

# Papel do Arquiteto de Software

- O Arquiteto de Software é responsável pelo design das aplicações conforme seu contexto ou escopo;
- É fundamental o conhecimento de padrões arquiteturais para design de aplicações, inclusive os mais adotados em pacotes de mercado otimizando nossas soluções de software e adotando mecanismos de alta disponibilidade;
- O Arquiteto de Software terá interações constantes com a arquitetura técnica, conhecendo e adotando de forma assertiva as tecnologias, frameworks e plataformas;
- O Arquiteto de Software deve conhecer todos os padrões de integração, estabelecer componentes que façam abstrações e segregação das responsabilidades entre as camadas;
- O Arquiteto de Software deve aplicar os conceitos e boas práticas de desenvolvimento de software, utilizar mecanismos de segurança e transporte de dados;
- É de responsabilidade do Arquiteto de Software dominar os requisitos não-funcionais garantindo a performance dos aplicativos, definindo métricas operacionais de Confiabilidade, Integridade e Disponibilidade.

## Objetivo do Desafio

Desenvolver uma arquitetura de software escalável e resiliente, garantindo alta disponibilidade, segurança e desempenho. Isso inclui a seleção adequada de padrões arquiteturais, integração de tecnologias e frameworks, além de otimização de requisitos não-funcionais. Deve abranger aspectos importantes, como design, integração, segurança e desempenho.

- **Escalabilidade:** Garanta que a arquitetura possa lidar com o aumento da carga de trabalho sem degradação significativa do desempenho. Considere dimensionamento horizontal, balanceamento de carga e estratégias de cache.
- **Resiliência:** Projete para a recuperação de falhas. Isso inclui redundância, failover, monitoramento proativo e estratégias de recuperação.
- **Segurança:** Proteja os dados e sistemas contra ameaças. Implemente autenticação, autorização, criptografia e mecanismos de proteção contra ataques.
- **Padrões Arquiteturais:** Escolha padrões adequados, como microsserviços, monolitos, SOA ou serverless. Considere trade-offs entre simplicidade e flexibilidade.
- **Integração:** Defina como os componentes se comunicarão. Avalie protocolos, formatos de mensagem e ferramentas de integração.
- **Requisitos Não-Funcionais:** Otimize para desempenho, disponibilidade e confiabilidade. Defina métricas e metas claras.
- **Documentação:** Registre decisões arquiteturais, diagramas e fluxos de dados. Isso facilita a comunicação e a manutenção.

Lembrando que não é necessário que todas essas premissas sejam apresentadas na codificação, mas nas decisões e representações arquiteturais do projeto. A intenção do desafio é analisar o seu conhecimento empírico, capacidade de tomada de decisão, aplicação de boas práticas, decomposição dos domínios e componentes, etc.

## Descritivo da Solução

Um comerciante precisa controlar o seu fluxo de caixa diário com os lançamentos (débitos e créditos), também precisa de um relatório que disponibilize o saldo diário consolidado.

## Requisitos de negócio

- Serviço que faça o controle de lançamentos
- Serviço do consolidado diário

## Requisitos técnicos obrigatórios

- Desenho da solução
- Deve ser feito usando C#
- Testes
- Boas praticas são bem vindas (Design Patterns, Padrões de Arquitetura, SOLID e etc)
- Readme com instruções claras de como a aplicação funciona, e como rodar localmente
- Hospedar em repositório publico (GitHub)
- Todas as documentações de projeto devem estar no repositório

Caso os requisitos técnicos obrigatórios não sejam minimamente atendidos, o teste será descartado.

## Requisitos não funcionais

O serviço de controle de lançamento não deve ficar indisponível se o sistema de consolidado diário cair. Em dias de picos, o serviço de consolidado diário recebe 50 requisições por segundo, com no máximo 5% de perda de requisições.

## Observações

Leve em consideração todos os critérios técnicos mencionados, mas não se prenda somente a eles. Use o teste para demonstrar sua habilidade em tomar decisões sobre o que é importante durante o desenvolvimento de aplicações.

Também são bem vindas descrições sobre o que você gostaria de ter implementado ou evoluções futuras para o sistema proposto. Entendemos que o tempo para a execução do projeto é limitado, então aproveite a documentação do seu projeto para apresentar pontos que possam demonstrar melhor suas capacidades, além do que é possível demonstrar através do desenvolvimento desse desafio.

Desejamos boa sorte para você!