

Cenário

Considere o cenário em que você é um engenheiro de sistemas e será responsável por analisar os sistemas legados do domínio de EMPRÉSTIMOS. Nossos sistemas legados possuem mais de 20 anos e estão suportados em diferentes tecnologias e plataformas (mainframe e distribuída).

As principais funções que temos hoje no sistema são: Simulação de Empréstimos, Quitação de saldo devedor de empréstimos, Liberação do valor contratado via: (DOC, TED, Crédito em Conta Corrente) e Cobrança de contratos e renegociação de dívidas. Estas funcionalidades executam no ambiente mainframe.

Uma das principais funções que temos no domínio de EMPRÉSTIMOS é a Simulação de Empréstimos. Atualmente ela executa no ambiente mainframe (COBOL).

O seu desafio como engenheiro de sistemas será propor e apresentar uma solução para modernizar esta funcionalidade, utilizando as técnicas de engenharia de software de forma que resolva a necessidade de negócio.

Perguntas

1. Qual é a sua estratégia para modernizar essa funcionalidade?
 - a. Proponho uma arquitetura de microsserviços, onde teríamos, pela visão passada nas informações importantes, a execução de toda a lógica do simulador, recebendo seus parâmetros funcionais;
 - b. Esse microsserviço possuiria dois adaptadores de entrada de informações, 1 *controller*, para receber requisições REST, e 1 como *listener*, recebendo mensagens de uma fila. A ideia aqui é demonstrar que poderão haver diversos adaptadores de entrada (SOAP, arquivo texto, arquivo xml etc);
 - c. Além disso, possuiria uma camada *core*, responsável pelo processamento em si da simulação de crédito. Essa camada é isolada, não requisitando ser alterada quando da necessidade de um novo adaptador de entrada;
 - d. Para dados recorrentes, utilizaria de técnicas de cache, a fim de evitar diversos I/Os desnecessários. Uma abordagem seria com a utilização do ElastiCache AWS, com Redis ou Memcached, de acordo com a necessidade da aplicação sendo elaborada;
 - e. Como adaptadores de saída de dados, definiria adaptadores de acesso à base, ou mesmo de saída para chamada a outro microsserviço externo ao de simulação de empréstimos. Um adaptador de saída que posso visualizar inicialmente seria um adaptador de base DB2, caso necessária a persistência de alguma informação na alta plataforma, ou mesmo um adaptador que se faça necessário para uma chamada a algum dos sistemas legados, mantendo assim a convivência entre os ecossistemas;
 - f. A fim de disponibilizar essa nova funcionalidade, criaria um pipeline Jenkins, passando pelo SonarQube (ou HP Fortify) a fim de garantir a qualidade do código, publicando, por

fim, num EC2 da AWS gerenciado por um *kubernetes*, ou mesmo no EKS, que gerencia e provisiona automaticamente;

- g. Realizadas as etapas de desenvolvimento, com cobertura mínima aceitável de 80% mediante uso de testes unitários (JUnit), documentar toda a nova funcionalidade no Jira Confluence (ou o software utilizado na empresa), além de entregar à equipe de QA para construção dos testes funcionais (Cucumber, Selenium).

2. Como será a convivência desta funcionalidade modernizada com o legado? (Visão técnica)

- a. Ao entender quais são as necessidades do legado, e sua forma de acesso, a nova funcionalidade conversará com o mesmo através da criação de adaptadores de saída de dados para persistência ou consumo de outra aplicação, conforme explicado acima, ou mesmo adaptadores de entrada de dados, caso alguma aplicação legada necessite de dados advindos do simulador de empréstimos.

3. Quais serão os fatores críticos para o sucesso da modernização dessa funcionalidade?

- a. Sinergia entre a equipe técnica e a área de negócios;
- b. Domínio da tecnologia a ser utilizada;
- c. Comprometimento da *squad* para com as entregas;
- d. Elevado senso de *ownership* do produto pela *squad*;
- e. Padronização e asseio de código (*clean code*);
- f. Documentação da aplicação.

4. Quais seriam as linguagem e tecnologias envolvidas na solução e por quê?

- a. Java, com SpringBoot ou C# com WebApi, dada a facilidade de desenvolvimento e disponibilização de profissionais no mercado;
- b. Servidor de aplicações em nuvem (Amazon EC2, Amazon EKS ou Azure App Service), objetivando eficiência, fácil e veloz escalabilidade, além de redução de custos em se manter uma arquitetura mainframe;
- c. Utilização do *kubernetes* a fim de orquestrar os contêineres, ou utilizar-se de Amazon EKS, que orquestra de forma automática e dinâmica;
- d. Base de dados relacional SQL Server, Oracle ou PostgreSQL, ou até mesmo uma base de dados NoSQL (CosmosDB, DynamoDB ou CassandraDB), após avaliar a dimensão da aplicação a ser modernizada;
- e. Uso de ElastiCache AWS para disponibilização de dados recorrentes, evitando I/Os desnecessários;
- f. Jenkins (ou VSTS) e SonarQube (ou HP Fortify), a fim de criar um *pipeline* para disponibilização contínua (*continuous deployment*)

Case

5. Como a performance será otimizada de um jeito que ficará fácil de escalar?
 - a. Através do uso de contêineres escaláveis, juntamente com utilização de *kubernetes* para realizar a governança desses contêineres.
6. Dado que é um projeto importantíssimo para a organização, como você faria para acelerar o desenvolvimento, entregando valor para o cliente em menor tempo?
 - a. Empregando conceitos de CD / CI, a fim de proporcionar a elaboração de *MVPs* a cada Sprint ou ciclo de desenvolvimento.

Informações importantes

1. Essa funcionalidade recebe os parâmetros Valor da operação, Prazo, Taxa e Data do primeiro vencimento.
2. Modernização da funcionalidade significa aplicar tecnologias novas e emergentes que simplifiquem e acelerem as soluções técnicas, valorizando o desacoplamento das funcionalidades com a criação de peças simples, performáticas e que tenham reuso.

Protótipo arquitetônico da solução proposta:

