

DOCUMENTO DE VISÃO DA SOLUÇÃO - CHALLENGE 2024 - 4SI – FIAP
TELEFÔNICA VIVO

O documento de visão deve ser objetivo, com as informações que vocês têm no momento, ou seja, deve apresentar o que pretendem realizar como solução ao problema proposto. Embora não haja um *template* fixo para o Documento de Visão, recomendamos alguns itens que um bom Documento precisa ter, minimamente.

Importante: cada item precisa trazer a quantidade de informações suficientes para seu entendimento, não economize na descrição, mas seja objetivo.

Nome e Descrição do Cliente (Empresa):

A empresa em foco no Challenge é a Vivo, uma operadora de telecomunicações que cresceu ao comprar várias empresas menores. Isso deixou a empresa com muitos sistemas diferentes, dificultando a gestão de uma grande base de clientes.

A Vivo oferece serviços como celular, telefone fixo, TV por assinatura e internet fibra óptica e afins. Seu aplicativo principal, o "Vivinho", enfrenta desafios para se conectar com esses sistemas antigos e para organizar os dados de forma eficaz.

Nome do projeto:

VivoConnect

Nome do time e integrantes:

Nome do time: VIVXS
Nome dos integrantes:
Thiemi Soubhia Doi
Yasmim Rodrigues Bussoni
Ana Flavia Furtado

Histórico de revisões deste documento:

Versão (0.0)	Data (DD/MM/YYYY)	Autor	Descrição (O que fez no doc?)
0.1	03/04/2024	Yasmim Bussoni	Planejando dos componentes da arquitetura (Api Gateway, virtualização e Data Mesh)
0.2	10/04/2024	Ana Flavia	Aprimoramento dos componentes visando melhor eficiência e escalabilidade.

0.3	12/04/202	Thiemi Soubhia	Ajustes no cronograma do projeto
-----	-----------	----------------	----------------------------------

Escopo (o que o desafio abrange e contempla):

O desafio é desenvolver uma solução tecnológica que melhore a forma como os diferentes sistemas antigos da Vivo funcionam juntos. O objetivo é tornar tudo mais rápido e confiável, garantir que os dados sejam consistentes e diminuir o tempo que o aplicativo "Vivinho" leva para acessar esses dados. A solução deve organizar os dados de forma padronizada e facilitar o acesso a eles por meio de APIs, que são as pontes de comunicação utilizadas pelo app "Vivinho".

Não Escopo (o que não será feito deve ser esclarecido para alinhar expectativas):

Não iremos fazer a substituição dos sistemas antigos, devemos melhorar a forma como eles interagem. Não iremos trabalhar o front-end e desenho de telas, tudo deve ser realizado e validado a arquitetura pelo back-end.

Oportunidades de Negócio identificadas (alto nível - preliminares):

Identificamos algumas oportunidades de negócio importantes para a Vivo. Ao melhorar a rapidez e a confiabilidade do app Vivinho, podemos aumentar a satisfação dos clientes e diminuir as reclamações. A organização e o acesso facilitado aos dados permitem que a Vivo ofereça serviços mais personalizados e tome decisões acertivas. Impulsionando inovação e crescimento.

Posicionamento

Descrição do problema

O problema detectado é	Muitos sistemas diferentes na Vivo tornam o app Vivinho lento e propenso a erros. O que dificulta a integração eficiente e aumenta a latência nas comunicações entre o app principal, Vivinho, e os sistemas de back-end
Afeta	A experiência dos usuários do app Vivinho, causando frustrações e reclamações. Além de impactar a eficiência operacional interna da Vivo.
O impacto é	Clientes insatisfeitos e custos operacionais mais altos para a Vivo, que precisa manter vários sistemas.
A solução ideal seria	Criar uma solução que simplifique e agilize a troca de informações entre o app Vivinho e os sistemas antigos da Vivo, melhorando a rapidez e confiabilidade do app.

--	--

Descrição e Visão Geral da solução (inspiração, motivação, propósito):

Nossa solução visa simplificar e otimizar o gerenciamento de dados, oferecendo uma plataforma robusta e eficiente. A inspiração vem da necessidade de lidar com a complexidade dos sistemas legados e garantir uma integração perfeita com novas tecnologias. Nosso propósito é proporcionar aos usuários uma experiência tranquila e segura, com acesso rápido e confiável às informações necessárias

Regras e/ou Restrições a considerar:

- 1 - Será realizado um monitoramento contínuo do sistema para identificar e corrigir eventuais problemas de desempenho, segurança ou disponibilidade em todas as etapas da arquitetura
- 2- Todas as comunicações entre os componentes do sistema serão criptografadas e autenticadas para garantir a segurança dos dados e serviços.
- 3- A disponibilidade do sistema será priorizada, com mecanismos de failover e fallback implementados para garantir a continuidade do serviço em caso de falhas.

Relação de Requisitos Funcionais e Não funcionais da solução:

Nº	Nome	Descrição
RF001	Caching Inteligente	Capacidade de armazenar dados em cache de forma inteligente para acelerar o acesso subsequente.
RF002	Compressão de Dados	Utilização de algoritmos de compressão para reduzir o tamanho dos dados e otimizar o uso do armazenamento e largura de banda.
RF003	Otimização de Queries	Melhoria no desempenho das consultas ao banco de dados, reduzindo o tempo de resposta.
RF004	Balanceamento de Carga	Distribuição equilibrada das solicitações entre os servidores para evitar sobrecarga e maximizar o desempenho.
RF005	Circuit Breakers	Mecanismos para interromper temporariamente a comunicação com um serviço que está sobrecarregado ou com problemas, evitando falhas em cascata.
RF006	Autoescala	Capacidade de aumentar ou diminuir automaticamente a capacidade dos recursos de acordo com a demanda.
RF007	Políticas de Acesso e Logs	Definição de políticas de acesso para controlar quem pode acessar quais recursos, juntamente com registros detalhados de todas as atividades do sistema.

Nº	Nome	Descrição
RF008	Roteamento Dinâmico	Capacidade de rotear dinamicamente as solicitações para os recursos mais adequados com base em várias métricas, como latência, carga e proximidade geográfica.
RF009	Monitoramento em Tempo Real	Monitoramento contínuo do sistema e das métricas de desempenho em tempo real, com alertas configuráveis para identificar e resolver problemas rapidamente.
RF010	Análise Preditiva com IA	Utilização de técnicas de inteligência artificial para prever tendências, identificar anomalias e tomar decisões proativas para otimizar o desempenho do sistema.
RF011	Logins: Validação de tokens de acesso	Verificação da autenticidade e autorização dos usuários por meio de tokens de acesso, garantindo a segurança do sistema.

Nº	Nome	Descrição
RNF001	Desempenho e Latência	Garantia de que o sistema responda rapidamente às solicitações dos usuários e mantenha baixa latência.
RNF002	Alta Disponibilidade e Resiliência	Capacidade de manter o serviço disponível mesmo em caso de falhas, com mecanismos de failover, fallback e redundância.
RNF003	Escalabilidade	Capacidade de aumentar ou diminuir a capacidade do sistema conforme necessário para lidar com diferentes volumes de tráfego.
RNF004	Garantia de Integridade de Dados	Asseguração de que os dados permaneçam consistentes e não sejam corrompidos, mesmo em situações de falha.
RNF005	Gestão de Dados	Implementação de políticas e práticas para garantir a qualidade, confiabilidade e segurança dos dados ao longo do tempo.
RNF006	Monitoramento e Troubleshooting	Monitoramento constante do sistema e capacidade de identificar, diagnosticar e resolver problemas rapidamente.
RNF007	Evolução Contínua e processo de correção de dados	Capacidade de realizar atualizações e melhorias no sistema de forma incremental e sem interrupções no serviço.
RNF008	Desacoplamento	Redução das dependências entre os componentes do sistema para facilitar a manutenção, atualização e escalabilidade.
RNF009	Segurança da Aplicação	Implementação de medidas de segurança para proteger o sistema contra ataques, vazamentos de dados e outras ameaças.

Os requisitos funcionais descrevem o comportamento da solução, ou seja, descrevem o trabalho que a solução deve realizar em itens, o que a solução faz, por exemplo, inserir, alterar dados, gerar relatório, consultar dados, relacionar dados, etc. Os requisitos não funcionais são os atributos de qualidade, por exemplo, velocidade, usabilidade, portabilidade, design, acessibilidade, etc.

Tecnologias e linguagens previstas para desenvolver a solução:

API Gateway: Gzip, AWS Elastic Load Balancer, Istio, Horizontal Pod Autoscaler (kubernetes), Prometheus, Google Cloud AI Platform e OAuth, NGINX

Middle de roteamento: Apache Kafta

Virtualização de dados: Denodo; Apache Spark MLlib; Apache ZooKeeper

Middleware de Roteamento: Amazon Redshift M; Drivers JDBC/ODBC; Apache Camel; : Apache Spark

Data Mesh: NoSQL, NLTK.

Linguagens de programação: Java Spring Boot e Python, Banco de dados Oracle

Benefícios previstos ou esperados com a solução:

Com a solução proposta, esperamos:

1. **Desempenho otimizado:** Respostas mais rápidas e eficientes (Graças ao caching inteligente, otimização de consultas e balanceamento de carga)
2. **Alta disponibilidade:** Sistema resiliente e sempre acessível (Com mecanismos de autoescalabilidade, circuit breakers e replicação de dados)
3. **Segurança reforçada:** Proteção avançada de dados. (A integração de políticas de acesso, logs e análise preditiva com IA)
4. **Manutenção simplificada:** Identificação rápida de problemas e resolução ágil.(Implementando roteamento dinâmico, circuit breakers e monitoramento em tempo real)
5. **Escalabilidade e flexibilidade:** Capacidade de adaptação a diferentes demandas.(arquitetura baseada em microsserviços)

Cronograma de execução:

Atividade	Março	Abril	Maió	Junho	Julho	Agosto	Setembro	Outubro
Definição dos requisitos da API Gateway	✓							
Planejamento da arquitetura inicial	✓							
Seleção das tecnologias a serem utilizadas	✓							
Desenvolvimento inicial da API Gateway		✓						
Configuração do balanceamento de carga e caching		✓						
Início do planejamento do Middleware de Roteamento		✓						
Continuação do desenvolvimento da API Gateway			✓					
Implementação das políticas de acesso e logs			✓					
Início do desenvolvimento do Middleware de Roteamento			✓					

Atividade	Março	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro
Finalização do desenvolvimento da API Gateway			✓					
Testes de integração e desempenho			✓					
Desenvolvimento do Middleware de Roteamento				✓				
Monitoramento e ajustes de desempenho				✓				
Início do desenvolvimento do Sistema de Virtualização de Dados				✓				
Configuração da replicação de dados e balanceamento de carga				✓				
Desenvolvimento do Middleware de Integração					✓			
Testes e ajustes finais no Sistema de Virtualização de Dados					✓			
Implantação do Middleware de Integração						✓		
Início do planejamento e desenvolvimento do Data Mesh						✓		
Finalização do desenvolvimento e implantação do Data Mesh							✓	
Testes finais de integração e documentação								✓