF, há algumas diferenças entre as soluções candidatas. A Atlassian possuía um modelo de licenciamento perpétuo para o Bitbucket Server, no entanto, a empresa descontinuou o produto e oferece agora o Bitbucket Data Center que é uma subscrição anual. A Microsoft oferece o licenciamento do Azure DevOps Server, no modelo perpétuo com a *Software Assurance* paga de forma recorrente, conjugado com *Client Access Licenses (CAL)* de usuários pagas de forma periódica. Por fim, o modelo de licenciamento *on-premise* da Gitlab é anual, para o nível Premium ou para o Ultimate, sendo que o cenário de não renovação do licenciamento gera o impacto menos danoso, quando comparado às outras soluções *self-managed*. O Gitlab continua em operação, após a expiração do licenciamento, com as funcionalidades do nível gratuito. O planejamento ágil seria bastante impactado, mas os *issues* ainda estariam disponíveis para se trabalhar. O repositório de código e os *pipelines* seriam menos impactados do que as atividades de planejamento.

2.6. Diferentes tipos de soluções em termos de especificação, composição ou características dos

A solução da Gitlab oferece um conjunto de funcionalidades que abrangem todas as etapas do ciclo de vida do desenvolvimento de software. A principal vantagem dessa plataforma em relação aos demais é a possibilidade de registrar artefatos e registrar *containers* diretamente. A solução da Atlassian não oferece nenhum dos dois recursos, o que torna necessária uma solução externa. Por seu turno, a solução da Microsoft oferece o registro de artefatos, mas não o registro de *containers*. No ecossistema Microsoft Azure, existe um registro de *containers* que só faz sentido quando a aplicação é implantada na nuvem. Considerando o cenário de implantação de aplicações apenas no *datacenter* da CLDF, o registro de imagens de *container* deve ser local, não em nuvem. Sendo assim, para compatibilizar a execução local de *pipelines* e a política de implantação com as soluções candidatas que não oferecem ambos os recursos, seria necessário manter os serviços do JFrog Artifactory e do Harbor.

As subseções a seguir destringem outras características e composições de cada solução candidata.

	Microsoft Azure DevOps	Gitlab	Atlassian Jira + Bitbucket
Arquitetura e implantação	Disponível como serviço em nuvem (Azure DevOps Services) e on-premise (Azure DevOps Server). A versão em nuvem é gerenciada pela Microsoft, enquanto a versão on-premise requer infraestrutura própria. Integração nativa com outros serviços da Microsoft Azure.	Oferece opções de nuvem (GitLab.com), auto-hospedagem (GitLab self-managed) e uma versão híbrida (GitLab Dedicated). A versão auto-hospedada permite maior controle e personalização. Arquitetura monolítica, com todos os serviços integrados em uma única aplicação.	Jira e Bitbucket estão disponíveis em nuvem (Atlassian Cloud) e on-premise. A versão on-premise oferece maior controle, mas requer mais manutenção. Arquitetura modular, permitindo integração entre diferentes produtos Atlassian.
Controle de versão	Suporta Git e Team Foundation Version Control (TFVC). Oferece repositórios privados ilimitados.	Focado exclusivamente em Git. Fornece recursos avançados de Git, como merge requests e revisão de código integrada.	Suporta Git e Mercurial. Oferece pipelines de CI/CD integrados aos repositórios.
Gerenciamento de projetos e rastreamento de issues	Utiliza Azure Boards para gerenciamento de projetos. Suporta metodologias ágeis e tradicionais. Oferece recursos de planejamento de sprints e backlogs.	Inclui funcionalidades de gerenciamento de projetos e issues integradas. Suporta quadros Kanban e planejamento ágil. Oferece recursos de épicas para gerenciamento de alto nível.	Especializado em gerenciamento de projetos e rastreamento de issues. Altamente personalizável com diversos workflows e tipos de issues. Oferece recursos avançados para metodologias ágeis e tradicionais.

Integração contínua e entrega contínua	Utiliza Azure Pipelines para CI/CD. Oferece integração nativa com serviços Azure para implantação. Suporta containers e ambientes de nuvem variados.	CI/CD integrado nativamente à plataforma. Oferece recursos de DevOps como monitoramento e segurança integrados. Suporta implantação em várias plataformas de nuvem e on-premise.	CI/CD integrado ao Bitbucket através do Bitbucket Pipelines. Integração com ferramentas de terceiros para implantação e monitoramento. Menos recursos nativos de DevOps comparado às outras soluções.
Segurança e conformidade	Conformidade com padrões como ISO 27001, SOC 1 e SOC 2. Recursos avançados de segurança e governança da Microsoft. Integração com Azure Active Directory para controle de acesso.	Oferece recursos de segurança integrados, incluindo SAST, DAST e varredura de dependências. Conformidade com padrões como ISO 27001, SOC 2, e HIPAA. Suporte a Single Sign-On (SSO) e autenticação de dois fatores.	Conformidade com padrões como ISO 27001, SOC 2, e HIPAA. Recursos de segurança variam entre Jira e Bitbucket. Oferece controle de acesso baseado em funções e autenticação de dois fatores.
Escalabilidade e desempenho	Alta escalabilidade na nuvem Azure. Desempenho pode variar dependendo da localização geográfica dos servidores Azure.	Escalável tanto na nuvem quanto on-premise. Pode enfrentar desafios de gerenciamento em instalações muito grandes devido à sua arquitetura.	Escalável na nuvem Atlassian. A versão on-premise pode requerer configurações adicionais para alta escalabilidade.
Integrações e extensibilidade	Amplo marketplace de extensões. Integração nativa com ferramentas Microsoft e Azure.	Abordagem "tudo-em-um" com menos necessidade de integrações externas. Oferece APIs e webhooks para extensibilidade.	Ecosistema vasto de plugins e integrações através do Atlassian Marketplace. Forte integração entre produtos Atlassian.

2.7. Possibilidade de aquisição na forma de bens ou contratação como serviço

Todos os fabricantes de soluções candidatas oferecem o modelo Software as a Service (SaaS) em nuvem e o modelo de instalação no *datacenter* da CLDF (*self-managed*). As características dos licenciamentos foram abordadas na seção 2.5.

2.8. Ampliação ou substituição da solução implantada

A ampliação da solução em nuvem é imediata, desde que haja viabilidade contratual para se incrementar o número de usuários de forma *pró-rata*, no modelo *true-up* ou no modelo de *pay-as-you-go*, ou para celebrar termo aditivo que aumente o número de usuários até o limite legal. A ampliação da

solução *self-managed* requer a mesma análise de viabilidade contratual citada anteriormente, porém o ambiente computacional da CLDF pode exigir alguma ampliação para acomodar os novos usuários, o volume de dados que eles geram e consomem e a execução de pipelines que eles disparam. Nesse sentido, soluções em nuvem podem levar vantagem, mas ela é irrelevante para o tamanho da equipe que adotará a solução.

2.9. Diferentes métricas de prestação do serviço e de pagamento

As soluções em nuvem mensuram a quantidade de usuários como principal fator de precificação do serviço, mas também podem cobrar por armazenamento, tráfego de rede e tempo de processamento, principalmente se excederem limites ofertados para a categoria de nível de serviço contratado.

Na solução Gitlab Premium em nuvem, o limite de armazenamento para o repositório Git e para *Large File Storage (LFS)* é 10 GiB, já o limite para tempo de processamento é 10.000 minutos por mês para a versão Premium e 50.000 minutos por mês para a versão Ultimate. Especificamente, a versão Ultimate segrega os usuários entre usuários normais (pagos) e usuários convidados (gratuitos), sendo que esses últimos podem criar e atribuir *issues*, observar certas análises e observar código.

A solução Bitbucket Standard concede 2.500 minutos de pipeline por mês, 4GB como limite de armazenamento de repositórios Git e 5GB como limite global de armazenamento LFS. Já a subscrição Premium em nuvem amplia para 3.500 minutos o limite de pipeline por mês, mantém os mesmos 4GB de limite de armazenamento de repositórios Git e sobe para 10GB o limite global de armazenamento LFS.

Por fim, a solução da Microsoft em nuvem diferencia a cobrança de usuários de acordo com seu perfil. O perfil *stakeholder* pode ser concedido para usuários de forma gratuita e permite a criação e modificação de itens de trabalho, aprovação de pipelines de entrega, mas não tem acesso ao repositório de código. Usuário com perfil básico é pago e pode executar todas as funcionalidades, exceto aquelas exclusivas do perfil test plan. Usuário com perfil test plan também é pago e gerencia planos e casos de teste. Além dessa política de preços relacionada aos perfis de usuário, a Microsoft também cobra pelo armazenamento de artefatos, quando excedem os 2GB gratuitos, e pela execução de pipelines em nuvem ou locais. Em nuvem, são concedidos 1.800 minutos gratuitos por mês e cada execução concorrente de pipelines é cobrada. Localmente, é concedida a execução ilimitada de um pipeline e cada execução concorrente é cobrada.

Por outro lado, as soluções *on-premise* mensuram a quantidade de usuários como principal fator de valoração do serviço. Pelo fato de usarem recursos do parque computacional do cliente, não são mensurados e cobrados armazenamento, tráfego de rede e tempo de processamento.

2.10. Soluções consideradas inviáveis

Solução	Justificativa da inviabilidade
Gitlab Premium (gitlab.com)	Embora o GitLab Premium ofereça um conjunto robusto de recursos para DevOps, ele é insuficiente em áreas críticas para o gerenciamento do ciclo de vida de desenvolvimento de software da CLDF. A ausência de funcionalidades, como épicos multinível, épicos vinculados e gerenciamento de requisitos, limita a capacidade de planejar e acompanhar com eficácia iniciativas complexas e concorrentes da fábrica de software. Além disso, o GitLab Premium carece de recursos de garantia de qualidade, incluindo a capacidade de definir e gerenciar casos de teste dentro da plataforma.

Solução	Justificativa da inviabilidade
Gitlab Premium (self-managed)	Essa limitação dificulta a capacitação da equipe em processos rigorosos de controle de qualidade e de colaborar de forma eficiente com partes interessadas externas. Além disso, a incapacidade de marcar requisitos como atendidos a partir dos pipelines de CI/CD representa uma oportunidade perdida para automatizar e agilizar os processos de desenvolvimento. Tais lacunas de funcionalidades, que são críticas para as atividades de planejamento e interação com a fábrica de software, só são supridas pela versão Ultimate da Gitlab.
Microsoft Azure DevOps Server	Como discorrido na seção "Necessidades de adequação do ambiente da CLDF para viabilizar a execução contratual", está em fase de planejamento a mudança no ambiente computacional da CLDF, que alterará a arquitetura dos servidores físicos para otimizar a dissipação de calor. Tal mudança de arquitetura afetará o cálculo do licenciamento do sistema operacional Microsoft Windows Server, de sorte que o custo de novas máquinas virtuais Windows aumentará substancialmente. Dessa forma, é necessário uma otimização dos serviços para reduzir a quantidade de máquinas virtuais Windows requeridas no parque computacional. A solução Microsoft Azure DevOps Server é necessariamente instalada em sistema operacional Windows. Sendo assim, a equipe de planejamento da contratação entende que ela é inviável nesse cenário de redução de máquinas virtuais Windows Server por questões de otimização de custos dado o planejamento atual de aquisição de servidores.

A solução Atlassian Jira Software Cloud Standard + Atlassian Bitbucket Cloud Standard não permitem planejar e rastrear trabalho através de múltiplos projetos e equipes, nem rastrear lançamentos, também não é possível proteger o repositório de código com autenticação de dois fatores e o fabricante não garante SLA algum para seu *uptime*, o que poderia prejudicar a disponibilidade do serviço. Isso já seria suficiente para considerar essa solução como inviável, no entanto, a equipe de planejamento manteve a solução para demonstrar sua inviabilidade na etapa de comparação de custo.

3. ANÁLISE COMPARATIVA DE CUSTOS (ART. 12, INC. III)

Nesta seção, é realizada a análise comparativa de custos, que considera apenas as soluções técnicas e funcionalmente viáveis.

3.1. Cálculo dos custos totais de propriedade

Nesta subseção, o Total Cost of Ownership é realizado por meio da obtenção dos custos inerentes ao ciclo de vida dos bens e serviços de cada solução, a exemplo dos valores de aquisição dos ativos, insumos, garantia técnica estendida, manutenção, migração e treinamento. Para todas as soluções, será considerado que o custo da capacitação é idêntico e precificado de acordo com o valor unitário da hora-aula de instrutor externo, conforme AMD nº 79/2020.

Para comparação das soluções, foi considerado o valor publicado pelo fabricante nas páginas públicas da plataforma e convertido para moeda nacional pela cotação 1 USD = 5,50 BRL.

	Gitlab Ultimate (gitlab.com)	Gitlab Ultimate (self-managed)	Microsoft Azure DevOps Services	Atlassian Jira Software Cloud Standard + Atlassian Bitbucket Cloud Standard	Atlassian Jira Software Cloud Premium + Atlassian Bitbucket Cloud Premium	Atlassian Jira Software Data Center + Atlassian Bitbucket Data Center
Preço publicado pelo fabricante	USD 99 por usuário por mês	USD 99 por usuário por mês	Basic: USD 6 por usuário por mês Basic+Test Plan: USD 52 por usuário por mês Pipeline local: USD 15 por mês	Jira: USD 8,15 por usuário por mês Bitbucket: USD 3,00 por usuário por mês	Jira: USD 16 por usuário por mês Bitbucket: USD 6 por usuário por mês	Jira: USD 44.000 por ano para 500 usuários (não é comercializado em quantidade inferior) Bitbucket: USD 7.600 por ano para 100 usuários
Quantitativo de usuários e serviços	122 usuários pagos para devs +30 usuários guest para stakeholders	122 usuários pagos para devs +30 usuários guest para stakeholders	110 usuários Basic +12 usuários Test Plan +30 usuários stakeholder +11 execuções concorrentes de pipeline on-premise +3 execuções concorrentes de pipeline na nuvem	122 usuários Jira e Bitbucket para devs +30 usuários Jira para stakeholders	122 usuários Jira e Bitbucket para devs +30 usuários Jira para stakeholders	100 usuários Jira e Bitbucket para devs +22 usuários Jira para devs +30 usuários Jira para stakeholders +348 usuários Jira desperdiçados porque a licença datacenter é cotada em lote mínimo de 500 usuários
Custo anual em reais	R\$ 797.148	R\$ 797.148	R\$ 103.554	R\$ 105.917	R\$ 208.824	R\$ 283.800
Capacitação no início do contrato	R\$ 3.702					
Custo total para 36 meses	R\$ 2.395.146	R\$ 2.395.146	R\$ 314.364	R\$ 321.452	R\$ 630.174	R\$ 855.102

3.2. Memória de cálculo que referenceie os preços e os custos utilizados na análise, com vistas a

permitir a verificação da origem dos dados

3.2.1. Gitlab Ultimate (gitlab.com) e Gitlab Ultimate (self-managed)

Calculadora oficial do fabricante disponível em: <https://about.gitlab.com/pricing/ultimate/>

Custo anual calculado como: [(122 usuários pagos) * (\$ 99 por usuário por mês) + (30 usuários guest) * (\$ 0 por usuário por mês)] * (12 meses) * (5,50 de taxa de conversão dólar-real)

3.2.2. Microsoft Azure DevOps Services

Calculadora oficial do fabricante disponível em: <https://azure.microsoft.com/en-us/pricing/details/devops/azure-devops-services/>

Custo anual calculado como: [(110 usuários Basic - 5 usuários gratuitos) * (\$ 6 por usuário por mês) + (12 usuários test plan) * (\$ 52 por usuário por mês) + (30 usuários stakeholder) * (\$ 0 por usuário por mês) + (11 execuções concorrentes de pipelines on-premise - 1 execução gratuita) * (\$ 15 por execução concorrente por mês) + (3 execuções concorrentes de pipeline na nuvem) * (\$ 40 por execução concorrente por mês)] * (12 meses) * (5,50 de taxa de conversão dólar-real)

3.2.3. Atlassian Jira Software Cloud Standard + Atlassian Bitbucket Cloud Standard

Calculadora oficial do fabricante disponível em: <https://www.atlassian.com/software/jira/pricing> e <https://www.atlassian.com/software/bitbucket/pricing?tab=cloud-tab>

Custo anual calculado como: ((122 usuários) * (\$ 8,15 por usuário Jira + \$ 3,00 por usuário Bitbucket) + (30 usuários Jira para stakeholders) * (\$ 8,15 por usuário Jira)) * (12 meses) * (5,50 de taxa de conversão dólar-real)

3.2.4. Atlassian Jira Software Cloud Premium + Atlassian Bitbucket Cloud Premium

Calculadora oficial do fabricante disponível em: <https://www.atlassian.com/software/jira/pricing> e <https://www.atlassian.com/software/bitbucket/pricing?tab=cloud-tab>

Custo anual calculado como: ((122 usuários) * (\$ 16 por usuário Jira + \$ 6 por usuário Bitbucket) + (30 usuários Jira para stakeholders) * (\$ 16 por usuário Jira)) * (12 meses) * (5,50 de taxa de conversão dólar-real)

3.2.5. Atlassian Jira Software Data Center + Atlassian Bitbucket Data Center

Calculadora oficial do fabricante disponível em: <https://www.atlassian.com/software/jira/pricing> e <https://www.atlassian.com/software/bitbucket/pricing?tab=self-manageddata-center>

Custo anual calculado como: [(\$ 44.000 por ano para 500 usuários Jira) + (\$ 7.600 por ano para 100 usuários Bitbucket)] * (5,50 de taxa de conversão dólar-real)

3.2.6. Capacitação

Custo da capacitação baseado no AMD nº 79/2020, que estabelece a remuneração por hora-aula do instrutor externo igual a 1,5% do vencimento básico da carreira de Consultor Técnico Legislativo, combinado com a Lei Distrital nº 7515/2024.

Memória de cálculo: 1,5% * (R\$ 20.566,52 do vencimento básico) * (12 horas-aula)

4. ESTIMATIVA DO CUSTO TOTAL DA CONTRATAÇÃO (ART. 12, INC. IV)

A solução Microsoft Azure DevOps Services demonstrou atender os requisitos de negócio e que possui o menor custo total de propriedade. A seguinte tabela sumariza os quantitativos, custos unitários e

custos totais para uma contratação de 36 meses.

Descrição	Quantidade máxima mensal a ser contratada	Valor máximo mensal estimado pela calculadora do fabricante	Valor do contrato para 36 meses
Azure Repos + Azure Boards para usuários com perfil Basic	105 usuários pagos +5 usuários gratuitos(*)	R\$ 3.567,06	R\$ 128.414,16
Azure Test Plan	12 usuários pagos	R\$ 3.533,09	R\$ 127.191,24
Azure Artifacts	0GB pagos +2GB gratuitos(*)	R\$ 0,00	R\$ 0,00
Azure Pipelines Self-hosted	10 pipelines pagos +1 pipeline gratuito(*)	R\$ 849,30	R\$ 30.574,8
Azure Pipelines Microsoft-hosted	3 pipeline pago +1 pipeline com 1.800 minutos gratuitos(*)	R\$ 679,44	R\$ 24.459,84
Capacitação 12 horas-aula			R\$ 3.701,97
TOTAL		R\$ 8.628,89	R\$ 314.342,01

(*) faixas ou franquias declaradas como gratuitas na política de preços praticada pelo provedor de serviços em nuvem, que deverão ser disponibilizados sem ônus ao órgão ou entidade contratante.

5. DECLARAÇÃO DA VIABILIDADE DA CONTRATAÇÃO (ART. 12, INC. V)

Apesar dos avanços significativos que a adoção de DevOps trará, persiste a necessidade de avaliar em um outro estudo técnico preliminar se a melhor estratégia é adotar uma solução de IA via API em nuvem, uma infraestrutura on-premises ou uma abordagem híbrida. A abordagem híbrida poderia usar modelos open-source on-premises para tarefas mais básicas e os modelos de ponta fornecidos pela nuvem para tarefas mais complexas. A decisão sobre a melhor estratégia será crucial para complementar os esforços de modernização e inovação na CLDF, proporcionando ferramentas avançadas para apoiar a eficiência, a qualidade e a eficácia das atividades legislativas e de fiscalização.

Ciente do estabelecido no AMD 71/2023, a equipe de planejamento da contratação, com base neste estudo técnico, declara viável esta contratação com a solução **Microsoft Azure DevOps Services**.

Conforme [AMD nº 71, de 2023](#), art. 12, § 2º, o Estudo Técnico Preliminar da Contratação será assinado pelos Integrantes Técnico e Requisitante da contratação e pelo Chefe da respectiva Área Técnica de TI e aprovado pelo Chefe da Área de TI. Caso o Chefe da Área Técnica de TI ou o Chefe da Área de TI venha a compor a Equipe de Planejamento da Contratação, a autoridade que assinará o Estudo Técnico Preliminar da Contratação juntamente com os Integrantes Técnico e Requisitante será aquela diretamente superior ao respectivo Chefe, conforme § 3º.



Documento assinado eletronicamente por **RANIERI JOSE DANTAS SEVERIANO - Matr. 18338**, Integrante Técnico, em 08/08/2024, às 15:16, conforme Art. 22, do Ato do Vice-Presidente nº 08, de 2019, publicado no Diário da Câmara Legislativa do Distrito Federal nº 214, de 14 de outubro de 2019.



Documento assinado eletronicamente por **LUIS FELIPE RABELLO TAVEIRA** - Matr. 22970, Integrante **Requisitante**, em 08/08/2024, às 20:31, conforme Art. 22, do Ato do Vice-Presidente nº 08, de 2019, publicado no Diário da Câmara Legislativa do Distrito Federal nº 214, de 14 de outubro de 2019.



Documento assinado eletronicamente por **RONALD TETSUO MIURA** - Matr. 18552, Consultor(a) **Técnico-Legislativo**, em 08/08/2024, às 20:40, conforme Art. 22, do Ato do Vice-Presidente nº 08, de 2019, publicado no Diário da Câmara Legislativa do Distrito Federal nº 214, de 14 de outubro de 2019.



Documento assinado eletronicamente por **JEFFERSON MOURA PARAVIDINE** - Matr. 22751, Diretor(a) de **Modernização e Inovação Digital**, em 08/08/2024, às 20:53, conforme Art. 22, do Ato do Vice-Presidente nº 08, de 2019, publicado no Diário da Câmara Legislativa do Distrito Federal nº 214, de 14 de outubro de 2019.



Documento assinado eletronicamente por **ABIMAEEL AMORIM DA SILVA ROMA** - Matr. 11363, Analista **Legislativo**, em 09/08/2024, às 15:00, conforme Art. 22, do Ato do Vice-Presidente nº 08, de 2019, publicado no Diário da Câmara Legislativa do Distrito Federal nº 214, de 14 de outubro de 2019.



A autenticidade do documento pode ser conferida no site:

http://sei.cl.df.gov.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0

Código Verificador: **1602803** Código CRC: **CE681E11**.

Praça Municipal, Quadra 2, Lote 5, 2º andar, Sala 2.15— CEP 70094-902— Brasília-DF— Telefone: (61)3348-8391
www.cl.df.gov.br - seasi@cl.df.gov.br

00001-00006100/2024-05

1602803v47