**DOCUMENTAÇÃO B-HEALTH**

Processo Seletivo | Especialista Tech Lead | Entrega Desafio

Rayans Carvalho Santos

**CASE**

Você é o novo tech lead da squad que cuida do sistema de agendamento de exames, que está com os seguintes problemas:

1. Diariamente todos os hospitais enviam via FTP um arquivo para atualizar os exames, que são processados por um robô que faz a classificação e inclusão no banco de dados. Neste processo ocorrem diversos erros, como processamento de arquivos muito grandes, concorrência de I/O, falha no upload/download do arquivo, que deixam os exames desatualizados. Ficou definido com os hospitais que a comunicação não deverá mais ser feito por arquivos.
   * Qual solução você daria para essa integração? Descreva a solução e a arquitetura que você faria.
2. Existe um front-end que o cliente final pode agendar exames, realizar o pagamento e consultar os dados do hospital. Da forma que foi construída, cada consulta dos exames abre uma conexão com o banco e nosso banco permite até 100 conexões simultâneas. Com novos hospitais sendo integrados, hoje já tem mais de 1000 usuários acessando o site ao mesmo tempo, com muitas reclamações de performance e erros de consulta.
   * Qual solução você daria para que suporte o aumento contínuo de usuários simultâneos, sem envolver alguma alteração de hardware (servidores e bancos)? Descreva a solução e a arquitetura que você faria.
3. Pensando nas soluções propostas dos itens anteriores, quais boas práticas você aplicaria no processo de entrega em produção, pensando nos conceitos de disponibilidade, qualidade e segurança?

**SOLUÇÃO 01**

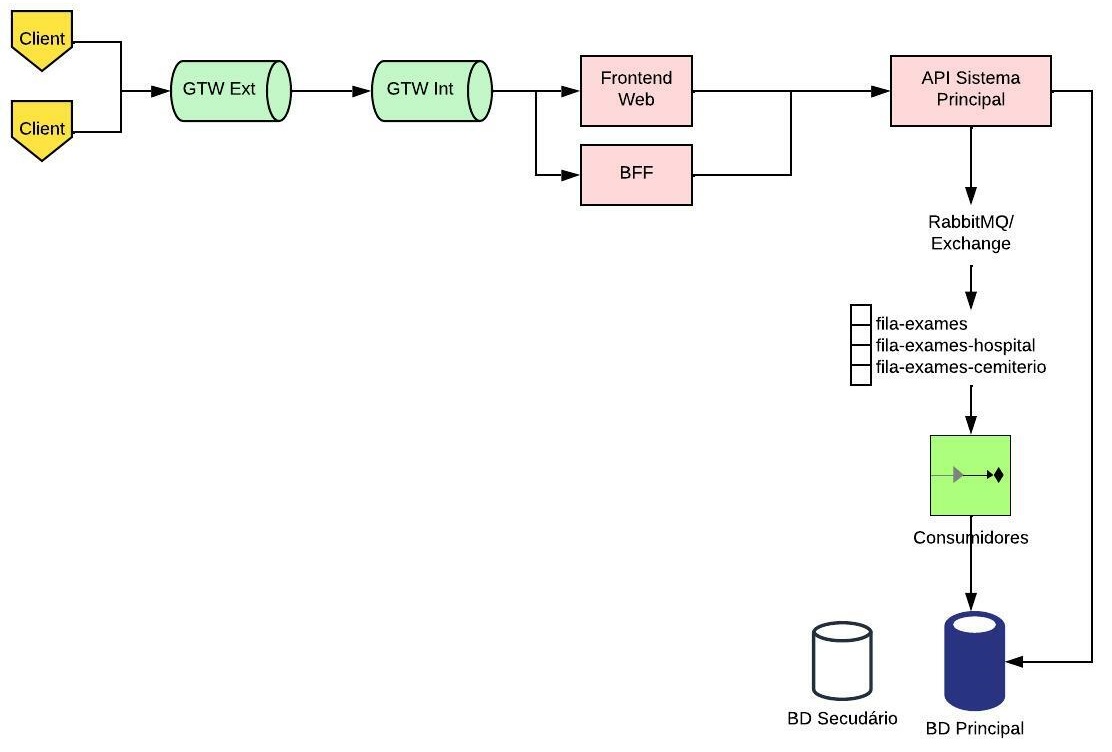
Foram desenvolvidos duas soluções baseadas na mesma arquitetura, porém com implementações diferentes.

A arquitetura utilizada:

* Cada hospital/Client iria acessar um frontend via WEB e/ou apps mobile.
* As rotas do front passariam por uma infra de gateway externo e interno.
* Há um BFF para os apps (solucao-01.1).
* As rotas referentes aos sistemas da empresa ficariam todos controlados pelo gateway interno.
* Com o backend do sistema apartado do frontend, as inclusões de exames seriam efetuadas por mensageria utilizando RabbitMQ.
* Utilizaríamos como parte da resiliência do sistema envelvendo a mensageria, fila de hospital e cemiterio.
* Há consumidores para a fila, podendo ser paralelizados suas instâncias.
* A persistência poderia ser feita em um banco SQL ou NoSQL, com bancos secundarios de backup, para questões de "Disaster Recovery".

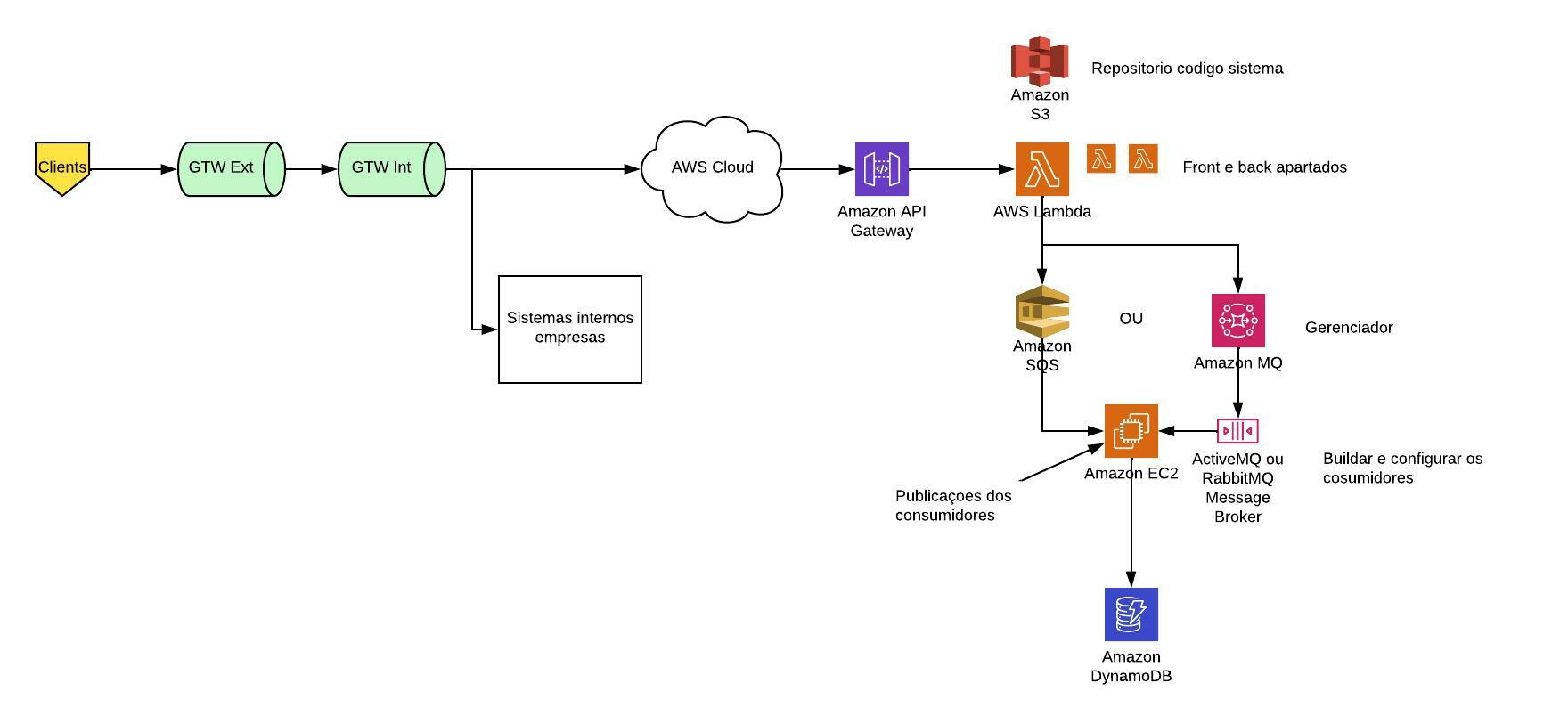
Solução 01.1

* Toda infra/sistemas são mantidos pela empresa.

Desenho [](https://github.com/rayanscs/b-health/blob/main/Solucao-01/Solucao-01-1-b-health.jpeg)

Solução 01.2

* Toda infra/sistemas são mantidos na AWS.

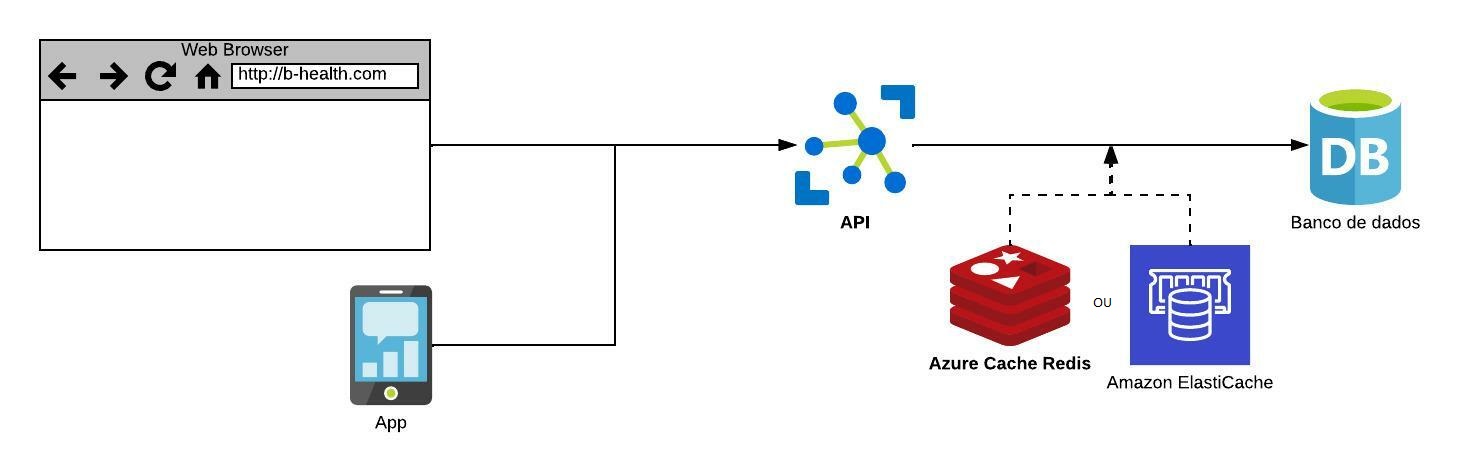
Desenho [](https://github.com/rayanscs/b-health/blob/main/Solucao-01/Solucao-01-2-b-health.jpeg)

Geral

* APIs desenvolvidos com modelagem de software em Domain Driven Design.
* Fronts desenvolvidos em arquitetura MVVM e modulação de consumo de serviços.

**Solução 02**

* Projeto de frontend apartado do backend.
* Desenvolver API para o agendamento de exames, pagamentos, consultas dos dados do hospital. Com isso, o front consome suas rotas.
* Aplicar o pattern Singleton para conexão com o banco de dados.
* Utilizar Repository Pattern para isolar a camada de acesso a dados (DAL), impulsionando o uso de injeção de dependência.
* Para CRUDs, utilização de algum Micro ORM para usar "query pura", caso seja um BD SQL.
* Criação de índices em banco de dados caso necessário.
* Para CRUDs NoSQL, otimização de estrutura json caso necessário (genérica adaptativa).
* Aplicação de cache no Redis.

Desenho [](https://github.com/rayanscs/b-health/blob/main/Solucao-02/Solucao-02-b-health.jpeg)

**Solução 03**

Boas práticas no processo de entrega em produção, pensando nos conceitos de disponibilidade, qualidade e segurança.

Ponto DISPONIBILIDADE:

* Servidores das aplicações com redundância e configurações de balanceamento de cargas (F5 por exemplo).
* Utilização da AWS ou qualquer outra solução em nuvem.
* Utilização de Docker para conteinerização.
* Persistencia de dados com boas prtáticas de Disaster Recovery.
* Utilização de monitoramento das aplicações e verificação de logs com ferramentas como New Relic, ElastichSearch/Kibana.
* Automação de acionamento de responsáveis pelas aplicações e/ou plantão.
* Esteira de CI/CD.

Ponto QUALIDADE

* Processos de CAB/Candidate para testes das aplicações, com isso efetuando teste regressivo, teste de carga.
* Reuniões de Chapter para disseminação de cenários ocorridos em PRD.
* Criação de testes unitários nos sistemas desenvolvidos.
* Execução de Code Review para análise de código.
* Aplicação de SonarQube para medição da qualidade do código desenvolvido.

Ponto SEGURANÇA

* Utilização de autenticação de serviço entre aplicações.
* Utilização de HTTPS.
* Dados sensiveis no header da reuqisição.
* Aplicação de replace token para senhas nas aplicações.