



A) Usabilidade, na subcaracterística de Facilidade de operação, de maneira que o front-end ajude na experiência do cliente;

Portabilidade, na subcaracterística de Facilidade de adaptação a novas plataformas, desenvolvendo os aplicativos em flutter, para criar uma única aplicação ao invés de criar uma aplicação para cada plataforma(IOS, Android eWEB)

Desempenho, na subcaracterística de Tempo Consumido, de modo que ao ter uma aplicação front-end, o acesso é muito mais rápido e acessível, pois o cliente pode ter o aplicativo na palma das mãos no seu celular.

B)

Efetividade:

Git: O Git é altamente eficaz como sistema de controle de versão distribuído. Ele permite que equipes de desenvolvimento colaborem de forma eficiente, rastreiem alterações no código-fonte e gerenciem fluxos de trabalho de desenvolvimento de software de forma eficaz.

JUnit: O JUnit é eficaz como framework de teste unitário para Java. Ele permite que os desenvolvedores escrevam e executem testes de unidade de forma automatizada, garantindo a qualidade e a robustez do código-fonte.

Eficácia:

Git: O Git é eficaz na gestão de alterações no código-fonte, facilitando o desenvolvimento colaborativo e a manutenção do histórico de versões do software.

JUnit: O JUnit é eficaz na identificação de erros e falhas no código através da execução de testes automatizados, garantindo que o software desenvolvido atenda aos requisitos de qualidade estabelecidos.

Confidencialidade:

Git: O Git oferece recursos para controlar o acesso ao repositório de código-fonte, permitindo que os desenvolvedores definam permissões de acesso e protejam informações confidenciais contidas no código.

JUnit: O JUnit em si não lida diretamente com questões de confidencialidade, pois é focado na execução de testes de unidade. No entanto, os testes automatizados podem ajudar a identificar possíveis vulnerabilidades de segurança que afetam a confidencialidade do software.

Integridade:

Git: O Git mantém a integridade do código-fonte através de um sistema robusto de controle de versão, garantindo que todas as alterações sejam rastreadas e registradas de forma precisa.

JUnit: O JUnit contribui para a integridade do software ao permitir que os desenvolvedores criem testes de unidade que verifiquem se as partes individuais do código estão funcionando conforme o esperado.

Disponibilidade:

Git: O Git é amplamente disponível e suportado em uma variedade de plataformas, o que facilita o acesso e a colaboração entre os membros da equipe de desenvolvimento.

JUnit: O JUnit está disponível como um framework de teste de código aberto e é amplamente utilizado na comunidade de desenvolvimento de software Java.

Conformidade:

Git: O Git pode ser configurado para atender a requisitos de conformidade específicos da organização, como controle de acesso e auditoria de alterações no código-fonte.

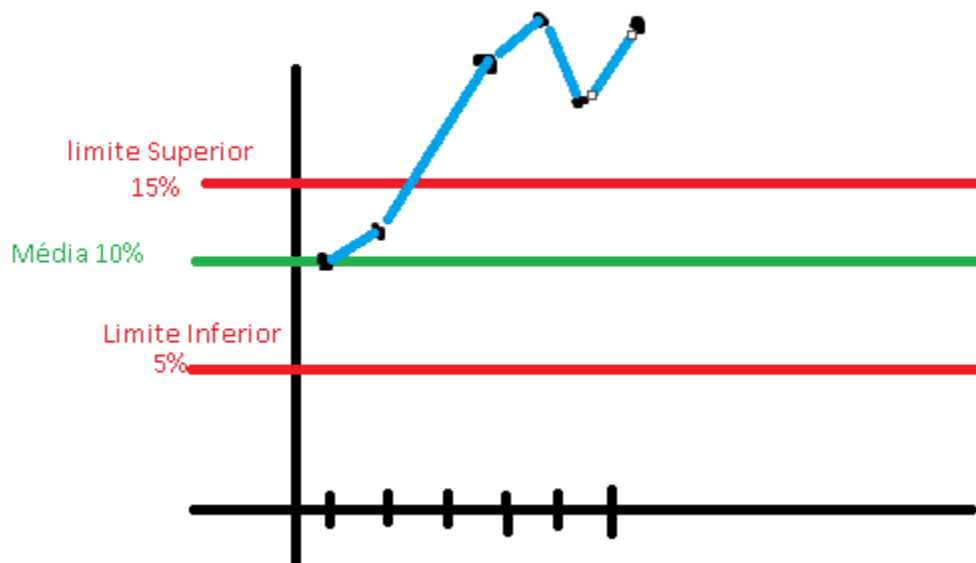
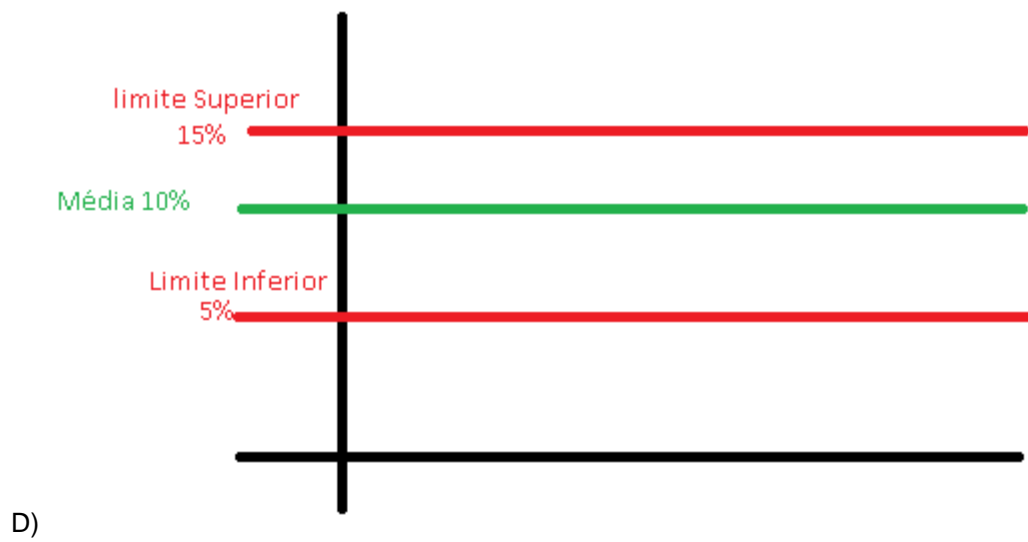
JUnit: O JUnit não lida diretamente com conformidade, mas pode ser usado como parte de um processo de desenvolvimento que segue padrões e práticas de conformidade específicos.

Confiabilidade:

Git: O Git é conhecido por sua confiabilidade e estabilidade como sistema de controle de versão, sendo amplamente utilizado em projetos de todos os tamanhos.

JUnit: O JUnit é confiável como framework de teste de unidade, permitindo que os desenvolvedores identifiquem e corrijam problemas no código de forma consistente e confiável.

C)



O processo não está controlado, pois há muitos indicadores acima do limite superior. Dessa maneira, é necessária uma verificação para filtrar o que está bom para continuar implementando na empresa