

ITEM	ALTERAÇÃO
2.3.8.1. A solução deve disponibilizar uma visualização de forma nativa e consolidada da quantidade de licenças contratadas, a quantidade efetivamente em uso, a quantidade disponível para uso e a métrica de licenciamento, de maneira individual por software e tipo de licença.	2.3.8.1. A solução deve disponibilizar uma visualização de forma nativa e consolidada da quantidade de licenças contratadas, a quantidade efetivamente em uso, a quantidade disponível para uso, de maneira individual por software e tipo de licença.
2.3.8.3. Se a monitoração de experiência de usuário tiver uma abordagem específica de licenciamento, então a ferramenta deve permitir, por meio de uma interface de gerenciamento, a configuração do percentual de usuários que terão as métricas de experiência de usuário coletadas/monitoradas, a fim de termos um maior controle do uso das licenças de monitoração de experiência do usuário.	2.3.8.3. Se a monitoração de experiência de usuário tiver uma abordagem específica de licenciamento, então a ferramenta deve permitir, por meio de uma interface de gerenciamento ou através de customização do agente na aplicação monitorada,, a configuração do percentual de usuários que terão as métricas de experiência de usuário coletadas/monitoradas, a fim de termos um maior controle do uso das licenças de monitoração de experiência do usuário.
2.3.11.3. A solução deve ter compatibilidade com o padrão Open Telemetry.	2.3.11.3. Remover.

2.3.12.4.1. Não deve necessitar de configuração manual do agente em função do tipo e versão de tecnologia, considerando as tecnologias suportadas.	2.3.12.4.1. Preferencialmente a configuração do agente deve ser automática em função do tipo e versão da tecnologia, considerando as tecnologias suportadas, podendo haver configuração manual do agente quando a opção automático não estiver disponível.
2.3.12.4.3. A solução deve prover um único agente por servidor monitorado, independentemente das tecnologias existentes neste servidor.	2.3.12.4.3. A solução deverá prover um único agente que gerencia os demais agentes da solução (se necessário) por servidor monitorado, independentemente das tecnologias existentes neste servidor.
2.3.12.5. A solução deve prover mecanismo nativo de gerenciamento de agentes, que permita de forma remota e em lote, operações como atualização de versão, consulta e alteração de configurações.	2.3.12.5. A solução deve prover mecanismo nativo de gerenciamento de agentes, que permita de forma remota, operações como atualização de versão, consulta e alteração de configurações.
2.3.12.5.1. A solução deve possuir interface de gestão centralizada com informações sobre as versões, situação atual e histórica (Health Check) de todos os agentes instalados no parque.	2.3.12.5.1. A solução deve possuir interface de gestão centralizada com informações sobre as versões, situação atual (Health Check) de todos os agentes instalados no parque.

2.3.13.6. A solução deve ser capaz de monitorar a experiência de usuários finais da aplicação, através de um código JavaScript injetado no front-end da aplicação web de maneira nativa e automática, a ser executado no ambiente/dispositivo/navegador do usuário final sem a necessidade de intervenções como instalação de agentes ou extensões de navegador.	O questionamento está no item 2.3.13.7
2.3.13.7. Em casos de aplicação desenvolvida em plataforma Lowcode onde a simples "injeção" automática de código JavaScript não é possível, a solução deverá possibilitar a "injeção" manual do código JavaScript no front-end da aplicação web.	2.3.13.7. Em casos de aplicação desenvolvida onde a simples "injeção" automática de código JavaScript não é possível, a solução deverá possibilitar a "injeção" manual do código JavaScript no front-end da aplicação web ou configuração de Servidores Web (Apache e NGINX por exemplo) para a inserção do JavaScript.
2.3.13.13.1.7. Kotlin;	2.3.13.13.1.7. Remover
2.3.13.13.4.7.ElasticSearch;	2.3.13.13.4.7. Remover
2.3.13.13.5.3. Nats;	2.3.13.13.5.3. Remover;
2.3.13.14. A solução deve monitorar de forma nativa (serviços ou recursos gerenciados) em, no mínimo, dois dos provedores de nuvem (AWS, Azure, Google cloud, OCI, Huawei, IBM Cloud)	2.3.13.14. A solução deve monitorar de forma nativa ou através de extensões próprias (serviços ou recursos gerenciados) em, no mínimo, dois dos provedores de nuvem (AWS, Azure, Google cloud, OCI, Huawei, IBM Cloud)

<p>2.3.15.1.3. A solução deve monitorar o comportamento de utilização de memória do servidor de aplicação JAVA, contendo, no mínimo, as seguintes métricas: percentual de heap utilizada, execução de garbage collector, CPU dos processos do Garbage Collection e promoção de objetos.</p>	<p>2.3.15.1.3. A solução deve monitorar o comportamento de utilização de memória do servidor de aplicação JAVA, contendo, no mínimo, as seguintes métricas: percentual de heap utilizada, execução de garbage collector, tempo gasto nas execuções dos processos do Garbage Collection e promoção de objetos.</p>
<p><u>2.3.17.1.4. A solução deve verificar se uma transação ou requisição WEB ou mobile (por exemplo, HTTP ou HTTPS) foi atendida do ponto de vista da experiência do usuário final, identificando a satisfação do usuário segundo APDEX (https://www.apdex.org/wp-content/uploads/2020/09/ApdexTechnicalSpecificationV11_000.pdf) ou similar que também seja uma referência de mercado.</u></p>	<p><u>2.3.17.1.4. A solução deve verificar se uma transação ou requisição WEB ou mobile (por exemplo, HTTP ou HTTPS) foi atendida do ponto de vista da experiência do usuário final, identificando a satisfação do usuário segundo APDEX (https://www.apdex.org/wp-content/uploads/2020/09/ApdexTechnicalSpecificationV11_000.pdf) ou método próprio que informe a satisfação do usuário.</u></p>

<p>2.3.17.1.5. A solução deve ser capaz de gerar automaticamente métricas de percentual de conversão para determinada transação de negócio, bem como gerar a visualização de todas as etapas e transações de negócio, com métricas de abandono, e com possibilidade de drill-down (detalhamento) para as transações de abandono.</p>	<p>2.3.17.1.5. A solução deve ser capaz de gerar automaticamente métricas de percentual de conversão para determinada transação de negócio, bem como gerar a visualização de todas as etapas e transações de negócio, com métricas de abandono, e com possibilidade de drill-down (detalhamento) para as transações da etapa.</p>
<p>2.3.19.1. A solução deve coletar dados de desempenho/métricas de performance no banco de dados, em tempo real, sem instalação de agente no servidor de banco de dados e sem a instalação de qualquer componente ou abertura de conexão específica com o banco, tais como:</p>	<p>2.3.19.1. A solução deve coletar dados de desempenho/métricas de performance no banco de dados, em tempo real, sem instalação de agente no servidor de banco de dados e sem a instalação de qualquer componente, tais como:</p>
<p>2.3.20.1.2. A solução deve ser capaz de realizar análise preditiva que permita a antecipação de problemas que possam impactar a aplicação.</p>	<p>2.3.20.1.2. A solução deve ser capaz de realizar análise preditiva que permita a antecipação de problemas e comportamentos que possam impactar a aplicação, seja por meio de técnicas como mineração de dados, estatísticas, modelagem e inteligência artificial/machine learning - AIOps, etc.</p>

<p>2.3.20.1.3.1. A solução deve apresentar de forma gráfica considerando a topologia da aplicação monitorada, a evolução ao longo do tempo em que um comportamento anômalo ou problema ocorrer, destacando os componentes da aplicação impactados e os relacionamentos entre eles. Essa funcionalidade deve permitir a visualização histórica dessa evolução dos comportamentos da aplicação com problema ao longo do tempo.</p>	<p>2.3.20.1.3.1 A solução deve permitir a identificação dos componentes da aplicação impactados e dos relacionamentos entre eles, destacando a evolução desse comportamento ao longo do tempo e com a opção de detalhamento (drill down) quando um comportamento anômalo ou problema ocorrer para a apresentação gráfica da topologia da aplicação monitorada.</p>
<p>2.3.21.1. A solução deve identificar, de forma automática e com uso de inteligência artificial, a causa raiz dos problemas nas aplicações monitoradas em tempo real, apontando as aplicações, componentes impactados pelo problema e manter o histórico dos problemas identificados.</p>	<p>2.3.21.1. A solução deve identificar, de forma automática e com uso de inteligência artificial, a causa raiz dos problemas nas aplicações monitoradas em tempo real, apontando as aplicações, componentes impactados pelo problema e manter o histórico dos problemas identificados.</p>

2.3.21.1.4. A solução deve realizar a correlação de eventos e métricas considerando todo o ecossistema da aplicação, não requisitando treinamento longo de IA (não maior que 24 horas), com objetivo de realizar a análise aprofundada do desempenho e disponibilidade das aplicações monitoradas, podendo chegar até o nível de classes e métodos da aplicação, ou equivalentes em linguagens não Java.	2.3.21.1.4. A solução deve realizar a correlação de eventos e/ou métricas considerando todo o ecossistema da aplicação, com objetivo de realizar a análise aprofundada do desempenho e disponibilidade das aplicações monitoradas, podendo chegar até o nível de classes e métodos da aplicação, ou equivalentes em linguagens não Java.
2.3.24.4.3. Mapas dinâmicos viabilizando a inclusão ou retirada (de maneira "Drag and Drop") das informações disponibilizadas. Inclusive, com a utilização de templates próprios.	2.3.24.4.3. Mapas dinâmicos viabilizando a inclusão ou retirada (de maneira "Drag and Drop") das informações disponibilizadas.
2.3.24.9. A solução deve exibir as informações de experiência de usuário coletadas em um mapa customizável por região demográfica, com capacidade de drill-down (detalhamento).	2.3.24.9. A solução deve exibir as informações de experiência de usuário coletadas em um mapa por região demográfica, com capacidade de drill-down (detalhamento).