

Requisitos não funcionais

Desempenho

1. O sistema deve possuir transições suaves e rápidas entre cada ação ou mudança de tela.
2. O sistema deve executar e responder às ações do usuário e operações (consultas, login, extrato, etc.) em até 2 segundos.
3. O processamento de uma transferência (Pix, Ted, etc) devem ser concluídas em até 5 segundos.
4. O sistema deve suportar até 10.000 acessos simultâneos sem que o seu desempenho seja afetado de forma perceptível.
5. O sistema deve funcionar de forma estável em conexões mais fracas com dados móveis 3G e acima.
6. O sistema não deve utilizar mais que 50Mb de memória RAM por cada consulta básica.
7. O sistema não deve ocupar mais de 400mb de espaço em armazenamento.

Segurança

1. O sistema deve permitir que o usuário crie e salve senhas fortes, alertando sobre os riscos de se utilizar senhas fracas.
2. O login deve possuir autenticações multifator (biometria, código por email, senha, etc.) no acesso ao aplicativo, tentativa de transações, entre outros.
3. O sistema deve utilizar criptografia em todas as comunicações entre servidor e cliente.
4. O sistema deve bloquear o acesso após 5 tentativas falhas de acesso por meio da senha ou biometria.
5. O sistema deve permitir sessões de uso máximas de 10 minutos, após exceder o usuário deve logar novamente.
6. O sistema deve permitir bloquear o cartão imediatamente de acordo com a necessidade do usuário.
7. Dados pessoais ou sensíveis (senhas, CPF, nº do cartão, etc) não devem ser exibidos por completo na tela.
8. O sistema deve bloquear capturas e gravações de tela. 9. O sistema deve realizar monitoramento em tempo real de fraudes, tentativas indevidas de login ou transações, etc.

Usabilidade

1. Interface intuitiva – o app deve permitir que usuários de diferentes perfis naveguem sem necessidade de treinamento.
2. Consistência visual e de interação – padrões de cores, ícones e fluxos devem ser uniformes em todas as telas.
3. Feedback imediato – cada ação do usuário (transferência, pagamento, login) deve gerar respostas visuais ou sonoras em tempo aceitável (< 1s para confirmações simples).
4. Acessibilidade – suporte a leitores de tela, alto contraste, fontes ajustáveis e navegação por voz.
5. Compatibilidade com múltiplas plataformas – a experiência deve ser similar em iOS, Android e web.
6. Redução de erros do usuário – validações de campos (ex.: número da conta, CPF, valor de transação) antes do envio.
7. Ajuda integrada – FAQs, chatbot ou tutoriais rápidos dentro do aplicativo.
8. Tempo de aprendizado baixo – um usuário iniciante deve conseguir realizar operações básicas (ex.: consultar saldo) em menos de 3 minutos após o primeiro uso.

Requisitos de confiabilidade

1. Disponibilidade alta – o sistema deve estar disponível 24/7, com uptime $\geq 99,9\%$.
2. Tolerância a falhas – em caso de falha em um serviço (ex.: API de pagamentos), o app deve exibir mensagem clara e tentar retentativas automáticas.
3. Recuperação de falhas – em caso de queda, o sistema deve se recuperar em menos de X minutos (definido por SLA).
4. Integridade dos dados – nenhuma transação deve ser perdida ou processada parcialmente.
5. Backup e redundância – dados críticos (saldos, histórico, transferências) devem ter cópias redundantes em servidores seguros.
6. Consistência das operações – valores exibidos (ex.: saldo, extrato) devem refletir sempre o estado real da conta.
7. Monitoramento e alertas – falhas, indisponibilidades ou anomalias devem ser detectadas e notificadas em tempo real.
8. Proteção contra corrupção de dados – uso de mecanismos de verificação (checksums, logs transacionais).

Escalabilidade

1. Elasticidade – O sistema deve ajustar os recursos conforme a demanda, evitando perda de desempenho.
2. Distribuição – O processamento deve ser dividido em múltiplos servidores ou instâncias.

3. Crescimento Horizontal – Permitir adicionar novos servidores sem grandes alterações no código.
4. Crescimento Vertical – Possibilidade de ampliar recursos de hardware em servidores já existentes.
5. Balanceamento de Carga – As requisições devem ser distribuídas para evitar sobrecargas.
6. Monitoramento – O sistema deve registrar métricas e prever pontos de falha.
7. Tolerância a Picos – Deve suportar acessos massivos em períodos críticos, como datas de pagamento.
8. Automatização – A infraestrutura deve permitir ajustes automáticos de recursos quando necessário.

Manutenibilidade

1. Modificabilidade - O código deve ser escrito de forma limpa, livre de gambiarras e erros, para que facilite mudanças e manutenções. Onde, o sistema permita alterações diretas no código.
2. Modularidade - Deve-se ter a capacidade do código ser dividido em módulos independentes. Onde será permitido a inclusão de novas funcionalidades e serviços sem que se altere os outros componentes do código.
3. Analisabilidade - O código deve possuir uma boa documentação, além de ser bem comentado, afim de facilitar a identificação de erros dentro de setores específicos. Registrando-se cada alteração em logs detalhados.
4. Reusabilidade - É interessante o sistema ter a capacidade de reutilizar funções e bibliotecas internas, dentro de outros componentes do código.
5. Testabilidade - A capacidade de testar cada modificação no código do sistema para validar uma nova versão, possuindo um ambiente propício para testes.