



JÁ VAI COMEÇAR

ARQUITETURA, INFRAESTRUTURA E STACK DE TECNOLOGIA

ENQUANTO ISSO...



**Escolha um
lugar
confortável**
para você
sentar e se
acomodar



**Que tal
pegar um
snack** para
matar a fome,
uma **água**,
um chá



Abra o chat,
envie um
"olá" e
#sentimento
de como
chega



Que tal pegar
caderno e
caneta para
anotações?



DIÓGENES CARVALHO

PM NO RD STATION

ARQUITETURA, INFRAESTRUTURA
E STACK DE TECNOLOGIA

AGENDA

- **Bloco 1:**
Como os produtos digitais funcionam?
- **Bloco 2:**
Almanaque de tecnologia para Product Managers
- **Bloco 3:**
Engenharia e Produto em conjunto

COMO OS PRODUTOS DIGITAIS FUNCIONAM?

**MAS E AÍ,
PORQUE ISSO É
IMPORTANTE?**



T

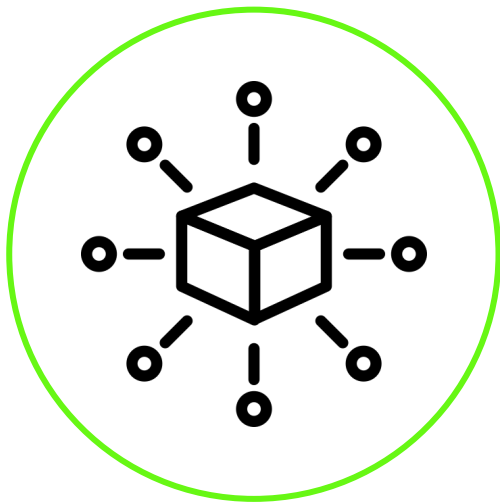
MAS E AÍ, POR QUE ISSO É IMPORTANTE?

CONHECER MELHOR
O SEU PRODUTO

ENTENDER O QUE OS
DESENVOLVEDORES
FALAM

AJUDAR O TIME

O QUE EU DEVO SABER SOBRE TECNOLOGIA?



ARQUITETURA

- Monolitos
- Microserviços

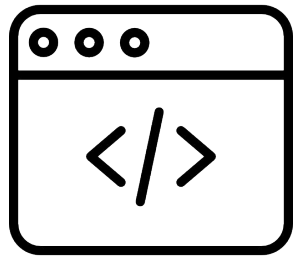
O QUE EU DEVO SABER SOBRE TECNOLOGIA?



INFRAESTRUTURA

- DevOps
- Banco de Dados
- API Gateway
- Cloud
- Um monte de 'as a Service'

O QUE EU DEVO SABER SOBRE TECNOLOGIA?



STACK DE DESENVOLVIMENTO

- Linguagens de Programação
- Frameworks
- Web, App, APIs

**TECNOLOGIA NÃO É MAIS
INÍCIO E FIM, MAS SIM
MEIO PARA TUDO!**



COMO ISSO TUDO FUNCIONA?

GOOGLE MAPS

OVERVIEW

- Traçar rotas
- Trânsito em tempo real
- Transporte público em tempo real
- Descobrir lugares
- Ver recomendações
- Criar listas de locais
- Conectar-se com os lugares (telefone, site, e-mail, etc)
- Reservar





COMO ISSO TUDO FUNCIONA?

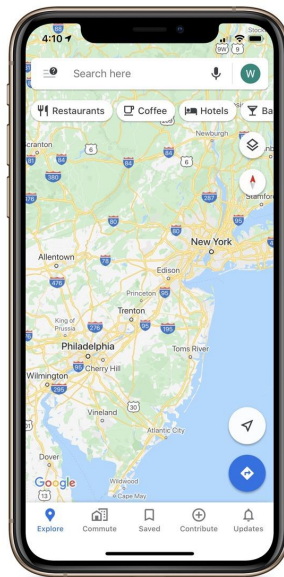
GOOGLE MAPS

COMPONENTES

- Client Side
- Backend
- Frontend (HTML, CSS, AJAX)
- Imagens
- Banco de Dados
- Apps (Android/iOS)
- Endereços (URLs)



Google Maps





COMO ISSO TUDO FUNCIONA?

GOOGLE MAPS

O QUE PROPORCIONA?

- Diferentes formatos
- Atualizações na interface
- Atualização das funcionalidades
- Correção de problemas





COMO ISSO TUDO FUNCIONA?

E NÃO SÓ ISSO...

APIs

Build amazing Apps and websites with the



Google Maps Platform

Family of APIs

Maps



JavaScript Maps



Static Maps



StreetView API

Routes



Directions API



Distance Matrix

Places



Places API
(points of interest)

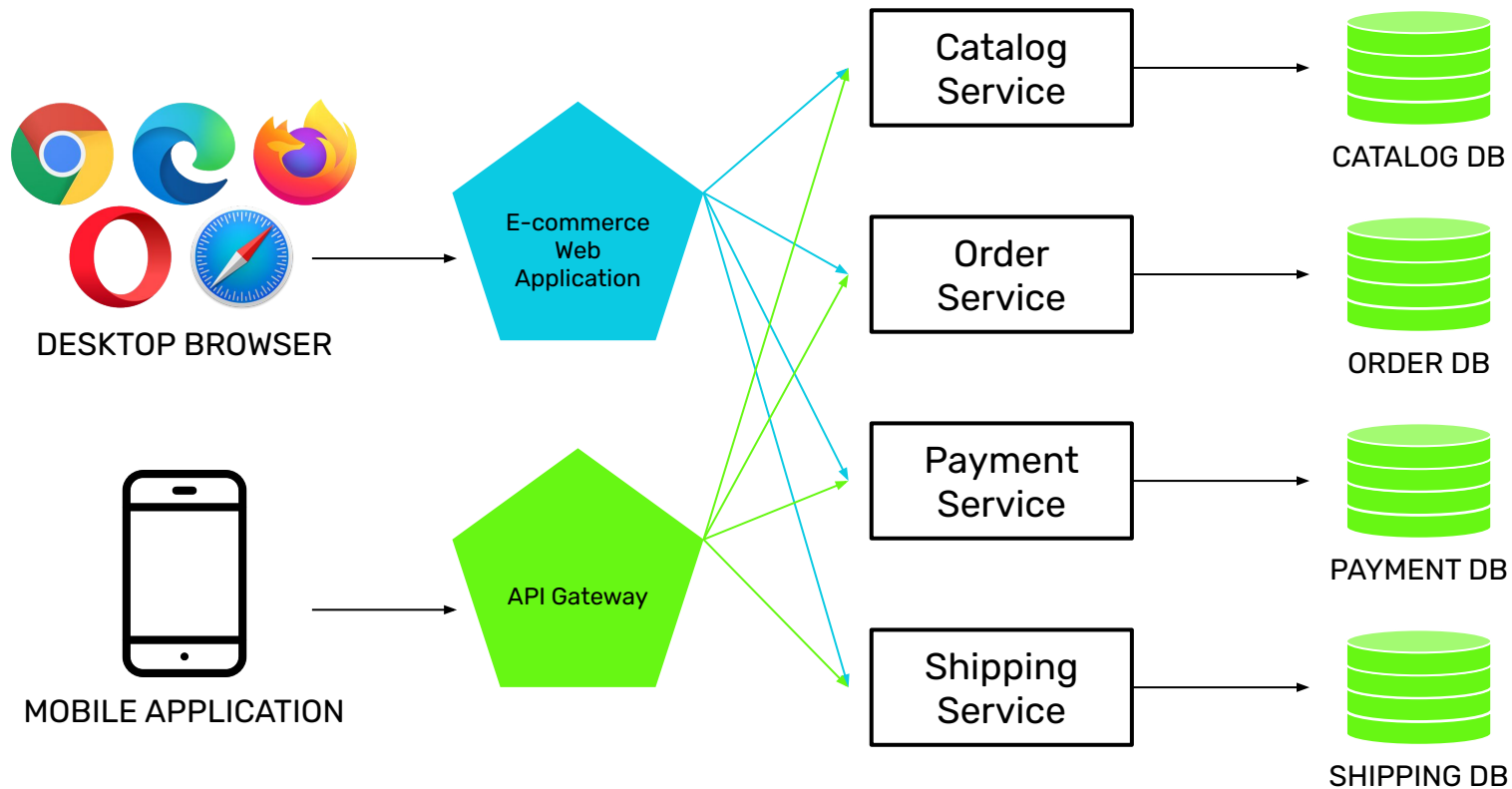


Geocoding



Elevation API

- Rotas
- Lugares
- etc.





I NA PRÁTICA

OVERVIEW UTILIZANDO A EXTENSÃO



Wappalyzer



Spotify

Medium

T

ALMANAQUE DE TECNOLOGIA PARA PRODUCT MANAGERS

ENTENDENDO A WEB E COMO OS PRODUTOS DIGITAIS FUNCIONAM

O QUE É UM DOMÍNIO NA INTERNET?

www.somostera.com

1ºNÍVEL

2ºNÍVEL

3ºNÍVEL

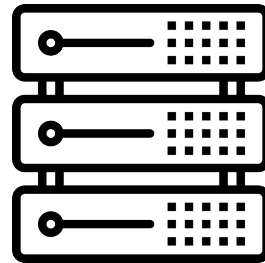
O QUE É HTTP?



BROWSER

HTTP request

method: GET
path: /index.html



SERVER

HTTP response

status: 200
content-type: text/html;
charset= UTF-8

O QUE SÃO APIs?

APIs (*Application Programming Interface*) são mais do que detalhes técnicos de *JSON* ou *XML*, *síncrono* ou *assíncronos*, etc. Eles são uma forma de conectar unidades de negócios entre si, para expor funcionalidades e conhecimentos importantes de uma forma que ajude as empresas a serem mais eficazes.

As *APIs* geralmente são uma forma de **desbloquear** o valor que já existe na organização, por exemplo, por meio da criação de novos aplicativos, habilitando novos fluxos de receita e iniciando novos negócios.

TIPOS DE APIs

TIPO	ACESSO	EXEMPLOS
Privada	Privado	Empresas que mantêm suas APIs para consumo interno
Parceiros	Limitado	Empresas que precisam compartilhar dados com parceiros
Pública	Público	Para consumo público e com restrições mínimas

O QUE É REST?

REST significa *Representational State Transfer*. Em português, **Transferência de Estado Representacional**. Trata-se de uma abstração da arquitetura Web.

Resumidamente, o REST consiste em princípios/regras que, quando seguidas, permitem a criação de um projeto com interfaces bem definidas. Desta forma, permitindo, por exemplo, que aplicações se comuniquem.

MAS E O RESTFUL... QUAL A DIFERENÇA?

Existe uma certa confusão quanto aos termos **REST** e **RESTful**.

Entretanto, ambos representam os mesmos princípios.

A diferença é apenas gramatical. Em outras palavras, sistemas que utilizam os princípios REST são chamados de RESTful.

MÉTODOS HTTP

REQUEST

get

Requisição utilizada para buscar os dados solicitados do servidor

post

Requisição utilizada para criar dados no servidor

put

Requisição utilizada para atualizar o dado indicado no servidor

patch

Requisição feita para atualização de somente uma parte de uma informação no servidor

delete

Requisição para excluir uma informação do servidor

MÉTODOS HTTP

RESPONSE

1xx – Informativos

a solicitação foi aceita ou o processo continua em andamento;

2xx – Indicativos de sucesso

a ação foi concluída ou entendida;

3xx – Redirecionamentos

indica que algo mais precisa ser feito ou precisou ser feito para completar a solicitação;

4xx – Erros do cliente

indica que a solicitação não pode ser concluída ou contém a sintaxe incorreta;

5xx – Erros no lado do servidor

o servidor falhou ao concluir a solicitação.

EXEMPLO

- Postman

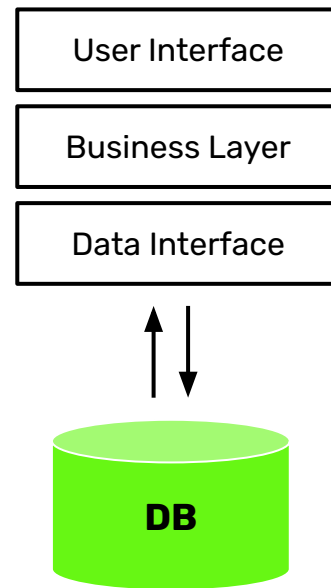


CONHECENDO ARQUITETURA DE SOFTWARE

MONOLITOS

Na engenharia de software, um aplicativo monolítico descreve um aplicativo de software de camada única em que a interface do usuário e o código de acesso aos dados são combinados em um único programa a partir de uma única plataforma.

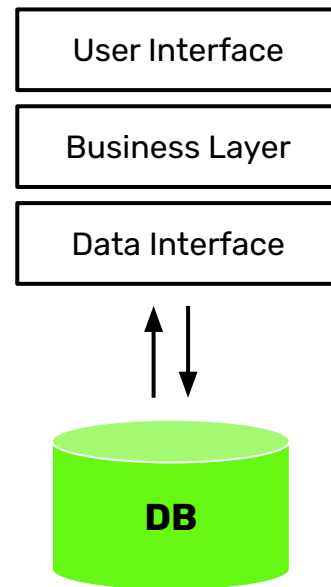
Um aplicativo monolítico é autocontido e independente de outros aplicativos de computação. A filosofia de design é que o aplicativo é responsável não apenas por uma tarefa específica, mas pode executar todas as etapas necessárias para completar uma função específica.



CONHECENDO ARQUITETURA DE SOFTWARE

CARACTERÍSTICAS DA ARQUITETURA MONOLÍTICA

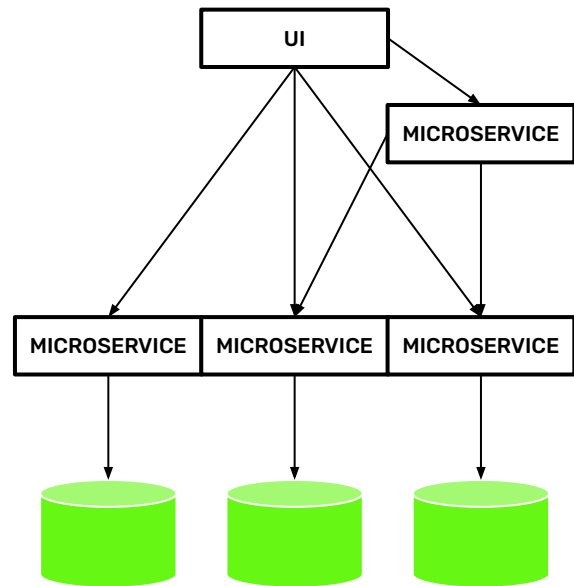
- Não pode ser escalada facilmente;
- Inflexível, pois precisa seguir a tecnologia atual;
- Alterações não podem ser empacotadas e implantadas ao mesmo tempo;
- Desenvolvimento lento;
- Se uma feature não funciona, o sistema todo não funciona;



CONHECENDO ARQUITETURA DE SOFTWARE

MICROSSERVIÇOS

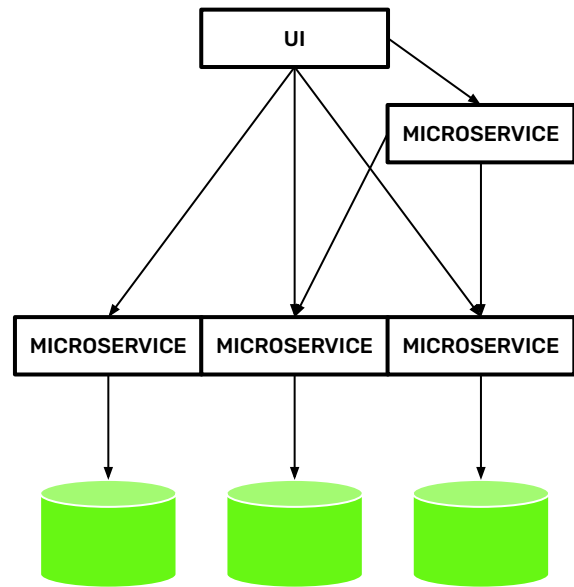
Microserviços é um padrão de arquitetura que estrutura um aplicativo como uma coleção de serviços pequenos e fracamente acoplados que operam juntos para atingir um objetivo comum. Como funcionam de forma independente, podem ser adicionados, removidos ou atualizados sem interferir em outros aplicativos, proporcionando teste mais fáceis, produtividade, flexibilidade e escalabilidade aprimoradas.



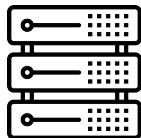
CONHECENDO ARQUITETURA DE SOFTWARE

CARACTERÍSTICAS DOS MICROSERVIÇOS:

- São amplamente desacoplados;
- São tratados como componentes independentes;
- Autonomia para os desenvolvedores e equipes trabalharem independente uns dos outros;
- Permite lançamentos frequentes através de entrega contínua;
- Liberdade para escolher tecnologias e ferramentas para o desenvolvimento;
- Se uma feature apresentar problemas, não afeta os demais (a não ser que seja uma feature core);
- Componentes podem escalar conforme necessidade individual;



INFRAESTRUTURA - ONDE TUDO FICA ARMAZENADO?



Tradicional

- Gerencia a própria instalação dos componentes
- Datacenters
- Alto custo
- Necessita de espaço físico e grande quantidade de hardware

vs



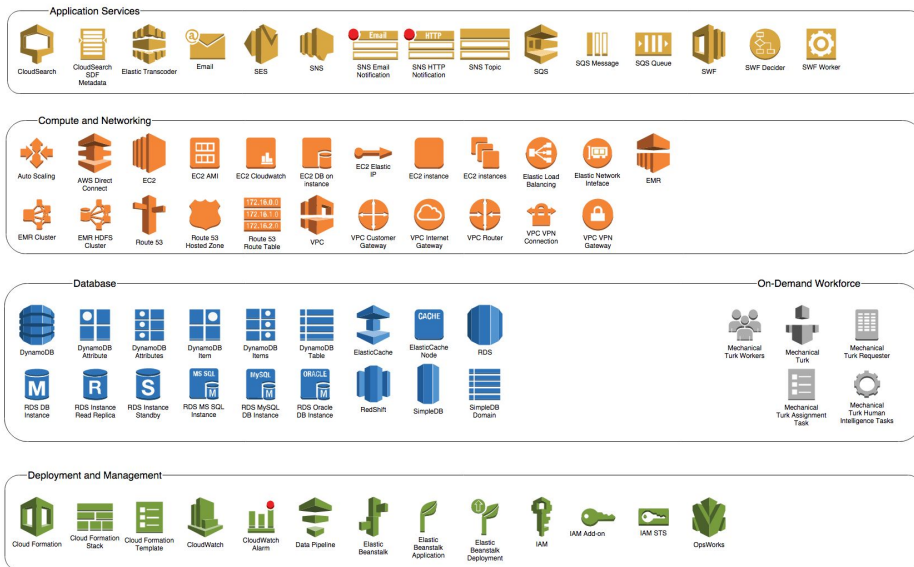
Cloud

- Não necessita de espaço físico e investimento em hardware
- Entrega on-demand
- Economia
- Escalabilidade
- Segurança
- etc

INFRAESTRUTURA - ONDE TUDO FICA ARMazenado?



INFRAESTRUTURA - ONDE TUDO FICA ARMAZENADO?



Compute



App Engine



Compute Engine



Container Engine

Storage



Cloud Storage



Cloud Datastore



Cloud SQL



Cloud Bigtable

Big Data



BigQuery



Cloud Dataflow



Cloud Dataproc



Cloud Pub/Sub

Services



Cloud Endpoints



Translate API



Prediction API

O QUE SÃO BANCO DE DADOS?

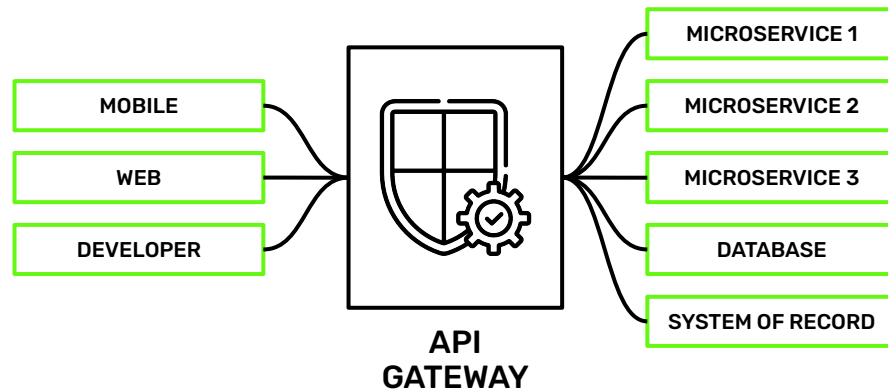
- É onde os serviços guardam as informações que recebem ou processam, podendo também recuperá-las quando necessário;
- Possui rotinas de backup de acordo com a necessidade do negócio;
- É a peça fundamental para o bom funcionamento de uma empresa/produto/serviço.

TIPOS DE BANCO DE DADOS?

- **Relacionais:** São criados no paradigma da orientação a conjuntos armazenados em tabelas, onde cada uma terá atributos, linhas ou registros responsáveis por organizar as informações.
 - Linguagem utilizada SQL (Structured Query Language)
- **Não Relacionais:** tem um esquema dinâmico para os dados não estruturados, podendo ser armazenado em várias formas (imagens, tabelas, documentos, grafos e etc).
 - Linguagem utilizada NoSQL (Not Only SQL)

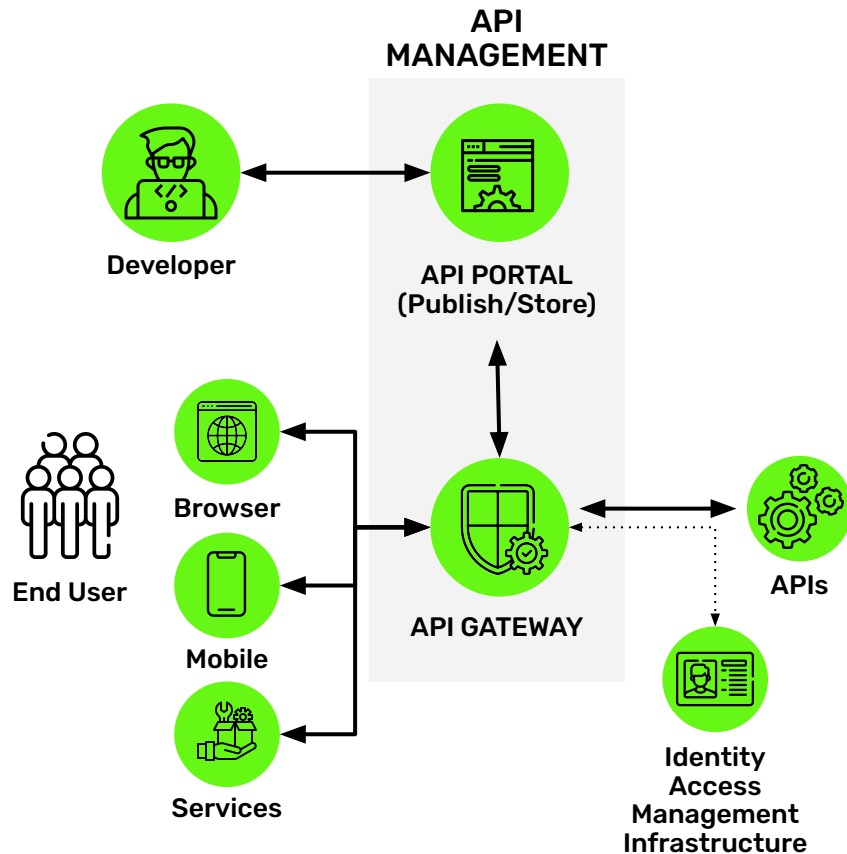
API GATEWAY E A SUA IMPORTÂNCIA

Um gateway de API é a interface que fica na frente de uma API e é o ponto de entrada única para APIs e microsserviços de back-end definidos (que podem ser internos e externos). atuando como protetor, reforçando a segurança, garantindo escalabilidade e alta disponibilidade.

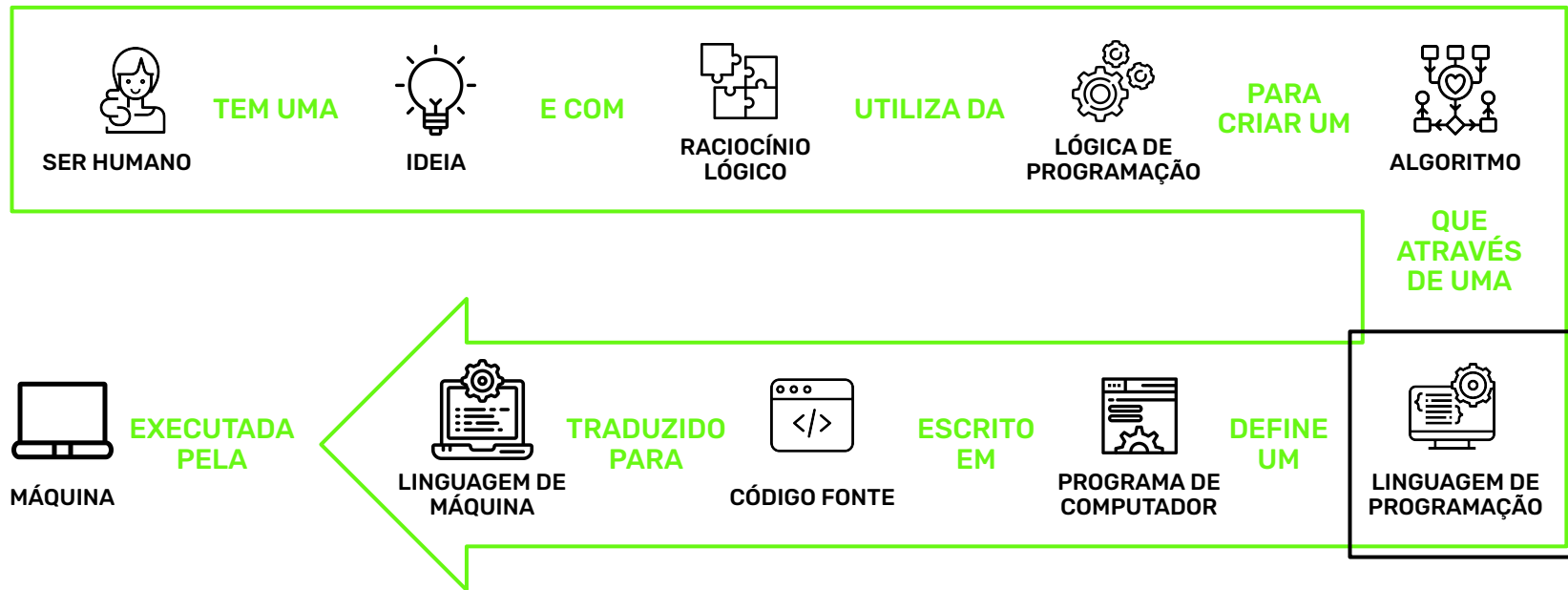


API MANAGER
















O gerenciamento de API é o processo de distribuição, controle e análise de APIs que conectam aplicações e dados por toda a empresa e nas clouds. O objetivo é permitir às organizações que criam APIs (ou utilizam APIs de terceiros) a monitorar atividades e garantir que as necessidades dos desenvolvedores e das aplicações sejam atendidas.



LINGUAGENS DE PROGRAMAÇÃO

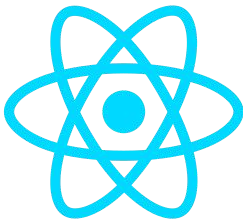


RANKING DE LINGUAGENS DE PROGRAMAÇÃO

Nov 2021	Nov 2020	Change	Programming Language	Ratings	Change
1	2	▲	 Python	11.77%	-0.35%
2	1	▼	 C	10.72%	-5.49%
3	3		 Java	10.72%	-0.96%
4	4		 C++	8.28%	+0.69%
5	5		 C#	6.06%	+1.39%
6	6		 Visual Basic	5.72%	+1.72%
7	7		 JavaScript	2.66%	+0.63%
8	16	▲	 Assembly language	2.52%	+1.35%
9	10	▲	 SQL	2.11%	+0.58%
10	8	▼	 PHP	1.81%	+0.02%
11	21	▲	 Classic Visual Basic	1.56%	+0.83%
12	11	▼	 Groovy	1.51%	-0.00%
13	15	▲	 Ruby	1.43%	+0.22%
14	14		 Swift	1.43%	+0.08%
15	9	▼	 R	1.28%	-0.36%

T

! O QUE SÃO FRAMEWORKS?



EXEMPLO

- Rapid API



TIPOS DE 'AS SERVICE1

IaaS

Infrastructure as a Service

PaaS

Platform as a Service

SaaS

Software as a Service

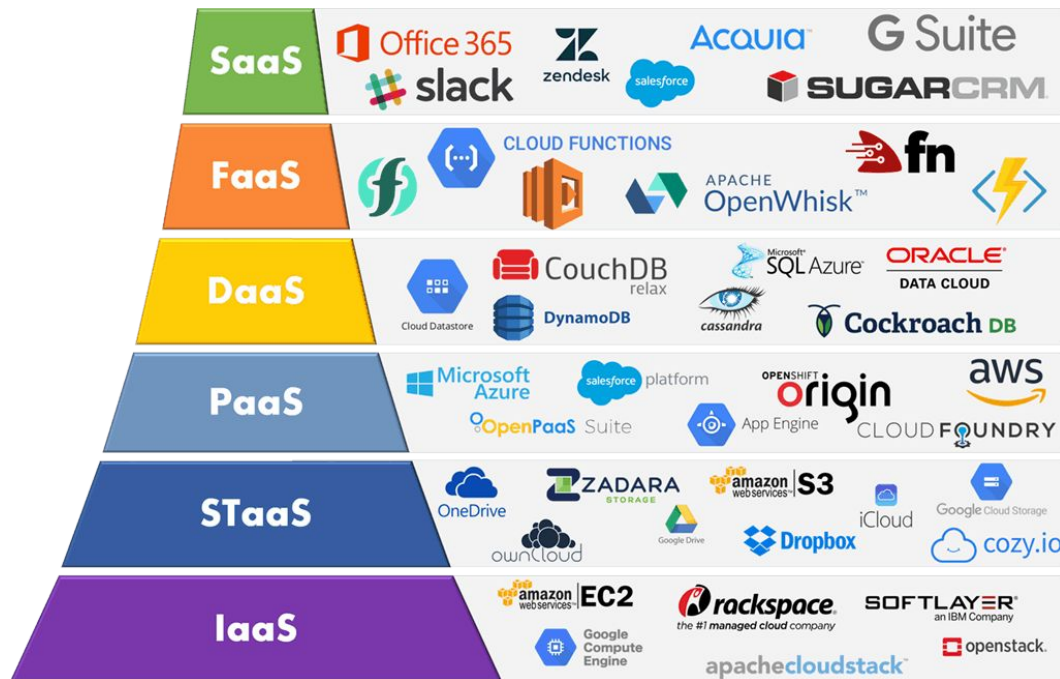
iPaaS

Integration Platform as a Service

STaaS

Storage as a Service

TIPOS DE 'AS SERVICE'



O QUE É CULTURA DevOps?

- **Integração Contínua:**

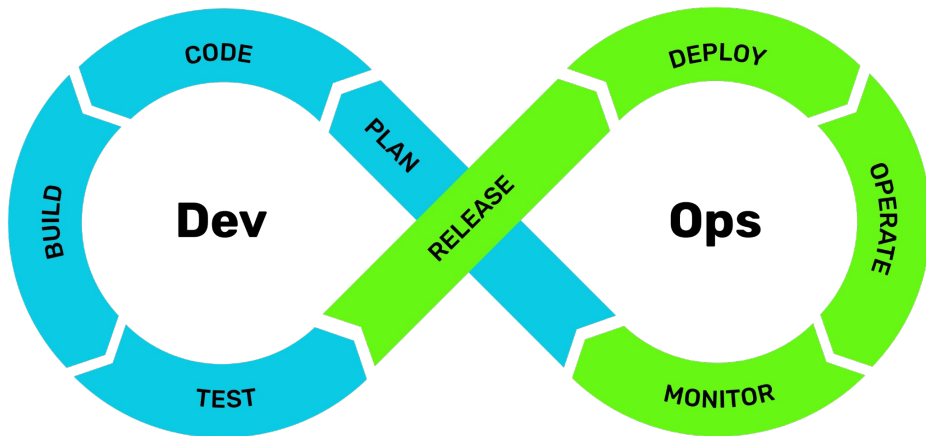
fácil transferência de conhecimento e experiências entre as áreas de Desenvolvimento, Operações e Apoio.

- **Implantação Contínua:**

liberação rápida e contínua de novas versões de software ou serviços.

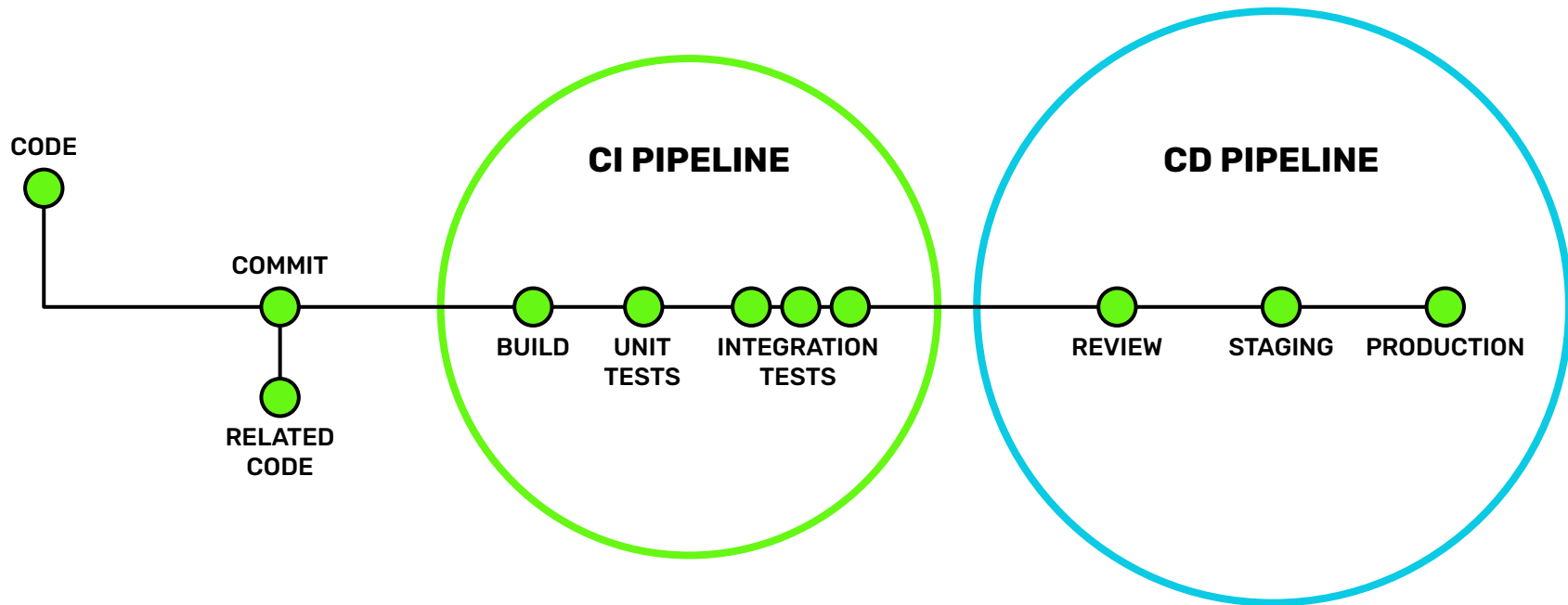
- **Feedback contínuo:**

feedbacks frequentes das equipes envolvidas em todas as fases do ciclo de vida do software ou serviço.



T

EXEMPLE



T

ENGENHARIA E PRODUTO EM CONJUNTO

SINERGIA COM O TIME DE DESENVOLVIMENTO

- Comunicação
- Escuta ativa
- Diagnósticos do Produto
- Acompanhamento de Métricas
- Gerenciamento/Mapeamento de Débitos Técnicos
- Cooperação
- Confiança
- Consenso





COMO GERENCIAR DÉBITOS TÉCNICOS?

VISIBILIDADE E COMUNICAÇÃO:

Dar visibilidade ao conceito e comunicá-lo em linguagem de negócios a executivos e demais stakeholders é fundamental.

CONSIDERAÇÃO:

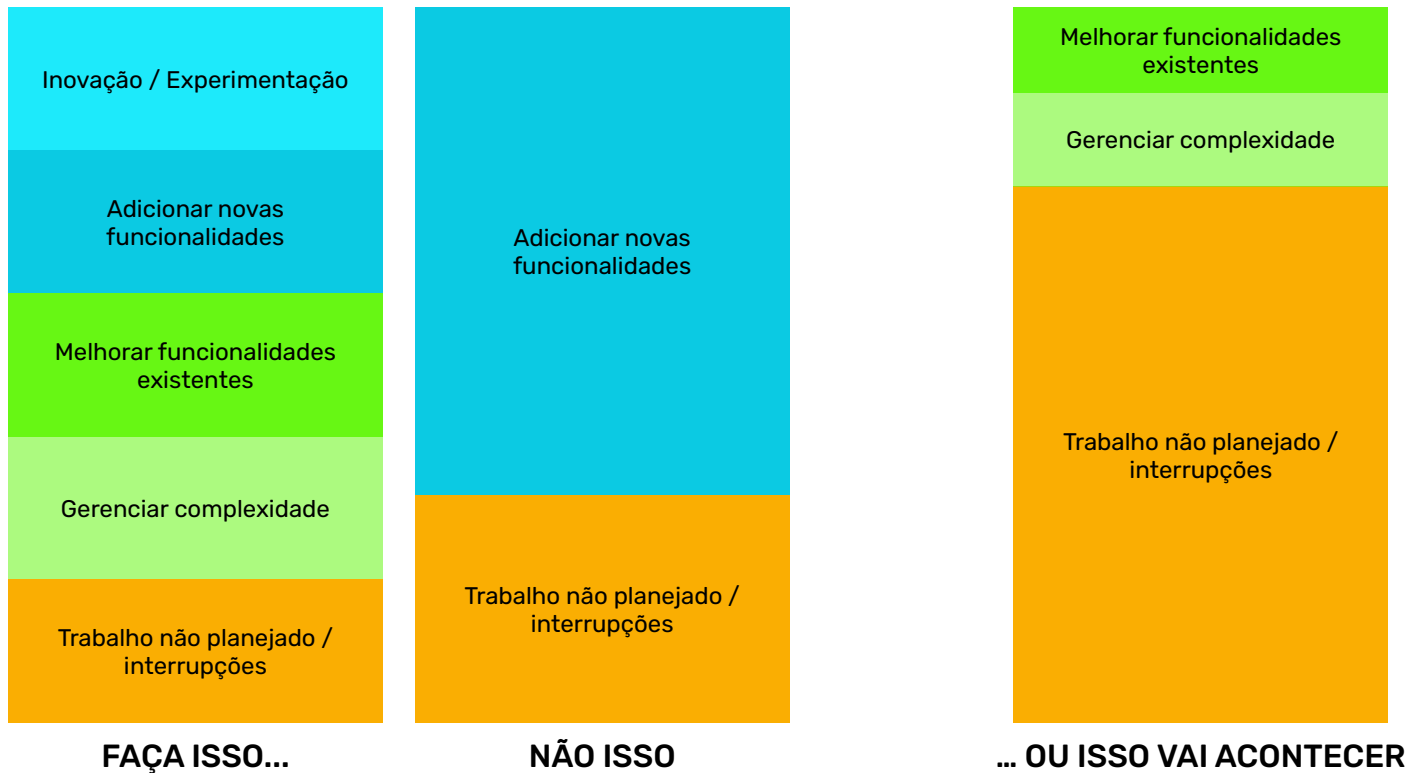
utilizar a visibilidade da dívida técnica para avaliar impactos dela tanto na performance da organização quanto no desenvolvimento de novos projetos, assim como embasar a tomada de decisão sobre um gerenciamento explícito, seja sobre a assunção de novas dívidas, análises de risco, necessidade de refatorar e etc.

GERENCIAMENTO EXPLÍCITO:

criação de frameworks e adoção de ferramentas para gestão de dívidas técnicas existentes, como pagamento em iterações mínimas e releases de limpeza, avaliação de riscos, mesmo que conscientemente assumidos.

MONITORAMENTO:

mensurar níveis de dívida técnica ao longo do tempo, identificar e reconhecer tendências, disseminar boas práticas de integração da dívida técnica ao processo de desenvolvimento a fim de torná-la produtiva etc.



T



RESUMO DA AULA

A vertical bar with a gradient from green at the top to blue at the bottom.

COMO OS PRODUTOS DIGITAIS SE COMUNICAM?

- Conhecer melhor o seu produto
- Entender o que os desenvolvedores falam
- Ajudar o time

ARQUITETURA

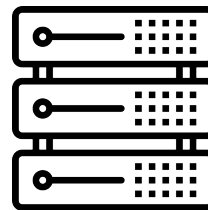
- APIs
 - Privadas
 - Parceiros
 - Públicas
- HTTP
- Métodos HTTP
- Códigos HTTP



BROWSER

HTTP request

method: GET
path: /index.html

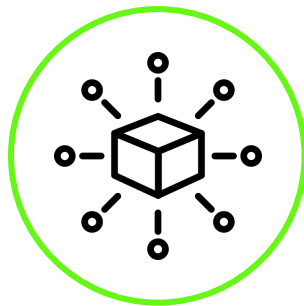


SERVER

HTTP response

status: 200
content-type: text/html;
charset= UTF-8

ARQUITETURA



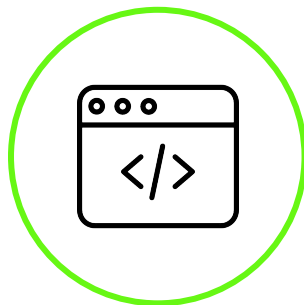
- Monolitos
- Microsserviços

INFRAESTRUTURA



- Cloud
- Banco de Dados
- API Gateway
- API Manager
- DevOps
- Um monte de 'as a Service'

STACK DE DESENVOLVIMENTO



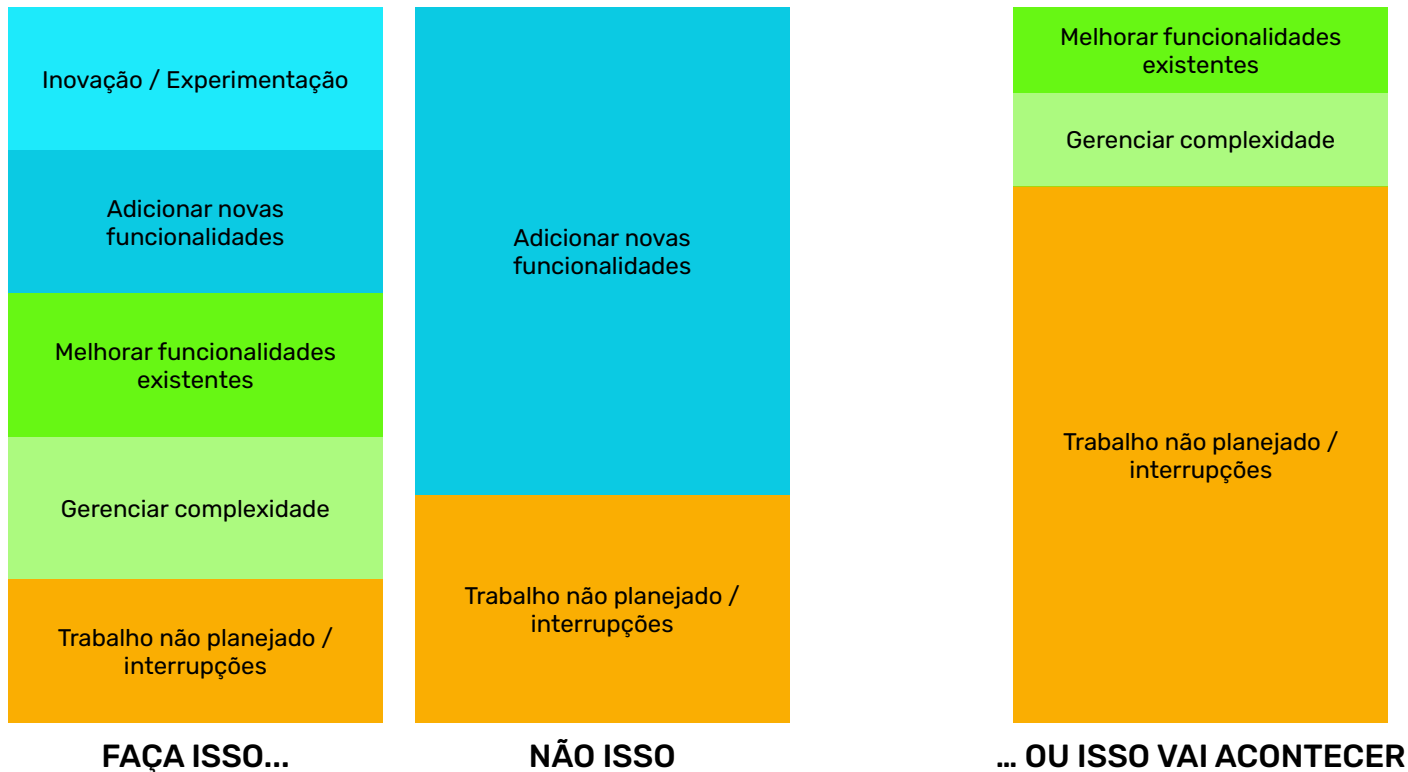
- Linguagens de Programação
- Frameworks
- Web, App, APIs

SINERGIA COM O TIME DE DESENVOLVIMENTO



- Comunicação
- Escuta ativa
- Diagnósticos do Produto
- Acompanhamento de Métricas
- Gerenciamento/Mapeamento de Débitos Técnicos
- Cooperação
- Confiança
- Consenso





**TECNOLOGIA NÃO É MAIS
INÍCIO E FIM, MAS SIM
MEIO PARA TUDO!**

