



Apresentação do Instrutor

1 Apresentação do Instrutor

Doutorando em Computação – UFES

Mestre em Informática – PUC-Rio

Pós-Graduando em Ciência de Dados – PUC-Rio

Pós-Graduado em Análise, Projeto e Gerência de Sistemas de Informação – IFF

Bacharel em Ciência da Computação – UCAM-Campos

Petrobras – 2006 – Atual

Engenheiro de Dados

Arquiteto de Software

Net-Campos – 2005 – 2006

Administrador de Redes

Slide 2

- 1 Criar tópico para apresentação da disciplina e levar para ele os slides 2 e 3. Será usado no vídeo referenciado na pág. 1 do roteiro.

Andreza Vieira; 05/10/2020



Apresentação da Disciplina

Infraestrutura de Desenvolvimento Web

Visão Geral da Disciplina

INFRAESTRUTURA PARA DESENVOLVIMENTO WEB

EMENTA: Introdução à Infraestrutura para desenvolvimento Web (front-end, back-end e servidores web). Cloud Computing (Containers com Kubernetes e Virtualização com Dockers). Ferramenta NPM. Apache Kafka. Estudo de caso.

CONTEÚDO PROGRAMÁTICO:

UNIDADE I: Introdução à Infraestrutura

- 1.1 Front-End;
- 1.2 Back-End;
- 1.3 Servidores Web.

UNIDADE II: Cloud Computing

- 2.1 Containers;
 - 2.1.1 Kubernetes
- 2.2 Virtualização
 - 2.2.1 Dockers

UNIDADE III: A Ferramenta NPM

- 3.1 Aspectos Gerais;
- 3.2 Download;
- 3.3 Criando Módulos;

UNIDADE IV: Apache Kafka

- 4.1 Aspectos Gerais;
- 4.2 Semântica;
- 4.3 Estudo de Caso.



Unidade 1 – Introdução à Unidade I - Introdução à Infraestrutura

Unidade 1 - Introdução à Infraestrutura

1.1 Front-End

- 1.1.1 Atividades no desenvolvimento de Front-end
- 1.1.2 Papéis no desenvolvimento de Front-end
- 1.1.3 Tecnologias empregadas no Front-end

1.2 Back-End

- 1.2.1 Atividades no desenvolvimento de Back-end
- 1.2.2 Papéis no desenvolvimento de Back-end
- 1.2.3 Tecnologias empregadas no Back-end

1.3 Servidores Web

- 1.3.1 Servidor Web
- 1.3.2 Servidor de Aplicação
- 1.3.3 Servidor de Banco de Dados
- 1.3.4 Infraestrutura para Serviços
- 1.3.5 Conclusão da Unidade



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Processos no desenvolvimento de Front-End

Definição de Front-end

O Front-end é a parte da aplicação responsável por interagir com o usuário.

É ela que é apresentada ao usuário e por meio dela o usuário consegue enviar requisições e obter respostas da aplicação.

Atividades

O desenvolvimento de Front-end tem evoluído com o passar dos anos. As pesquisas em IHC (Interação Humano Computador) têm trazido inovações e novas motivações para o aperfeiçoamento das atividades relacionadas a interface com o usuário.

Em linhas gerais as atividades são classificadas em:

- Gestão de Requisitos / Entendimento do Problema
- Prototipação
- Design de Interface
- Gestão de Conteúdo

No entanto, cada atividade pode ser detalhada em inúmeras tarefas.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Papéis desempenhados no Front-End

Papéis

- Antigamente
 - Web Designer
- Hoje
 - Business/Strategic Designer (Estratégico)
 - User Researcher (Entendimento)
 - Estrategista de Conteúdo (Informação)
 - Arquiteto da Informação (Organização)
 - Designer de Interação (Comportamento)
 - Visual/Graphic Designer (Estética)
 - Analista de Usabilidade e Métricas (Medição)
 - Testador
 - Gerente de Configuração e Mudança
 - Desenvolvedor

Papéis

Com a evolução das atividades, novos papéis foram surgindo.

Antigamente, a interface gráfica era pouco valorizada, mas as evidências dos dias atuais mostram o quanto o Front-end agrega valor a solução.

Papéis

É importante ressaltar que em times que utilizam metodologia ágil os papéis citados não são suprimidos, porém desempenhados por membros do time auto organizável.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Tecnologias empregadas no Front-End

Construção de Páginas Web

HTML e HTML5

Linguagem de marcação utilizada para criação de páginas na internet. HTML5 é a quinta versão de HTML.

XHTML

Uma extensão de HTML combinada com a linguagem de marcação XML.

CSS

Mecanismo utilizado para adicionar o estilo nas páginas web. Com ele é possível definir fonte, tamanho, cores, espaçamento, etc no HTML.

Transporte de Informação

XML

Linguagem de marcação que utiliza tags para categorizar e armazenar informações de forma que possam ser transportadas por diferentes plataformas.

JSON

O JSON é um formato de troca de dados entre sistemas independente de linguagem de programação derivado do JavaScript.

DOM

Convenção multiplataforma para representação e interação com objetos em documentos HTML, XHTML e, XML.

OWL

OWL foi projetada para disponibilizar uma forma comum para o processamento de conteúdo semântico da informação na Web.

RDF

Vem sendo utilizada como um método geral para a descrição conceitual ou de modelagem de informação, implementada em recursos Web.

Frameworks

React

React é uma biblioteca de JavaScript open-source para construção de interface gráfica e componentes de interface. É mantida pelo Facebook, uma comunidade de desenvolvedores autônomos bem como algumas empresas.

Pode ser utilizado para construção de uma simples página ou aplicações para celular.

Angular

Angular é um framework para aplicações web open-source mantido pelo time Angular na Google.

Vue

Vue foi construído por um ex-integrante da equipe do Angular (Google) visando prover um framework open-source que fosse mais leve que o Angular.

Script

Javascript

AJAX

AngularJS

Tanto AJAX quanto AngularJS são baseadas em Javascript que é uma linguagem de alto nível, interpretada com tipagem fraca.

jQuery

jQuery é uma biblioteca de código aberto que utiliza a licença MIT em seu código-fonte. A sintaxe do jQuery foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos, desenvolver aplicações AJAX e criação de plugins sobre ela.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Atividades no desenvolvimento de Back-End

Definição

Desenvolvedores de Back-end são responsáveis por escrever a aplicação no lado do servidor, tais como web services, APIs, componentes e as demais funcionalidades que serão utilizadas pelo front-end e aplicações mobile.

Todas as regras de negócio precisam ser implementadas pelo desenvolvedor de back-end. Mesmo que algumas regras possam ser validadas na interface gráfica ou até mesmo na camada de armazenamento, a lógica do negócio precisa ser garantida no back-end da aplicação.

Em alguns modelos arquiteturais, o back-end pode ser visto como camada de negócio da aplicação, porém as arquiteturas atuais estendem essa camada para serviços, brokers e até mesmo componentes externos.

Atividades

Dentre as atividades de um desenvolvedor de back-end podemos destacar:

- Gerência de Configuração e Mudança
- Testes
- Deploy
- Integração
- Implementação
- Métrica



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Papéis desempenhados no Back-End

Papéis

Muitos papéis são desempenhados por um desenvolvedor de back-end. Abaixo listamos alguns deles, ressaltando que não são somente estes papéis:

- Programador

- Testador

- Projetista

- Arquiteto

- Inspetor de Qualidade

- Gerente de Requisitos

- Implantador

- Gerente de Configuração e Mudança

Da mesma forma que citamos os múltiplos papéis exercidos pelo time ágil no desenvolvimento de front-end, aplica-se ao back-end, pois as atividades existem independente do processo de desenvolvimento.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Tecnologias empregadas no Back-End



Introdução a Infraestrutura

Linguagens de Programação

Java

C# .NET

Python

Ruby

PHP

VB .NET

Frameworks

Spring

Um dos mais utilizados frameworks do mercado. Além de uma gama de clientes expressivos que o utilizam, sua linguagem, Java, tem uma forte presença na atividade de desenvolvimento de software

Django

O framework popularizou bastante o uso de Python, pois foi concebido pelo conceito de disponibilizar “tudo” que necessário para o desenvolvedor

Rails

Framework para a linguagem Ruby cujo conceito se assemelha ao Django, ou seja, entregar o máximo possível de coisas que um desenvolvedor venha precisar.

Laravel

Voltado para o desenvolvimento PHP, segue o padrão arquitetural MVC e possui uma variedade de ferramentas que facilitam o desenvolvimento, além de aumentar a produtividade.

Pyramid

Apesar de pouco tempo no mercado, se apresenta como um framework de bastante representatividade no desenvolvimento Python



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Servidor Web

Definição

Responsável por responder as requisições HTTP do cliente (navegador) no mesmo protocolo.

Em linhas gerais, ele disponibiliza um serviço capaz de permitir o acesso às páginas armazenadas no servidor através de uma porta específica (geralmente 80)

Para algumas linguagens, o servidor web é capaz de interpretar o código da página dinâmica e apresentar para o cliente (navegador) uma versão estática do conteúdo

Servidores Web

Apache Tomcat

Criado em 1995, é o principal produto da Apache Foundation. Este servidor web além de ser um dos mais antigos é um dos mais utilizados no mercado. Dele foram derivados várias outras soluções, tais como Oracle HTTP Server, IBM HTTP Server, etc

IIS (Internet Information Services)

Solução Microsoft para prover acesso a páginas. Além de servidor Web o IIS também é servidor de aplicação

WAMP

Servidor utilizado para soluções PHP, uma vez que a linguagem é de script e precisa ser interpretada antes de apresentar ao navegador do usuário.

Também utilizado para soluções Python e Perl



Obrigado!

✉ prof.guylerme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylerme-figueiredo-b8497939](https://www.linkedin.com/in/guylerme-figueiredo-b8497939)



Unidade 1 – Servidor de Aplicação

Definição

Servidor que disponibiliza um ambiente para instalação e execução de softwares.

Na evolução das arquiteturas de aplicações, os sistemas tendem a utilizar uma arquitetura mais centralizada, retirando das máquinas cliente a responsabilidade pela execução da aplicação.

Com isso, tornou-se necessário ter servidores centralizados que pudessem disponibilizar o serviço para os usuários.

Em linhas gerais, o servidor web ele centraliza a execução de um software, sem ser necessário o mesmo estar instalado em cada máquina cliente.

Servidores de Aplicação

Uma vez que o servidor de aplicação irá executar o software, ele é fortemente vinculado a linguagem de programação.

Desta forma, os servidores de aplicação com maior expressividade no mercado são:

Para Java

- Red Hat JBoss
- Glassfish
- Apache Tomcat
- Oracle WebLogic
- IBM WebSphere

Para .NET (C#, VB, ASP, etc)

- IIS

Para Python

- Zope

Para PHP

- Zend



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Servidor de Banco de Dados

Definição

O Servidor de Banco de Dados por muitos é considerado o coração do sistema. Uma vez que é responsável por armazenar o dado da aplicação, este elemento na arquitetura tem sua relevância em evidência.

Em geral, um SGBD (Sistema de Gerenciamento de Banco de Dados) precisa garantir o acrônimo ACID - Atomicidade, Consistência, Isolamento e Durabilidade, ou seja, o dado a ser armazenado precisa sempre ser persistido na totalidade (Atomicidade), mantendo o SGBD consistente, impedindo que transações paralelas interfiram (Isolamento) e por fim fazendo que o dado permaneça persistido (Durabilidade)

Este serviço é um dos mais antigos no desenvolvimento de software.

Servidor de Banco de Dados

Nos dias atuais, muitas tecnologias de banco de dados têm sido utilizadas na construção de aplicações. Para cada necessidade do negócio, existe uma alternativa para SGBD que seja mais compatível.

Em uma visão resumida, podemos dividir as soluções em quatro grupos:

- Banco de Dados Relacional: Armazena dados em formato de tabelas. Por exemplo: Oracle, Microsoft SQL Server, MySQL, Postgres, dentre muitos outros.
- Banco de Dados em Grafo: Armazena os dados em formato de grafo, ou seja, arestas e vértices. Ex: Neo4J, ArangoDB
- Banco de Dados NoSQL: Utilizado para dados não-estruturados ou semi-estruturados, tais como fotos, documentos, etc. Ex.: MongoDB, Cassandra, Virtuoso, ElasticSearch, etc.
- Filas de Mensagem: Utilizado para armazenar mensagens, ou seja, elementos de pequeno tamanho que pode possuir uma ordem e uma relação entre si. Tem a característica de ser um dados mais contínuo. Ex. Kafka, SQS, etc.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 1 – Infraestrutura para Serviços

Arquitetura

- As principais arquiteturas separam as responsabilidades em camadas
- Cada camada pode estar em uma ou mais máquinas diferentes
- O número de camadas (contado do lado servidor) varia em função da complexidade a ser lidada
- Sistemas cliente-servidor tradicionais têm somente 1 camada no lado servidor
 - Armazenamento
- Arquiteturas em 2 camadas têm separação das entidades em
 - Apresentação
 - Armazenamento
- Arquiteturas em 3 camadas têm separação das entidades em
 - Apresentação
 - Aplicação
 - Armazenamento

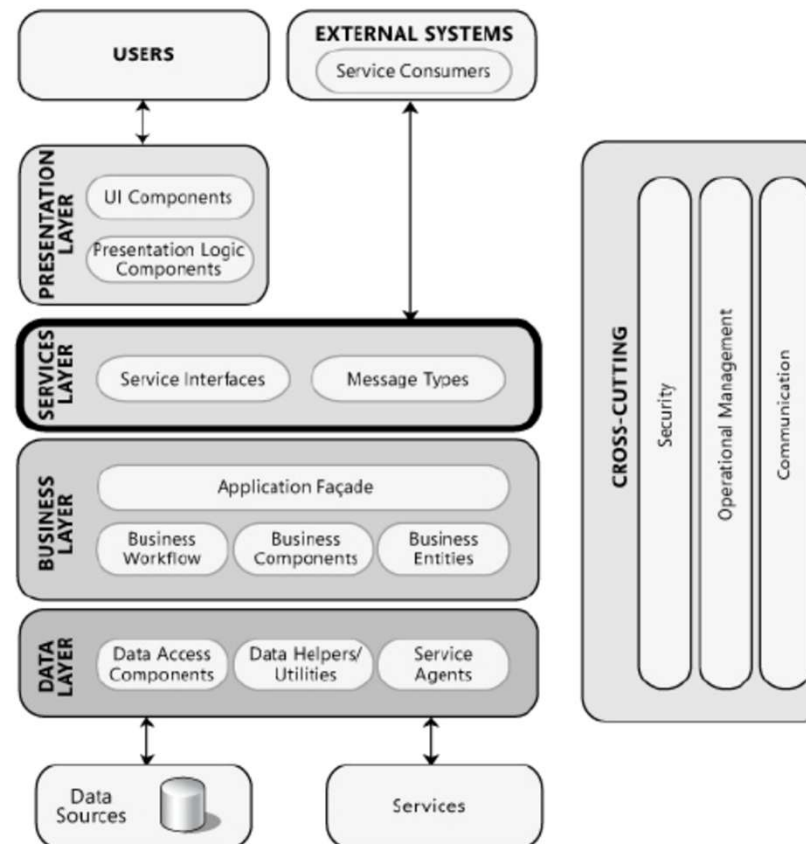
MVC

- Divide aplicação interativa em 3 partes
 - Model: regras de negócio e dados (core)
 - View: apresenta informações ao usuário
 - Controller: trata entrada de usuário e manipula o modelo

N-Tier

- Padrão geral de distribuição
- Componentes são separados em servidores diferentes
- Comumente, escolhemos um entre os padrões 2-tier, 3-tier ou 4-tier

Arquitetura Baseada em Serviços (SOA)





Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



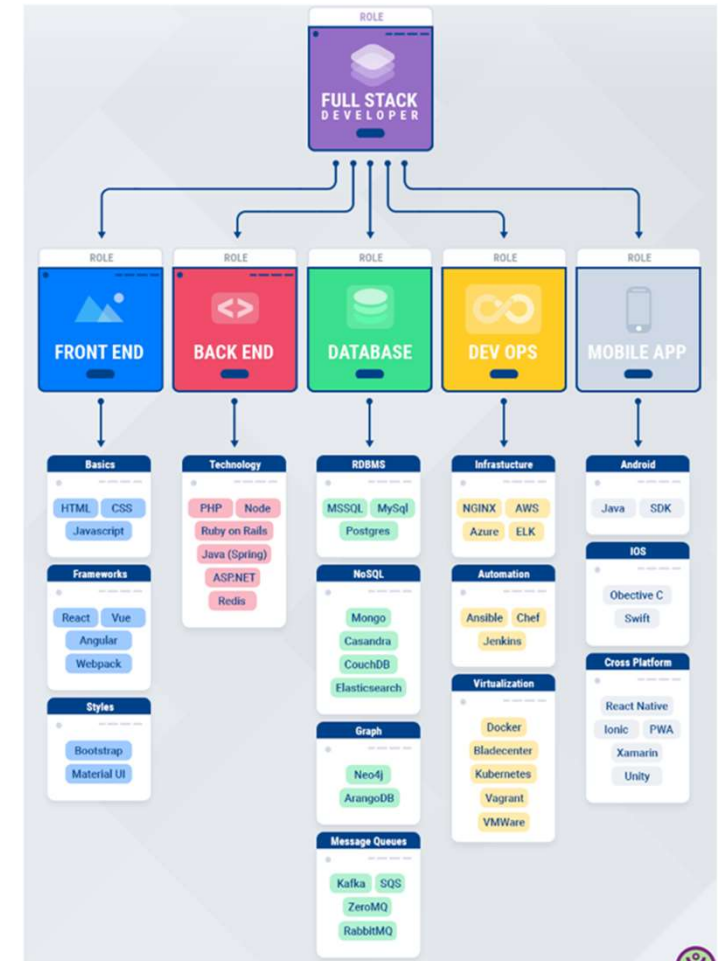
Unidade 1 – Conclusão da Unidade

Desenvolvedor Front-end e Back-end

Considera-se um desenvolvedor Full Stack aquele que é capaz de desempenhar tanto o papel de Desenvolvedor Front-End quanto Back-End

Esse perfil requer muito estudo e prática nas diversas disciplinas de desenvolvimento de software.

A figura ao lado mostra as áreas de conhecimento para cada perfil, conforme abordamos nesta unidade.



Servidores

Por fim, mostramos que quanto mais complexa a solução, mais componentes arquiteturais ela demanda, e consequentemente, mais servidores para desempenhar os papéis.

De toda as opções que o mercado oferece, o importante é harmonizar os servidores web, de aplicação e de banco de dados de forma que contribuam de maneira eficaz para o desempenho da aplicação

Na próxima unidade iremos abordar o tema Cloud Computing e como estes diversos serviços podem estar disponíveis de forma rápida, barata e escalável.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Introdução à Unidade II - Cloud Computing



Cloud Computing

Unidade II - Cloud Computing

- Introdução a Cloud Computing
- Arquitetura em Nuvem
- Introdução a Containers
- Tipos de Containers
- Introdução a Dockers
- Estudo de Caso
- Introdução a Kubernetes
- Estudo de Caso
- Introdução a Virtualização
- Tipos de Virtualização
- Conclusão da Unidade



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Introdução à Cloud Computing

Cloud Computing

Introdução a Cloud Computing



Imagem: Shutterstock

Em linhas gerais, computação em nuvem é um conjunto de data centers provendo serviços computacionais

Papéis na Nuvem

- Em geral, temos 3 papéis em uma prestação de serviço em nuvem:
- 1- Provedor de Serviços: detentor dos recursos computacionais que “aluga” seus recursos para seus clientes
- 2- Broker: Agente intermediário na contratação e prestação de serviço em nuvem.
- 3- Usuário/Cliente: quem contrata os recursos do provedor para exercer sua atividade fim

Vale ressaltar que nesta relação de contratação de serviço, não é obrigado a contratação via broker

O broker age como um facilitador, ou até mesmo um centralizador na prestação de serviço multi-cloud, ou seja, um broker pode intermediar a relação entre um usuário e 2 ou mais provedores de serviço.

Tipos de Nuvem

- Existem 3 tipos de nuvem:
- Nuvem Privada: Uma nuvem privada pertence a uma organização e é operada por ela. Eles criam um ambiente de nuvem no próprio datacenter e provê acesso self-service aos recursos computacionais aos seus usuários.
- Nuvem Pública: A nuvem pública pertence a um provedor de serviços de nuvem. Ele provê recursos e serviços a múltiplas organizações e usuários.
- Nuvem Híbrida: A nuvem híbrida combina ambas: pública e privada, e escolhe o melhor local para executar seus serviços e aplicações.

Cloud Computing

Tipos de Nuvem

- Muitas empresas têm provido serviço de nuvem



Google Cloud Platform

 Alibaba Cloud



Azure



IBM Cloud

Unyleya 
EDUCACIONAL



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)

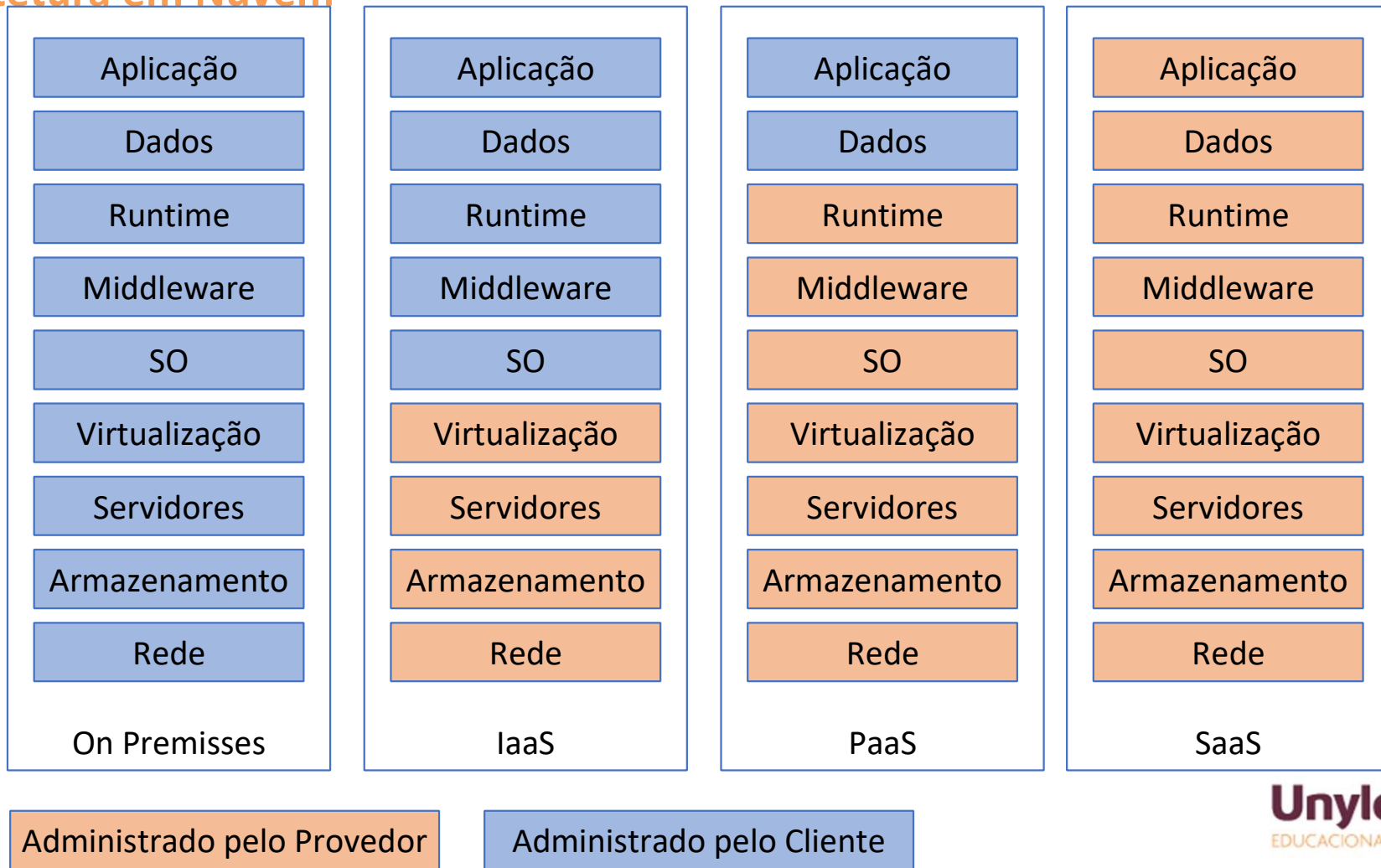


Unyleya
EDUCACIONAL



Unidade 2 – Arquitetura em Nuvem

Arquitetura em Nuvem



Infrastructure as a Service



- Neste modelo você aluga servidores de infraestrutura e máquinas virtuais do provedor de serviço. O recurso de infraestrutura é provisionado e gerenciado pela internet
- Modelo de contratação mínimo junto ao provedor

Plataform as a Service



- Neste modelo você aluga servidores de aplicação, ou seja, uma máquina completamente gerenciada e administrada pelo provedor de serviços de nuvem.
- Neste modelo o usuário implanta uma aplicação e gerencia esta aplicação e seus dados

Software as a Service



- Neste modelo você utiliza uma aplicação fornecida por um provedor, ou seja, os usuários conectam a aplicações baseadas na nuvem, tais como e-mail, calendário, ferramentas de edição de texto, etc.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Introdução a Virtualização



Cloud Computing

Virtualização

- É a técnica que permite um mesmo computador físico executar vários computadores virtuais ao mesmo tempo
- No geral, a virtualização pode ser provida via software ou via hardware.
- No caso de virtualização via software, o computador hospedeiro possui um SO e o virtualizador é executado sobre o SO

Benefícios

- Otimização de recursos: utilizar um mesmo computador para disponibilizar vários serviços e servidores é uma boa forma de aproveitar ao máximo os recursos oferecidos pelo computador
- Otimização de espaço: Desta forma não se faz necessário ter uma dezena de servidores para provimento de uma quantidade pequena de serviços
- Escalabilidade: para aumentar as capacidades da máquina virtual basta ajustar a configuração. Enquanto a máquina hospedeira tiver recurso, pode ser escalado as máquinas virtuais.
- Produtividade: a implementação é bem rápida na virtualização, uma vez que a infraestrutura já está instalada
- Múltiplas plataformas: é possível disponibilizar em uma mesma máquina física várias outras máquinas virtuais utilizando plataformas diferentes.

Cloud Computing

Soluções de Virtualização





Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Tipos de Virtualização

Tipos de Virtualização

- Além de ser o método de virtualização mais antigo, a virtualização de máquinas permite prover vários tipos de serviço, tais como:
 - Virtualização de Servidor Web
 - Virtualização de Servidor de Banco de Dados
 - Virtualização de Desktop
 - Entre outras

Tipos de Virtualização

- No entanto, existem outras opções de virtualização
 - Virtualização de hardware: O hypervisor cria versões virtuais de computadores e sistemas operacionais e os consolida em um servidor físico. Desta forma, a utilização do hardware é otimizada
 - Virtualização de rede: combinação dos equipamentos físicos de rede em um único recurso baseado em software.
 - Virtualização de armazenamento: consiste na compilação de discos rígidos físicos em um único cluster.

Tipos de Virtualização

- Outra opção de virtualização é a utilização de container. Com ele, empacotamos uma solução completa para ser executada em qualquer máquina.
- Na próxima unidade, iremos abordar este assunto.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Introdução a Containers

Introdução a Containers

- Os containers são ambientes de virtualização.
- Diferente das máquinas virtuais, o container não se administra o sistema operacional
- Eles são leves e projetados para serem criados, escalados e parados dinamicamente
- Containers permitem respostas rápidas em casos de mudança na demanda, além de reiniciar o serviço rapidamente em caso de problema no sistema ou no hardware



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Tipos de Containers

Tipos de Container

- Em geral os containers podem ser divididos em dois grupo: soluções open-source e soluções proprietárias
- Das soluções open-source temos:
 - Docker
 - OpenShift
 - Apache Mesos
 - Entre outros
- Destes, o Docker foi a primeira, dentre as maiores, a ter uma solução open-source implementando o conceito de container de fato.

Tipos de Container

- O mais importante na escolha da solução para container é a aderência à sua infraestrutura, ou seja, buscar soluções que de fato isolem a aplicação compartimentalizada e ofereça escalabilidade e flexibilidade
- Muitos clientes optam por soluções proprietárias devido a estar relacionado com uma prestação de serviço de suporte, porém, soluções como Docker possui uma comunidade bastante considerável contribuindo.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Introdução a Dockers

Introdução a Dockers

- Docker é um gerenciador de Containers Open-Source criado pela DotCloud e disponibilizado para a comunidade
- Em linhas gerais o Docker é um empacotador de ambiente, ou seja, consegue empacotar não só uma aplicação como também o ambiente necessário para ela ser executada
- Com isso, o deploy da aplicação fica mais fácil, pois não se tem “surpresas” como ambiente diferente, variável de ambiente do SO não configurada e outros detalhes particulares de um ambiente que influencia a execução da aplicação



Cloud Computing

Dockers

- Além de facilitar o deploy, a escalabilidade de serviços de back-end é mais fácil, pois funciona como se fosse uma máquina isolada
- Permite múltiplas instâncias utilizando o mesmo kernel



Cloud Computing

Dockers x Virtual Machine

- Vantagens de utilização de Docker sobre VM
 - Não é necessário administrar o Sistema Operacional
 - Menor complexidade no gerenciamento
 - Componentes mais atômicos



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Estudo de Caso

Instalando Docker em Unix

- Instalação
 - `curl -ssl https://get.docker.com | sh`
- Inicializando o serviço
 - `/etc/init.d/docker start`



Cloud Computing

Listando as Imagens e executar uma

- Listar imagens
 - `docker images`
- Executar uma imagem
 - `docker run -it IMAGEM PROCESSO`

Sair do Container

- Sair do Container finalizando o docker
 - CTRL + D
- Sair do container mantendo o docker ativo
 - CTRL + PQ
- Acessar o container ativo
 - docker attach CONTAINER ID



Cloud Computing

Finalizando

- Parar a execução da imagem
 - `docker stop CONTAINERID`
- Deletando a imagem
 - `docker rm CONTAINERID`



Obrigado!

✉ prof.guylerme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylerme-figueiredo-b8497939](https://www.linkedin.com/in/guylerme-figueiredo-b8497939)



Unidade 2 – Introdução a Kubernetes



Cloud Computing

Introdução a Kubernetes

- É uma plataforma de orquestração de contêineres para agendar e automatizar a implantação, o gerenciamento e o dimensionamento de aplicativos em contêiner.
- Desenvolvido por engenheiros da Google em 2014

Kubernetes

- O Kubernetes agenda e automatiza tarefas relacionadas ao container, tais como:
 - Implantação: implantar um ou mais containers em um host
 - Deploy: gerir o ciclo de vida do container
 - Descoberta de Serviço: export um container para ser encontrado por outros
 - Provisionamento de Armazenamento: montar um local de armazenamento para os containers
 - Balanceamento de carga: distribuir a carga sobre os recursos para manutenção da estabilidade



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Estudo de Caso

Inicializando um serviço

- Logar no PWK
- Clicar no botão <ADD NEW INSTANCE>
- Inicializar o node master do cluster
 - `kubeadm init --apiserver-advertise-address $(hostname -i)`
- Inicializar cluster em rede
 - `kubectl apply -n kube-system -f / "http://cloud.weave.works/k8s/net?k8s-version=\$\(kubectl version |`



Cloud Computing

Verificando

- Verificar os nós geridos:
 - `kubectl get nodes`
- Acessar o Dashboard
 - `kubectl -n kube-system edit service kubernetes-dashboard`



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 2 – Conclusão da Unidade

Conclusão

- Nesta unidade nos aprofundamos nos conceitos de cloud computing, bem como um dos recursos mais úteis neste ambiente: a virtualização
- A virtualização se mostrou um recurso poderoso para expandir as capacidades dos serviços ao tempo que é uma abordagem mais elástica
- Dentro das opções de virtualização, temos muitas abordagens, sendo a mais utilizada nos dias atuais a abordagem de container através da utilização do Docker, pois permite uma solução mais atômica, escalável e tolerante a falhas

Conclusão

- A gestão e orquestração de containers com Kubernetes é uma forma centralizada e eficaz de garantir interoperabilidade, gestão centralizada e domínio sobre o ambiente.
- Por fim, é importante destacar que os provedores de serviços na nuvem entregam soluções para todas as atividades e serviços citados até o momento. Em alguns, eles lançam produtos próprios com a mesma finalidade, em outros acabam lançando o serviço original



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Introdução à Unidade III - A Ferramenta NPM



A Ferramenta NPM

A Ferramenta NPM

- Introdução a NPM
- Noções Básicas de Node.js
- Servidor Web
- Express.js
- Aplicação
- Módulos: Criação e Gerenciamento
- Publicação de Módulo no NPM
- Linha de Comando e Comandos CLI
- Configurações NPM
- Scripts NPM para automação de tarefas



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Introdução a NPM



A Ferramenta NPM

Introdução

NPM (Node Package Manager) é um gerenciador de pacote para JavaScript

É o gerenciador de pacotes padrão para Node.js

A NPM Inc é uma subsidiária da GitHub



A Ferramenta NPM

Gerenciador de Pacotes

Gerenciador de pacotes é um conjunto de softwares utilizado para automatizar o processo de instalação, atualização, configuração e remoção de software de forma consistente

Os pacotes contém os metadados do software, tais como descrição, versão, checksum e lista de dependências para o software ser executado adequadamente



A Ferramenta NPM

Gerenciador de Pacotes

Temos no mercado diversos gerenciadores de pacotes, por exemplo:

DPKG → utilizado pelo Ubuntu

Homebrew → utilizado pelo MacOS

RPM → utilizado pelo Red Hat

Steam → Múltiplas plataformas: Windows, Linux, OSX, etc



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Noções Básicas de Node.js



A Ferramenta NPM

Noções Básicas de Node.js

É um ambiente de tempo de execução do lado do servidor de código aberto criado no mecanismo JavaScript V8 do Chrome.

Leve e eficiente

Orientado a eventos

Pode atender milhares de requisições simultâneas

A Ferramenta NPM

Noções Básicas de Node.js

Cada conexão a função de callback é chamada → alternativa que contrasta com o modelo de concorrência mais comum, onde são utilizadas threads do SO

Usuários do Node.js não precisam se preocupar com deadlock de processos, pois não existem locks

Usado para criar diferentes tipos de aplicativos, como aplicativo de linha de comando, aplicativo Web, aplicativo de bate-papo, servidor de API REST etc.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```



A Ferramenta NPM

Noções Básicas de Node.js

Permite executar código JavaScript de backend fora do navegador

Este código é executado no servidor utilizando o Google V8 Engine



Obrigado!

✉ prof.guylerme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylerme-figueiredo-b8497939](https://www.linkedin.com/in/guylerme-figueiredo-b8497939)



Unidade 3 – Servidor Web



A Ferramenta NPM

Servidor Web

Os servidores de aplicações Web geram o processo de ligação dos utilizadores de informação aos dados onde estão armazenados.

Envolvem tecnologia orientada a objetos na forma de objetos tradicionais e componentes de software.



A Ferramenta NPM

Servidor Web

Objetos e componentes (nem todo componente é um objeto) são softwares encapsulados e que podem ser reutilizáveis

O mundo dos servidores de aplicações baseia-se fortemente na estrutura e nas interfaces bem definidas dos objetos e dos componentes



A Ferramenta NPM

Servidor Web

Esta arquitetura rigorosa permite a interoperabilidade entre os produtos de vários fornecedores.

Fornecem a infraestrutura de runtime (tempo de execução) e serviços necessários para implementar aplicações ou componentes numa arquitetura de várias camadas, que suportam a Web e outras interfaces com o cliente.



A Ferramenta NPM

Servidor Web

São partes de uma infraestrutura que deve permitir que os programadores Web construam aplicações, que trabalhem com uma grande variedade de hardware e softwares-cliente e que possuam funcionalidades para serem integrados com recursos existentes.



A Ferramenta NPM

Servidor Web

Por outro lado, têm de ser de fácil utilização e capazes de fornecer mais valor de desenvolvimento do que produtos de middleware (produtos separados que servem de ligação entre duas aplicações).



A Ferramenta NPM

Servidor Web

Devem proporcionar a construção de aplicações mais elaboradas, que assegurem níveis aceitáveis de integridade, confiabilidade, disponibilidade, desempenho e segurança



A Ferramenta NPM

Servidor Web

O servidor de aplicação Web contém a lógica da aplicação e situa-se entre o servidor Web e o banco de dados.

Ele controla a conexão com o banco de dados em função das requisições do browser.



A Ferramenta NPM

Servidor Web

O ganho obtido com o servidor de aplicação é que a Web (originariamente) é um sistema que não preserva o estado, quando um cliente se conecta nela.

O protocolo HTTP que transporta HTML) é um protocolo sem estado e o próprio servidor Web (originariamente) é um servidor que não preserva o estado de conexões.



A Ferramenta NPM

Servidor Web

Web é sem estado (stateless):

Ou seja, o servidor Web “esquece” a primeira interação com o banco de dados, mesmo se cliente retornar apenas alguns segundos depois.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Express.js



A Ferramenta NPM

Express.js

Express.js é um framework para backend de aplicações web em Node.js

Express.js juntamente com MongoDB e AngularJS compõem a pilha de solução MEAN (**M**ongoDB Express.js **AN**gularJS.)



A Ferramenta NPM

Framework

É um template com diversas funções que podem ser usadas pelo desenvolvedor.

Em outras palavras, ele é uma estrutura base, uma plataforma de desenvolvimento, como uma espécie de arcabouço.

Ele contém ferramentas, guias, sistemas e componentes que agilizam o processo de desenvolvimento de soluções

Possui também um conjunto de bibliotecas, que permitem aos desenvolvedores trabalharem sobre eles para operações maiores.

Os frameworks fazem com que você não tenha que se preocupar em ficar reescrevendo códigos



A Ferramenta NPM

Framework

As funções do framework têm uma grande variedade de parâmetros, garantindo ao desenvolvedor a possibilidade de fazer personalizações, de acordo com as necessidades do projeto.

Vale ressaltar que a integração entre as diferentes funções dessa ferramenta é uma de suas principais características.

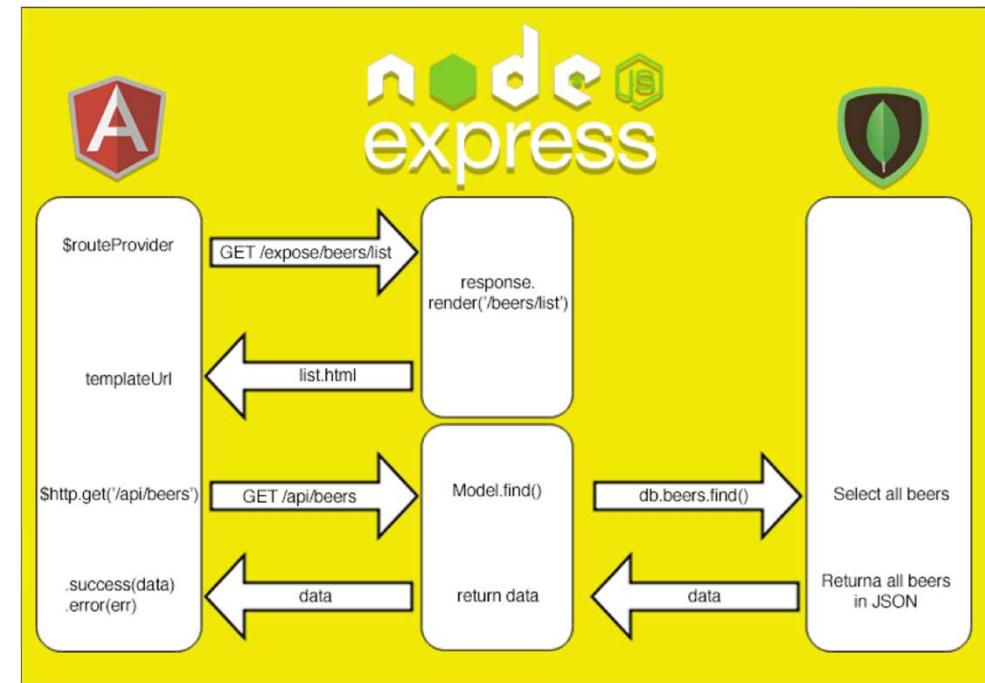
Os pontos providos pelo framework são chamados de frozen spots ou hook points. Já a instanciação e a personalização criadas pelo desenvolvedor são denominadas de hot spots.

A Ferramenta NPM

Express.js

Como framework, fornece um conjunto de funcionalidades promovendo aumento de produtividade no desenvolvimento em Node.js.

Na figura ao lado, os métodos `response.render()` e `Model.find()` são providos pelo Express.js



<https://github.com/suissa/workshop-be-mean-online-junho>



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Aplicação



A Ferramenta NPM

Criando uma primeira aplicação - Hands-On

Depois de instalado o Node.js.

Verifique a instalação:

```
$node -v
```

Este comando retornará a versão do Node.js instalada



A Ferramenta NPM

Criando uma primeira aplicação - Hands-On

Crie uma pasta o projeto

Dentro dela crie um arquivo chamado olamundo.js com o seguinte código:

```
var http = require('http');
http.createServer(function(req,res) {
    res.writeHead(200, { 'Content-Type': 'text/plain; charset=utf-8' });
    res.write('Olá mundo!');
    res.end();
}).listen(3000);
console.log('Servidor iniciado!');
```




A Ferramenta NPM

Criando uma primeira aplicação - Hands-On

Para executar, acesse a pasta do projeto e execute o comando
\$ node olamundo.js

Para visualizar seu programa vá ao seu navegador e acesse localhost:3000



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Módulos: Criação e Gerenciamento



A Ferramenta NPM

Módulos

Um módulo é um conjunto de funções e objetos do JavaScript que podem ser utilizados por aplicativos externos.

Qualquer arquivo Node.js pode ser considerado um módulo caso suas funções e dados sejam feitos para programas externos.



A Ferramenta NPM

Funções

As funções são conjuntos de códigos que produzem algum resultado. Considerado elemento essencial em Node.js

Pode ter atributos e propriedades

Pode ser tratado como uma classe em JavaScript.

Módulos

O uso de módulos promove:

- Reuso de soluções
- Organização das funções
- Possibilidade de extensão sem "quebrar" o programa, ou seja, pode ser incluídas novas funções sem prejudicar o programa principal
- Compartimentalização
- Viabilidade de contribuir com a comunidade Node.js com soluções

Tipos de Módulos

Node.js inclui três tipos de módulos:

- Módulos principais:
 - Funcionalidades mínimas básicas do Node.js
 - Carregados automaticamente quando o Node.js é iniciado.
 - Precisa importar o módulo principal primeiro para usá-lo em seu aplicativo
- Módulos locais:
 - São módulos criados localmente no seu aplicativo
 - Pode empacotá-lo e distribuí-lo via NPM
- Módulos de terceiros:
 - Módulos desenvolvidos por terceiros e utilizado no seu projeto

Criando Módulos - Hands-On

- Crie uma pasta colors
 - mkdir colors
- Dentro da pasta crie um arquivo index.js
 - cd colors
 - nano index.js
- No arquivo index.js iremos colocar o código
 - ```
class Color {
 constructor(name, code) {
 this.name = name;
 this.code = code;
 }
}
```
  - ```
const allColors = [  
  new Color('brightred', '#E74C3C'),  
  new Color('soothingpurple', '#9B59B6'),  
  new Color('skyblue', '#5DADE2'),  
  new Color('leafygreen', '#48C9B0'),  
  new Color('sunkissedyellow', '#F4D03F'),  
  new Color('groovygray', '#D7DBDD'),  
];
```
 - ```
exports.getRandomColor = () => {
 return allColors[Math.floor(Math.random() * allColors.length)];
}
```
  - ```
exports.allColors = allColors;
```




A Ferramenta NPM

Criando Módulos - Hands-On

- Teste o módulo criado
 - \$node
 - colors = require('./index');
 - colors.getRandomColor();
 - .exit

Criando Módulos - Hands-On

- Incluir módulo em projeto
 - `$npm init -y`
 - `npm install --save ../color`
 - `nano package.json`
 - Visualizando o arquivo `package.json` você irá verificar se o módulo foi incluído
 - Crie o arquivo `index.js` no seu projeto com o código
 - `const colors = require('colors');`
 - `const chosenColor = colors.getRandomColor();`
 - `console.log(`Sua cor é ${chosenColor.name}`);`
 - Depois execute o script
 - `node index.js`



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Publicação de Módulo no NPM



A Ferramenta NPM

Publicação de Módulo no NPM

O processo de publicação consiste em um conjunto de atividades relacionadas a fim de disponibilizar para o consumidor o produto de software.

Dependendo da linguagem e tecnologia, este processo pode ter atividades distintas

A automação desta atividade tem sido perseguida e até mesmo obtida em algumas tecnologias.



A Ferramenta NPM

Publicação de Módulo no NPM

Um dos propósitos do NPM é facilitar este processo para produtos de software desenvolvidos em Node.js

Em conjunto com soluções como Grunt é possível automatizar o processo de publicação



A Ferramenta NPM

Publicando - Hands-On

Para esta prática sugerimos utilizar o projeto realizado na aula anterior, no entanto, pode utilizar qualquer outro caso prefira.

É importante destacar que é publicada toda pasta que contém o arquivo **package.json**



A Ferramenta NPM

Publicando - Hands-On

Antes de publicar, você precisa ter uma conta no npm.js.

Existem duas maneiras de criar um comando ou através do site npm.js.

O link para criar uma conta está aqui: <https://www.npmjs.com/signup>



A Ferramenta NPM

Publicando - Hands-On

Vá ao terminal e execute o comando:
`npm adduser`

Desta forma você configura o usuário e senha



A Ferramenta NPM

Publicando - Hands-On

Agora, finalmente, execute o comando para publicar o projeto
npm publish



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Linha de Comando e Comandos CLI



A Ferramenta NPM

CLI

Command Line Interface (Interface de Linha de Comando) é um programa que permite que os usuários digitem comandos de texto dando instruções a um computador para fazer funções específicas.

Nos anos 60, as pessoas tinham apenas teclados como um dispositivo de entrada e a tela do computador mostrava apenas informações em texto. Sendo assim usavam o CLI como a interface de usuário padrão.



A Ferramenta NPM

CLI

Com a evolução dos dispositivos, a interação com o computador passou a ser feita de outras formas.

Primeiramente o mouse, dando início a era do *point-and-click*

E nos dias atuais temos vários outros instrumentos, como touch-screen, eye-tracking, interação baseada em gestos, etc.

No entanto, apesar das evoluções dos mecanismos de IHC (Interação Humano Computador), a interface CLI se mostra bastante produtiva. São algumas vantagens:

- Menos Recursos: programas baseados em textos consomem bem poucos recursos de um computador
- Alta Precisão: Você pode usar um comando específico para alcançar os objetivos que quiser
- Tarefas Repetitivas Amigavelmente: Permite que você use um único comando para fazer a automação de uma tarefa aplicada a milhares de arquivos, por exemplo.
- Poder de Sobra: Sistemas operacionais contam com um sistema de proteção do sistema. Com isso, você não vai poder fazer certas tarefas que estão protegidas pelo próprio sistema. É por isso que, com o CLI, você ganha controle total sobre o seu sistema de escolha.

CLI

No link <https://docs.npmjs.com/cli-documentation/cli-commands> temos a lista de todos os comandos CLI para o NPM.

Destacamos aqui os mais essenciais:

- adduser
- install
- config
- publish

Nos exercícios práticos das aulas anteriores utilizamos a interface CLI para execução das nossas tarefas.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Configurações NPM



A Ferramenta NPM

Configurações NPM

Já falamos em aulas anteriores a importância do NPM dentro do processo de desenvolvimento Node.js

Da mesma forma, a sua correta configuração é essencial para todo o processo

As configurações podem ser detalhadas em: <https://docs.npmjs.com/cli-documentation/configuring-npm>



A Ferramenta NPM

Configurações NPM

Em linhas gerais as configurações NPM são:

- Gestão de dependências
- Execução de tarefas

Configurações NPM

- Para inclusão de módulos utilizamos o comando
 - `npm install [MODULO]`
 - Com isso, o arquivo de configuração `package.json` é escrito automaticamente pelo NPM informando o novo recurso de dependência do projeto
- Se você utilizar uma dependência apenas durante o desenvolvimento da aplicação, é recomendado que você a salve como uma "development dependency". Dessa forma, o pacote não será utilizado no ambiente de produção
 - `npm install [MODULO] --save-dev`

Configurações NPM

- Além de definir e buscar dependências, você também pode nomear scripts dentro do seu arquivo package.json e chamar o NPM para executá-lo a partir de um run-script command.
- Essa abordagem é comum para automatizar testes e tarefas ao longo do desenvolvimento
- Para isto é preciso incluir no package.json o seguinte bloco:
 - "scripts": {
 - ...
 - "lint": "[MODULO] [PASTA]"
 - ...
 - }
- Vale ressaltar que no exemplo acima o texto "lint" é um atalho atribuído para simplificar o uso. Pode ser considerado qualquer nome
- Por fim, o comando pode ser executado:
 - npm run-script lint

Configurações NPM

- 2 ● Em geral, as configurações ficam no arquivo package.json, sendo possível editá-lo e ajustar as configurações da forma que for mais conveniente.

```
• {
•   "name": "test-project",
•   "version": "1.0.0",
•   "description": "A Node.js project",
•   "main": "src/main.js",
•   "private": true,
•   "scripts": {
•     "dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js",
•     "start": "npm run dev",
•     "test": "npm run unit",
•     "lint": "eslint --ext .js,.vue src test/unit",
•     "build": "node build/build.js"
•   },
•   "dependencies": {
•     "vue": "^2.5.2"
•   },
•   "devDependencies": {
•     "autoprefixer": "^7.1.2",
•   },
•   "engines": {
•     "node": ">= 6.0.0",
•     "npm": ">= 3.0.0"
•   },
•   "browserslist": [
•     "> 1%",
•     "last 2 versions",
•     "not ie <= 8"
•   ]
• }
```



- 2** Caso queira, vc pode criar estes arquivos e códigos-fonte para deixar disponível para o aluno na Biblioteca da sala. Basta inserir os arquivos na pasta e no roteiro referenciá-los na seção de Material Complementar.

Andreza Vieira; 15/10/2020

Configurações NPM

- As propriedades mais comuns:
 - name: define o nome da aplicação ou pacote;
 - version: indica a versão atual;
 - description: é um resumo da sua aplicação/pacote;
 - main: define o ponto de entrada da aplicação;
 - private (true): previne a sua aplicação de ser publicada acidentalmente no npm;
 - scripts: define um conjunto de scripts Node para você executar;
 - dependencies: define uma lista de pacotes npm instalados como dependências;
 - devDependencies: define uma lista de pacotes npm instalados como dependências de desenvolvimento;
 - engines: define quais versões de Node este pacote/aplicação funciona;
 - browserslist: é usado para dizer quais browsers (e versões) você quer suportar;



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Scripts NPM para automação de tarefas



A Ferramenta NPM

Scripts NPM para automação de tarefas

No NPM é possível definir um conjunto de scripts que serão executados automaticamente.

O grande benefício é que tarefas rotineiras podem ser configuradas e poupar tempo, além de reduzir a possibilidade de erros



A Ferramenta NPM

Scripts NPM para automação de tarefas

Na seção "scripts" do arquivo package.json é onde podemos definir os comandos a serem executados

Tarefas que podem ser automatizadas:

- Implantação em desenvolvimento
- Teste unitário e de integração
- Implantação em produção
- Build

Scripts NPM para automação de tarefas

Utilizando o exemplo da aula anterior:

```
"scripts": {  
  "dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js",  
  "start": "npm run dev",  
  "unit": "jest --config test/unit/jest.conf.js --coverage",  
  "test": "npm run unit",  
  "lint": "eslint --ext .js,.vue src test/unit",  
  "build": "node build/build.js"  
}
```



A Ferramenta NPM

Scripts NPM para automação de tarefas

Basicamente definimos um apelido e o comando:

[ALIAS] : [COMANDO]

Nesta linha temos o comando de deploy em desenvolvimento:

```
"dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js",
```



A Ferramenta NPM

Scripts NPM para automação de tarefas

Neste comando utilizamos o npm para executar a linha de comando definida anteriormente e apelidada por dev

```
"start": "npm run dev",
```




A Ferramenta NPM

Scripts NPM para automação de tarefas

Aqui definimos o comando para execução de testes

```
"unit": "jest --config test/unit/jest.conf.js --coverage"
```



A Ferramenta NPM

Scripts NPM para automação de tarefas

Agora utilizamos o npm para executar o comando de teste definido anteriormente

```
"test": "npm run unit"
```



A Ferramenta NPM

Scripts NPM para automação de tarefas

Neste, está sendo utilizado o ESLint para checagem do código

```
"lint": "eslint --ext .js,.vue src test/unit"
```



A Ferramenta NPM

Scripts NPM para automação de tarefas

Por fim, o build da aplicação:

```
"build": "node build/build.js"
```



A Ferramenta NPM

Scripts NPM para automação de tarefas

Aqui apresentamos alguns exemplos de scripts que podem ser configurados no NPM.

Muitas outras opções podem ser consideradas.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 3 – Conclusão da Unidade



A Ferramenta NPM

Conclusão

Nesta unidade apresentamos o NPM e suas possibilidades

Construímos uma aplicação exemplo em Node.js e fizemos o processo de configuração e implantação da mesma usando recursos de NPM



A Ferramenta NPM

Conclusão

NPM se mostrou uma ferramenta bastante eficaz que pode não só auxiliar na garantia de qualidade do software, como também na produtividade da equipe

Com a automação de tarefas percebemos o quanto pode ser obtido em termos de produtividade.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Introdução à Unidade IV - Apache Kafka



Apache Kafka

Apache Kafka

- Introdução à Unidade IV - Apache Kafka
- Introdução ao Apache Kafka
- Introdução a Streaming de Dados
- Kafka em Cluster
- Kafka - Vantagens e Desvantagens
- Semântica
- Estudo de Caso
- Estudo de Caso - Netflix
- Estudo de Caso - Uber
- Criando fluxo no Kafka - Hands-On
- Conclusão da Unidade
- Conclusão da Disciplina



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Introdução ao Apache Kafka



Apache Kafka

Introdução

Apache Kafka é uma solução de código aberto para processamento de dados em streaming desenvolvido pela Fundação de Software Apache.

Kafka pode conectar sistemas externos e prover um fluxo de dados.



Apache Kafka

Introdução

O nome Kafka foi inspirado em um escritor chamado Franz Kafka

Isso faz sentido, uma vez que o Kafka é um serviço de mensageria e streaming de dados

Originalmente criado pelo LinkedIn, depois se tornou open-source

Introdução

Pode compor o Ecossistema Hadoop como elemento para processamento em tempo real

Não foi projetado especificamente para o Hadoop. O ecossistema Hadoop é apenas um dos possíveis consumidores.

Ecossistema Hadoop





Apache Kafka

Introdução

É um sistema de mensageria no modelo publicação / assinatura

Isto significa que é muito fácil adicionar um grande número de consumidores sem afetar o desempenho e sem tempo de inatividade.

Isso ocorre porque o Kafka não rastreia quais mensagens no tópico foram consumidas pelos consumidores. Ele simplesmente mantém todas as mensagens no tópico dentro de um período configurável. É de responsabilidade do consumidor fazer o rastreamento por meio de offset.



Apache Kafka

Introdução

Com isso, oferece durabilidade, escalabilidade e suporte à tolerância a falhas.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Introdução a Streaming de Dados



Apache Kafka

Streaming de Dados

São dados que:

- Chegam continuamente
- Em grande velocidade
- Com intervalos variados
- Em um fluxo ilimitado

Exemplos:

- Monitoração
- Dados de sensores



Apache Kafka

Streaming de Dados

Os SGBDs tradicionais não são preparados para armazenar os dados de forma contínua e rápida, além de suportar consultas contínuas sobre estes dados

Os SGBDs tradicionais utilizam lock na escrita de registros. Quando se tem várias transações em paralelo, este lock pode fazer com que uma transação "impeça" outra de ser realizada, gerando filas e acúmulo de requisições

Quanto maior o volume de entrada, mais gargalo na operação de escrita

Streaming de Dados

Os Streaming de Dados são:

- Fluxos contínuos
- Não existe controle de ordem de chegada para ser processado
- Depois de processado, geralmente são descartados
- As consultas precisam ser feitas em tempo real

Estes dados têm a característica de que se ele ficar muito obsoleto ele perde seu valor, por exemplo: um sensor meteorológico para previsão de tornado. Se a velocidade do vento capturada for processada um mês depois, o tornado já aconteceu e esta informação já não serve para mais nada.



Apache Kafka

Streaming de Dados

Para esse perfil de produção e consumo de dados novas tecnologias devem prover suporte

As tecnologias Big Data justamente atuam nos casos de dados produzidos em grande volume, velocidade e variedade

Apache Kafka faz parte deste conjunto de tecnologías Big Data



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



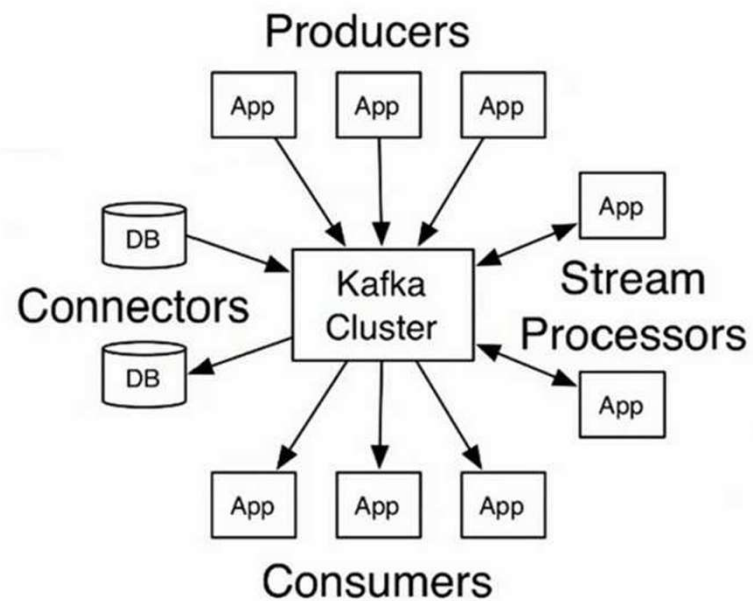
Unidade 4 – Kafka em Cluster

Apache Kafka

Kafka em Cluster

O Kafka permite ser utilizado tanto em um único nó com em cluster

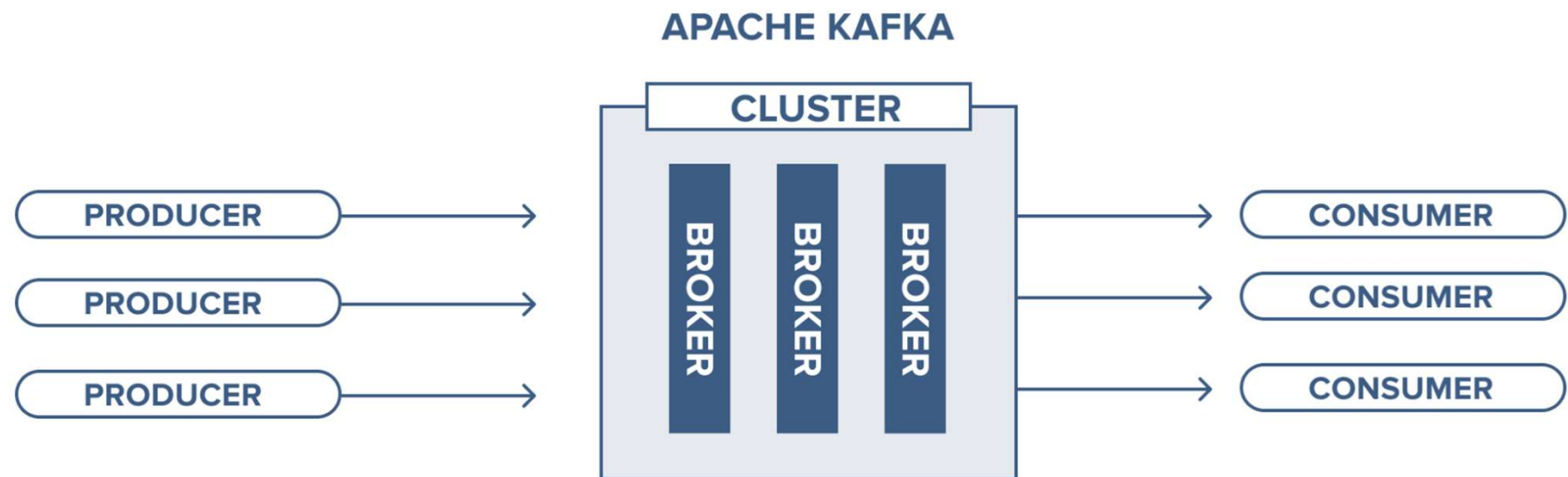
Com o cluster, aumenta-se a escala e o potencial da ferramenta.



Apache Kafka

Kafka em Cluster

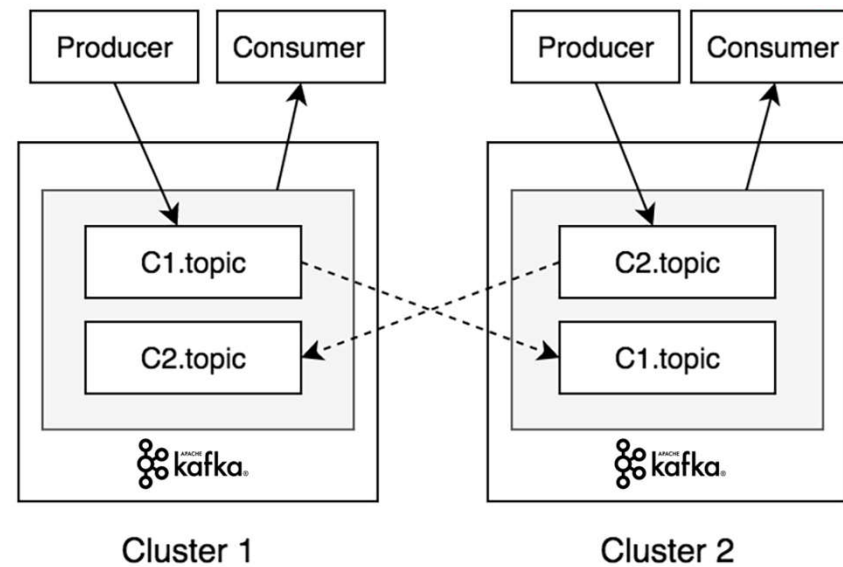
O cluster é a composição de várias máquinas gerenciadas



Apache Kafka

Kafka em Cluster

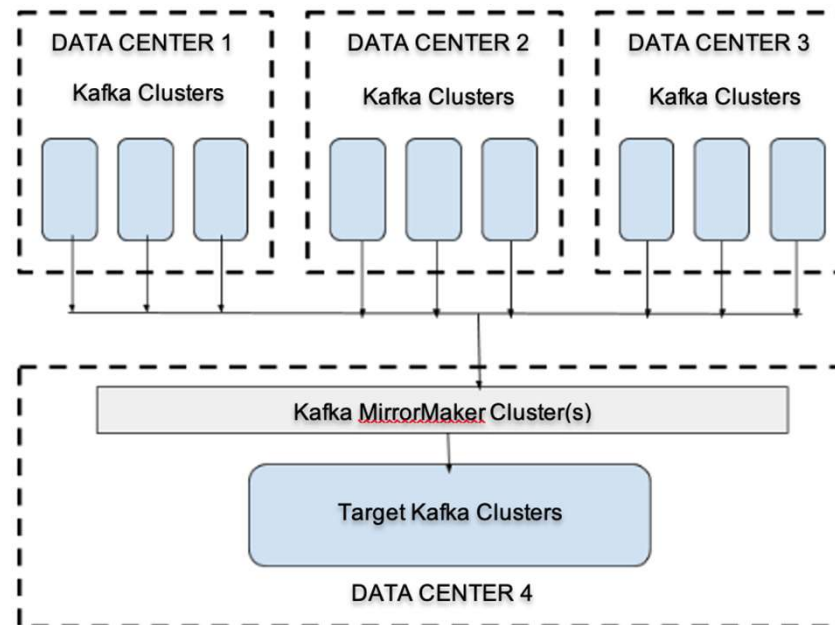
Pode-se ter vários clusters trabalhando em cooperação



Apache Kafka

Mirror Maker

O MirrorMaker usa um consumidor Kafka de alto nível para buscar os dados do cluster de origem e, em seguida, alimenta esses dados em um produtor Kafka para despejá-los no cluster de destino.





Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Kafka - Vantagens e Desvantagens

Vantagens

- Alto rendimento: suporta o processamento de milhares de mensagens por segundo
- Baixa latência: lida com mensagens de latência muito baixa em intervalo de milissegundos
- Escalabilidade: pode ser expandido on-the-fly, sem incorrer em tempo de inatividade, adicionando novos nós.
- Distribuído: suporta arquitetura distribuída
- Alta simultaneidade: capaz de lidar com milhares de mensagens por segundo e também em condições de baixa latência com alta taxa de transferência.
- Consumidor: pode ser integrado com uma variedade de consumidores.
- Apache Kafka é open source

Desvantagens

- Não possui um conjunto completo de ferramentas de gerenciamento e monitoramento.
- O Kafka Broker usa certas chamadas do sistema para entregar mensagens ao consumidor, mas se a mensagem precisar de alguns ajustes, isso reduz significativamente o desempenho do Kafka
- O Kafka corresponde apenas ao nome do tópico exato e não suporta seleção de tópico curinga
- As APIs necessárias a outras linguagens de programação são mantidas por diferentes pessoas e empresas
- O Kafka Broker e sua abordagem são frequentemente atribuídos a serem realmente simples e descomplicados por natureza. Por isso, outros componentes são usados para atender a determinados requisitos, como Zookeeper (coordenação de estado) e MirrorMaker (comunicação entre clusters)
- Conforme o tamanho da mensagem aumenta, os agentes e consumidores começam a compactar essas mensagens e, por causa disso, a memória do nó é usada lentamente quando a descompactação e a compactação ocorre e quando os dados fluem no pipeline.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Semântica



Apache Kafka

Semântica

Em um sistema de mensagens de publicação/assinatura distribuída, os computadores que compõem o sistema sempre podem falhar independentemente um do outro.

Dependendo da ação que o produtor executa para lidar com essa falha, você pode obter semânticas diferentes

Semântica pelo menos uma vez

Se o produtor recebe uma confirmação (ack) do broker Kafka e acks = all, significa que a mensagem foi gravada exatamente uma vez no tópico Kafka.

No entanto, se um ack do produtor atingir o tempo limite ou receber um erro, ele poderá tentar enviar a mensagem novamente, assumindo que a mensagem não foi gravada no tópico Kafka.

Se o broker falhou pouco antes de enviar a confirmação, mas após a mensagem ter sido gravada com sucesso no tópico Kafka, essa nova tentativa levará a mensagem a ser gravada duas vezes e, portanto, entregue mais de uma vez ao consumidor final. E todo mundo adora um doador alegre, mas essa abordagem pode levar a trabalho duplicado e resultados incorretos



Apache Kafka

Semântica no máximo uma vez

Se o produtor não tentar novamente quando uma confirmação exceder o tempo limite ou retornar um erro, a mensagem poderá acabar não sendo gravada no tópico Kafka e, portanto, não entregue ao consumidor.

Na maioria dos casos, será, mas, a fim de evitar a possibilidade de duplicação, aceitamos que algumas vezes as mensagens não sejam enviadas.

Semântica exatamente uma vez

Mesmo que um produtor tente enviar uma mensagem novamente, isso leva a mensagem a ser entregue exatamente uma vez ao consumidor final.

A semântica exata uma vez é a garantia mais desejável, mas também mal compreendida. Isso ocorre porque requer uma cooperação entre o próprio sistema de mensagens e o aplicativo que produz e consome as mensagens.

Por exemplo, se, após consumir uma mensagem com êxito, você retroceder seu consumidor Kafka para um deslocamento anterior, receberá todas as mensagens desse deslocamento para a mais recente, novamente. Isso mostra por que o sistema de mensagens e o aplicativo cliente devem cooperar para que a semântica exata exista.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Estudo de Caso



Apache Kafka

Estudo de Caso

Apache Kafka tem sido amplamente adotado por empresas, fazendo parte de um dos pilares da arquitetura de serviços.

Pois é uma plataforma de serviço de mensagens pub-sub (publisher-subscriber) que permite enviar mensagens de um extremo ao outro enquanto lida com grande parte dos dados.

Também funciona bem para os tipos de serviços de mensagens online e offline.

Apache Kafka

Estudo de Caso

Empresas que precisam lidar com Big Data, ou seja, dados em grande volume produzidos em alta velocidade e de vários formatos diferentes, adotaram o Kafka como opção para manipular streaming de dados.

LinkedIn

NETFLIX

Spotify

The New York Times

Uber

twitter



Rabobank

Unyleya
EDUCACIONAL

Apache Kafka

Estudo de Caso



O LinkedIn processa bem mais de um trilhão de mensagens por dia. Esse número é surpreendente e o LinkedIn registrou um aumento de mais de 1200x nos últimos anos.

O LinkedIn usa clusters diferentes para aplicativos diferentes para evitar o colapso da falha de um aplicativo, o que levaria a prejudicar os outros aplicativos no cluster. Os clusters do Broker Kafka no LinkedIn os ajudam a diferenciar e listar certos usuários para permitir uma maior largura de banda e garantir a experiência perfeita do usuário.

Apache Kafka

Estudo de Caso



O Spotify, que é a maior plataforma de música do mundo, possui um enorme banco de dados para manter 200 milhões de usuários e 40 milhões de faixas pagas.

O Apache Kafka foi usado para notificar os usuários que recomendam as listas de reprodução, colocando anúncios direcionados entre muitos outros recursos importantes.



Apache Kafka

Estudo de Caso

The New York Times

O New York Times, um dos mais antigos meios de comunicação.

Sempre que um artigo é publicado no NYT, ele precisa ser disponibilizado em todos os tipos de plataformas e entregue aos seus assinantes em pouco tempo. Para resolver esse problema, o NYT desenvolveu um projeto chamado de Pipeline de publicação, no qual o Apache Kafka é usado para remover problemas baseados em API por meio de sua arquitetura baseada em log.



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

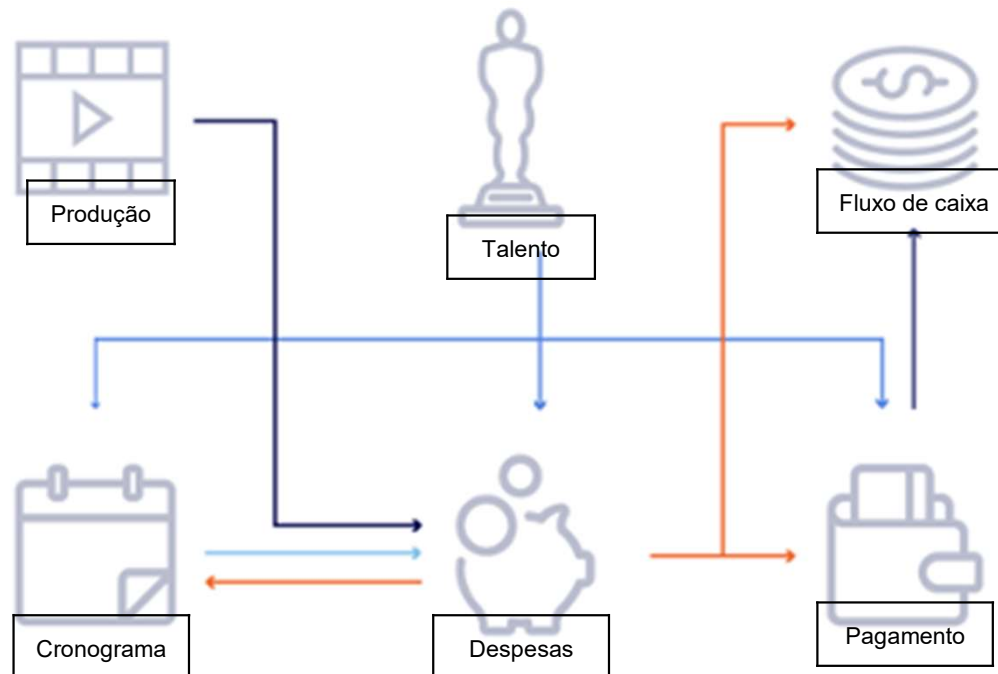
in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Estudo de Caso - Netflix

Estudo de Caso - Netflix

A Netflix adota a governança distribuída e incentiva uma abordagem de aplicativos orientada por microsserviços.



**Estudo de caso retirado de (Confluent, 2020).*

Estudo de Caso - Netflix

Geralmente, em uma arquitetura de microsserviço, uma porcentagem de falha é aceitável. No entanto, uma falha em qualquer uma das chamadas de microsserviço para Engenharia de Finanças de Conteúdo poderia resultar em dados perdidos em milhões de dólares.

A mudança para um modelo de publicação / assinatura permite que todos os serviços publiquem suas alterações como eventos em um barramento de mensagens, que pode ser consumido por outro serviço de interesse que precisa ajustar seu estado do mundo.



Apache Kafka

Estudo de Caso - Netflix

O enriquecimento pode ser executado como um microserviço separado, que é responsável por materializar conjuntos de dados.



Estudo de Caso - Netflix

Uma vez que as mensagens são disponibilizadas em qualquer ordem, os consumidores podem fazer a leitura no momento que for mais adequado.

Desta forma, evita a criação de filas e bloqueios devido a falta de sincronização em algum dos processos

Essa dinâmica garante velocidade e robustez no sistema





Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Estudo de Caso - Uber



Apache Kafka

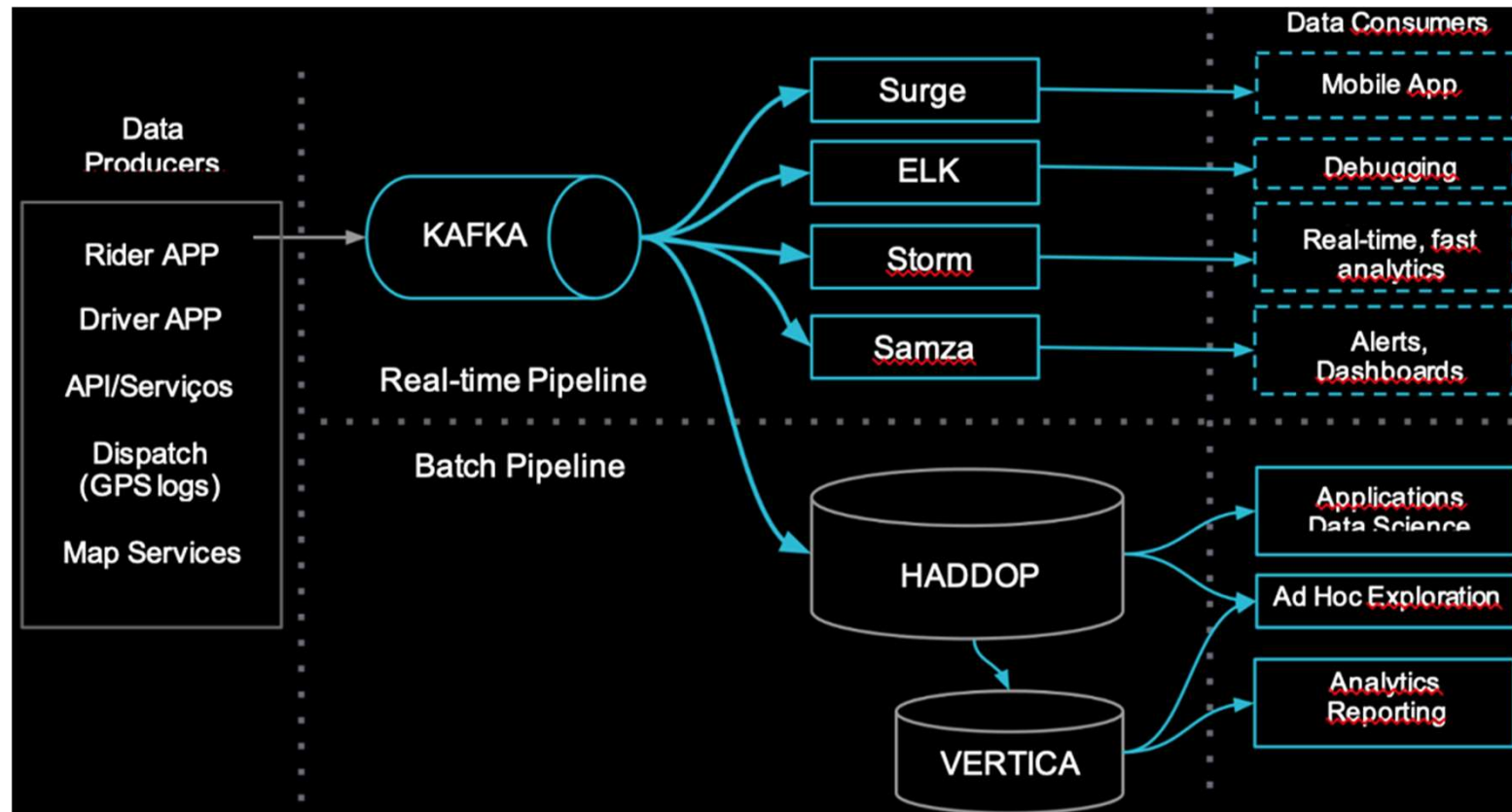
Estudo de Caso - Uber

No Uber, o Apache Kafka é usado como um barramento de mensagens para conectar diferentes partes do ecossistema.

É coletado logs do sistema e de aplicativos, bem como dados de eventos dos aplicativos de piloto e motorista. Em seguida, é disponibilizado esses dados para uma variedade de consumidores a jusante via Kafka.

**Estudo de caso retirado de (SOMAN; et al., 2016).*

Estudo de Caso - Uber

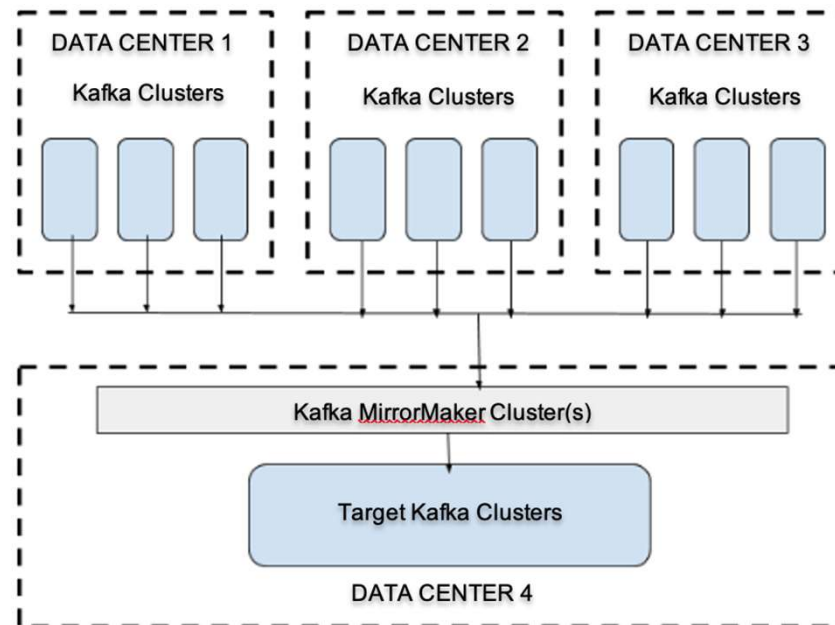


Apache Kafka

Estudo de Caso - Uber

Os dados no Kafka alimentam os pipelines em tempo real e os pipelines em lote.

O MirrorMaker garante a unidade dos clusters





Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Criando fluxo no Kafka - Hands-On



Apache Kafka

Hands-On

Primeiro passo é instalar o kafka ou utilizar um docker. No nosso caso, optaremos pelo docker:

Utilizando um dos tanto containers disponibilizados pela comunidade. A Confluent (empresa fundada pelo criadores do Kafka e que oferece serviços de Kafka-as-a-service) disponibilizou um tutorial mostrando como utilizar o container mantido por eles:

<https://docs.confluent.io/current/installation/docker/docs/quickstart.html#getting-started-with-docker-compose>



Apache Kafka

Hands-On

Criando um tópico no kafka:

```
docker-compose exec kafka \  
kafka-topics --create --topic meu-topico-legal --partitions 1 --replication-factor 1 --if-not-exists --  
zookeeper localhost:32181
```



Apache Kafka

Hands-On

Produzindo mensagens

```
docker-compose exec kafka \  
bash -c "seq 100 | kafka-console-producer --request-required-acks 1 --broker-list localhost:29092 -  
-topic meu-topico-legal && echo 'Produced 100 messages.'"
```




Apache Kafka

Hands-On

Consumindo mensagens

```
docker-compose exec kafka \  
kafka-console-consumer --bootstrap-server localhost:29092 --topic meu-topico-legal --from-  
beginning --max-messages 100
```



Apache Kafka

Hands-On

Agora é programar!

Pode ser usado Java, Python, Node.js ou qualquer linguagem que tiver mais afinidade



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)



Unidade 4 – Conclusão da Unidade



Apache Kafka

Conclusão da Unidade

Nesta unidade abordamos o Kafka, desde a introdução do conceito de serviço de streaming até sua implementação

É importante cada etapa que passamos para entendermos por completo o funcionamento deste serviço



Apache Kafka

Conclusão da Unidade

O Kafka é uma implementação de um serviço de mensageria para tratar fluxos de dados

Problemas como condição de corrida ou deadlocks precisam ser observados, apesar do Kafka optar por locks mais flexíveis para evitar isso

A implantação pode ser instalado na máquina o utilizando container



Apache Kafka

Conclusão da Unidade

A operação do Kafka pode ser monitorado pelo Zookeeper

Apesar de não ser uma solução nascida na infraestrutura Hadoop, ele tem grande aderência no ecossistema

Por fim, vimos o quanto o Kafka pode ser útil e o quanto tem sido absorvido pelas grandes empresas, sendo um componente importante na arquitetura tecnológica da empresa



Obrigado!

✉ prof.guylerme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

🌐 [linkedin.com/in/guylerme-figueiredo-b8497939](https://www.linkedin.com/in/guylerme-figueiredo-b8497939)



Unidade 4 – Conclusão da Disciplina



Apache Kafka

Conclusão da Disciplina

Neste curso percorremos por vários elementos e estruturas fundamentais para o desenvolvimento Web.

Percebemos que para diferentes perfis de desenvolvimento (Frontend e Backend) temos necessidades e ferramental diferentes.

Os servidores que viabilizam tanto o desenvolvimento quanto a execução da aplicação podem ser providos por diversos provedores de nuvem, tais como Azure, Google Cloud e AWS.

Conclusão da Disciplina

A Cloud Computing tem sido a grande opção que as empresas têm escolhido, pois os benefícios e vantagens obtidos com a nuvem são vários

Desde elasticidade e expansibilidade da infraestrutura à velocidade no provimento de recursos, o uso de nuvem tem aumentado consideravelmente nos últimos anos.

Sobre a virtualização, agora tem se utilizado containers para garantir uma solução que seja de fácil migração e expansão de recursos. O uso de docker, seja em soluções on cloud, seja soluções on premisses, tem compartimentalizado soluções a fim de garantir que a mesma seja executada sobre qualquer ambiente



Apache Kafka

Conclusão da Disciplina

O desenvolvimento utilizando recursos da nuvem tem direcionado as melhores práticas do mercado.

A fim de potencializar a produtividade, ferramentas para automação de tarefas tem composto este conjunto de aceleradores

Neste ínterim, ferramentas como NPM tem se mostrado bastante produtiva, eficaz e simples de utilizar

O NPM automatiza o deploy de soluções Node.js, que é uma tecnologia que permitiu o uso de JavaScript no desenvolvimento de backend



Apache Kafka

Conclusão da Disciplina

Por fim, para finalizar o conjunto de elementos para a infraestrutura de desenvolvimento web, temos o Apache Kafka

Este serviço de mensageria se colocou como umas das ferramentas mais fundamentais para streaming de dados

Com o Kafka é possível enviar e consumir dados contínuos em alta velocidade

Várias empresas têm adotado ele como importante elemento na arquitetura tecnológica



Obrigado!

✉ prof.guylherme@gmail.com

🐦 [@guylermef](https://twitter.com/guylermef)

in [linkedin.com/in/guylherme-figueiredo-b8497939](https://www.linkedin.com/in/guylherme-figueiredo-b8497939)