

# Desafio 2022-04-20 – Qintess

Sistema de captura de dados para credit-score de clientes

## Índice

Introdução.....	1
Modelos .....	3
Base A.....	3
Base B.....	3
Base C.....	3
Arquitetura.....	4
Modelo AWS .....	4
Adaptando .....	4
CI/CD - Code Pipeline .....	5
Políticas de qualidade de software – Sonar-cube.....	5
Build pipeline Jenkins.....	6
Arquitetura do micro serviço .....	6
Conclusão.....	8

Autor	data	Versão
Oswaldo.junior@qintess.com	20-abr-22	1.0

## Introdução

Hoje, a Empresa de Scores, como dito anteriormente, é o maior Bureau de crédito do Brasil. Aqui trabalhamos constantemente com grande volume e complexidade de dados. Sabendo disso, precisamos que você elabore uma solução que ofereça armazenamento, processamento e disponibilização desses dados, sempre considerando que tudo deve estar conforme as boas práticas de segurança em TI. Afinal, nosso principal ativo são os dados sensíveis dos consumidores brasileiros.

Este documento descreve de forma sucinta uma sugestão de construção do sistema

A



Base A, é extremamente sensível e deve ser protegida com os maiores níveis de segurança, mas o acesso a esses dados não precisa ser tão performática. Dados pessoais devem ter acesso restrito e protegido, usando criptografia de ponta a ponta, usando chave simétrica de acessos – Dados Pessoais.

B



Base B que também possui dados críticos, mas ao contrário da Base A, o acesso precisa ser um pouco mais rápido. Uma outra característica da Base B é que além de consultas ela é utilizada para extração de dados por meio de algoritmos de aprendizado de máquina – Dados Financeiros

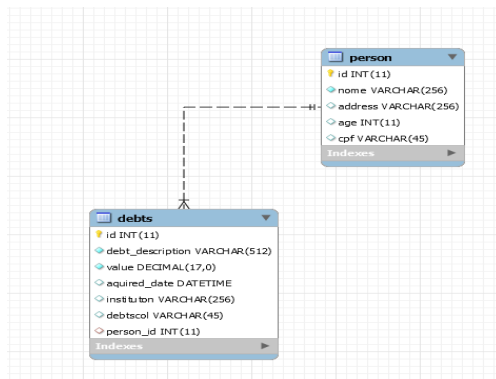
C



Base C, que não possui nenhum tipo de dado crítico, mas precisa de um acesso extremamente rápido – Dados de Movimentações

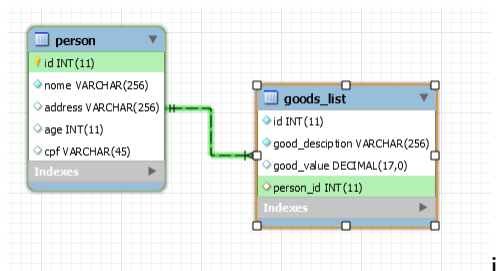
# Modelos

## Base A



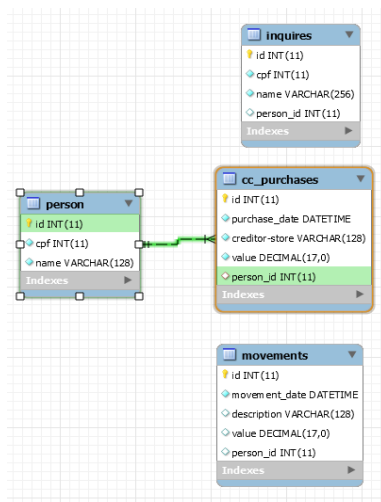
Base de dados pessoais deve ter acesso restrito e protegido contra invasões, usar Transparente Data Encryption – obriga o uso de uma chave de acesso seguro e confinada.

## Base B



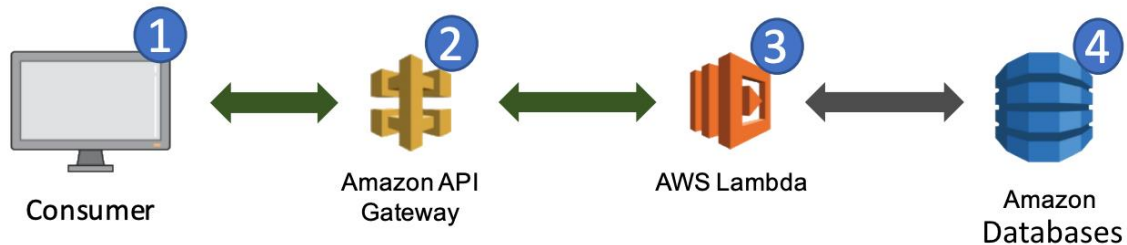
implementada em uma base Spark SQL que em combinação com a base C permite a obtenção estatísticas e conclusões relevantes sobre score de crédito e eventuais mudanças de hábitos que podem revelar tentativas de fraudes.

## Base C



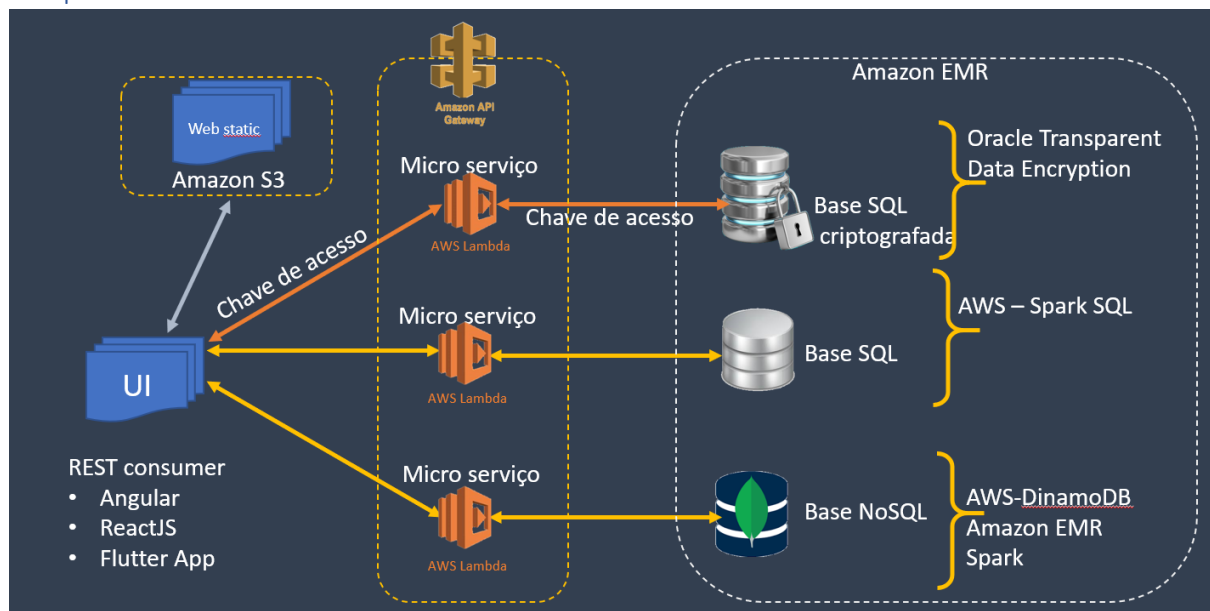
Esta base contém os dados brutos de movimentações e devem ter um acesso rápido e simplificado como o uso de NoSQL , por exemplo DinamoDB.

## Arquitetura Modelo AWS



Visando criar uma aplicação escalável e segura adotamos uma arquitetura em camadas baseadas nos produtos CLOUD da Amazon – AWS.

## Adaptando



O acesso às respectivas bases é feito por micro serviços (AWS lambda) que garantem maior flexibilidade de implementação bem como escalabilidade de acessos.

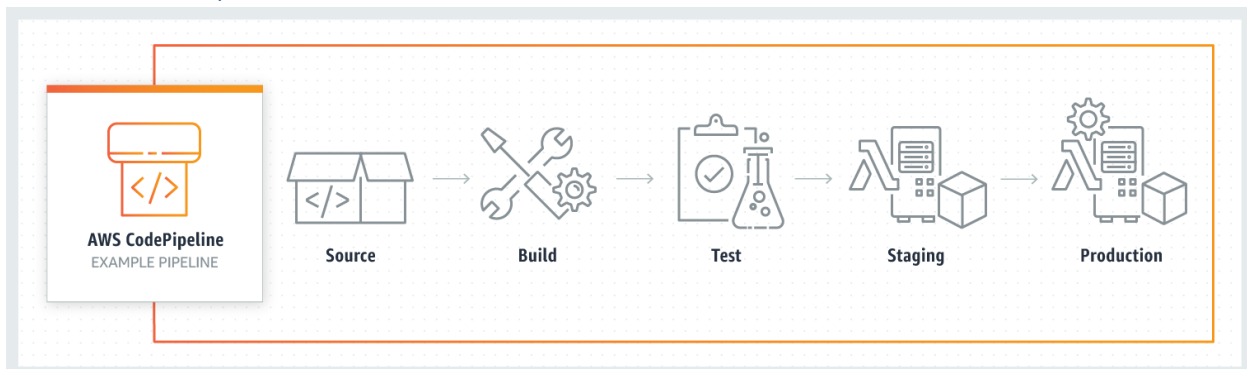
As aplicações de acesso (UI) devem consumir os micro serviços REST aqui implementados na tecnologia AWS Lambda (serverless) que definem um único ponto de acesso aos micro serviços e usam os mecanismos de controle de acesso do Amazon API Gateway que também gerencia as instâncias e expõe API de atualizações (Developer Portal) para implementar o CI/CD com integração ao Workflow da metodologia AGIL usando por exemplo o Atlassian - JIRA e o Jenkins, permitindo um modelo *Canary Release* com migrações controladas de versões.

Para a gestão de acessos e identificar atividades suspeitas um CloudTrail permite criar, extrair e analisar os LOGS. ( <https://docs.aws.amazon.com/apigateway/latest/developerguide/cloudtrail.html> )

A gestão de acesso e autorizações é feita por Lambda Authorizers que permitem implementar o controle de acesso via JWT/AUTH2 na aplicações de usuários (<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-use-lambda-authorizer.html> ).

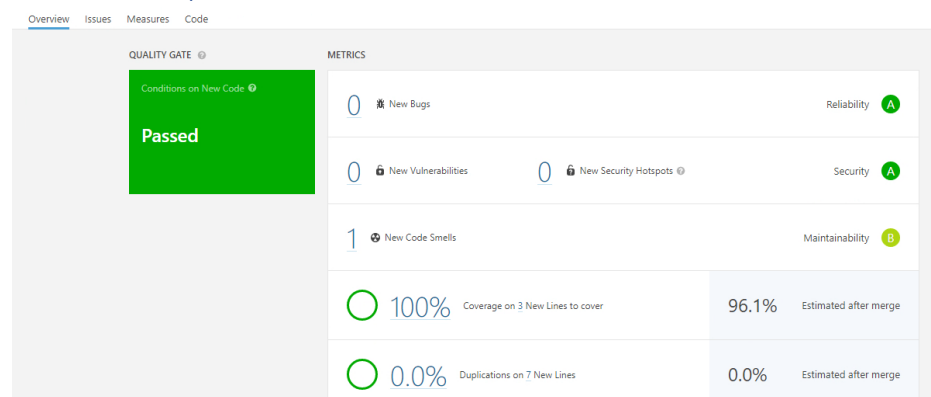
Amazon EMR – serviço de banco de dados – BIGDATA – abstrai as questões de disponibilidade e escalabilidade além de garantir e disponibilizar ferramentas de acesso e API que viabilizam acessos administrativos e bem como amostragens e aprendizados para ferramentas de B.I. e I.A.

## CI/CD - Code Pipeline

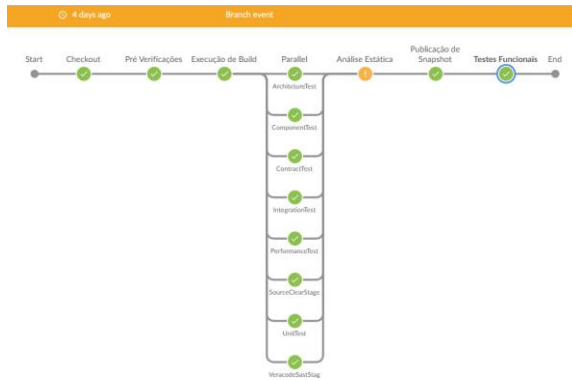


No processo de criação e instalação dos micro serviços é importante o controle de versões e a viabilização de uma evolução constante com o mínimo de impacto nas versões existentes (CI/CD) o Code Pipeline da AWS propõe facilidades para a criação de esteira de desenvolvimento atrelada das tarefas definidas em um contexto de desenvolvimento definidos nas equipes ágeis (AGILE) sendo já integrado ao ambiente AWS.

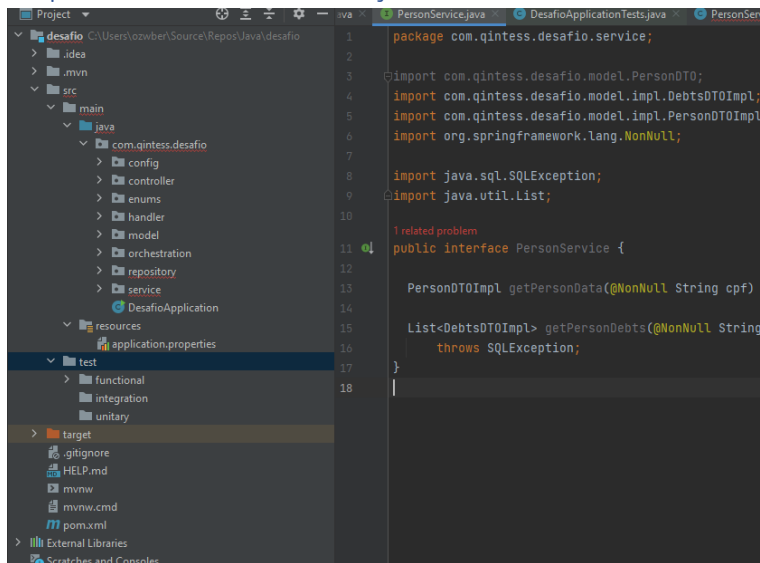
## Políticas de qualidade de software – Sonar-cube



## Build pipeline Jenkins



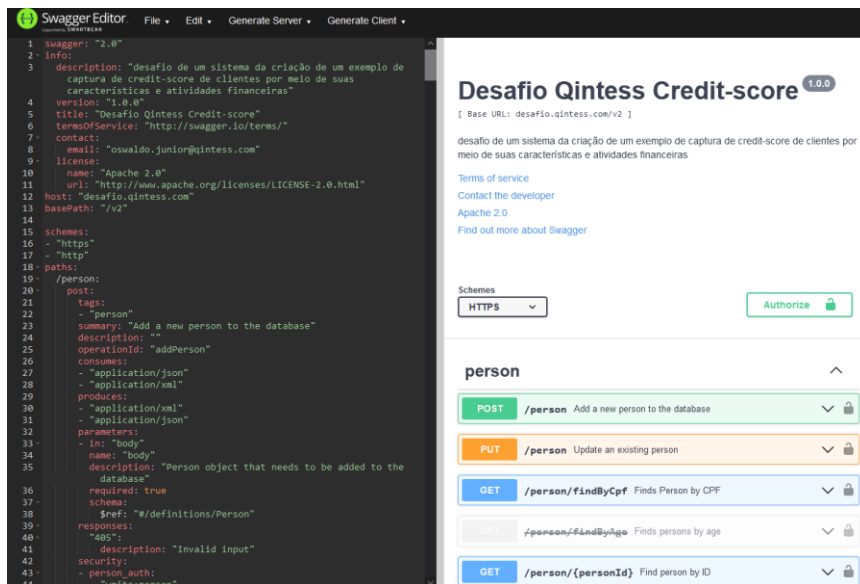
## Arquitetura do micro serviço



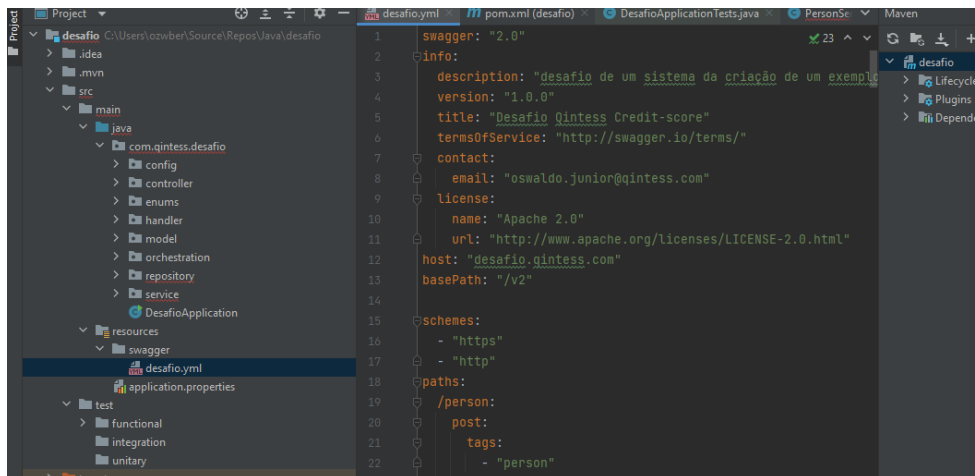
A separação em camadas bem definidas garante a “manutibilidade” do sistema com um menor impacto.

Nesta estrutura ficam separadas as classes de configuração, interface com cliente, as constantes enum e etc. de maneira a criar um padrão de desenvolvimento facilmente reconhecível através de pastas dedicadas.

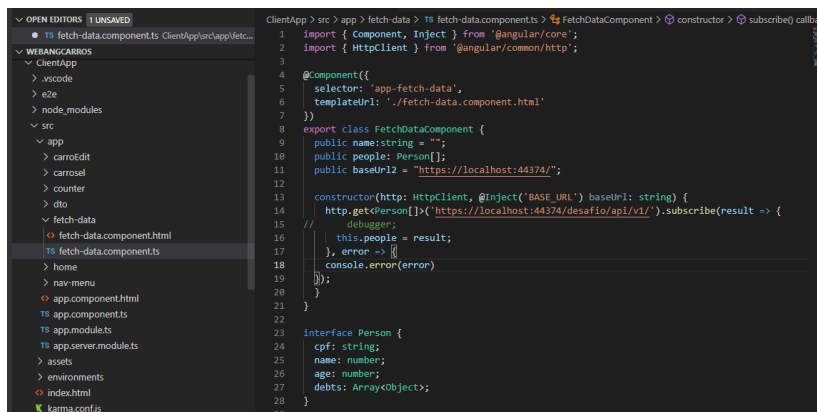
As interfaces REST podem ser definidas por arquivos YAML na ferramenta **SWAGGER.IO**



O arquivo YAML gerado pode ser adicionado ao projeto e depois, usando o plugin Swagger-codegen são geradas as interfaces das classes que definirão as chamadas REST do projeto.



Uma vez publicado o endpoint do micro serviço o mesmo pode ser integrado à aplicação, por exemplo Angular



## Conclusão

Usando os recursos aqui apresentados teremos um sistema robusto e estruturado para atender a grandes e crescentes demandas de acessos aliadas a flexibilidade de implementação e facilmente adaptável à futuras evoluções tecnológicas.

<b>Autor</b>	<b>data</b>	<b>Versão</b>
Oswaldo.junior@qintess.com	20-abr-22	1.0