

Parte 5: Identificando se a Tecnologia Pode Escalar com o Negócio

Objetivo

Capacitar os analistas a identificar se a tecnologia, arquitetura e infraestrutura de uma startup são capazes de suportar o crescimento acelerado. Além disso, compreender o impacto de escalabilidade, débito técnico e infraestrutura nas operações futuras.

1. Discussão de Case do Mercado Livre

Case

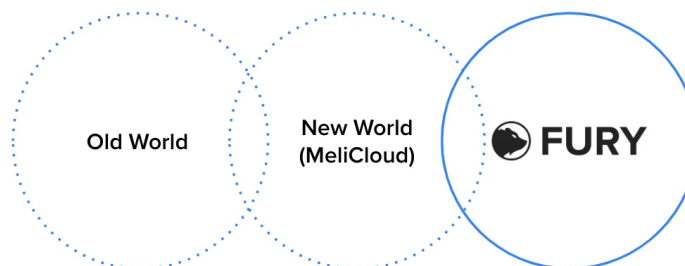
[The+Rise+of+Mercado+Libre.pdf](#)

Resumo do Case

- **Tecnologia Proprietária como Diferencial:** Desde o início, o Mercado Libre optou por desenvolver sua própria tecnologia ao invés de usar soluções prontas, garantindo flexibilidade para adaptar-se às necessidades específicas do mercado latino-americano.
- **Transformação Digital Estratégica:** O projeto "New World" em 2009 foi uma decisão crucial de atacar os desafios de escalabilidade de uma empresa em rápida expansão com o Mercado Libre, e transformar a arquitetura monolítica em uma plataforma baseada em APIs e microserviços, permitindo maior velocidade de inovação e escalabilidade.
- **Expansão Além do Marketplace:** A arquitetura de plataforma permitiu a expansão para novos negócios como:
 - Mercado Pago (fintech)
 - Mercado Envios (logística)
 - Mercado Crédito (empréstimos)
 - Mercado Shops (lojas online)
- **Vantagens Competitivas Sustentáveis:**
 - Capacidade de adaptar soluções para realidades locais
 - Integração entre diferentes serviços do ecossistema
 - Rapidez na implementação de novas funcionalidades
 - Escalabilidade para suportar crescimento acelerado
- **Lições Aprendidas:**
 - Uma tecnologia bem planejada permite expansão geográfica e diversificação de serviços.
 - Investimentos em infraestrutura e reestruturação técnica podem ser decisivos para escalar sem comprometer a inovação.



- A história continua.



Atividade em sala

MercadoLibre compra a su rival DeRemate en US\$ 40 millones

La empresa ligada a eBay pagará US\$22 millones en efectivo y emitirá un pagaré a la adquirida por los restantes US\$18 millones.

- Por que a decisão de desenvolver tecnologia proprietária foi tão crucial para o sucesso do Mercado Libre, em vez de usar soluções prontas como fez seu concorrente DeRemate?
- Como a transformação do "New World" em 2009 se relaciona com a capacidade da empresa de competir com a Amazon anos depois?
- Quais foram os trade-offs entre manter o sistema monolítico vs. migrar para microserviços?

2. O Que é Escalabilidade e Por Que é Importante?

Definição

Escalabilidade refere-se à capacidade de um sistema crescer para atender ao aumento da demanda sem degradação significativa no desempenho ou aumento proporcional dos custos.

Por Que É Crucial?

- Startups frequentemente enfrentam crescimento explosivo de usuários ou transações, exigindo sistemas que acompanhem esse ritmo.

- Sistemas não escaláveis podem gerar falhas críticas, insatisfação do cliente e perda de receita.
- O planejamento inadequado da escalabilidade pode comprometer a capacidade da startup de capturar oportunidades de mercado e crescimento.

Fatores-Chave na Escalabilidade

1. Performance

- Capacidade de manter tempos de resposta baixos mesmo com picos de tráfego.

2. Custos Operacionais

- Escalabilidade eficiente deve permitir crescimento sem custos desproporcionais.

3. Confiabilidade

- Sistemas devem manter operação estável mesmo com falhas parciais ou manutenções.

4. Disponibilidade

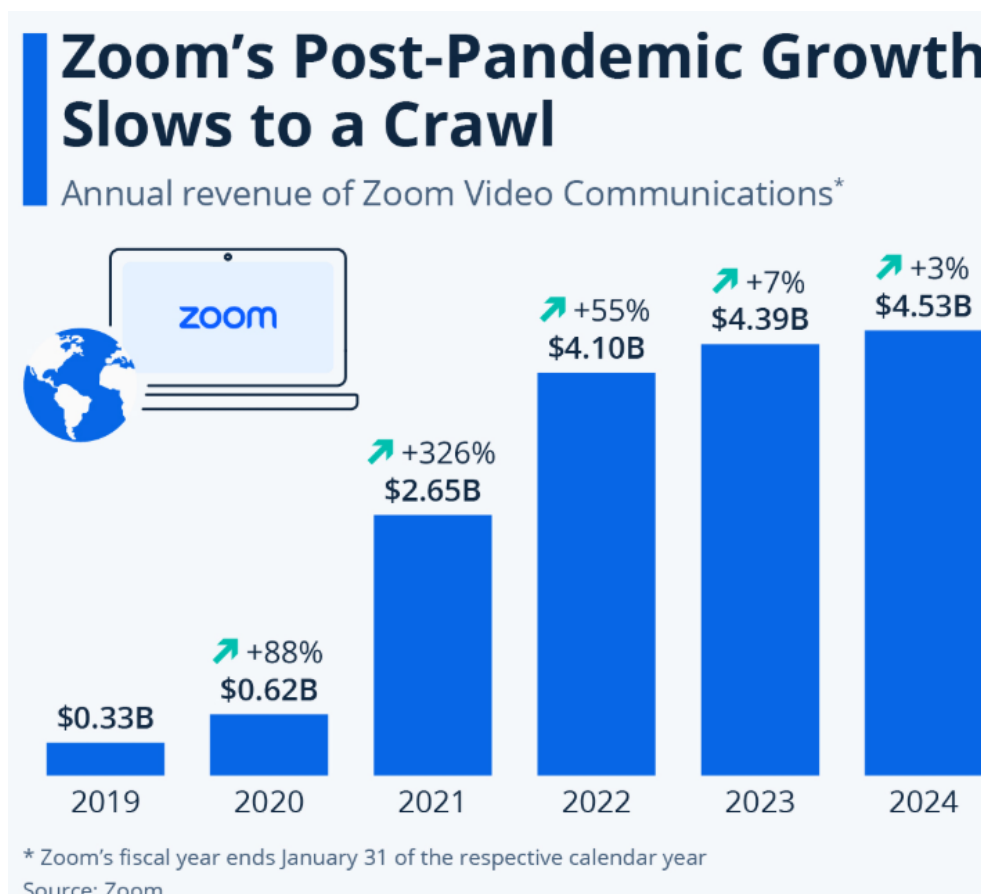
- Capacidade de manter o serviço acessível 24/7 com redundância e recuperação rápida de falhas.

• Flexibilidade

- Sistemas devem suportar mudanças rápidas, como entrada em novos mercados ou lançamento de novos produtos.

Case: Zoom durante a Pandemia (2020-2022)

Empresas SaaS como Zoom, que enfrentaram aumento exponencial de usuários durante a pandemia, demonstraram como a escalabilidade técnica permite atender picos extremos de demanda como durante a pandemia.



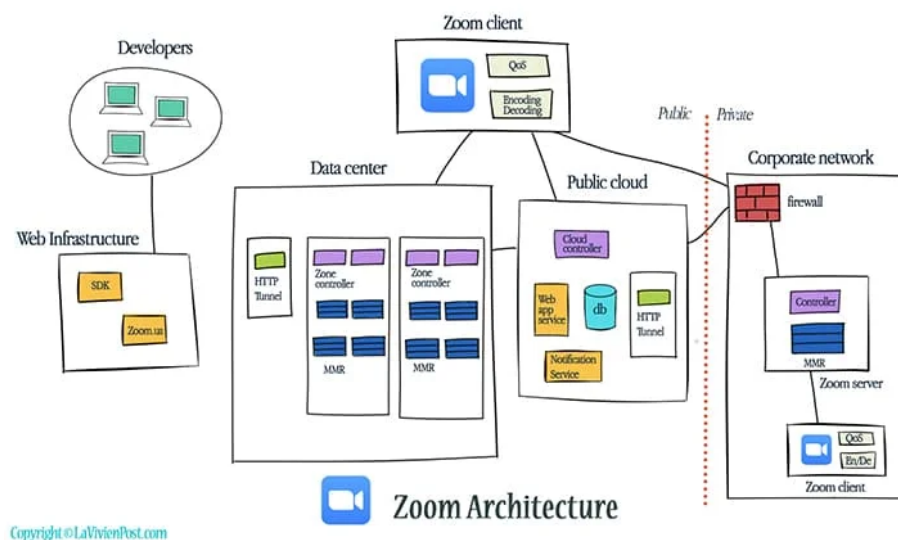
Embora o crescimento do Zoom tenha desacelerado a partir de 2023, 2021 e 2022 representaram uma explosão de crescimento que só foi possível por que estavam preparados.

O vídeo abaixo fornece mais detalhes sobre como a arquitetura do Zoom funciona:

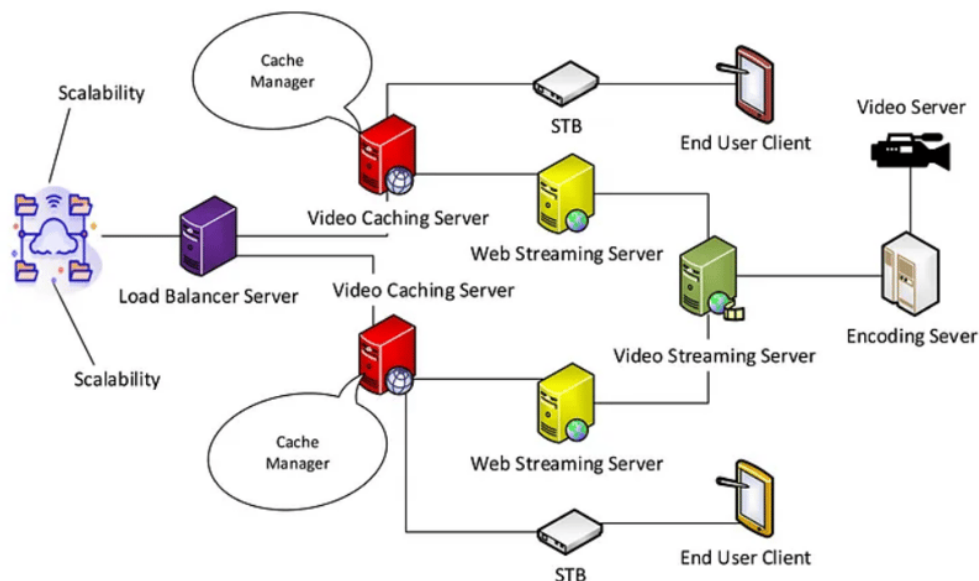
https://www.youtube.com/watch?v=CgriozF_U20

Capacidade do Zoom durante a pandemia:

# reuniões por dia	300 milhões
# participantes por dia	100~300 bilhões
# participantes por reunião	(Enterprise users) 300~500 (Basic or Pro users) 100~200
# reuniões por VM	100~200
# de data centers	17



Arquitetura que permitiu o Zoom escalar durante o pico reuniões ativas durante a pandemia



Detalhamento da parte de streaming e metadados da arquitetura do Zoom

3. Arquitetura de Software: Uma Introdução Indispensável

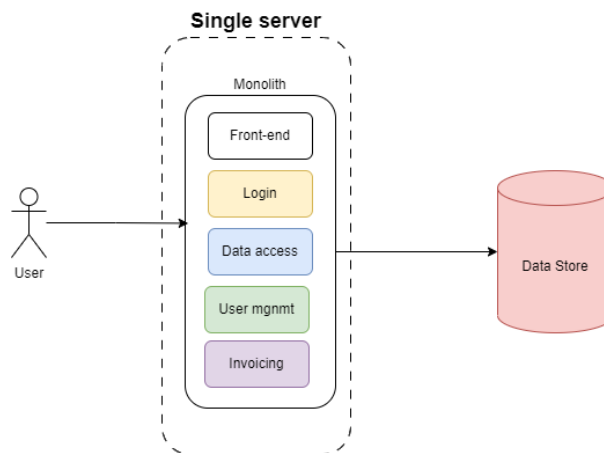
O Que É Arquitetura de Software?

- É a estrutura geral dos componentes de um sistema, incluindo como eles interagem entre si e com o ambiente externo.

Modelos de Arquitetura Comuns

Existem mais de 15 padrões de arquitetura, porém vamos detalhar somente os mais comuns.

1. Monolito



Abordagem tradicional onde a aplicação é construída como uma única unidade, com todos os componentes integrados em uma base de código comum.

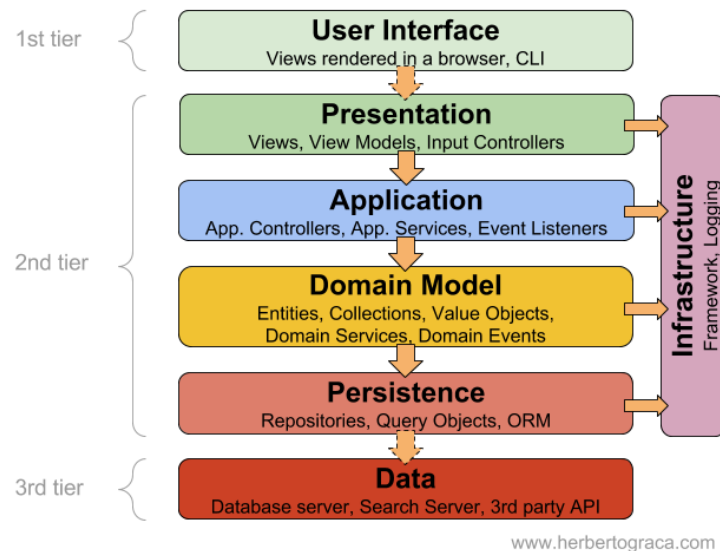
Casos de Uso:

- Aplicações Pequenas: Ideal para projetos simples e MVPs
- Prototipagem: Desenvolvimento inicial rápido
- Sistemas Legados: Aplicações antigas em produção

Desvantagens:

- Difícil de escalar, pois requer duplicação completa
- Manutenção complexa devido ao alto acoplamento
- Atualizações arriscadas por afetar todo o sistema

2. Em Camadas



Arquitetura em camadas organiza o software horizontalmente, dividindo em níveis como apresentação, lógica e dados. É modular e comum em aplicações web.

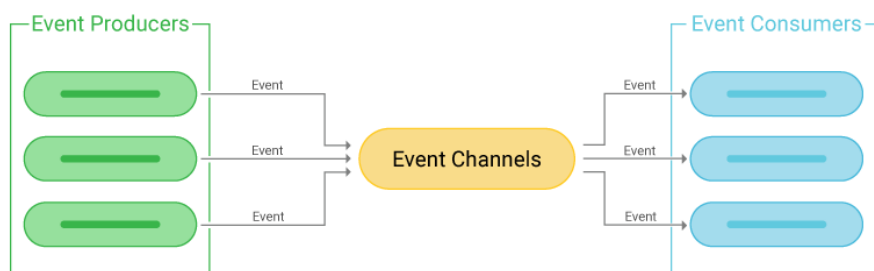
Casos de uso:

- E-commerce: Separa interface, lógica e dados para atualizações eficientes.
- Bancos: Separa interações, transações e armazenamento.
- CMS: Divide apresentação, gerenciamento e armazenamento.

Desvantagens:

- Comunicação entre camadas pode afetar performance.
- Risco de acoplamento forte entre camadas.
- Complexidade aumenta com múltiplas camadas.

3. Orientado à Eventos



Usa eventos assíncronos para comunicação entre componentes, respondendo a ações do usuário ou mudanças nos dados. Ideal para sistemas em tempo real.

Casos de Uso:

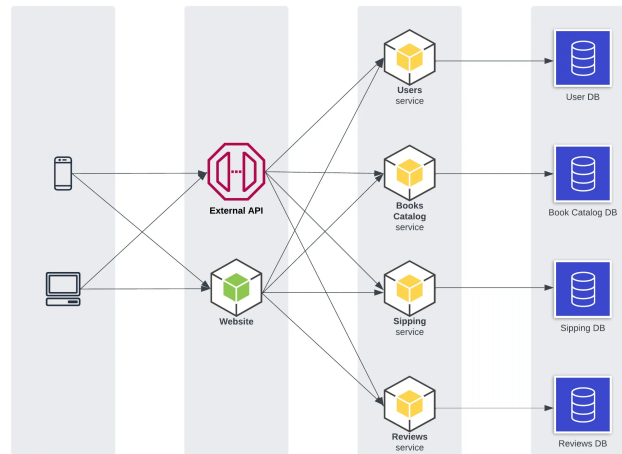
- Mídia Social: Atualizações em tempo real para posts, curtidas e comentários.
- Negociação de Ações: Execução automática de ordens baseada em eventos do mercado.
- Casa Inteligente: Automação baseada em sensores e comandos do usuário.

Desvantagens:

- Depuração complexa por fluxos não lineares

- Problemas de timing (race conditions) podem causar comportamentos inesperados
- Excesso de eventos pode complicar a arquitetura

4. Microsserviços



Divide aplicações em serviços independentes e pequenos que podem ser implantados separadamente, oferecendo melhor escalabilidade. Popular em sistemas cloud.

Casos de Uso:

- E-commerce: Microsserviços para usuários, produtos, pagamentos e pedidos.
- Transporte: Serviços para autenticação, corridas, rastreamento e pagamentos.
- Streaming: Serviços para conteúdo, perfis, recomendações e cobranças.

Desvantagens:

- Complexidade no gerenciamento distribuído
- Desafios na consistência entre serviços
- Impacto de performance na comunicação

5. Híbrido

Mistura diferentes padrões de arquitetura para atender necessidades específicas, como microsserviços com eventos.

Casos de Uso:

- Sistemas Empresariais: Combina padrões para atender diferentes necessidades.
- Aplicações Web: Une microsserviços e eventos para melhor responsividade.
- Análises em Tempo Real: Integra eventos e processamento espacial eficiente.

Desvantagens:

- Complexidade no gerenciamento.
- Desafios de integração.
- Necessita planejamento cuidadoso.

Tabela de Resumo

Padrão de Arquitetura	Agilidade	Facilidade de desenvolvimento	Testabilidade	Performance	Escalabilidade	Facilidade de deploy
Monolito	Baixa - componentes altamente acoplados atrapalham agilidade	Alta - um único repositório facilita o desenvolvimento	Alta - componentes podem ser testados dentro do monolito	Variável - Chamadas de métodos diretamente pode gerar eficiência	Baixa - significa duplicar o monolito	Baixa - necessita deployar todo o monolito
Em Camadas	Alta - promove modularidade e separação de responsabilidades	Alta - clara separação possibilita desenvolvimento focado	Alta - Isolamento de camadas facilita testes unitários	Baixa - Pode introduzir sobrecarga por causa das múltiplas camadas	Baixa - pode ser restrito devido às dependências entre as camadas	Alta - separações claras facilitam deploy
Orientado à Eventos	Alta - componentes desacoplados flexibilizam mudanças	Alta - componentes reagem a eventos de forma independente	Alta - eventos podem ser simulados para teste	Alta - eficiente para sistemas assíncronos e reativos	Alta - pode ser escalado distribuindo o processamento de eventos	Alta - processadores de eventos podem ser atualizados de forma independente
Microserviços	Alta - serviços podem ser desenvolvidos independentes	Alta - times pequenos podem focar nos serviços	Alta - serviços podem ser testados isoladamente	Variável - depende da interação entre os serviços	Alta - serviços podem ser escalados de forma independente	Alta - serviços podem ser deployados separadamente
Híbrido	Alta - combina as vantagens de múltiplas arquiteturas	Variável - depende da abordagem	Variável - depende dos componentes	Variável - depende da composição	Variável - depende da estrutura	Variável - depende da configuração

Por Que Isso Importa?

- Arquiteturas escaláveis, como microserviços, permitem que startups adicionem funcionalidades sem comprometer o sistema principal.
- Escolhas erradas de arquitetura podem levar ao acúmulo de débito técnico e problemas de performance.

4. Lei de Conway, Topologia de Times e Arquitetura de Software

Lei de Conway

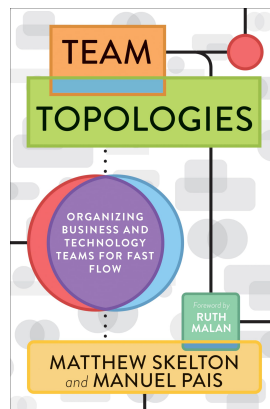
A Lei de Melvin Conway descreve a conexão entre a estruturas organizacionais e os sistemas que elas projetam:

“Organizações projetam sistemas que refletem as estruturas de comunicação de suas equipes.”

- Exemplo: Empresas com times separados de front-end e back-end tendem a ter arquiteturas que espelham essa divisão. O Spotify demonstra isso com suas "squads" autônomas que naturalmente evoluíram para microserviços independentes.

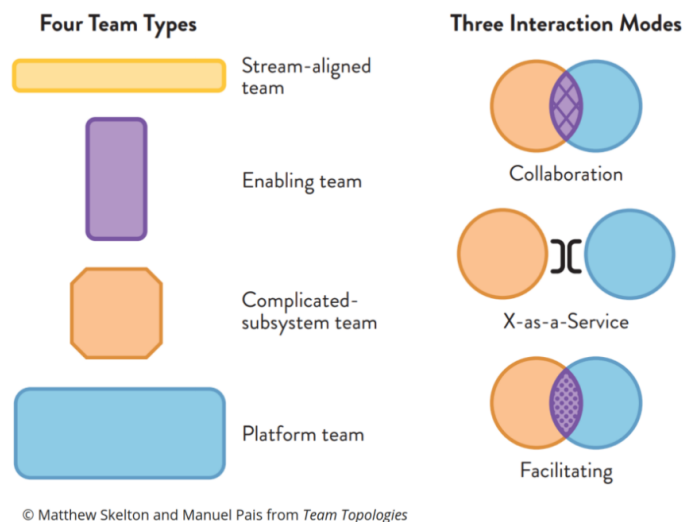
A Lei de Conway estabelece que a estrutura de um sistema reflete a organização que o criou, pois os desenvolvedores precisam se comunicar para garantir compatibilidade entre componentes. Assim, produtos complexos tendem a espelhar a estrutura organizacional de suas equipes. Embora seja mais aplicada em arquitetura de software, seus princípios se estendem a outros campos técnicos.

Topologias de Times (Livro *Team Topologies*)



Livro Team Topologies

Articula padrões de desenho organizacionais (tipos de times e modos de interação) para garantir que os sistemas sejam produzidos como projetados com o mínimo de gargalos e barreiras.



Tipos de times e modos de interações entre times

- **Tipos de Times**

1. **Times Alinhados ao Fluxo de Valor:** Focados em entregar valor direto ao cliente.
2. **Times de Plataforma:** Fornecem ferramentas e infraestrutura para outros times.
3. **Times Facilitadores:** Auxiliam outros times com expertise técnica.
4. **Times de Subsistemas Complexos:** Especialistas em componentes complexos específicos.

- **Modos de Interação**

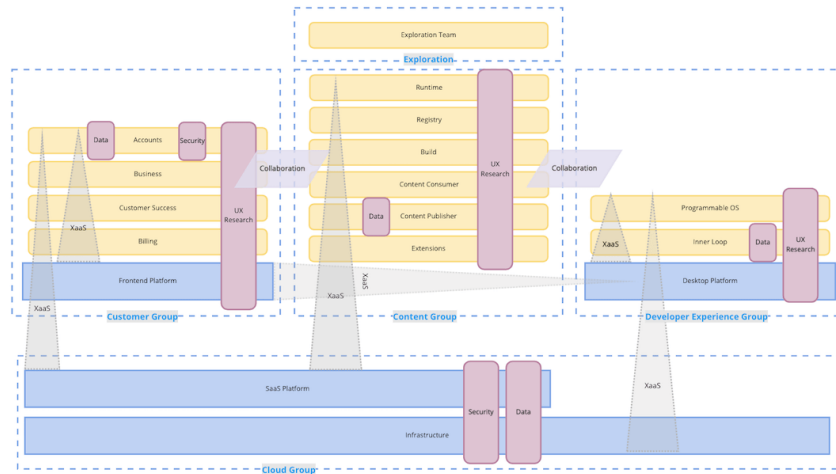
1. **X-as-a-Service:** Times trabalham juntos como prestadores de serviço.
2. **Facilitação:** Times compartilham conhecimento e boas práticas.
3. **Colaboração:** Times trabalham juntos em objetivos compartilhados.

Alinhamento de Estrutura e Arquitetura

É fundamental que as lideranças técnicas possuam conhecimentos de arquitetura de software ou trabalhem em parceria com arquitetos experientes ao desenvolver estratégias e redesenhos organizacionais. Para isso, existem

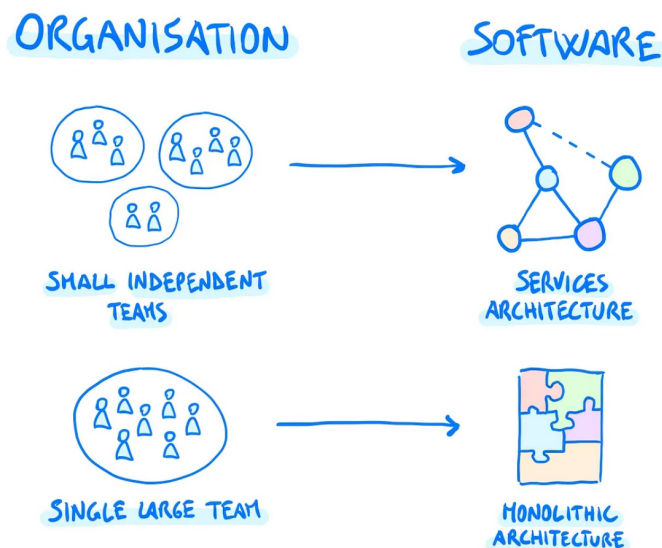
ferramentas específicas, como os padrões do Team Topologies e o "Reverse Conway Maneuver".

O "Reverse Conway Maneuver" é uma estratégia na qual a organização é estruturada para espelhar a arquitetura de sistema desejada. Por exemplo, quando uma empresa planeja migrar para microsserviços, primeiro reorganiza suas equipes em unidades pequenas e autônomas. Essa abordagem facilita a transição técnica, pois a estrutura organizacional já está alinhada com a arquitetura pretendida.



Estrutura organizacional e arquitetura do Docker.

Red Flags Comuns



Times organizados em Operação e Inovação

A separação entre times de "operação" e "inovação" cria silos prejudiciais e diminui a eficiência organizacional. Os times de operação ficam desmotivados e desconectados do desenvolvimento do produto, enquanto os times de inovação tendem a criar soluções que não refletem as necessidades operacionais reais. Essa divisão artificial gera conflitos de prioridade e fragmenta o conhecimento institucional.

Times funcionais

A organização em times puramente funcionais (front-end, back-end, mobile, etc.) pode criar silos técnicos que dificultam a entrega de valor ao cliente. Essa estrutura tende a gerar gargalos de comunicação e dependências

entre equipes, prejudicando a velocidade de desenvolvimento. Além disso, times funcionais frequentemente perdem a visão do produto como um todo, focando apenas em suas especialidades técnicas.

Monolito vs Múltiplos Times

Ter múltiplos times trabalhando em um monolito pode gerar conflitos de desenvolvimento e integração, pois várias equipes precisam coordenar mudanças no mesmo código-fonte. Isso frequentemente resulta em ciclos de desenvolvimento mais lentos e maior risco de introdução de bugs. A situação é agravada quando não há uma clara governança do código ou processos bem definidos para gerenciar as contribuições de diferentes times.

Time pequeno vs Muitos Microserviços

Times pequenos gerenciando muitos microserviços podem ficar sobrecarregados com a complexidade operacional. Isso pode levar a problemas de manutenção, monitoramento inadequado e resposta lenta a incidentes. A sobrecarga cognitiva de lidar com múltiplos serviços pode comprometer a qualidade do código e a capacidade de inovação do time.

Por Que Isso Importa?

Compreender a Lei de Conway e as topologias de times é crucial para startups em crescimento porque:

- Permite alinhar a estrutura organizacional com a arquitetura técnica desejada, reduzindo barreiras de comunicação e aumentando a eficiência.
- Ajuda a identificar e resolver problemas de coordenação antes que impactem o desenvolvimento do produto.
- Facilita a evolução do sistema conforme a empresa cresce, mantendo a agilidade e qualidade técnica.

5. Infraestrutura e Impacto na Escalabilidade

Infraestrutura de TI: O que é?

Infraestrutura de TI refere-se ao conjunto de componentes tecnológicos fundamentais que suportam as operações de uma empresa, incluindo hardware, software, redes, data centers e sistemas de armazenamento. É a base que permite que aplicações e serviços funcionem de maneira eficiente e segura. Uma infraestrutura bem planejada é essencial para garantir disponibilidade, performance e escalabilidade dos sistemas.

Infraestrutura de TI: Por que é importante?

A infraestrutura de TI é crucial para garantir várias dimensões importante de um negócio de tecnologia, dentre estas dimensões estão:

1. **Escalabilidade:** Capacidade de aumentar recursos e performance conforme o crescimento da demanda do negócio, permitindo que a infraestrutura se adapte de forma eficiente às necessidades em constante evolução.
2. **Agilidade e Flexibilidade:** Capacidade de responder rapidamente às mudanças do mercado e demandas dos usuários, permitindo que a empresa implemente novas soluções e adapte seus sistemas de forma ágil, mantendo-se competitiva no mercado.
3. **Integração e Interoperabilidade:** Capacidade de diferentes sistemas e aplicações se comunicarem e trabalharem juntos de forma eficiente, permitindo o compartilhamento de dados e recursos entre diferentes departamentos e sistemas, além de facilitar a integração com parceiros externos e novos serviços.
4. **Melhor Tomada de Decisão:** Uma infraestrutura robusta fornece dados precisos e em tempo real, permitindo que gestores tomem decisões mais informadas e estratégicas com base em análises confiáveis e insights acionáveis.
5. **Inovação e Diferenciação:** Uma infraestrutura robusta permite que empresas experimentem novas tecnologias, desenvolvam produtos inovadores e se diferenciem da concorrência através de recursos técnicos avançados. Isso possibilita a rápida implementação de novas funcionalidades e a adaptação ágil às mudanças do mercado.

6. **Melhor Experiência do Cliente:** Uma infraestrutura bem projetada oferece serviços rápidos e confiáveis, com alta disponibilidade e tempo de resposta reduzido. Isso resulta em uma experiência fluida para os clientes, minimizando frustrações e aumentando a retenção. O desempenho consistente fortalece a confiança na marca.
7. **Eficiência de Custo:** Uma infraestrutura bem planejada reduz custos operacionais através de automação e gerenciamento eficiente de recursos. O dimensionamento dinâmico evita gastos com capacidade ociosa, enquanto a consolidação de sistemas elimina redundâncias desnecessárias.
8. **Melhor Segurança e Conformidade:** Uma infraestrutura robusta permite implementar controles de segurança eficazes e manter conformidade regulatória. Isso inclui proteção contra ameaças, gestão de identidade e auditoria. A segurança integrada facilita adaptação a novas regulamentações conforme o negócio cresce.

Infraestrutura de TI: Os Componentes

Os principais componentes de uma infraestrutura de TI incluem:

- **Hardware:** Servidores, computadores, dispositivos de rede e equipamentos físicos
- **Software:** Sistemas operacionais, aplicações e ferramentas de gerenciamento
- **Redes:** Conectividade, switches, roteadores e firewalls
- **Armazenamento:** Sistemas de backup, bancos de dados e soluções de storage
- **Segurança:** Ferramentas de proteção, autenticação e monitoramento

Infraestrutura de TI: Os Tipos

On-Premise (Tradicional):

- **Definição:** Infraestrutura onde todos os recursos de TI são mantidos e gerenciados internamente pela organização em suas próprias instalações físicas, incluindo servidores, redes, armazenamento e software.
- **Vantagens:**
 - **Controle total:** As organizações são proprietárias de seus edifícios e equipamentos, o que permite personalizar sua TI e instalações para atender às suas necessidades específicas.
 - **Custo-benefício a longo prazo:** Apesar do alto investimento inicial em data centers, os custos operacionais tendem a ser menores e a otimização da infraestrutura pode gerar economia a longo prazo.
 - **Alta visibilidade:** A supervisão e o acesso direto à infraestrutura facilitam a detecção e resposta mais rápidas a problemas e vulnerabilidades.
- **Desvantagens:**
 - **Altos custos iniciais.** O investimento em instalações, hardware, software e pessoal pode não ser viável para todas as organizações.
 - **Gestão das instalações.** É necessária uma equipe dedicada com habilidades especializadas para manter o edifício e solucionar problemas. É frequentemente necessário investimento regular em nova infraestrutura e tecnologia.
 - **Infraestrutura fixa.** Os recursos são tipicamente provisionados para demanda máxima, tornando impraticável e custoso reduzir a estrutura devido aos significativos processos de reconfiguração e desativação.
- **Casos de uso:**
 - **Grandes empresas** com ambientes de TI complexos, altos requisitos de segurança e capacidade de arcar com os custos iniciais.
 - **Agências governamentais** que lidam com dados sensíveis que não podem ser armazenados externamente.
 - **Instituições financeiras** ou outras organizações que processam dados de cartões e precisam cumprir as regulamentações PCI-DSS.

- **Provedores de saúde** que armazenam dados de pacientes e precisam cumprir as regulamentações HIPAA.
- **Instituições de pesquisa** com equipamentos especializados e necessidades de computação de alto desempenho.

Colocated (Datacenters):

- **Definição:** Modelo onde a empresa aluga espaço em um datacenter profissional para hospedar seus próprios servidores e equipamentos. O provedor do datacenter fornece energia, refrigeração, segurança física e conectividade, enquanto a empresa mantém controle total sobre seu hardware e software.
- **Vantagens:**
 - **Instalações de primeira classe:** Organizações sem expertise em instalações ainda podem alugar infraestruturas de última geração com sistemas avançados de refrigeração, energia redundante e medidas robustas de segurança.
 - **Flexibilidade:** Os recursos podem ser facilmente expandidos ou reduzidos de acordo com as necessidades.
 - **Baixos custos iniciais:** Além do investimento inicial em equipamentos de TI, o restante do custo da colocação se dá na forma de pagamentos mensais previsíveis.
- **Desvantagens:**
 - **Falta de visibilidade.** Os locatários têm pouco controle sobre sua infraestrutura física e devem confiar em seu provedor para executar as ordens de serviço com precisão.
 - **Desafios de migração.** Mover equipamentos para um datacenter de colocação requer planejamento meticuloso, um inventário preciso e a capacidade de rastrear cada mudança.
 - **Custos contínuos.** Custos adicionais com consumo de energia, largura de banda de rede, serviços de suporte remoto e interconexões podem aumentar rapidamente.
- **Casos de uso:**
 - **Pequenas e médias empresas** que necessitam de infraestrutura confiável para armazenar e gerenciar grandes volumes de dados.
 - **Empresas em rápido crescimento** que precisam escalar rapidamente sua infraestrutura de data center para atender à demanda dos clientes.
 - **Organizações com requisitos de conformidade** que podem ser atendidos por um provedor de colocation respeitável.
 - **Serviços online** como empresas de e-commerce e mídia digital que desejam reduzir a latência e melhorar a experiência do cliente ao localizar equipamentos próximos às áreas que atendem.

Nuvem:

- **Definição:** Modelo de computação em nuvem onde recursos de TI (servidores, armazenamento, bancos de dados, rede, software) são fornecidos pela internet sob demanda, com pagamento baseado no uso. Os recursos podem ser rapidamente provisionados e liberados com mínimo esforço de gerenciamento.
- **Vantagens:**
 - **Escalabilidade elástica:** Capacidade de aumentar ou diminuir recursos instantaneamente conforme a demanda.
 - **Custo otimizado:** Pagamento apenas pelos recursos utilizados, sem necessidade de grandes investimentos iniciais.
 - **Agilidade:** Rápida implementação e atualização de recursos e serviços.
- **Desvantagens:**
 - **Preocupações com segurança:** Confiar dados sensíveis a um provedor terceirizado pode levantar preocupações sobre privacidade de dados, possíveis violações ou conformidade regulatória.

- **Questões de soberania de dados:** Os dados podem ser armazenados em diferentes jurisdições, potencialmente levando a complexidades de conformidade, conflitos regulatórios ou limitações de acesso.
- **Sem controle:** Os provedores de nuvem são proprietários e operam seus próprios data centers, deixando os clientes sem poder de decisão sobre a infraestrutura ou medidas de segurança.
- **Casos de uso:**
 - **Startups e empresas em crescimento:** Que precisam de flexibilidade e escalabilidade.
 - **Grandes empresas** que utilizam serviços em nuvem para big data, IA, machine learning e colaboração em tempo real, além de backup e recuperação de desastres.
 - **Empresas sazonais** que precisam escalar rapidamente para atender a picos de tráfego ou outras organizações com cargas de trabalho imprevisíveis ou flutuantes.
 - **Desenvolvimento e teste:** Ambientes que requerem provisionamento rápido.

Híbrido:

- **Definição:** Combinação de diferentes tipos de infraestrutura de TI, onde uma organização utiliza tanto recursos on-premise quanto em nuvem, ou uma mistura de diferentes modelos, para otimizar custos, performance e segurança de acordo com necessidades específicas.
- **Vantagens:**
 - **Flexibilidade:** Permite escolher a melhor solução para cada tipo de workload ou aplicação.
 - **Otimização de custos:** Equilibra investimentos entre infraestrutura própria e serviços em nuvem.
 - **Segurança e conformidade:** Dados críticos e sensíveis podem ser mantidos localmente, garantindo segurança e conformidade regulatória.
- **Desvantagens:**
 - **Complexidade:** Gerenciar múltiplos ambientes requer expertise técnica mais ampla.
 - **Integração:** Necessidade de garantir comunicação eficiente entre diferentes ambientes.
 - **Custos de gestão:** Manter equipes capacitadas para diferentes tecnologias.
- **Casos de uso:**
 - **Empresas com necessidades diversas de TI** que já investiram em data centers próprios e desejam migrar gradualmente algumas cargas de trabalho para a nuvem.
 - **Organizações com requisitos de conformidade** como entidades governamentais, financeiras, de saúde e jurídicas que precisam manter dados sensíveis em ambientes próprios ou em instalações de colocação.
 - **Empresas que buscam otimização de custos** que podem aproveitar diferentes opções de hospedagem com base nas características de suas cargas de trabalho.
 - **Empresas que priorizam a continuidade dos negócios** que utilizam a nuvem para replicação e backup de dados.

Cloud Computing: The Good, The Bad & The Ugly

Cloud: Os Modelos

- **Cloud Infrastructure as a Service (IaaS):** As empresas precisam gerenciar aplicações, dados, middleware e sistema operacional. (Ex.: AWS EC2, que permite configurar e gerenciar máquinas virtuais com total controle sobre o ambiente de computação)
- **Cloud Platform as a Service (PaaS):** As empresas precisam gerenciar apenas aplicações e dados; o restante é gerenciado pelo provedor do serviço. (Ex.: Heroku, que oferece uma plataforma para desenvolver, executar e gerenciar aplicativos sem se preocupar com a infraestrutura subjacente)
- **Software as a Service (SaaS):** O modelo mais abstrato, onde todo o ambiente é gerenciado pelo provedor e os usuários apenas consomem o software via internet. O provedor é responsável por toda a infraestrutura,

manutenção e atualizações. (Ex.: Salesforce, que oferece CRM como serviço, ou Microsoft 365, que disponibiliza aplicativos de produtividade na nuvem)

Cloud: Os Players

- **Amazon Web Services (AWS)** Um provedor popular de serviços em nuvem que oferece uma variedade de serviços, incluindo armazenamento de banco de dados, poder computacional e redes. A AWS é conhecida por seus recursos de segurança e arquitetura de data center.
- **Google Cloud Platform (GCP)** Um provedor de serviços em nuvem escalável que oferece soluções para gerenciamento de dados, desenvolvimento de aplicações e infraestrutura. O GCP é conhecido por sua infraestrutura global, recursos de segurança e capacidades de IA e machine learning.
- **Oracle Cloud** Um fornecedor de nuvem que oferece serviços IaaS, PaaS e SaaS. O Oracle Cloud é conhecido por seu foco em soluções empresariais e integração com ambientes computacionais Oracle existentes.
- **Microsoft Azure** Um provedor de serviços em nuvem que oferece
- **Outros:** Alibaba Cloud, Salesforce, Dell, IBM, Digital Ocean, Tencent Cloud, and Huawei Cloud...

Cloud: As Abordagens

- **Multi-cloud:** Uso simultâneo de múltiplos provedores de nuvem para distribuir cargas de trabalho, reduzir dependência e otimizar custos. Oferece redundância e flexibilidade, mas aumenta a complexidade.
- **Hybrid-cloud:** Combinação de infraestrutura local (on-premises) com serviços em nuvem pública, permitindo que as organizações mantenham dados sensíveis internamente enquanto aproveitam a escalabilidade da nuvem para outras cargas de trabalho. Esta abordagem oferece maior flexibilidade e controle sobre onde os dados e aplicações residem.
- **Edge:** Processamento e armazenamento de dados mais próximo da fonte de geração, reduzindo latência e custos de bandwidth. Especialmente útil para IoT, streaming de mídia e aplicações que requerem resposta em tempo real.

Cloud: Os serviços

- **Maquinas Virtuais:** Serviços de virtualização que permitem criar e gerenciar VMs na nuvem com recursos configuráveis de CPU, memória e armazenamento. Exemplos: Amazon EC2, Google Compute Engine e Azure VMs.
- **Balanceador de Carga:** Serviço que distribui o tráfego entre servidores para garantir disponibilidade e performance. AWS, GCP e Azure oferecem balanceadores que mantêm aplicações estáveis durante picos.
- **Autoscaling:** Serviço que ajusta a capacidade computacional automaticamente com base em métricas como CPU, memória ou tráfego, mantendo performance e otimizando custos ao adicionar ou remover recursos conforme a demanda.
- **VPC:** Virtual Private Cloud (VPC) é um ambiente isolado na nuvem para gerenciar recursos de rede como sub-redes e gateways. Oferece controle total sobre configurações de rede, incluindo IPs e roteamento, garantindo segurança e isolamento dos recursos.
- **CDN:** Content Delivery Network distribui conteúdo através de servidores globais para reduzir latência e melhorar performance. Serviços como Amazon CloudFront, Google Cloud CDN e Azure CDN entregam conteúdo estático (imagens, vídeos, JavaScript) rapidamente aos usuários.
- **File Storage:** Serviços de armazenamento que oferecem opções flexíveis para arquivos na nuvem, como Amazon S3, Google Cloud Storage e Azure Blob Storage. Fornecem armazenamento seguro e escalável para dados não estruturados (documentos, imagens, backups), com diferentes classes de armazenamento baseadas na frequência de acesso.
- **Banco de Dados Gerenciado:** Serviços de bancos de dados gerenciados na nuvem (Amazon RDS, Google Cloud SQL, Azure Database) que fornecem MySQL, PostgreSQL e MongoDB com backup automático e alta disponibilidade. Permite que startups foquem no desenvolvimento sem administrar infraestrutura.

- **Sistema de Mensageria/Notificação:** Serviços para troca de mensagens entre componentes de aplicações. Amazon SQS, Google Cloud Pub/Sub e Azure Service Bus oferecem comunicação assíncrona e processamento em lote para sistemas distribuídos.
- **Serverless Backends:** Serviços para executar código sem gerenciar servidores (AWS Lambda, Google Cloud Functions, Azure Functions). Escala automaticamente com a demanda e cobra apenas pelo uso real.

Cloud: As Histórias de Sucesso

- **Netflix:**
 - A Netflix é um excelente exemplo de transformação bem-sucedida para a nuvem. Como líder em streaming, ela utiliza a AWS para atender milhões de espectadores globais.
 - Antes da migração para a nuvem, a Netflix enfrentava dificuldades com a escalabilidade da infraestrutura. A computação elástica da AWS permitiu um escalonamento contínuo para um serviço de streaming ininterrupto.
 - Usando a infraestrutura global e CDN da AWS, a Netflix melhorou a velocidade de entrega de conteúdo e reduziu a latência em todo o mundo. Além disso, os recursos de machine learning da AWS possibilitaram recomendações personalizadas para os espectadores.
 - Essa transformação aprimorou a escalabilidade, a entrega de conteúdo e a experiência do usuário da Netflix.
- **AirBnB:**
 - O Airbnb representa outro caso bem-sucedido de transformação para a nuvem. Como um marketplace global de hospedagem, a empresa precisava de uma infraestrutura robusta para lidar com suas operações massivas de dados.
 - Após enfrentar desafios de escalabilidade, o Airbnb fez parceria com a AWS, aproveitando sua infraestrutura global para expandir capacidades e gerenciar a crescente demanda dos usuários. A AWS permitiu o provisionamento rápido de recursos, garantindo experiências de reserva sem interrupções.
 - A migração proporcionou recursos aprimorados de segurança e capacidades de análise de dados. Isso permitiu ao Airbnb proteger as informações dos usuários enquanto obtinha insights valiosos para otimizar preços e melhorar a experiência do cliente.
 - Por meio dessa transformação, o Airbnb alcançou melhor escalabilidade, segurança e capacidades de tomada de decisão baseada em dados.
- **Capital One:**
 - O Capital One fez parceria com a AWS para transformação digital visando manter-se competitivo no setor bancário. A empresa modernizou sua infraestrutura, migrando de data centers tradicionais para soluções baseadas em nuvem para reduzir custos e melhorar a eficiência.
 - Usando os serviços da AWS, o Capital One conseguiu desenvolver e lançar novos produtos rapidamente, mantendo rigorosos requisitos de segurança e conformidade. As capacidades de machine learning da plataforma em nuvem aprimoraram a detecção de fraudes e o gerenciamento de riscos.
 - Essa transformação permitiu ao Capital One otimizar operações, inovar mais rapidamente e oferecer melhores experiências aos clientes, garantindo a segurança dos dados.

O Mau e o Feio

- **A Grande Repartição:** Está acontecendo uma “repatriação” de empresas de tecnologia, saindo da Cloud e voltando para o On-premise (dentre outras estratégias ainda mais polemicas). E isso está acontecendo por alguns motivos:
 - **Custo:** A repatriação para fora da nuvem é principalmente motivada pelos custos. [Uma pesquisa da Citrix descobriu que 43% dos líderes de TI enfrentaram custos de nuvem mais altos do que o esperado.](#) Embora os serviços em nuvem ofereçam custos iniciais baixos, o modelo de pagamento conforme o uso frequentemente leva a despesas inesperadas, especialmente para cargas de trabalho empresariais não otimizadas.

How an empty S3 bucket can make your AWS bill explode

Cloudflare took down our website after trying to force us to pay 120k\$ within 24h

How we've saved 98% in cloud costs by writing our own database

Scenario	AWS	HorizonIQ	Cost Savings
MODERATE USE			
Web (2)	\$1,110.12	\$189.00	83%
App (2)	\$930.88	\$382.00	59%
DB	\$1,701.76	\$499.00	71%
Total	\$5,783.76/mo	\$1,641.00/mo	72%
LIGHT USE			
Web	\$155.80	\$89.00	43%
App	\$280.68	\$189.00	83%
DB	\$1,605.89	\$307.00	81%
Total	\$2,042.37/mo	\$585.00/mo	71%

- **Segurança:** Preocupações com segurança também estão impulsionando a repatriação da nuvem, com 33% das organizações citando segurança como um fator-chave segundo a Citrix. Apesar das medidas de segurança dos provedores de nuvem, as empresas preferem gerenciar sua própria infraestrutura para ter melhor controle, especialmente nos setores financeiro e de saúde que lidam com dados sensíveis.
- **Expectativas Não Atendidas:** Muitas empresas descobrem que os serviços em nuvem não atendem às suas expectativas internas, com 24% citando isso como motivo para deixar a nuvem, segundo a Citrix. Apesar das promessas de integração perfeita e melhor desempenho, problemas com compatibilidade, tempo de inatividade e custos reais levam à insatisfação.
- **As Histórias dos Rebeldes**
 - **Basecamp:** Uma empresa de software de gerenciamento de projetos, migrou dos serviços AWS/GCP para sua própria instalação de colocation. Isso economizou milhões. O CTO David Heinemeier Hansson citou a carga de trabalho previsível e a equipe qualificada como razões para tornar a hospedagem local mais econômica, demonstrando como as escolhas de infraestrutura devem se alinhar com as capacidades organizacionais.

**Leaving the cloud will save us
\$7 million over five years.**

- **DropBox:** Dropbox moveu 90% de seus dados da nuvem para seus próprios data centers, economizando \$75M em dois anos. A mudança foi motivada pela necessidade de controle da infraestrutura e melhor alinhamento com requisitos específicos.

- **Ahrefs:** A Ahrefs economizou US\$ 400M em 3 anos usando infraestrutura própria ao invés da nuvem. Com 1.000 servidores processando 100 TB diários, o CEO revelou que custos em nuvem seriam 6-8x maiores. Para empresas com cargas de trabalho previsíveis e equipe técnica capacitada, servidores próprios podem ser mais econômicos que cloud.

Owned servers on premises		AWS EC2 instances	
Monthly cost for 850 servers	\$ 1,317,301	Monthly cost for 850 servers	\$ 14,923,154
Cost for 30 months	\$ 39,519,025	Cost for 30 months	\$ 447,694,623
		AWS vs On-Prem	\$ (408,175,598)

Impacto na Escalabilidade

A escolha entre cloud, on-premise ou soluções híbridas tem um impacto direto na capacidade de uma empresa escalar. Enquanto a cloud oferece elasticidade e rápido provisionamento de recursos, soluções on-premise podem proporcionar maior controle e potencial economia em escala para cargas de trabalho previsíveis. A decisão deve ser baseada não apenas nos custos, mas também nos requisitos específicos de performance, segurança e conformidade do negócio.

Por que isso é importante para os analistas?

Entender a infraestrutura tecnológica e sua capacidade de escalar é crucial para os analistas por três razões principais. Primeiro, permite avaliar se a startup tem base tecnológica para suportar o crescimento projetado sem interrupções significativas. Segundo, ajuda a identificar potenciais gargalos técnicos que podem impactar a execução do plano de negócios e, terceiro, possibilita antecipar necessidades de investimento em tecnologia que podem afetar significativamente as projeções financeiras.

6. O Conceito de Débito Técnico

O Que é?

Débito técnico refere-se às consequências de escolhas de implementação que priorizam velocidade sobre qualidade de código ou arquitetura. É similar a uma dívida financeira: tomar atalhos no desenvolvimento pode trazer benefícios imediatos, mas acumula "juros" na forma de maior complexidade e custos de manutenção futuros. Quando não gerenciado adequadamente, o débito técnico pode tornar o sistema cada vez mais difícil de modificar e manter.

Por que aparecem e crescem?

O débito técnico geralmente surge e cresce devido a pressões de prazo e decisões de curto prazo. Startups frequentemente priorizam o lançamento rápido de funcionalidades para atender demandas do mercado ou investidores, sacrificando a qualidade do código. Além disso, mudanças frequentes nos requisitos de negócio, falta de documentação adequada e rotatividade da equipe técnica contribuem para o acúmulo progressivo de débito técnico.

Impacto

- Aumenta custos de manutenção.
- Pode limitar a escalabilidade e introduzir vulnerabilidades.
- Dificulta a implementação de novas funcionalidades e inovações.
- Pode causar problemas de performance e experiência do usuário.
- Aumenta o risco de falhas e downtime do sistema.

Como Medir

- **Número de Bugs:** O aumento no número de bugs reportados sinaliza problemas no código. Uma tendência crescente indica acúmulo de débito técnico. Métricas de bugs por linha de código ou funcionalidade ajudam

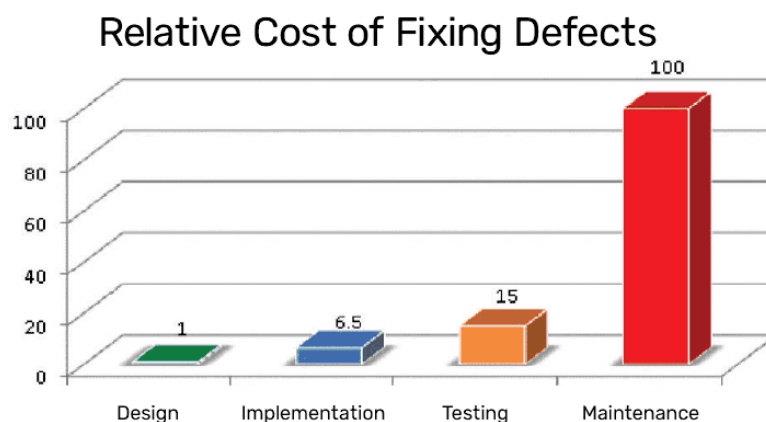
na quantificação.

- **Velocidade de Desenvolvimento:** A velocidade reduzida no desenvolvimento de novas funcionalidades indica débito técnico. Quando desenvolvedores passam mais tempo com código legado e correções, a produtividade cai. Isso pode ser medido pelo tempo de implementação de features ou entregas por sprint.
- **Tempo de Correção de Problemas:** O tempo de correção de problemas indica a complexidade do código. Tempos crescentes de resolução sugerem acúmulo de débito técnico, medido pelo tempo médio de resolução de bugs críticos.
- **Cobertura de Testes:** Baixa cobertura de testes automatizados aumenta o risco de bugs. Medida como percentual do código testado, cobertura menor que 70-80% em código crítico indica débito técnico.
- **Frequência de Regressões:** Aumento no número de problemas que reaparecem após correções indica fragilidade no código.

Testes

Os testes são uma parte fundamental da infraestrutura de qualidade de software moderna. Não se tratam de uma bateria de testes manuais realizados por um time de Q.A. ou PMs, mas sim de testes automatizados escritos pelos próprios desenvolvedores.

Testes automatizados são o cerne da cultura de DevOps e padrões implantação contínua que é comum em muitos times de engenharia hoje. Eles garantem que bugs sejam pegos o quanto antes e não cheguem em produção, onde afetam o usuário e o negócio. Um bug encontrado em produção custa até 100 vezes mais que na etapa de projeto.

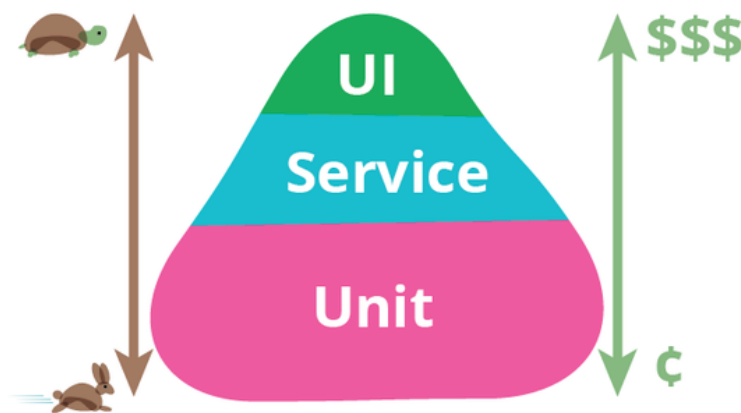


O custo relativo de resolver bugs mais tarde no ciclo de vida de desenvolvimento é altíssimo.

De forma simplificada existem 3 tipos de testes automatizados:

- **Testes unitários:** um tipo de teste para validar diretamente o comportamento esperado de uma unidade conceitual de um software (componente ou módulo), que são partes de código que podem ser isolados de um sistema maior e mais complexo;
- **Testes de integração ou serviço:** um tipo de teste para validar a interação coerente entre dois ou mais componentes (unidades) ou serviços, garantindo a coesão desta combinação de componentes;
- **Testes funcionais ou ponta-a-ponta:** um tipo de teste para verificar a funcionalidade e a performance do software por completo, do início ao fim, simulando usuários reais com dados do mundo real.

Esses tipos de teste revelam um espectro de ROI e velocidade de execução, onde testes unitários tem baixo ROI e rápida execução, e testes ponta-a-ponta possuem alto ROI e execução lenta.



O espectro de velocidade de execução vs ROI dos tipos de teste (UI = ponta-a-ponta).

Impacto na Escalabilidade

O débito técnico tem um impacto direto na capacidade de escalabilidade de um sistema. Quando não gerenciado adequadamente, pode resultar em gargalos de performance, dificuldade em adicionar novos recursos e aumento exponencial do tempo necessário para implementar mudanças conforme o sistema cresce. Isso é especialmente crítico em startups em fase de hipercrecimento, onde a necessidade de escalar rapidamente pode ser comprometida por decisões técnicas anteriores que priorizaram velocidade sobre qualidade.

Por que isso é importante para os analistas?

Para analistas, compreender o débito técnico é fundamental por três aspectos principais. Primeiro, permite avaliar riscos técnicos ocultos que podem impactar a execução do plano de negócios. Segundo, ajuda a identificar necessidades futuras de investimento em refatoração e modernização de sistemas, e terceiro, possibilita uma avaliação mais precisa da real velocidade de desenvolvimento e capacidade de inovação da empresa.

7. Trade-offs: Velocidade de Desenvolvimento vs. Débito Técnico

O equilíbrio entre velocidade de desenvolvimento e débito técnico é um dos principais desafios enfrentados por startups em crescimento. Enquanto a pressão por entregas rápidas e novas funcionalidades é constante, decisões apressadas podem resultar em problemas técnicos que comprometem a escalabilidade futura. A chave está em encontrar um balanço que permita inovação rápida sem comprometer a sustentabilidade técnica do produto.

Existem três principais aspectos a serem considerados neste trade-off:

- **Time to Market vs. Qualidade:** Lançar funcionalidades rapidamente pode dar vantagem competitiva, mas código mal estruturado aumenta o débito técnico.
- **Custo Imediato vs. Custo Futuro:** Atalhos técnicos podem economizar tempo e dinheiro no curto prazo, mas frequentemente resultam em custos maiores de manutenção e refatoração no futuro.
- **Flexibilidade vs. Estabilidade:** Mudanças rápidas podem atender demandas do mercado, mas podem comprometer a estabilidade e confiabilidade do sistema.

Cenários

Foco na Robustez

- **Benefícios:**
 - Maior confiabilidade e estabilidade do sistema
 - Menor risco de falhas em produção
 - Facilidade de manutenção a longo prazo

- Melhor segurança e proteção de dados
- Código mais limpo e bem documentado
- **Riscos:**
 - Desenvolvimento mais lento de novas funcionalidades
 - Maior custo inicial de desenvolvimento
 - Pode perder oportunidades de mercado devido ao tempo extra
 - Resistência da equipe a processos mais rigorosos
 - Possível frustração de stakeholders com prazos mais longos

Foco na Velocidade

- **Benefícios:**
 - Lançamento mais rápido de produtos no mercado
 - Capacidade de resposta ágil às demandas do mercado
 - Maior velocidade na captura de oportunidades de negócio
 - Feedback mais rápido dos usuários
 - Potencial vantagem competitiva inicial
- **Riscos:**
 - Acúmulo rápido de débito técnico que pode paralisar o desenvolvimento futuro
 - Maior probabilidade de bugs e falhas em produção
 - Dificuldade crescente em manter e escalar o sistema
 - Custos mais altos de manutenção a longo prazo
 - Possível perda de desenvolvedores devido à baixa qualidade do código

Por que isso é importante para os analistas?

Para analistas, entender o trade-off entre velocidade e débito técnico é crucial por permitir uma avaliação mais precisa do risco-retorno das decisões tecnológicas. Esta compreensão ajuda a avaliar se as escolhas técnicas da startup estão alinhadas com seus objetivos de crescimento e necessidades do mercado. Além disso, permite aos analistas identificarem sinais de alerta quando o balanço entre velocidade e qualidade técnica está muito desequilibrado, podendo comprometer o sucesso futuro da empresa.

Resultado Esperado

Ao final desta seção, os analistas serão capazes de:

- Avaliar se a tecnologia de uma empresa pode suportar crescimento rápido.
- Reconhecer arquiteturas e infraestruturas escaláveis.
- Identificar e questionar decisões que aumentam o débito técnico.
- Compreender os trade-offs entre velocidade de desenvolvimento e escalabilidade.

Esses insights permitirão aos analistas prever se a startup está preparada tecnicamente para escalar ou se enfrentará gargalos que podem limitar seu crescimento.

Apresentação

https://docs.google.com/presentation/d/1_kiRSDh3pd7NyFK0yyMQasLh5z7UwEhzd2uhWxME-YM/preview#slide=id.g2d4e8f707bc_0_351

Recursos e Referências

- <https://www.df.cl/empresas/mercadolibre-compra-a-su-rival-deremate-en-us-40-millones>
- [https://medium.com/mercadolibre-tech/the-technological-evolution-at-mercado-libre-fb269776a4e8\](https://medium.com/mercadolibre-tech/the-technological-evolution-at-mercado-libre-fb269776a4e8)
- https://www.youtube.com/watch?v=CgriozF_U20
- <https://www.lavivienpost.com/how-zoom-works-architecture/>
- <https://dev.to/somadevtoo/10-scalability-lessons-from-zooms-software-architecture-6g3>
- <https://www.turing.com/blog/software-architecture-patterns-types>
- <https://www.amazon.com.br/Team-Topologies-Organizing-Business-Technology/dp/1942788819>
- <https://www.docker.com/blog/building-stronger-happier-engineering-teams-with-team-topologies/>
- <https://www.atlassian.com/itsm/it-operations-management/it-infrastructure>
- <https://www.sunbirdcim.com/blog/prem-vs-colo-vs-cloud-vs-hybrid-how-choose>
- <https://appinventiv.com/blog/it-infrastructure-software-development/>
- <https://medium.com/codex/aws-case-studies-success-stories-of-cloud-transformation-e10a14a0cf95>
- <https://ip-specialist.medium.com/top-10-most-used-aws-services-in-2024-you-should-know-71b7ebc6a440>
- <https://www.horizoniq.com/blog/companies-are-leaving-cloud/>
- <https://world.hey.com/dhh/why-we-re-leaving-the-cloud-654b47e0>