

a (peso 2) Quais atributos da qualidade segundo a ISO 25010 podem ser desenvolvidos pela PAPA LEGUAS para superar o concorrente que é forte nas funcionalidades de consulta e parcerias, as quais sua empresa não oferecerá de antemão? Em quais atributos da ISO você pode buscar diferenciação competitiva? Explique ao menos 3 características e 1 subcaracterística de cada uma dessas características escolhidas, onde você pode superar seu concorrente.

Exemplo de resposta esperada (mostrando uma característica – lembre-se que são 3 que você deve fazer): “Confiabilidade, na subcaracterística de Tolerância a falhas, garantindo que xxx aconteça.”

R: Para superar o concorrente que se destaca em funcionalidades de consulta e parcerias, a **PAPA LEGUAS** pode se concentrar nos seguintes atributos da qualidade da **ISO 25010**:

1. Usabilidade:

- **Facilidade de aprendizagem:**
 - Interface intuitiva e amigável para operadores de drones, facilitando o aprendizado rápido e a produtividade.
 - Tutoriais interativos e documentação abrangente para auxiliar no uso da plataforma.
- **Operabilidade:**
 - Navegação fluida e eficiente, minimizando o tempo e esforço necessários para realizar tarefas.
 - Design consistente e intuitivo, reduzindo a necessidade de memorização e treinamento.
 - Acessibilidade para usuários com diferentes níveis de experiência e habilidades técnicas.
- **Compreensibilidade:**
 - Feedback claro e conciso sobre o status das entregas e voos dos drones.
 - Visualizações de dados informativas e intuitivas para facilitar a tomada de decisões.
 - Mensagens de erro explicativas e soluções para problemas comuns.

2. Segurança:

- **Confidencialidade:**
 - Proteção dos dados dos clientes e operadores de drones contra acesso não autorizado.
 - Criptografia robusta para garantir a segurança das comunicações entre a plataforma e os drones.
 - Controle de acesso granular para restringir o acesso às informações apenas aos usuários autorizados.
- **Integridade:**
 - Proteção contra adulteração ou modificação não autorizada de dados de entrega e voo.
 - Mecanismos de detecção e prevenção de intrusões para proteger a plataforma contra ataques cibernéticos.
 - Validação de dados robusta para garantir a precisão e confiabilidade das informações.
- **Disponibilidade:**
 - Alta disponibilidade da plataforma para garantir o acesso contínuo aos serviços de roteirização de drones.
 - Planos de recuperação de desastres para garantir a rápida retomada das operações em caso de falhas.
 - Monitoramento proativo da plataforma para identificar e resolver problemas em potencial antes que afetem os usuários.

3. Eficiência de desempenho:

- **Velocidade:**

- Processamento rápido de solicitações de voo, minimizando o tempo de espera para os operadores de drones.
- Algoritmos eficientes de roteirização para otimizar o tempo de voo e reduzir custos.
- Suporte para múltiplos drones simultâneos para aumentar a capacidade de entrega.
- **Precisão:**
 - Cálculos precisos de rotas para garantir entregas eficientes e evitar atrasos.
 - Atualizações em tempo real das condições do tráfego aéreo para otimizar rotas e evitar colisões.
 - Consideração de fatores como clima e condições do terreno para garantir a segurança das entregas.
- **Recursos:**
 - Uso eficiente de recursos de computação e infraestrutura para minimizar custos.
 - Otimização do consumo de bateria dos drones para aumentar o tempo de voo e reduzir a necessidade de recargas.
 - Monitoramento do desempenho da plataforma para identificar e otimizar áreas de consumo excessivo de recursos.

b (peso 2) Quais domínios de processos do COBIT estão ligados com o uso dos recursos GIT e JUNIT? Liste-os.

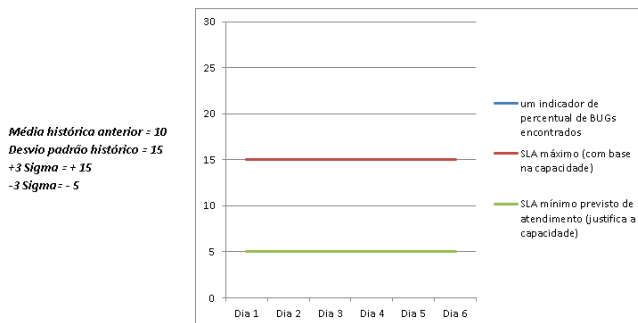
b. Domínios de processos do COBIT ligados com GIT e JUNIT:

- **Planejar e Organizar (PO):**
 - PO7 - Gerenciar recursos humanos.
- **Adquirir e Implementar (AI):**
 - AI2 - Adquirir e manter software aplicativo.
 - AI6 - Gerenciar mudanças.

c (peso 2) Para um indicador de percentual de BUGs encontrados por aplicação liberada por programadores de uma fábrica de software, considere que a média histórica de projetos anteriores é de 10% com desvio padrão que aponta um limite superior de controle de 15% e inferior de 5%, desenhe o gráfico de controle desse processo de controle de BUGs.

Medições: Gráfico de Controle de BUGs

	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6
um indicador de percentual de BUGs encontrados						
SLA máximo (com base na capacidade)	15	15	15	15	15	15
SLA mínimo previsto de atendimento (justifica a capacidade)	5	5	5	5	5	5



INDICADOR: OBSERVAÇÃO DE CONTROLE DE BUGS
OBJETIVO: Acompanhar a ocorrência de bugs em um sistema
DEMANDANTE: Gerente de Desenvolvimento de Software.
CÁLCULO: O percentual de BUGs é calculado dividindo o número de BUGs encontrados pelo número total de testes realizados e multiplicando por 100 para obter a porcentagem.
REPORTE: Os resultados são reportados diariamente em um relatório específico de controle de qualidade de software.
RESPONSÁVEL PELA APURAÇÃO: Equipe de Controle de Qualidade de Software
DEMONSTRAÇÃO: Os resultados são demonstrados em um gráfico de controle estatístico que mostra a evolução do percentual de BUGs ao longo do tempo, comparando-os com os limites de controle estabelecidos.

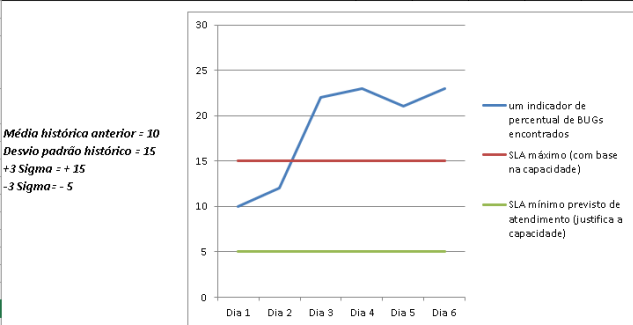
d (peso 2) Considere que foram feitas as seguintes medições recentes, realizadas durante o desenvolvimento do projeto de controle de drones. Os desenvolvedores estão realizando entregas com um percentual de bugs registrados por dia, conforme a distribuição a seguir:

- Dia 1: 10%
- Dia 2: 12%
- Dia 3: 22%
- Dia 4: 23%
- Dia 5: 21%
- Dia 6: 23%

Marque as observações no gráfico de controle e indique se o processo está controlado ou não.

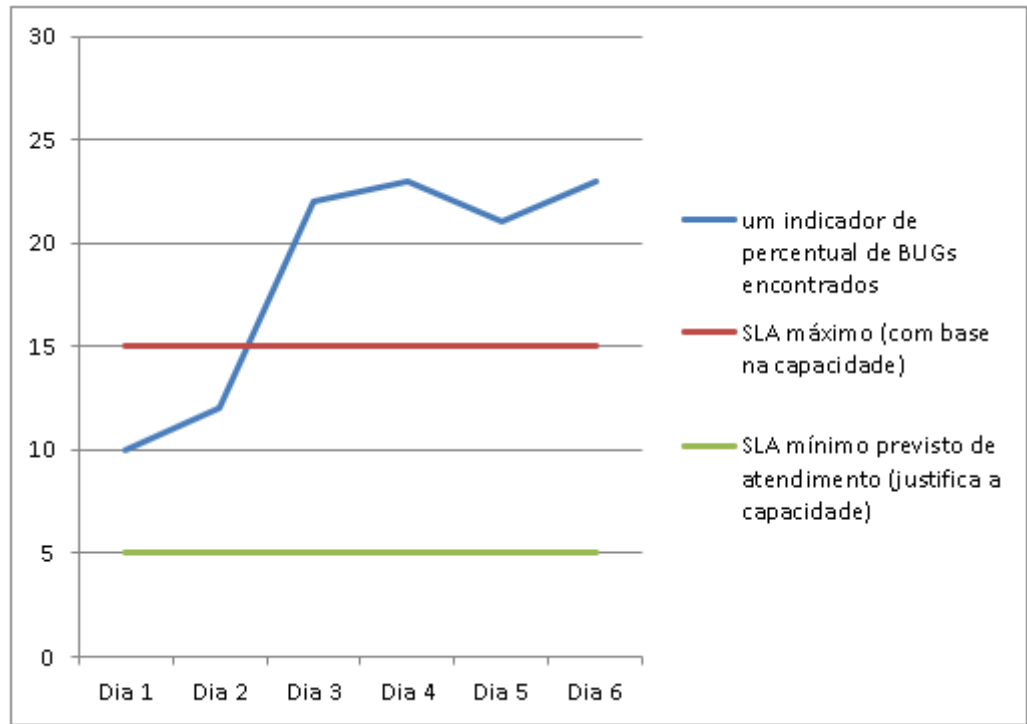
Medições: Gráfico de Controle de BUGs

	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6
um indicador de percentual de BUGs encontrados	10	12	22	23	21	23
SLA máximo (com base na capacidade)	15	15	15	15	15	15
SLA mínimo previsto de atendimento (justifica a capacidade)	5	5	5	5	5	5



INDICADOR: OBSERVAÇÃO DE CONTROLE DE BUGS
OBJETIVO: Acompanhar a ocorrência de bugs em um sistema
DEMANDANTE: Gerente de Desenvolvimento de Software.
CÁLCULO: O percentual de BUGs é calculado dividindo o número de BUGs encontrados pelo número total de testes realizados e multiplicando por 100 para obter a porcentagem.
REPORTE: Os resultados são reportados diariamente em um relatório específico de controle de qualidade de software.
RESPONSÁVEL PELA APURAÇÃO: Equipe de Controle de Qualidade de Software
DEMONSTRAÇÃO: Os resultados são demonstrados em um gráfico de controle estatístico que mostra a evolução do percentual de BUGs ao longo do tempo, comparando-os com os limites de controle estabelecidos.

Dia 1 e Dia 2: Os percentuais de BUGs encontrados estão abaixo do limite superior de controle (LSC), portanto, essas observações estão dentro do controle.
Dia 3, Dia 4, Dia 5 e Dia 6: Os percentuais de BUGs encontrados estão acima do limite superior de controle (LSC), indicando que o processo está fora de controle.



Dia 1 e Dia 2: Os percentuais de BUGs encontrados estão abaixo do limite superior de controle (LSC), portanto, essas observações estão **dentro do controle**.

Dia 3, Dia 4, Dia 5 e Dia 6: Os percentuais de BUGs encontrados estão acima do limite superior de controle (LSC), indicando que o processo está **fora de controle**.

e (peso 2) Ao terminar o seu documento de prova, gere um PDF e suba em um repositório GITHUB **público**, seu, numa Branch develop, dentro de uma pasta chamada “DocumentosCheckpoint”. De preferencia, faça as operações com o GIT Flow.

Feito!