



Consulta pública eletrônica no modelo *Request For Information* - RFI para identificar modelo de negócio e de licenciamento para subsidiar estudo para futura contratação de solução de Low Code.



1.Objeto.....	2
2.Descrição do Objeto .....	2
3. Da Resposta a Consulta Pública.....	16
4. Da Transparência.....	17

# 1.Objeto

- 1.1. Consulta pública eletrônica no modelo Request For Information - RFI para identificar modelo de negócio e de licenciamento que subsidiará estudo para futura contratação de solução de Low Code.

## 2.Descrição do Objeto

2.1. Consulta pública eletrônica para Identificar os modelos de negócios e formas de licenciamentos oferecidos pelo mercado, que sejam mais adequados às necessidades do SERPRO, que subsidiará estudo para futura contratação de solução de solução integrada de tecnologia da informação LOW CODE que apoie nas atividades de desenvolvimento, manutenção, monitoramento e operação para possibilitar a criação de softwares WEB e MOBILE de forma rápida em todo o seu ciclo de vida, aderente aos principais modelos arquiteturais de projetos de software disponíveis no mercado, e que possua recursos de aceleração da produtividade do desenvolvimento.

### 2.1.1. Requisitos Funcionais

2.1.1.1. A solução de Low Code deve possuir Ambiente de desenvolvimento integrado que implemente, no mínimo, os seguintes padrões de desenvolvimento:

2.1.1.1.1. Permitir a criação e manutenção de aplicativos WEB e MOBILE com o auxílio de ferramentas visuais sem a necessidade de codificação manual;

2.1.1.1.2. Ser totalmente integrada, garantindo total interoperabilidade entre todos os seus módulos sem a necessidade de implementação/manutenção de integrações entre as ferramentas que compõem a plataforma, assim como as ferramentas deverão ser de um único fabricante a fim de garantir a manutenção desta interoperabilidade à medida que as ferramentas e tecnologias evoluam.

2.1.1.1.3. A solução LOW CODE deve ser independente de tecnologias proprietárias de terceiros, como por exemplo, interpretadores, sendo admitidos servidores de aplicação Java (TOM CAT ou JBOSS).

2.1.1.1.4. Possuir mecanismo de geração automática de código, sem componentes proprietários, que verifique a consistência do código gerado, realizando a otimização e atualização do código gerado para acompanhar a evolução da tecnologia utilizada (ex. versão do framework), compilando e publicando o código gerado em servidores de aplicações padrão de mercado.

2.1.1.1.5. Permitir realizar a catalogação dos componentes, controlando o versionamento das aplicações e disponibilizando um ambiente de gerenciamento de configuração sem a necessidade de software adicional à plataforma, além de possibilitar a distribuição dos aplicativos publicados em vários ambientes de forma automatizada e parametrizada, indicando as etapas de publicação e o detalhe da execução da publicação.

2.1.1.1.6. Permitir a extração de todo o código fonte desenvolvido para um código aberto padrão, sem componentes proprietários, organizado e otimizado na linguagem Java ou .NET que pode ser mantido por meio de plataformas de desenvolvimento em uma das duas linguagens, com os direitos autorais exclusivos ao SERPRO ou a seus clientes quando por força de contratação dos serviços do SERPRO.

2.1.1.1.7. Permitir acoplar ferramentas para realizar a análise e prevenção de erros de codificação a partir de validações sintática e semântica de uma aplicação em desenvolvimento. O processo de validação deverá considerar possíveis falhas de lógica de negócio, conflito de regras de negócio, modelo de dados, integridade de dados, ações errôneas do desenvolvedor na construção da aplicação. A solução deverá corrigir ou alertar sobre os erros encontrados na validação da construção da aplicação, não permitindo a compilação ou publicação da aplicação até que sejam solucionados os erros apresentados.

2.1.1.1.8. Permitir a depuração do aplicativo de forma visual de maneira que o usuário possa acompanhar o passo a passo da execução do seu aplicativo dentro dos próprios modelos visuais e fluxogramas que foram implementados para a geração dos aplicativos e de forma isolada dos demais membros da equipe, permitindo que vários usuários realizem testes e depuração de código isoladamente sem afetar um ao outro.

2.1.1.1.9. Permitir a construção e manutenção de aplicativos mobile e web por meio da utilização de modelos visuais que possibilitem a definição das camadas de interface do usuário, lógica de negócios, processos de negócios e modelos de dados, além da criação de webservices, reutilização de componentes, definição de perfis e regras de acesso e programação de tarefas agendadas.

2.1.1.1.10. Permitir a criação de componentes objetivando o reuso de comportamentos durante o desenvolvimento de aplicativos, visando a integração com aplicativos externos ou integração com outras bases de dados.

2.1.1.1.11. Permitir a exportação de métricas dos aplicativos desenvolvidos de maneira que seja possível verificar a ocorrência de erros, consultas a banco de dados, chamadas a webservices SOAP e REST, envios de mensagens de e-mail, entre outras funções, para ser utilizado por aplicativo de software livre usado para monitoramento de eventos e alertas.

2.1.1.1.12. Possuir aceleradores que permitam que o usuário crie telas com recursos de listagem, visualização, inclusão, alteração e exclusão de registros automaticamente a partir do modelo de dados definido, seguindo padrões de design e layout de tela predefinidos, mas que possam ser modificados pelos usuários para criação destas telas seguindo uma identidade visual personalizada.

2.1.1.1.13. Possuir uma série de layouts de telas padrão com interfaces mobile ou web ricas, organizados em temas que possam ser reutilizadas para criação de novos padrões ou completamente modificados apenas por meio de definição e modificação do CSS Stylesheet (folhas de estilo CSS), para todo o aplicativo ou para telas específicas, garantindo total flexibilidade na criação de diferentes padrões visuais por parte dos usuários.

2.1.1.1.14. Permitir o desenvolvimento de aplicativos com design responsivo, utilizando tecnologias HTML5 e CSS3 ou mais avançadas que possibilitem o redimensionamento automático para adequação às dimensões de tela, de forma que os aplicativos sejam desenvolvidos de forma única e que possam ser executados/suportados por diversos ambientes operacionais como IOS e Android, e diversos tipos de dispositivos, como smartphones, tablets e desktops, sem a necessidade de escrever código nativo específico para cada plataforma.

2.1.1.1.15. Permitir a implementação da usabilidade dos aplicativos por meio da utilização de recursos visuais e componentes no modelo drag-and-drop (arrastar e soltar) que permitam a abstração de recursos técnicos de programação específicos como programação em linguagem JavaScript, requisições assíncronas de HTTP ou manipulação de arquivos XML para operações de atualização AJAX ou outros frameworks/tecnologias equivalentes.

2.1.1.1.16. Possuir recursos que possibilitem a implementação do layout, usabilidade e experiência dos usuários em aplicativos web e mobile por meio do uso de modelos, padrões de interface e usabilidade, componentes visuais, animações e transições de telas pré-constituídas e disponíveis na plataforma para construção das telas por meio da utilização destes elementos.

2.1.1.1.17. Permitir a criação de aplicativos com recursos de multi-idíomas, de forma que a plataforma realize o mapeamento dos textos utilizados nas interfaces para que sejam traduzidos para os idíomas desejados e que o usuário necessite apenas realizar a tradução dos textos e elementos e a plataforma possua recursos para realizar a apresentação do idíoma correto a partir da seleção do usuário durante o desenvolvimento ou execução da aplicação.

2.1.1.1.18. Permitir a definição do nível de segurança de páginas por meio do uso de HTTPS e SSL, com ou sem certificado do cliente, assim como alertar sobre possíveis vulnerabilidades em relação a injeção de código (SQL, HTML e JavaScript), proteger o aplicativo contra ataques de força bruta na autenticação, gerenciar a autenticação no dispositivo móvel para garantir a segurança das transações com o servidor, entre outros recursos.

2.1.1.1.19. Permitir a definição da estrutura de dados por meio da criação das tabelas, campos, chaves (primária e estrangeiras), índices e demais propriedades por meio da modelagem visual, de forma que seja possível a organização da estrutura de dados em diagramas relacionais, assim como realizar a atualização da base de dados automaticamente por meio da aplicação dos scripts diretamente na base de dados para criação e atualização dos objetos.

2.1.1.1.20. Possuir aceleradores que permitem a criação automática da estrutura de dados em uma base de dados a partir da importação de arquivos, incluindo a criação automática da lógica necessária para importação dos dados contidos no arquivo importado.

2.1.1.1.21. Possuir mecanismos visuais de criação de pesquisas/consultas dinâmicas e otimizadas a banco de dados, sem a necessidade de escrita manual de código em linguagem SQL, contendo recursos de agrupamentos, ordenação, contagens e realização de filtros, assim como possibilitando a apresentação do resultado das consultas durante o desenvolvimento a fim de possibilitar a visualização prévia dos dados consultados e permitir a validação/auditoria da qualidade da consulta realizada.

2.1.1.1.22. Permitir realizar a otimização do código no nível de consultas a banco de dados, quantidade de registros retornados pelas consultas, utilização de índices, paginação e cache de dados sem que o usuário precise tratar destes pontos isoladamente nos aplicativos desenvolvidos.

2.1.1.1.23. Possuir recursos que permitam a criação de webservices nos padrões SOAP e REST por meio dos recursos visuais da plataforma, de forma que seja possível definir os métodos, as assinaturas e a lógica dos serviços sem que seja necessário realizar a implementação de código para publicar os serviços.

2.1.1.1.24. Permitir o consumo/acesso a webservices nos padrões SOAP e REST por meio do uso de recursos visuais da plataforma, de forma que seja possível acessar os métodos por meio do seu endereço (URL) e definições, como o WSDL para o SOAP, e utilizá-los na aplicação apenas por meio da chamada aos métodos, sem a necessidade de implementação de código para consumir os serviços.

2.1.1.1.25. Permitir o desenvolvimento de aplicativos mobile híbridos, utilizando o desenvolvimento visual e sem a necessidade de desenvolvimento adicional à plataforma, de forma que seja possível executar lógica e armazenar dados diretamente no dispositivo móvel, atualizar dados no servidor de forma assíncrona e que seja possível realizar mudanças e atualizações na lógica, interface e estrutura de dados no aplicativo de forma imediata, sem submeter novamente para as lojas de aplicativos.

2.1.1.1.26. Permitir o desenvolvimento visual de aplicativos mobile com acesso e utilização aos diversos tipos de sensores e recursos nativos dos dispositivos como acelerômetro, geolocalização, biometria, câmera, entre outros, por meio de recursos nativos da plataforma (pré construídos), de componentes disponíveis em repositórios de código aberto ou por meio do desenvolvimento de novos componentes utilizando frameworks de desenvolvimento de código aberto para aplicativos móveis.

2.1.1.1.27. Permitir a criação de novos componentes para o desenvolvimento de aplicativos mobile ou web para estender as capacidades da plataforma a partir do desenvolvimento utilizando linguagens como Java ou .NET, Objective-C ou Swift, de forma que estes componentes sejam incorporados ao aplicativo mobile ou web desenvolvido na plataforma por meio de recursos visuais.

2.1.1.1.28. Possuir recursos e utilizar tecnologias que otimizem e possibilitem a implementação de controles para garantir o desempenho dos aplicativos mobile desenvolvidos com a plataforma como a utilização de interfaces reativas, que se atualizam automaticamente a partir dos dados, execução de lógica no próprio dispositivo, validação e alertas para implementações que não sigam boas práticas de escalabilidade, configuração de limites de tempo (timeouts) para respostas do servidor, entre outros recursos.

2.1.1.1.29. Permitir o reuso de lógica e dados entre aplicativos web e mobile desenvolvidos por meio dos recursos visuais da plataforma, de forma que aplicativos web e mobile que compartilhem de uma estrutura de dados semelhante e de uma mesma lógica para suas ações possam ser implementadas reaproveitando o que foi desenvolvido em ambos os aplicativos.

2.1.1.1.30. Permitir a simulação da execução do aplicativo mobile desenvolvido por meio da plataforma em um navegador de internet com a possibilidade de simular os gestos e ações realizados no dispositivo, de forma que seja possível realizar testes no aplicativo mobile e validar a experiência de uso junto aos usuários.

2.1.1.1.31. Possuir uma camada de processos de negócios nativa à plataforma e integrada com as demais camadas de desenvolvimento, totalmente visual, de forma que o usuário possa visualizar os processos de negócios integrados ao aplicativo gerado e que estes aplicativos controlem todas as atividades a serem realizadas dentro de um processo, acompanhem os status destas atividades, a alocação das atividades para os usuários e o gerenciamento destas atividades e armazenem informações da execução destas atividades de forma a permitir a extração de métricas e geração de indicadores dos processos.

2.1.1.1.32. Permitir o mapeamento e definição de processos de negócio de forma totalmente integrada ao desenvolvimento de aplicativos, de maneira que o usuário possa implementar os processos de negócios, com subprocessos, tarefas manuais e automáticas, definição de fluxos alternativos, entre outros recursos, que possam ser executados pelos aplicativos desenvolvidos por meio da plataforma e que possam ser relacionados às telas e dados gerenciados pelos aplicativos criados para execução dos processos.

2.1.1.1.33. Permitir a definição do início da execução de processos de negócios a partir da interação do usuário com o aplicativo, como por meio de ações em tela, gravação de informações, envio de informações, atualização de dados ou de algum evento gerado a partir de um aplicativo ou API externos.

2.1.1.1.34. Permitir Validar as mudanças realizadas em processos de negócio já definidos e que tenham processos em andamento, de forma que, a partir das alterações realizadas nos processos de negócio, seja possível identificar conflitos no aplicativo desenvolvido, de forma automática e sem a necessidade de esforço de desenvolvimento adicional a plataforma, sendo necessário apenas a modelagem do processo de negócio na plataforma.

2.1.1.1.35. Permitir a interação com o usuário de forma a exibir todas as atividades pendentes que estão associadas ao usuário para que o mesmo possa realizar o acompanhamento das suas atividades, assim como para que possa acessar diretamente, a partir de um mecanismo no qual deve indicar as tarefas pendentes e as telas relacionadas a essas tarefas, para facilitar sua execução.

2.1.1.1.36. Permitir realizar a coleta de informações sobre os processos de negócios executados a fim de permitir o registro do histórico da execução dos processos e permitir que seja realizado



o acompanhamento do status e andamento de cada processo por meio de relatórios e monitoramento.

2.1.1.1.37. Possuir recursos para a realização de processamento paralelo em lote (Parallel Batch Processing) permitindo maior flexibilidade em ambientes que necessitem de grandes processamentos para extração, transformação e carga de dados, armazenamento e consolidação de informações.

2.1.1.1.38. Permitir a criação de conteúdos de e-mails a serem utilizados para criação de aplicativos colaborativos, onde os designs dos e-mails criados são compatíveis com os principais leitores de e-mail como Outlook, Gmail, entre outros.

2.1.1.1.39. Possuir recurso para realizar autenticação e controle de acesso dos usuários de forma centralizada e unificada (single sign-on), sem que seja necessário esforço de desenvolvimento adicional à plataforma, por meio da capacidade nativa ou do uso de acesso ao LDAP, assim como permitir a extensão da plataforma para implementação de outras formas de controle de acesso, por meio de integração com outras ferramentas disponíveis no mercado ou soluções de controle do SERPRO.

2.1.1.1.40. Permitir a realização de cache no servidor para resultados de consultas a banco de dados, interfaces de usuário, lógicas de negócio ou referências da web, de forma a otimizar a utilização dos recursos e prover maior desempenho dos aplicativos.

2.1.1.1.41. Permitir realizar a validação sintática e semântica de todo o aplicativo em desenvolvimento, incluindo todas as referências, de forma que todas as alterações realizadas, seja na interface do usuário, lógica de negócios, processos de negócio, modelo de dados, regras de acesso ou em outro ponto do aplicativo, sejam validadas e que os ajustes necessários sejam automaticamente corrigidos ou alertados para que sejam modificados, assim como deve bloquear a publicação do aplicativo em caso de erro sintático ou semântico, a fim de garantir que o aplicativo gerado seja livre de erros.

2.1.1.1.42. Possuir assistentes automatizados (wizards) e aceleradores nativos da plataforma, sem a necessidade de esforço de modelagem ou lógica de programação adicional a plataforma, que auxiliem o usuário na criação de componentes para integração, por meio de conexão e mapeamento de estruturas de dados em bases de dados externas da plataforma, importação de código ou importação de funções de sistemas.

2.1.1.1.43. Permitir a reutilização de códigos desenvolvidos em linguagem externa compatível com a plataforma adquirida, disponibilizando-os como componentes visuais da plataforma, de forma que seja possível o reaproveitamento de códigos já existentes.

2.1.1.1.44. Possuir mecanismo, no aplicativo desenvolvido, de forma automática e sem a necessidade de esforço de desenvolvimento adicional à plataforma, sendo necessária apenas a modelagem do processo de negócio na plataforma, que permita a interação com o usuário de forma a exibir todas as atividades pendentes que estão associadas aos usuários, para que os mesmos possam realizar o acompanhamento das suas atividades, assim como para que possam acessar diretamente, a partir deste mecanismo, as telas relacionadas às atividades que devem ser executadas.

2.1.1.1.45. Permitir a conexão e o mapeamento de estruturas de dados em bases de dados no sistema de gerenciamento de banco de dados (SGBD), de forma que seja possível acessar as estruturas de dados existentes em bases de dados (já existentes) e utilizá-las no ambiente de desenvolvimento da plataforma para construção de novos aplicativos, por meio da manipulação e atualização dos dados, mas sem a necessidade de implementar uma integração específica para este acesso, respeitando as políticas de segurança definidas para acesso aos dados e sem a possibilidade de realizar alterações que modifiquem a estrutura de dados das bases legadas.

2.1.1.1.46. Permitir realizar a conexão com bases de dados já existentes (bases legadas) para acessar a estrutura de dados e utilizá-la sem violar as políticas de segurança, acesso e alterações estruturais nos objetos desta base de dados legada.

2.1.1.1.47. Permitir a gestão incremental das aplicações desenvolvidas de forma colaborativa pela equipe técnica de desenvolvimento, contando com recursos de comparação e mescla (merge) de soluções por meio de modelos visuais, permitindo aos usuários a visualização das diferenças existentes em cada fluxo de negócio, modelo de dados, propriedades, telas, entre outros elementos, e permitindo que o merge das diferenças seja realizado também de forma visual.

2.1.1.1.48. Permitir a criação de um repositório de componentes de integração, estruturas de dados, bibliotecas e outras soluções, que podem ser compartilhados e reutilizados no desenvolvimento de diversos aplicativos, sem a necessidade de duplicação do código desenvolvido.

2.1.1.1.49. Permitir realizar o armazenamento de todas as versões dos aplicativos e componentes desenvolvidos em um repositório central, permitindo ao usuário navegar entre todas as versões, verificar informações de quando foram publicadas e por quem, publicar ou reverter a publicação (rollback) de versões e ainda gerenciar detalhes de configuração destes aplicativos.

2.1.1.1.50. Permitir a definição de privilégios de acesso aos aplicativos em desenvolvimento para a equipe de desenvolvimento, de forma que seja possível isolar as equipes de desenvolvimento ou restringir o acesso por aplicativo, módulo ou área específica, provendo um maior controle e segurança ao processo de desenvolvimento.

2.1.1.1.51. Permitir a definição e o gerenciamento de regras de controle de acesso da equipe de desenvolvimento aos aplicativos e ambientes, de forma que seja possível definir os privilégios da equipe de desenvolvimento em relação a criação, alteração, referência e acesso a aplicativos e componentes, assim como para realização de operações de publicação para cada ambiente, a fim de garantir a governança do ambiente.

2.1.1.1.52. Permitir gerar e atualizar os principais artefatos que compõem a documentação técnica dos aplicativos automaticamente, incluindo a arquitetura e as APIs, disponibilizando os diagramas e dicionário de dados, referências e processos de forma visual e fornecendo informações detalhadas dos principais elementos, de forma que a documentação esteja sempre atualizada e que possa ser disponibilizada para os desenvolvedores para garantir a transferência de conhecimento e facilitar o entendimento sobre os aplicativos desenvolvidos.

2.1.1.1.53. Permitir o gerenciamento do portfólio de aplicativos de maneira que o usuário possa planejar e executar a publicação de aplicações completas entre diferentes ambientes, desde o ambiente de desenvolvimento até o ambiente de produção, considerando toda a análise de impactos e dependências entre todos os aplicativos e componentes em cada versão, de forma a garantir a publicação dos aplicativos de forma completa, correta e com integridade, tudo por meio de um console centralizado.

2.1.1.1.54. Possuir a capacidade de gerenciar os ambientes de servidores (desenvolvimento, qualidade e produção) e garantir a transferência do código-fonte dos aplicativos entre os diversos ambientes de forma visual e com garantia de inexistência de erro de dependências de aplicativos, pacotes ou componentes acessórios durante as publicações entre ambientes (staging), por meio de um ambiente de gerenciamento de configuração único, parametrizável e sem a necessidade de software adicional à plataforma, além de permitir a configuração de perfis de acesso para as diversas fases de publicação dos aplicativos, possibilitando governança, segurança e auditoria das alterações antes da publicação em produção.

2.1.1.1.55. Permitir a integração com qualquer esteira DEVOPS, permitindo a publicação e a reversão da publicação (rollback) de aplicativos de forma automatizada evitando a publicação dos aplicativos em fases.

2.1.1.1.56. Permitir realizar o controle do versionamento dos aplicativos mobile desenvolvidos por meio da plataforma e publicados nas lojas de aplicativos, permitindo também o mapeamento e controle das versões dos pacotes gerados pela plataforma em relação às versões publicadas nas lojas de aplicativos.

2.1.1.1.57. Possuir uma arquitetura que possibilite a atualização tecnológica dos aplicativos desenvolvidos de forma simples e direta por meio da plataforma para suportar novas tecnologias de mercado, assim como atualizações das tecnologias utilizadas, como por exemplo, novas versões de frameworks sem que seja necessário construir novamente ou realizar grande reestruturação (refactoring) dos aplicativos desenvolvidos.

## **2.1.2. Requisitos Não Funcionais**

### **2.1.2.1. Licenciamento**

2.1.2.1.1. A solução deve oferecer de licenças para o perfil desenvolvedor.

2.1.2.1.2. A solução deve oferecer uso gratuito e ilimitado de licenças para usuários CIDADÃO, que fazem consultas as aplicações de governo, ou que forneçam informações de consumo das aplicações ou para geração de informações, mesmo quando autenticados, haja visto ser estimado utilização por toda a população brasileira, sendo estimados 220 milhões de brasileiros, não simultâneos.

2.1.2.1.3. A solução deve oferecer licenças para aproximadamente 3.000 usuários identificados dos sistemas, com perfil para consulta ou para inserção de dados/informações.

### **2.1.2.2. Direito de Propriedade**

2.1.2.2.1. A solução de LOW CODE contratada deverá possibilitar que sistemas e aplicativos nele desenvolvidos sejam mantidos independentemente da atualização da plataforma quando instalados na Nuvem Privada do SERPRO.

2.1.2.2.2. A solução de LOW CODE contratada deverá possibilitar que sistemas e aplicativos nela desenvolvidas possam ter seus códigos e modelos baixados (detach) de modo a dar continuidade em outras plataformas ou softwares.

2.1.2.2.3. Os direitos de propriedade intelectual e os direitos autorais da solução de tecnologia da informação sobre os diversos artefatos e produtos produzidos ao longo do contrato, incluindo toda a documentação gerada, o código-fonte das aplicações, os modelos de dados e as bases de

dados, serão de propriedade do SERPRO, e por consequência dos clientes do SERPRO para as soluções contratadas.

2.1.2.2.4. A solução de LOW CODE não deve possuir nenhum tipo de vendor lock-in.

### **2.1.2.3. Documentação**

2.1.2.3.1. Possuir documentação on-line da plataforma, no idioma português (PT-BR), que permitam ao usuário ter perfeito conhecimento da plataforma.

### **2.1.2.4. Modelo de Implementação da Solução**

2.1.2.4.1. A solução deverá ser totalmente implantada no ambiente de nuvem privada do SERPRO, utilizando os recursos e a infraestrutura de TI disponíveis nessa nuvem.

2.1.2.4.2. Todas as licenças de softwares que componham a solução e não estejam disponíveis na nuvem privada do SERPRO, inclusive sistemas operacionais e bancos de dados, deverão ser fornecidas, com suporte e atualizações durante toda a vigência do contrato.

2.1.2.4.3. A arquitetura da solução deverá ser dimensionada e informada ao SERPRO de forma a atender aos requisitos expostos.

2.1.2.4.4. A arquitetura do ambiente de produção da solução deverá ser dimensionada prevendo uma taxa de disponibilidade de 99,5%, operando 24 (vinte e quatro) horas por dia, 7 (sete) dias por semana.

2.1.2.4.5. A arquitetura da solução deverá ser dimensionada prevendo a utilização de zonas de disponibilidade redundantes, garantindo que caso uma zona deixe de funcionar ou tenha degradação do tempo de resposta a segunda seja acionada para atendimento das requisições.

2.1.2.4.6. Deverão ser previstos ambientes de homologação e produção.

2.1.2.4.7. A solução deverá ser implantada sobre plataforma comum de mercado, sem qualquer dependência ou presença de hardwares específicos. Caso haja algum requisito obrigatório para o funcionamento da solução, como token de hardware ou instrução específica de processador, a possibilidade de adoção deverá ser avaliada previamente pelo SERPRO.

2.1.2.4.8. A instalação e configuração da solução na nuvem privada do SERPRO deverá ser realizada de forma assistida pelos times do SERPRO e com base em procedimentos operacionais previamente fornecidos.

2.1.2.4.9. A solução deve ser expansível.

2.1.2.4.10. A instalação como qualquer outra operação pertinente à administração da mesma deverá estar documentada e entregue ao SERPRO.

#### **2.1.2.5. Requisitos de Compatibilidade da Nuvem Privada do SERPRO**

2.1.2.5.1. O ambiente de nuvem privada do SERPRO é composto por um conjunto de ferramentas do programa VMware Cloud Provider (ou VMware Cloud Provider Program – VCPP), de modo que a solução deverá adotar tais ferramentas como infraestrutura base, mantendo compatibilidade com as versões de produção dos seguintes componentes:

2.1.2.5.1.1. VMware Cloud Director (vCD);

2.1.2.5.1.2. Container Service Extension (CSE);

2.1.2.5.1.3. VMware vSphere/ESXi.

2.1.2.5.2. A solução deverá ser totalmente construída em torno do protocolo IPv4 e suportar o protocolo IPv6 para comunicação com o usuário final.

2.1.2.5.3. A intercomunicação entre os componentes internos da solução deverá ocorrer de forma criptografada.

2.1.2.5.4. A solução deverá ser capaz de interoperar em ambientes com NAT (Network Address Translation) e Firewalls.

2.1.2.5.5. A solução deverá atender os seguintes requisitos de automação:

2.1.2.5.5.1. A criação do ambiente deverá ser realizada de forma automatizada, em modelo de autosserviço dentro do ambiente de nuvem privada do SERPRO;

2.1.2.5.5.2. A ferramenta de automação selecionada deverá ser compatível com os componentes da nuvem privada do SERPRO;

2.1.2.5.5.3. No momento da construção da arquitetura da solução, deverá ser observada e utilizada a versão mais atual da ferramenta de automação;

2.1.2.5.5.4. O código utilizado para automação deverá ser documentado e cedido como parte da solução;

2.1.2.5.5.5. Tanto a documentação, quanto a construção do código, deverão seguir as melhores práticas de mercado;

2.1.2.5.5.6. Fazer uso efetivo de um sistema de controle de versão.

2.1.2.5.6. A arquitetura da solução deverá observar os seguintes requisitos:

2.1.2.5.6. 1. Aproveitar os recursos de auto scaling disponíveis no VMware Cloud Director de nuvem para adequar a demanda ao consumo de recursos;

2.1.2.5.6.2. Informar a quantidade média de recursos (CPU e RAM) que a solução assumirá quando ociosa;

2.1.2.5.6.3. Apresentar uma matriz de projeção da quantidade de recursos (CPU e RAM) para atender ao crescimento da demanda e facilitar cálculos futuros de capacidade;

2.1.2.5.7. A solução poderá ser baseada em máquina virtual (VM) ou Contêiner, sendo que:

2.1.2.5.7.1. Se for VM, a arquitetura da solução deverá explorar o uso de VMs efêmeras e a persistência dos dados (da aplicação e logs) deverá ficar desvinculada do ciclo de vida da máquina virtual;

2.1.2.5.7.2. Se for contêiner, a solução deve possuir compatibilidade com pelo menos uma das tecnologias: VMware Tanzu ou Red Hat OpenShift. Somada a isto, deverá aderir às boas práticas para execução em ambientes baseados em contêineres, como:

2.1.2.5.7.2.1. Uso de namespaces;

2.1.2.5.7.2.2. Uso de auto scaling (elasticidade);

2.1.2.5.7.2.3. Uso de limites de requisições e recursos;

2.1.2.5.7.2.4. Uso balanceado dos nós disponíveis (regras de afinidade e anti-afinidade);

2.1.2.5.7.2.5. Uso de Role-Based Access Control (RBAC);

2.1.2.5.7.2.6. Uso de contêineres de tamanho reduzido;

2.1.2.5.7.2.7. Uso de rótulos (labels) para organização dos objetos;

2.1.2.5.7.2.8. Uso de annotation para extensão de funcionalidades;

2.1.2.5.7.2.9. Uso efetivo de políticas de rede e segurança;

2.1.2.5.7.2.10. Uso de contêineres inerentemente efêmeros;

2.1.2.5.7.2.11. Uso de volumes externos aos contêineres para persistência de dados;

2.1.2.5.7.2.12. Uso do paradigma de desenvolvimento para microserviços.

2.1.2.5.8. Caso utilize armazenamento de objetos, a solução deverá observar os seguintes requisitos:

2.1.2.5.8.1. Ser compatível com a solução de armazenamento de objetos disponível na nuvem privada do SERPRO, sendo esta compatível com o protocolo S3-like;

2.1.2.5.8.2. Utilizar a API de programação disponível com a plataforma de armazenamento de objetos;

2.1.2.5.8.3. Os buckets e objetos criados deverão utilizar o nível mais restrito de acesso possível, reduzindo assim a exposição de informações;

2.1.2.5.9. A solução deverá adotar apenas os sistemas operacionais e bancos de dados suportados pelo SERPRO para composição da sua arquitetura, a saber:

2.1.2.5.9.1. Windows Server, Red Hat, CentOS e/ou Rocky/Alma Linux, nas suas versões mais atuais;

2.1.2.5.9.2. Oracle, Microsoft SQL Server, Mariadb, Mysql, Postgress e/ou Elastic, nas suas versões mais atuais.



2.1.2.5.9.3. Caso seja necessária a adoção de qualquer solução diferente das apontadas acima, o SERPRO deverá ser consultado previamente sobre a viabilidade de manutenção posterior, sendo que em caso de soluções proprietárias, o fornecedor deverá incluir o suporte do fabricante.

## 3. Da Resposta a Consulta Pública

3.1. As empresas interessadas devem responder à Consulta Pública, por meio do endereço eletrônico: [consulta publica.supec@SERPRO.gov.br](mailto:consulta publica.supec@SERPRO.gov.br), com o título “Solução Low Code – consulta nº <número da consulta pública>”, com as seguintes informações:

3.2. Às empresas interessadas devem responder à consulta pública com as seguintes informações:

### 3.3.1. Identificação da Empresa

3.3.1.1. Nome completo e fantasia.

3.3.1.2. CNPJ.

3.3.1.3. Endereço completo.

3.3.1.4. Site WEB (www).

3.3.2. Contato:

3.3.2.1. Nome completo do responsável pela resposta desta Consulta Pública.

3.3.2.2. Cargo, telefones e endereço de e-mail.

### 3.3.3. Da Solução de Low Code

3.3.3.1. Nome da solução oferecida, objeto desta consulta pública.

3.3.3.2. Site WEB do fabricante da solução (www).

3.3.3.3. Descrição detalhada da solução e seus componentes (Documentos/datasheet, etc).

3.3.3.4. Descrição detalhada do(s) modelo(s) de negócio da solução.

3.3.3.4. Descrição detalhada do (s) modelo(s) de licenciamento da solução.

3.3.3.4. Estudo de preços de referência (Licença perpétua /subscrição anual, e outras), conforme exemplo abaixo:

Part Number	Descrição da Solução	Licença de uso perpétuo / subscrição anual /Outras	Unidade / Métrica	Faixa / Quantidade	Estudo de referência do Valor Unitário (R\$)

3.3.3.5. Forma e condições de pagamento da solução e seus componentes (Licença perpétua /subscrição anual, etc).

#### 3.3.4. Base de Clientes:

3.3.4.1. Quantidade de clientes no Brasil.

3.3.4.2. Nomes dos entes públicos que já adquiriram a solução.

#### 3.3.5. Experiência e Suporte Técnico

3.3.5.1. O suporte é prestado pelo fabricante ou parceiro?

2.3.5.2. Quais os níveis de serviços ofertados para a solução (Tempo de atendimento, tempo de solução, etc).

2.3.5.3. Informar a forma de repasse de conhecimento, resumos das grades e carga horária.

## 4. Da Transparência

4.1. Todos os documentos e informações relacionados ao processo de contratação do SERPRO e desta consulta pública estão disponíveis no site:

<https://www.transparencia.SERPRO.gov.br/acesso-a-informacao/licitacoes-e-contratos>

4.2. Para este processo foi observado a política de integridade de acordo com art. 32, inc.

V, da Lei nº 13.303/2016, Programa Corporativo de Integridade do SERPRO - PCINT

(TR - 138/2022) e a Cartilha de Integridade do Processo de Aquisições e Contratações.

4.2.1. Para conhecimento das regras de conduta no relacionamento entre fornecedores e empregados do SERPRO, acesse a Cartilha de Integridade do Processo de Aquisições e

Contratações, disponível no link: [https://www.transparencia.SERPRO.gov.br/acesso-a-informacao/licitacoes-e-contratos/documentos/Cartilha\\_paq\\_verso\\_final\\_diagramada.pdf](https://www.transparencia.SERPRO.gov.br/acesso-a-informacao/licitacoes-e-contratos/documentos/Cartilha_paq_verso_final_diagramada.pdf)

4.3. Ressaltamos que o SERPRO não concede ou autoriza nenhum tipo de registro de oportunidade em seus processos de contratação.