Tema 1

Princípios de Big Data

A área de Tecnologia da Informação (TI) é uma das que mais cresce atualmente e precisa de especialistas com sólida formação. Se você pretende atuar nessa área, é muito importante conhecer os conceitos e as tecnologias de Big Data. Esse é o grande diferencial do profissional de TI.

Tempo total de leitura
106 min.

Créditos

F
Professor (a)
FERNANDO DURIER

Introdução

Você já ouviu falar sobre o termo Big Data? Ele é usado para referenciar aplicações que envolvem grandes volumes de dados. Mais do que isso: é um conjunto de tecnologias que gerenciam aplicações, as quais, além do grande volume de dados, trabalham com dados que podem ser gerados com alta velocidade, de diversas fontes e diferentes formatos.

Ao longo deste conteúdo, você entenderá os conceitos relativos à tecnologia de Big Data e como ela se relaciona com outras tecnologias, como computação na nuvem e Internet das Coisas (IoT). Além disso, você aprenderá a executar um exemplo prático em Python. Preparado?

Preparação

Antes de iniciar o conteúdo, você vai precisar criar uma conta no Gmail para acessar o Google Colab e executar o exemplo prático na linguagem Python.

Objetivos

Ao final desta aula, você será capaz de:

- Reconhecer os conceitos e as aplicações de Big Data.
- Definir os conceitos de Internet das Coisas e computação distribuída.
- Analisar plataformas em nuvem e aplicações de streaming de Big Data.
- Empregar a linguagem Python em uma aplicação no mercado financeiro.

01. Aplicações de Big Data

Fundamentos da tecnologia de Big Data

Contextualização

Desde a popularização da Internet, com a criação da World Wide Web na década de 1990, cada vez mais, utilizamos aplicações e serviços que armazenam nossos dados e os aproveitam para fazer previsões sobre nosso comportamento. Não é à toa que muitas empresas nos fazem ofertas que, de fato, coincidem com nossos interesses.

Esse crescimento do volume de dados e de toda a complexidade que os envolve demandou tratamento especializado de armazenamento, gerenciamento e análise, popularmente conhecido como **Big Data.**

Os dados precisam ser tratados por um ciclo de vida, para que possamos extrair informações úteis deles e, depois, transformá-las em conhecimento. Como consequência desse processo, áreas como Ciência de Dados e Aprendizado de

Máquina – mais conhecidas como **Data Science** e **Machine Learning** – cresceram muito nos últimos anos.

Comentário

Devido ao potencial de valor que as aplicações de Big Data podem gerar, tanto empresas quanto agências governamentais têm investido nessa área por meio do desenvolvimento de soluções que capturem dados com mais qualidade para facilitar as etapas posteriores de armazenamento, gerenciamento e análise.

Dados que vêm de fontes distintas permitem fazer um mapeamento muito detalhado do comportamento das pessoas. Isso também desperta discussões nos campos ético e legal.

No Brasil, há disposições constitucionais sobre a inviolabilidade do sigilo de dados e das comunicações. Por exemplo, a Lei Geral de Proteção de Dados (LGPD) – Lei nº 13.709/2018 – visa proteger os cidadãos quanto ao uso indevido de seus dados.

Mas ainda há muito a ser feito, o que acaba gerando novas oportunidades de pesquisa e desenvolvimento de projetos envolvendo segurança e privacidade.

Arquitetura básica de Big Data

A complexidade que envolve o gerenciamento de todas as características do Big Data exige que tratemos sua arquitetura de modo específico. Mais uma vez, isso o diferencia dos sistemas de banco de dados tradicionais, que teriam dificuldade em lidar com operações de dados em sistemas heterogêneos.

Esses sistemas são chamados de data lake (lago de dados). Basicamente, é um enorme repositório de arquivos e objetos de dados. Portanto, as soluções da arquitetura de Big Data precisam ser eficientes, para que possam produzir resultados em tempos de resposta aceitáveis.

São componentes da arquitetura de Big Data:

Fontes de dados (data sources)

Além das fontes de dados tradicionais, os sistemas de Big Data podem ser alimentados por meio de dados que estão na nuvem e produzidos por sistemas de Internet das Coisas (IoT). Em muitos casos, esse processo, chamado de aquisição de dados, ocorre em tempo real.

Armazenamento de dados (data storage)

Os dados precisam ser armazenados de modo eficiente para otimizar seu acesso e sua segurança. Esse armazenamento pode ser feito de diversas formas – na nuvem, em bancos de dados estruturados ou em bancos de dados não estruturados que tenham:

- Escalabilidade – capacidade de crescer com consistência.
- Disponibilidade – prontos para acesso sempre que forem demandados.
- Segurança – mecanismos que garantam privacidade e restrição de acesso.
- Padronização – armazenamento seguindo um padrão que facilite sua recuperação posteriormente.

Processamento em lote (batch processing)

Armazenamento dos dados em lotes, para, então, realizar seu processamento. Isso é feito para lidar com grandes volumes de dados, quando não é viável fazer o processamento dos dados em fluxos.

Ingestão de mensagens (message ingestion)

Agrupamento dos dados, trazendo-os para um sistema de processamento, no qual podem ser armazenados, analisados e acessados.

Processamento de fluxo (stream processing)

Processamento de dados à medida que são produzidos ou recebidos. Essa situação ocorre com frequência em processos de eventos produzidos por sensores, atividades do usuário em um site e negociações financeiras, que têm em comum o fato de os dados serem criados como uma série de eventos de fluxo contínuo.

Armazenamento de dados analíticos (analytical data store)

Processo de armazenamento de dados de negócios, mercado e clientes para posterior análise. As aplicações desses dados são chamadas de business intelligence (BI) – inteligência de negócios. Os bancos de dados analíticos são otimizados para consultas rápidas.

Análise e relatórios (analysis and reporting)

Os relatórios são uma organização dos dados com o objetivo de fazer resumos informativos e monitorar o desempenho de diferentes áreas da empresa. A análise consiste em explorar dados e relatórios para extrair informações que agreguem valor e que possam ser usadas para compreender melhor os negócios e aperfeiçoar seu desempenho.

Atividade

Questão 1

Gerenciar um projeto de Big Data é uma tarefa complexa devido as suas características. Além de lidar com grandes volumes de dados, também é necessário tratar diversas questões da arquitetura que envolve o projeto. Considerando essa complexidade, assinale a alternativa correta a respeito da arquitetura de um projeto de Big Data:

A — Entre os aspectos que devem ser considerados em um projeto de Big Data está a necessidade de garantir a privacidade dos dados, para que apenas as pessoas autorizadas tenham acesso a eles.

B — Um dos fatores que precisam ser tratados na arquitetura de um projeto de Big Data é a padronização dos dados, para que possam ser armazenados em tabelas.

C — As fontes de dados constituem a base da arquitetura dos projetos de Big Data, pois garantem que os dados não sejam corrompidos.

D — Como os projetos de Big Data podem crescer rapidamente, é fundamental tratar aspectos relacionados às fontes de dados.

E — A complexidade da arquitetura de um projeto de Big Data está relacionada a dois fatores – volume e diversidade dos dados.

Parabéns! A alternativa A está correta.

Os projetos de Big Data são complexos, pois possuem muitas variáveis, como a diversidade e o volume dos dados, e a velocidade com que são gerados. Além disso, é necessário considerar aspectos como as diversas tecnologias envolvidas e a segurança dos dados.

Os 5 Vs do Big Data

Introdução aos 5Vs

Uma forma de definir a complexidade do Big Data é descrevendo suas características.

Em 2001, o analista Doug Laney, da empresa META (atual Gartner Group), apresentou um relatório de pesquisa no qual tratou desafios e oportunidades trazidos pelo aumento de dados com um modelo 3 Vs: **Volume**, **Velocidade** e **Variedade** (LANEY, 2001). Esse modelo foi usado durante muitos anos para descrever a tecnologia de Big Data.

Em 2011, um relatório publicado pela International Data Corporation (IDC) associou Big Data ao conjunto de tecnologias e arquiteturas projetadas para extrair valor de grandes volumes e variedades de dados, permitindo captura, descoberta e análise de alta velocidade (GANTZ; REINSEL, 2011). Assim, mais um V foi incluído no modelo anterior: **Valor**.

Atualmente, a forma mais comum de encontrarmos uma definição sobre Big Data contempla outro V: **Veracidade** (RUSSOM, 2011).

Resumindo

Como você viu, o conceito de Big Data sofreu uma evolução ao longo do tempo, porque é um ecossistema complexo que envolve aspectos tecnológicos de software e hardware, além de questões econômicas, sociais e éticas que ainda estão sendo compreendidas até hoje.

Agora, vamos analisar mais detalhadamente os 5 Vs que compõem essa tecnologia.

Volume

Esta característica está relacionada com a escala da geração e coleta de massas de dados.

A percepção de grandes volumes de dados está relacionada com a tecnologia disponível em determinado momento. Portanto, precisamos conhecer como o volume de dados é medido.

Basicamente, temos:

- **byte (B)** – unidade de informação digital, também chamada de octeto, que consiste em uma sequência de 8 bits (binary digits);
- **byte kilobyte (KB)** – 1 KB = 2^{10} B= 1024 bytes;
- **megabyte (MB)** – 1 M = 2^{10} KB = 2^{20} B;
- **gigabyte (GB)** – 1 GB = 2^{10} MB= $2^{1=20}$ KB = 2^{30} B;
- **terabyte (TB)** – 1 TB = 2^{10} GB;
- **petabyte (PB)** – 1 PB = 2^{10} TB;
- **exabyte (EB)** – 1 EB = 2^{10} PB;
- **zettabyte (ZB)** – 1 ZB = 2^{10} EB;
- **yottabyte (YB)** – 1 YB = 2^{10} ZB.

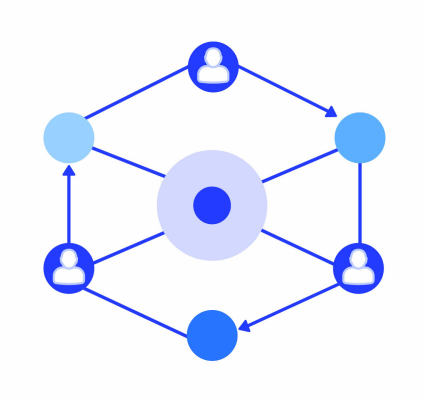
Quando nos referimos ao volume de uma aplicação de Big Data, normalmente, estamos tratando de petabytes (PB) de dados.

Velocidade

Esta característica corresponde a dois aspectos:

- Velocidade da geração de dados.
- Rapidez com que os dados são gerados e processados.

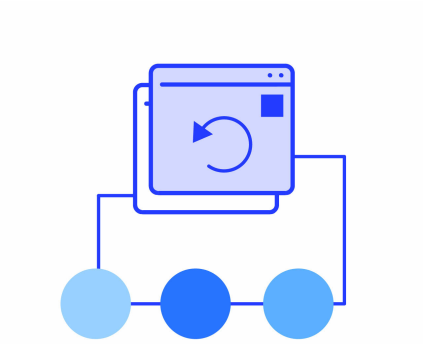
Basicamente, temos o problema clássico de computação: **produtor x consumidor**.



O **consumidor** representa o papel do analista que precisa fazer consultas rapidamente, mas pode sofrer limitações do tempo de resposta do **produtor**. Ou seja, o sistema pode possuir um ritmo mais lento para disponibilizar os dados para consulta.

Um projeto de Big Data precisa equilibrar os tempos de consumo e geração de dados.

Variedade



Um projeto de Big Data pode ter vários tipos de dados, como áudio, vídeo, página da Web e texto, e tabelas de bancos de dados tradicionais. Esses tipos de dados podem ser classificados como:

- Dados estuturados;
- Dados não estuturados;
- Dados semiestruturados.

Vamos entendê-los com mais detalhes:

Dados estruturados

São armazenados de maneira organizada e fáceis de processar e analisar. Normalmente, são dados numéricos ou texto que podem ser armazenados em um banco de dados relacional e manipulados usando a linguagem SQL (Structured Query Language).

Dados não estruturados

Não possuem uma estrutura predefinida. Como exemplo, temos as imagens e os arquivos de áudio. São armazenados em um banco de dados não relacional denominado NoSQL (Not Only SQL).

Dados semiestruturados

Mesclam as duas formas de dados anteriores. Como exemplo, temos arquivos nos formatos XML (eXtended Markup Language) e JSON (Java Script Object Notation).

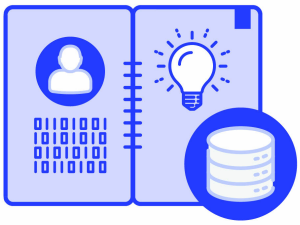
Veracidade

Esta característica está relacionada à qualidade dos dados. Isso é essencial, especialmente do ponto de vista de suporte para a tomada de decisão, pois é a veracidade dos dados que nos dá o grau de confiança para fazermos o que precisamos com base na integridade e precisão dos dados.

Um projeto de Big Data precisa utilizar técnicas que façam limpeza dos dados e garantam a qualidade deles, para que possam ser consumidos pelo processo de

análise.

Valor



Esta é a principal característica de um projeto de Big Data, que justifica todo o seu trabalho: extrair valor dos dados.

Os dados são a matéria-prima do negócio. Por isso, precisam passar por diversas etapas de tratamento e gerenciamento até que possam ser consumidos pelo processo de análise.

Podemos aplicar técnicas de Ciência de Dados e machine learning para obter informações e conhecimentos que vão direcionar ações para as diversas frentes de um negócio.

Atividade

Questão 1

O termo Big Data é bastante popular atualmente. Um dos motivos dessa popularidade é a propagação do uso das aplicações que funcionam na Internet. Nesse sentido, assinale a alternativa correta a respeito das aplicações de Big Data.

A | Uma das dificuldades atuais associadas aos projetos de Big Data é o uso para prestação de serviços públicos, uma vez que são caros, e seu benefício não é facilmente quantificável.

B

Dispositivos eletrônicos podem ser conectados diretamente à Internet, transmitindo dados sem a necessidade de garantir a qualidade deles, pois esta será tratada pela aplicação de Big Data.

C

A tecnologia de Big Data pode ser usada para monitorar os sinais vitais de pacientes, capazes de ser transmitidos via Internet.

D

Apesar de ainda não serem aplicados na área de entretenimento, os projetos de Big Data têm grande potencial de uso para proporcionar experiências específicas de acordo com o perfil do usuário.

E

Uma possível aplicação de Big Data é na prestação de serviços de utilidade pública, mas os benefícios só poderão ser percebidos se houver total integração entre todos os sistemas dos diversos setores que compõem o Estado.

Parabéns! A alternativa C está correta.

Muitos benefícios podem ser obtidos pela utilização de projetos de Big Data para prestação de serviços públicos, entretenimento, segurança e aplicações na área da saúde, entre tantas outras possibilidades. O potencial desses benefícios aumentará sempre que for possível usar diversas fontes de dados, pois essa diversidade permite identificar padrões complexos que dificilmente seriam detectados de outra forma.

02. IoT e computação distribuída

Aspectos da computação distribuída

Componentes de IoT

A tecnologia de IoT consiste na coexistência colaborativa de quatro componentes, que são:

Objetos físicos (coisas)

São os componentes eletrônicos e sensores responsáveis pela coleta de dados e aplicação de ações. Exemplo: termostatos usados para controlar a temperatura de um ambiente.

Protocolos de comunicação

Viabilizam a troca de dados via Internet entre os objetos físicos e outros sistemas. Um exemplo é o protocolo HTTP.

Computação

Faz o gerenciamento do ciclo de vida dos dados, desde a coleta e o armazenamento até o processamento. Por exemplo, podemos usar o Arduino para coletar os dados dos sensores.

Serviços

Provêm autenticação e gerenciamento de dispositivos, além de oferecer infraestrutura. Um exemplo são os serviços na nuvem.

Princípios da computação distribuída

Para tratar da integração dos componentes de IoT, utilizamos a computação distribuída: um modelo mais adequado para gerenciar essas unidades não centralizadas pelo compartilhamento de responsabilidades e riscos. Para alcançar esse objetivo, a computação distribuída segue alguns princípios-chave. São eles:

Distribuição de armazenamento e processamento de dados entre os nós da rede

Isso permite otimizar a eficiência dos processos.

Transferência de dados e análises conforme necessidade

Diferentes níveis de processamentos podem ser realizados pelos nós da rede. Isso significa que o custo global de processamento e análise dos dados é minimizado, pois os nós menos onerosos realizam pré-processamentos que reduzem o custo do processamento final dos nós mais caros da rede.

Tolerância a falhas

Muito provavelmente, há intermitência da operação dos nós das redes. Portanto, a política de computação distribuída já deve estar preparada para reorganizar o fluxo de dados na rede, de modo que possam ser roteados de outra maneira, e que a rede continue em operação.

Otimização dos recursos computacionais da rede

Especialmente no caso da IoT, em que os dispositivos possuem uma restrição de recursos de memória e processamento, a computação distribuída trabalha com níveis de consumo baixo de energia.

Arquitetura básica de computação distribuída de IoT

Em um projeto de Big Data, precisamos coletar grande quantidade de dados, armazená-los, processá-los e analisá-los para detectar padrões relevantes que demandem algum tipo de ação, quando necessário.

Agora, quando aplicamos Big Data para IoT, precisamos tratar a complexidade das características de seus componentes, ou seja, utilizar uma solução que dê suporte ao alto volume de dados e consiga se comunicar com os dispositivos.

> **A computação distribuída é a solução mais adequada para distribuir a computação aos nós da IoT.**

Uma arquitetura básica de computação distribuída de IoT é composta pelas seguintes camadas:

Computação em nuvem (cloud computing)

É a tecnologia que permite o uso de recursos computacionais de software e hardware remotamente. Por exemplo, quando utilizamos repositórios na Internet para armazenar dados ou servidores de aplicação, estamos trabalhando com computação em nuvem. Essa camada é responsável por:

- Processamento de Big Data;
- Lógica de negócios;
- Armazenamento de dados (data warehousing).

Computação em névoa (fog computing)

É uma extensão da camada de nuvem que aproxima servidores aos dispositivos de IoT. Esses servidores podem colaborar entre si por meio de trocas de dados e realizar processamentos que vão otimizar a operação do sistema como um todo. Um exemplo desse tipo de computação ocorre quando utilizamos um sensor de temperatura conectado a uma aplicação na nuvem. Entre as principais características da computação em névoa estão:

- Redes locais de computadores;
- Análise e redução de dados;
- Controle de respostas;
- Virtualização e padronização.

Computação de borda (edge computing)

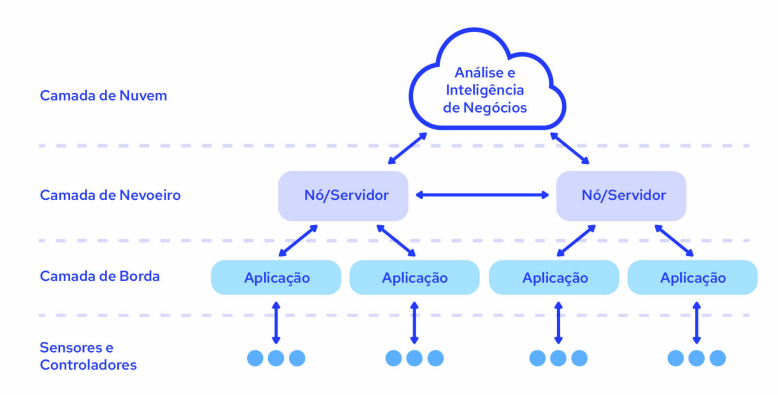
Relaciona-se diretamente com os sensores e controladores que ficam na borda da arquitetura. Assim, os dados podem ser armazenados e processados para, então, ser enviados à camada de névoa. Um exemplo desse tipo de computação

ocorre quando utilizamos dispositivos para conectar uma rede local (LAN) interna à Internet.

Sensores e controladores

São os dispositivos responsáveis por gerar os dados e, quando acionados, realizar ações. Por exemplo, em um sistema de irrigação, há sensores que monitoram a umidade do solo e controladores que o irrigam até obter o nível adequado de umidade.

Veja a arquitetura básica de computação distribuída aplicada para IoT:



Arquitetura básica de IoT.

Ao longo dessa arquitetura, observamos a mudança da velocidade do fluxo de dados. Na parte inferior, vemos os dados gerados pelos sensores a uma velocidade superior à medida que vamos avançando até a camada de nuvem.

Atividade

Questão 1

As aplicações de Internet das Coisas (IoT) estão cada vez mais presentes em nosso dia a dia. Algumas características dos projetos de IoT são a produção de grande volume de dados e o uso de computação distribuída. Por isso, devem ser tratados

como projetos de Big Data. Em relação às tecnologias de IoT e à computação distribuída, assinale a alternativa correta.

A

A camada de computação em nuvem é responsável por tratar diretamente da qualidade dos dados produzidos pelos dispositivos de IoT e transmiti-los para os servidores de aplicações de Big Data.

B

Um dos aspectos da arquitetura de computação distribuída é utilizar camadas responsáveis por atividades específicas, como é o caso da camada de computação em névoa.

C

As camadas da arquitetura de computação distribuída são equivalentes quanto ao tratamento dos dados e diferenciam-se apenas pela tecnologia que utilizam.

D

Uma das vantagens da computação distribuída é padronizar a tecnologia utilizada em um projeto de IoT.

E

Projetos de IoT são considerados complexos devido à grande quantidade de tecnologias neles envolvidas. Por isso, a arquitetura de computação distribuída deve ser aplicada apenas com duas camadas – de nuvem e de dispositivos.

Parabéns! A alternativa B está correta.

A arquitetura de computação distribuída aplicada para projetos de IoT envolve camadas especializadas em tratar determinados aspectos da gestão de dados, para que eles possam trafegar na rede com segurança e qualidade. As camadas da arquitetura de computação distribuída para IoT são: computação em nuvem,

computação em névoa, computação de borda e dispositivos de sensores e controladores.

Tecnologias de IoT

Protocolos de comunicação

Os sistemas de IoT precisam de protocolos que permitam que os dispositivos eletrônicos possam se comunicar com outros nós da rede. Um nó pode ser um dispositivo eletrônico, um computador ou servidor.

Alguns dos principais protocolos de comunicação de IoT são:

HTTP

O HTTP (Hyper Text Transport Protocol) é o Protocolo de Transporte de Hipertexto e o modelo cliente-servidor mais importante utilizado na Web. Nele, a comunicação entre um cliente e um servidor ocorre por meio de uma mensagem do tipo solicitação x resposta. A dinâmica básica da comunicação segue os seguintes passos:

- O cliente envia uma mensagem de solicitação HTTP.
- O servidor retorna uma mensagem de resposta, contendo o recurso solicitado, caso a solicitação tenha sido aceita.

MQTT

O MQTT (Message Queuing Telemetry Transport) é o protocolo de Transporte de Filas de Mensagem de Telemetria, lançado em 1999. Sua primeira aplicação foi para monitorar sensores em oleodutos. Como é um protocolo aberto, sua comunicação é baseada em um servidor que faz a publicação e o recebimento de dados com o padrão de mensagens publicação x assinatura. Esse padrão, chamado de broker, faz o trabalho intermediário de recebimento das mensagens dos nós da rede e as envia aos nós de destino. O MQTT é executado em um protocolo de transporte TCP (Transmission Control Protocol), o que garante a confiabilidade do tráfego de dados.

CoAP

O CoAP (Constrained Application Protocol) é o Protocolo de Aplicação Restrita. Utiliza a arquitetura REST (Representational State Transfer ou Transferência de Estado Representacional) e oferece suporte ao paradigma de solicitação x resposta, exatamente como ocorre no caso REST/HTTP. Além disso, é executado em um protocolo de transporte UDP (User Datagram Protocol).

XMPP-IOT

O XMPP-IOT (Extensible Messaging and Presence Protocol for the IoT) é o Protocolo de Mensagem Extensível e de presença para a IoT. Também é um protocolo aberto que foi projetado para trocas de mensagens instantâneas. Ele usa a arquitetura cliente-servidor, rodando sobre TCP. Sua comunicação é baseada em XML e possui extensões que possibilitam o uso do modelo de **publicação x assinatura.**

Plataformas para IoT

Quando trabalhamos com um sistema de IoT, é necessário desenvolver programas para que os dispositivos possam operar de forma adequada e enviar dados para a rede. Para isso, precisamos de plataformas de desenvolvimento que nos oferecem recursos de software e hardware, os quais nos ajudam a trabalhar com a interoperabilidade e a conectividade dos dispositivos à rede.

Vamos conhecer a seguir, algumas das principais plataformas de desenvolvimento para dispositivos de IoT.

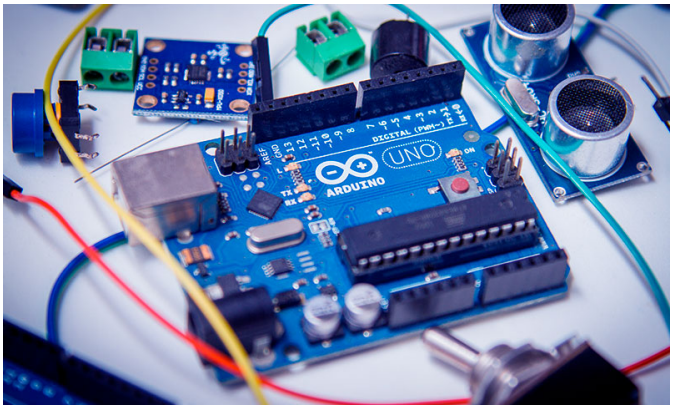
Arduino

Foi criado no Ivrea Interaction Design Institute, em 2002. Oferece um ecossistema de hardware, linguagem de programação, bibliotecas e dispositivos que nos ajuda a desenvolver projetos capazes de ter diversas aplicações.

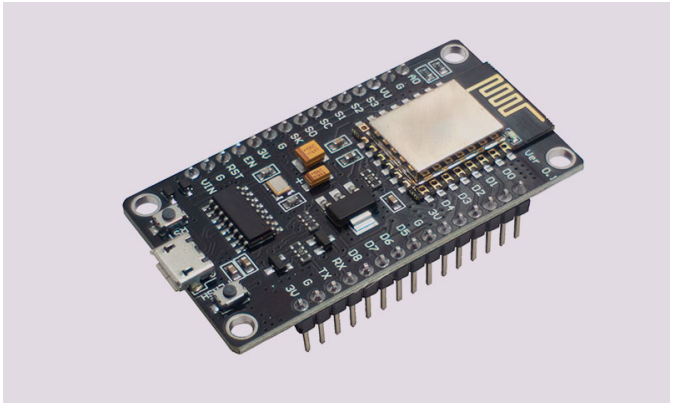
Uma das principais características do Arduino é que todas as suas placas e softwares

são de código aberto.

Essa característica ajudou a popularizar essa plataforma, que possui uma comunidade de desenvolvedores engajada em divulgar projetos e conhecimentos em fóruns on-line.



NodeMCU



É um dos principais kits eletrônicos de código aberto para desenvolvimento de aplicações de IoT.

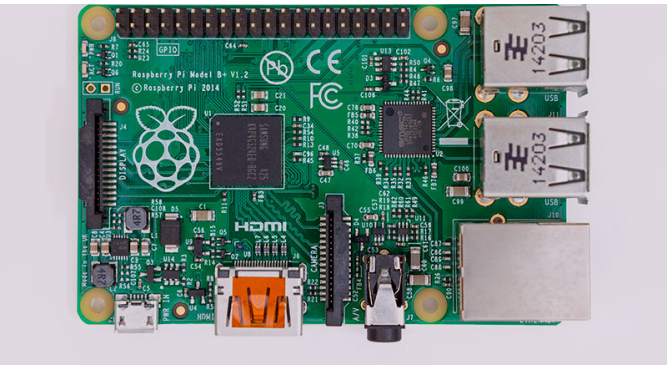
É baseado na família do microcontrolador ESP8266 e possui recursos que facilitam trabalhar com dispositivos conectados à Internet para monitoramento e controle.

Raspberry PI

É uma plataforma de computação de placa única.

Seu propósito inicial foi a aplicação no ensino de Ciência da Computação, evoluindo para funções mais amplas.

Possui uma interface de baixo nível de controle auto-operado por portas de entrada-saída chamado de GPIO (General Purpose Input-Output) e usa Linux como seu sistema operacional padrão.



Atividade

Questão 1

Os projetos de Internet das Coisas (IoT) têm sido utilizados com sucesso em diversas áreas. De forma simplificada, os sensores geram dados que são enviados para servidores de aplicação por meio da tecnologia de Internet. Nesse sentido, assinale a alternativa correta sobre os protocolos para aplicações de IoT.

A | Projetos de IoT são exemplos típicos de aplicações de Big Data e, portanto, devem ser desenvolvidos com o protocolo UDP, como é o caso do XMPP-IOT.

B | HTTP é o protocolo padrão para aplicações de IoT, utilizado por todos os demais protocolos como uma camada intermediária que garante a qualidade dos dados.

C | Dispositivos de IoT são caracterizados por possuírem muitos recursos de memória e processamento para tratar do grande volume e da diversidade dos dados. Por isso, utilizam protocolos como o HTTP e MQTT.

D | MQTT é um protocolo de IoT que usa uma estrutura de comunicação, na qual os dispositivos publicam seus dados. Estes, por sua vez, são

consumidos por um broker que os transmite para determinadas aplicações.

E

Alguns protocolos usados pelos projetos de IoT são construídos com tecnologias proprietárias que são mais adequadas para tratar a diversidade de dados, como é o caso do CoAP.

Parabéns! A alternativa D está correta.

O MQTT é um protocolo aberto de IoT baseado no padrão publicação x assinatura. Na prática, isso significa que os dispositivos publicam seus dados, e as aplicações que vão consumi-los fazem isso por meio de uma formalização (assinatura). Esse processo de recebimento e transmissão de dados é intermediado por uma aplicação chamada de broker.

03. Plataformas em nuvem para aplicações de Big Data

Modelos e plataformas de serviços na nuvem

Modelos de prestação de serviços na nuvem

Computação em nuvem (Cloud Computing) é o termo usado para se referir a uma categoria de serviços de computação sob demanda disponíveis na Internet.

Além de reduzir os custos necessários para oferecer os serviços, a tecnologia de computação em nuvem também aumenta a confiabilidade do sistema. Por isso, é cada vez mais comum encontrarmos aplicações que integram as diversas tecnologias e que oferecem os meios para que programas e dispositivos possam se comunicar na Internet.

Os modelos mais comuns de prestação de serviços na nuvem são:

SaaS (Software as a Service)

Ocorre quando uma aplicação é oferecida via Internet, e seu preço é dado de acordo com as necessidades de uso da parte contratante, como a quantidade de licenças, por exemplo. Esse tipo de serviço é bastante interessante para o cliente, pois vai pagar apenas as funcionalidades do sistema que lhe serão úteis. Além disso, não é necessário que o usuário se preocupe com instalação, ambiente para execução, manutenção e atualizações, pois tudo isso fica sob a responsabilidade do prestador de serviço.

PaaS (Platform as a Service)

Disponibiliza o sistema operacional e um ambiente de desenvolvimento na nuvem para o contratante. Dessa forma, ele pode criar seus próprios programas com acesso a ferramentas adequadas, bibliotecas e bancos de dados.

IaaS (Infrastructure as a Service)

Disponibiliza servidores de armazenamento e serviços de firewall e segurança da rede para os contratantes.

DaaS (Desktop as a Service)

Oferece computadores (desktops) virtuais aos usuários finais pela Internet, que são licenciados com uma assinatura por usuário. A forma como os dados podem ser persistidos nas máquinas virtuais também é tratada por esses serviços.

Assim, os computadores podem ser:

- Persistentes – os usuários podem personalizar e salvar uma área de trabalho para que mantenha a aparência sempre que fizer logon na máquina.
- Não persistentes – os desktops são apagados cada vez que o usuário se desconecta, pois são apenas um meio de acessar os serviços de nuvem compartilhados.

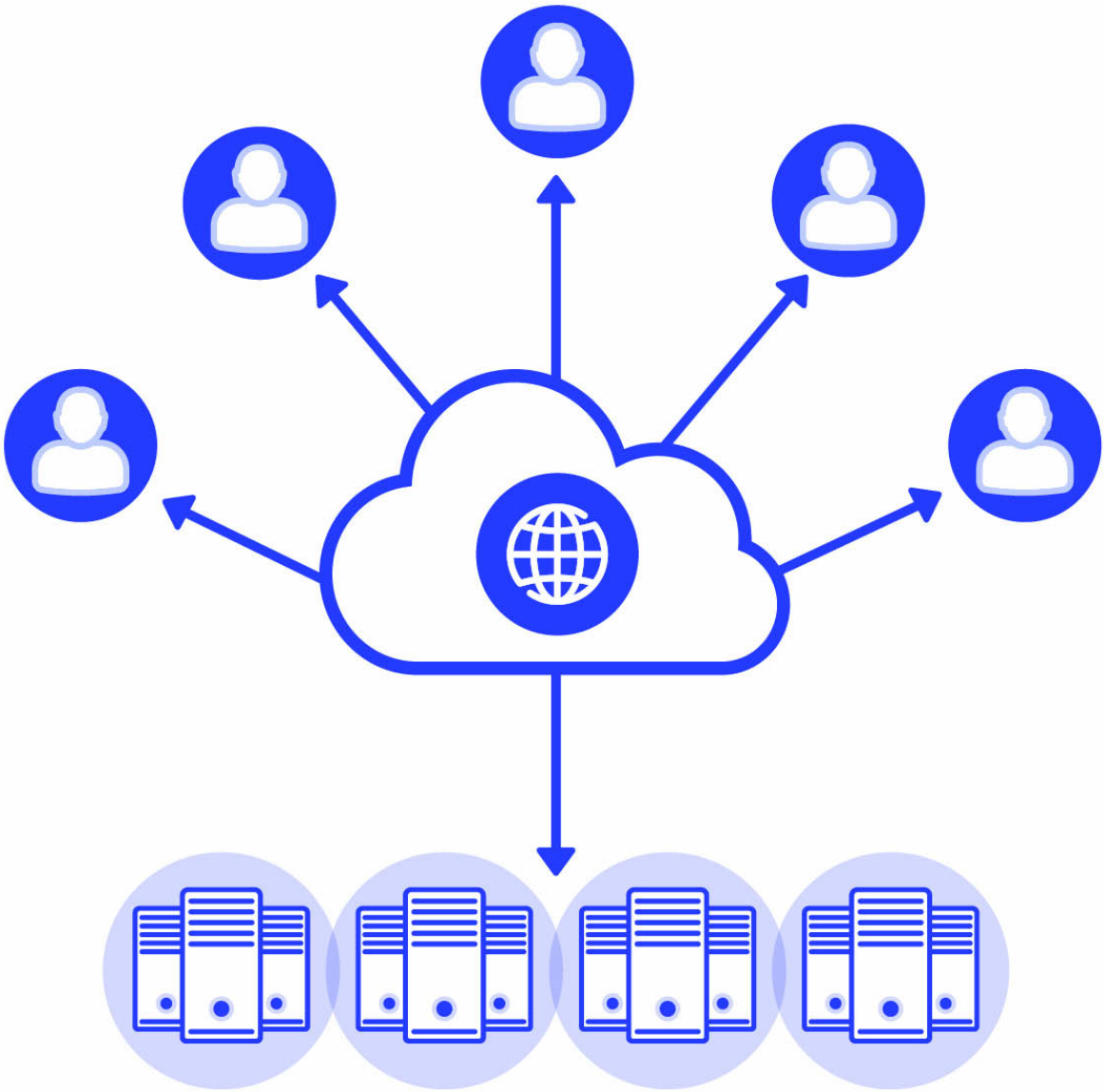
XaaS (Everything as a Service)

É um termo geral usado para se referir à entrega de qualquer coisa como um serviço. Entre os exemplos de XaaS, podemos citar:

- Modelos gerais – como Software como Serviço (SaaS), Plataforma como Serviço (PaaS) e Infraestrutura como Serviço (IaaS).
- Modelos especializados – como comunicação como um serviço (CaaS), monitoramento como um serviço (MaaS), recuperação de desastres como um serviço (DRaaS) e redes como um serviço (NaaS).

Tipos de nuvem

Existem três diferentes maneiras de implantar uma infraestrutura de nuvem e de disponibilizar programas que possuem vantagens e desvantagens associadas ao contexto em que serão utilizados. Os três tipos de nuvens são:



Nuvem pública

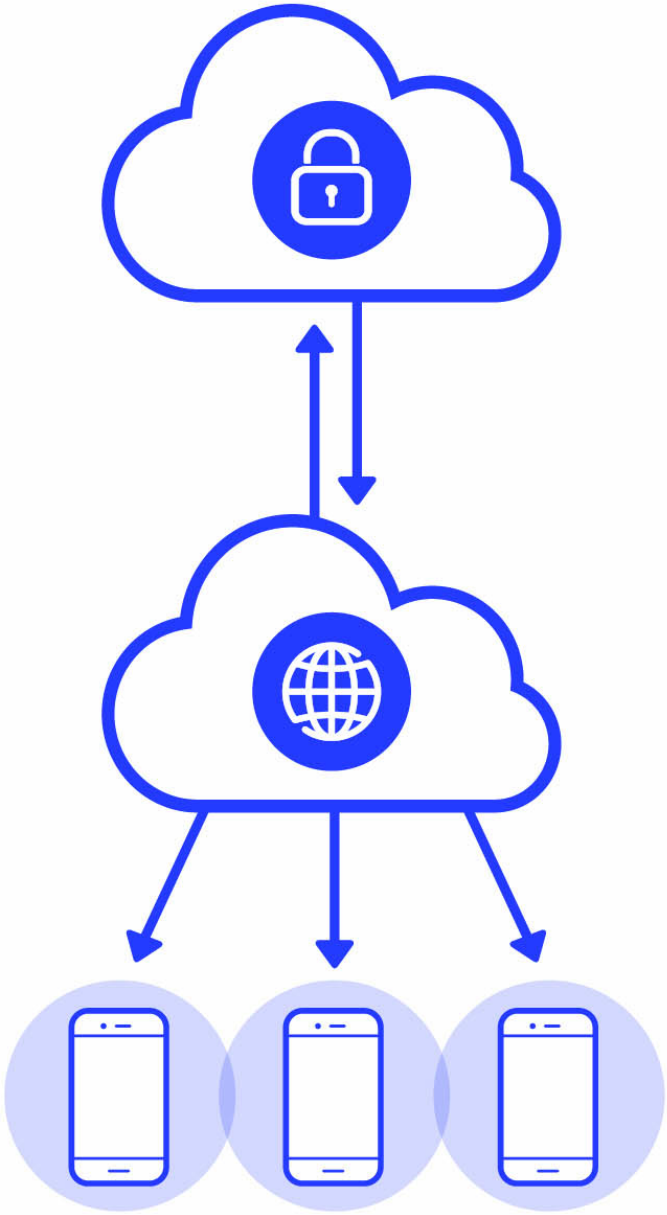
Essa configuração é adequada para as empresas que ainda estão na etapa de crescimento de sua infraestrutura, em que a demanda por serviços é instável, ou seja, muito baixa em alguns momentos e muito alta em outros.

Assim, as empresas podem pagar apenas pelo que estão usando e, se for necessário, podem ajustar sua infraestrutura na nuvem com base na demanda, sem a necessidade de fazer um investimento inicial em hardware. Isso garante economia de dinheiro e de tempo de configuração.

Nuvem privada

Todos os serviços são executados por servidores dedicados que dão ao contratante total controle sobre a gestão dos programas e da segurança da rede. Na prática, o usuário contratante pode monitorar e otimizar o desempenho da execução dos serviços de acordo com suas necessidades..

O principal valor desse tipo de nuvem é a privacidade que oferece. Essa característica é especialmente interessante para empresas que trabalham com dados confidenciais e que querem isolamento da Internet aberta.



Nuvem híbrida

Este tipo de nuvem combina aspectos das implementações de nuvem pública e privada. Por exemplo, os dados confidenciais permanecem na nuvem privada devido à segurança que ela oferece. Já as operações que não usam dados confidenciais são feitas na nuvem pública, na qual as empresas contratantes podem dimensionar a infraestrutura para atender às suas demandas com custos reduzidos. No caso de operações de Big Data, as nuvens híbridas podem ser utilizadas para atuar com

dados não confidenciais na nuvem pública e manter os dados confidenciais protegidos na nuvem privada.

Atividade

Questão 1

Os serviços de nuvem oferecem diversas facilidades para projetos de Big Data. Eles são uma combinação de tecnologias que envolvem hardware e software a partir da Internet. Nesse sentido, assinale a alternativa correta sobre os modelos de serviços na nuvem.

A | Os serviços de nuvem são utilizados apenas para transmissão e recepção de dados, ficando sob a responsabilidade do contratante o armazenamento e processamento dos dados.

B | Quando contratamos um modelo PAAS, esperamos que sejam disponibilizadas aplicações para gerenciar os dados.

C | Os modelos de serviço de nuvem só podem ser usados para projetos de Big Data voltados a aplicações de Internet das Coisas.

D | Apesar da redução de custos para montar uma infraestrutura, os serviços de nuvem têm como desvantagem a dificuldade para expandir o uso de novas tecnologias em um projeto de Big Data.

E | Os serviços de nuvem de software tratam de diversos aspectos, como rede, servidores, virtualização, sistema operacional, dados e aplicações.

Parabéns! A alternativa E está correta.

Os serviços de nuvem são muito úteis para projetos de Big Data, pois flexibilizam o uso de tecnologias e a adequação do tamanho da infraestrutura para atender às demandas dos clientes. Existem vários modelos, como o SaaS (software como serviço), PaaS (plataforma como serviço) e IaaS (infraestrutura como serviço).

Tecnologias de plataformas na nuvem

Plataformas de Big Data na nuvem

Uma plataforma de Big Data na nuvem é um conjunto de tecnologias de software e hardware que permite ao usuário contratante gerenciar projetos de Big Data a partir de aplicações para desenvolvimento, implantação e operação de programas, além do controle de uma infraestrutura voltada para Big Data.

> Do ponto de vista econômico, essa estratégia é bastante interessante, pois o contratante não precisa se preocupar com vários detalhes operacionais que, nesse modelo, ficam sob a responsabilidade do prestador de serviços.

Ao longo dos anos, a demanda por soluções de Big Data tem aumentado, e a oferta de serviços acompanhou esse processo. As soluções das plataformas de Big Data tratam de:

Gestão de dados

Disponibilização de servidores de banco de dados para gerenciamento de Big Data.

Análise de dados

Inteligência de negócios por meio de programas utilitários para tratamento e extração de dados de Big Data.

Ferramentas de desenvolvimento

Oferta de ambientes de desenvolvimento de programas para fazer análises personalizadas que podem se integrar com outros sistemas.

Além de todos esses aspectos, a plataforma oferece os serviços de segurança e proteção aos dados por meio do controle de acesso. É um modelo muito interessante para quem trabalha com Big Data devido à redução de complexidade da gestão de tantos detalhes e da possibilidade de se concentrar no negócio em si.

Profissionais da área

Toda a facilidade oferecida por uma plataforma de Big Data ajuda os profissionais a se concentrarem na excelência de seus trabalhos, em especial, porque eles vão atuar com conjuntos de dados de grande volume.

Alguns perfis de profissionais que trabalham com essas plataformas são:

Engenheiros de dados

Profissionais que fazem toda a gestão do fluxo dos dados, incluindo coleta, agregação, limpeza e estruturação, para que possam ser utilizados em análises.

Cientistas de dados

Profissionais que utilizam a plataforma para estudar padrões e descobrir relacionamentos em grandes conjuntos de dados.

Normalmente, existem dois tipos distintos de análise em **Ciência de Dados**, que incluem:

- **Análise exploratória e visualização de dados** – análise dos dados por meio de técnicas estatísticas.
- **Análise de algoritmos de aprendizado de máquina** – análise dos dados com o objetivo de encontrar associações não triviais que podem ser úteis para desenvolver estratégias de negócios, como aumentar engajamento de clientes e potencializar vendas.

Exemplos de plataformas na nuvem

Vamos conhecer, agora, algumas das principais plataformas na nuvem, mas, antes disso, precisamos entender um conceito muito importante de Big Data: o **data lake**.

Trata-se de um repositório centralizado, no qual é possível armazenar grandes volumes de dados estruturados e não estruturados. É um recurso muito útil para armazenar os dados sem precisar estruturá-los.

Além disso, existe a possibilidade de executar diferentes tipos de análises de Big Data com painéis que facilitam as visualizações e funcionam como suporte para a tomada de decisão.

Comentário

Esse recurso é essencial nas plataformas de Big Data, pois as organizações utilizam os dados como base para realizar análises e desenvolver estratégias que as ajudem a

potencializar seus negócios.

Cada plataforma oferece uma tecnologia de data lake. As principais plataformas de Big Data para nuvem são:

Amazon AWS

É a plataforma cuja primeira oferta como serviço ocorreu em 2006. Seu modelo é usado como referência por outras plataformas de armazenamento e computação em nuvem.

Microsoft Azure

É a plataforma de nuvem da Microsoft que foi lançada em 2010. Oferece ferramentas e serviços projetados para permitir a organizações que trabalham com grandes conjuntos de dados realizar todas as suas operações na nuvem.

Google Cloud Platform

É a plataforma de nuvem do Google. Utiliza a mesma tecnologia dos serviços de Big Data proprietários do Google, como YouTube e pesquisa Google.

Oracle Cloud

É a plataforma de banco de dados da Oracle na nuvem. Seu serviço de nuvem inclui armazenamento flexível e escalável junto com os serviços de análise e processamento de dados.

IBM Cloud

É a plataforma de nuvem da IBM. Oferece várias soluções de data lake com o objetivo de atender aos diferentes perfis de necessidades de seus clientes.

Atividade

Questão 1

A tecnologia de computação na nuvem é um importante recurso para projetos de Big Data. Para atender a essa demanda de mercado, grandes empresas da Internet oferecem plataformas com soluções de hardware e software. Sobre as plataformas de Big Data na nuvem, assinale a alternativa correta.

A | Ao utilizar plataformas na nuvem, os contratantes podem fazer análises personalizadas com o uso de programas especializados que são úteis para dar suporte à área de negócios de uma organização.

B | As plataformas de nuvem são protocolos de comunicação que fazem a intermediação entre as aplicações responsáveis pela coleta de dados até o processamento analítico, permitindo a elaboração de relatórios sofisticados.

C | A Amazon é uma das gigantes da Internet que disponibiliza uma plataforma de nuvem chamada de MQTT, a qual pode ser utilizada para projetos de Internet das Coisas.

D | Um dos perfis dos profissionais que trabalham com plataformas de Big

D Data na nuvem e o de engenheiro de dados, o qual se caracteriza por desenvolver aplicações de aprendizado de máquina.

E As plataformas de Big Data na nuvem são utilizadas para desenvolver exclusivamente aplicações voltadas para gestão do ciclo de vida dos dados, caracterizadas, principalmente, pelo uso da tecnologia de data lake.

Parabéns! A alternativa A está correta.

Os principais fornecedores de plataformas de Big Data na nuvem são a Amazon, Microsoft, Google, Oracle e IBM. Suas plataformas cobrem aspectos de hardware e software, em que o contratante usa um data lake e, posteriormente, pode empregar ferramentas analíticas para detecção de padrões que apoiam o desenvolvimento de estratégias de negócios.

Processamento e streaming de dados

O que é streaming de dados?

É o processo de transmissão de um fluxo contínuo de dados, o qual, por sua vez, é formado por diversos elementos de dados ordenados no tempo.

Exemplo

Em uma transmissão de dados de uma gravação de vídeo, as imagens são séries de dados que seguem uma ordem cronológica.

Assim, os dados representam que algo ocorreu – o chamado evento. Ou seja, houve uma mudança de estado sobre um processo que pode fornecer informações úteis. Por isso, muitas organizações investem para obter, processar e analisar esses dados. Podemos encontrar exemplos típicos de fluxos de dados nas seguintes situações:.

- Dados de sensores embarcados em equipamentos.
- Arquivos de logs de atividades de navegadores da Web.
- Logs de transações financeiras.
- Monitores de saúde pessoais.
- Sistemas de segurança patrimonial.

Atualmente, o fluxo e o processamento de dados aumentaram sua importância devido ao crescimento da IoT: o fluxo de dados dessas aplicações é muito grande e precisa de um tratamento específico.

Os sistemas de IoT podem ter vários sensores para monitorar diferentes etapas de um processo. Esses sensores geram um fluxo de dados transmitido de forma contínua para uma infraestrutura de processamento que monitora qualquer atividade inesperada em tempo real ou salva os dados para analisar padrões mais difíceis de detectar depois.

Características e desafios do processamento de fluxos de dados

As aplicações de Big Data sempre precisam considerar a complexidade de seus contextos de uso. Isso também vale para os dados de streaming de sensores,

navegadores da Web e outros sistemas de monitoramento, cujas características precisam ser tratadas de modo diferente em relação aos dados históricos tradicionais..

Devido aos aspectos que envolvem o processamento de fluxo de dados, podemos destacar algumas características. São elas:

Sensibilidade ao tempo

Independentemente de onde sejam aplicados, os elementos em um fluxo de dados estão associados a uma localização de tempo a partir de data e hora. Essa característica é usada junto com o contexto de aplicação para medir o valor do dado. Por exemplo, os dados de um sistema de monitoramento de saúde de pacientes que indiquem uma mudança grave dos níveis vitais devem ser analisados e tratados dentro de um curtíssimo período, para preservar a integridade da saúde dos pacientes e, assim, permanecer relevantes.

Continuidade

Especialmente para processos de tempo real, os fluxos de dados são contínuos e acontecem sempre que um evento é disparado ou quando ocorre uma mudança de estado no sistema. Portanto, o sistema de processamento deve estar preparado para ser acionado sempre que for requisitado. Por exemplo, um sistema pode acompanhar em tempo real a velocidade de uma broca de perfuração de rochas.

Heterogeneidade

Os dados de fluxo podem vir de diferentes fontes com distintos formatos e estar geograficamente distantes. Como já vimos, uma das características de Big Data é a variedade que abrange todas essas situações: formatos, fontes de dados e localização geográfica. Por exemplo, podemos receber dados estruturados em tabelas ou em arquivos de texto semiestruturados.

Imperfeição

Muitos fatores podem influenciar que os elementos de um fluxo de dados sejam prejudicados por perda e corrupção. Devido à variedade das fontes e dos formatos, esse processo é mais complexo de ser gerenciado. Ainda há a possibilidade de que os elementos de dados em um fluxo possam chegar fora de ordem. Isso implica que o sistema também precisa considerar essas falhas e ter uma medida de tolerância para fazer ajustes, quando for possível, e o processamento dos dados. Um exemplo disso ocorre quando utilizamos sistemas que compensam perdas de dados por meio de algoritmos de Inteligência Artificial.

Volatilidade

Os elementos de fluxo de dados são gerados em tempo real e representam estados de um sistema que está sob monitoramento. Isso implica que a recuperação desses dados é bastante difícil, quando ocorre uma falha de transmissão. Não se trata apenas de retransmitir os dados, mas da impossibilidade de reproduzir o estado do sistema quando os dados foram gerados. Portanto, é necessário desenvolver estratégias que minimizem esse problema, como redundâncias de monitoramento e armazenamento de dados. Um exemplo disso ocorre quando monitoramos o funcionamento de equipamentos e precisamos decidir até quando vamos guardar os dados.

Atividade

Questão 1

Projetos de Big Data são complexos, pois muitos aspectos devem ser considerados. Um deles é o fluxo de dados conhecidos como streamings. Nesse sentido, assinale a alternativa correta sobre as características e os desafios em relação ao processamento de fluxo de dados em projetos de Big Data.

A

Quando um sistema de fluxo de dados de Big Data falha, é possível recuperar os dados, reiniciando o sistema.

B

Aplicações de streaming são caracterizadas por fluxos não contínuos de dados. Por isso, é um desafio dimensionar uma infraestrutura para evitar a ociosidade do sistema.

C

Os fluxos de dados de aplicações de tempo real precisam de garantia de qualidade de serviço, pois não é possível fazer análises confiáveis com dados voláteis.

D

Muitas aplicações de Big Data que utilizam fluxos de dados são de tempo real, cujos dados precisam ser processados com muita velocidade, pois, em muitos casos, seu valor é reduzido ao longo do tempo.

E

Uma das vantagens de trabalhar com sistemas de fluxos de dados é o fato de eles serem oriundos da mesma fonte, o que reduz a complexidade da infraestrutura necessária para o processamento.

Parabéns! A alternativa D está correta.

Em aplicações como monitoramento de sinais vitais de pacientes e de segurança, é necessário processar os dados com grande velocidade. Por exemplo, depois de algum tempo, o paciente pode sofrer graves consequências por não ter sido atendido. E uma

equipe de segurança pode perder a oportunidade de intervir contra uma atividade criminosa. Projetos desse tipo são muito complexos, pois precisam garantir a disponibilidade dos dados e a velocidade de sua transmissão e de seu processamento, para detectar padrões e permitir que ações sejam tomadas dentro de um tempo adequado.

04. Uma aplicação no mercado financeiro

Preparação do ambiente

Preparando o ambiente para desenvolver um projeto na linguagem Python

1. Crie uma conta no Gmail.
2. Crie um projeto no Google Colab.
3. 3. No Google Colab, crie uma célula de código, escreva e execute o comando

   ```
   pip install yfinance
   ```

   .
4. Crie outra célula de código e execute o comando `pip install plotly`.
5. Crie outra célula de código e execute o comando `pip install seaborn`.
6. Crie outra célula de código e execute o comando `pip install plotly`.
7. Crie outra célula de código e execute o comando `pip install seaborn`.
8. Teste se a instalação dos pacotes está correta, criando outra célula de código para importar o pacote e verificar a versão. Por exemplo, para testar a instalação do yfinance, execute os seguintes comandos na mesma célula de código:

   ```
   import yfinance as yf
      a. print(yf.__version__)
   ```

Atividade

Questão 1

Agora que você está com seu ambiente de trabalho pronto, é sua vez de testar se os demais pacotes foram instalados corretamente. Você precisa fazer esse teste para ganhar mais confiança no desenvolvimento de sua aplicação. Realize o teste dos pacotes Pandas, Plotly e Seaborn.

Abrir solução

O procedimento para testar os demais pacotes é o mesmo que realizamos anteriormente. Ou seja, para cada pacote, você precisa criar outra célula de código para importar o pacote e verificar a versão.

No caso do Pandas, faça:

```
import pandas as pd
print(pd.__version__)
```

Para o Plotly, faça:

```
import plotly as ply
print(ply.__version__)
```

Por fim, para o Seaborn, faça:

```
import seaborn as sns
print(sns.__version__)
```

Parabéns por ter chegado até aqui! Agora, você já domina o processo de instalação dos pacotes do Python e de verificação para saber se está tudo certo. Isso será útil para qualquer projeto Python que você realize.

Aquisição dos dados

Realizando a aquisição dos dados

1. Crie uma célula para importar os pacotes pandas e yfinance. Para isso, execute os seguintes comandos:
    1.1. `import pandas as pd`
    1.2. `import yfinance as yf`
2. Crie outra célula e importe os dados de uma empresa cujas ações são negociadas na Bolsa de Valores de São Paulo (Bovespa) para determinado período. Para isso, execute o seguinte comando:

    2.1. `df = yf.download('PETR4.SA', start='2019-11-01', end='2020-06-01' group_by="ticker");`
3. Teste se a importação dos dados está correta. Para isso, você precisa criar outra célula de código e executar o comando `df.head()`.

Atividade

Questão 1

Você já conseguiu importar os dados corretamente, mas o teste que realizou exibiu apenas os cinco primeiros registros. Agora, seu desafio é exibir também os cinco últimos registros. Vamos lá!

Abrir solução

O procedimento para exibir os cinco primeiros registros e os cinco últimos é bem simples. Basta criar uma célula de código e executar o comando:

`df`

A saída que você vai ver será semelhante a esta:

Date	Open	High	Low	Close	Adj Close	Volume
2019-11-01 00:00:00-03:00	30.590000	31.230000	29.840000	30.430000	16.185854	101210200
2019-11-04 00:00:00-03:00	30.889999	31.219999	29.959999	30.360001	16.148621	81023400
2019-11-05 00:00:00-03:00	30.410000	30.600000	29.580000	29.650000	15.770967	92980900
2019-11-06 00:00:00-03:00	30.049999	30.700001	28.100000	29.709999	15.802880	154003100
2019-11-07 00:00:00-03:00	30.000000	31.070000	29.540001	30.900000	16.435850	96329000
...	...	...	...	...	...	...
2020-05-25 00:00:00-03:00	19.480000	19.559999	19.260000	19.480000	10.575249	38033000
2020-05-26 00:00:00-03:00	19.980000	20.090000	19.330000	19.670000	10.678396	68716900
2020-05-27 00:00:00-03:00	19.799999	19.930000	19.150000	19.930000	10.819545	74790800
2020-05-28 00:00:00-03:00	19.690001	20.080000	19.450001	19.770000	10.732684	65023000
2020-05-29 00:00:00-03:00	19.549999	20.340000	19.299999	20.340000	11.042123	128832600

Captura de tela da execução do Python no Colab exibindo os cinco primeiros e os cinco últimos registros do DataFrame.

Os dados que você está vendo correspondem aos valores da ação por dia. As colunas trazem informações sobre o preço de abertura da ação, o maior e o menor preço negociado, os preços de fechamento e ajustados, e o volume negociado. Muito interessante, não é?

Agora, você pode seguir para o próximo passo de nossa aplicação.

Visualização dos dados

Realizando a aquisição dos dados

1. Crie uma célula para visualizar os dados em um histograma. Para isso, execute os seguintes comandos:

```
import seaborn as sns
sns.set_theme(style="darkgrid")
sns.displot(df['Close'].dropna(),kde=True)
```

2. Crie outra célula para visualizar os dados em uma série temporal. Para isso, execute os seguintes comandos:

```
import plotly.offline as py
import plotly.graph_objs as go
dados = [go.Scatter(x=df.index, y=df['Close'])]
layout = go.Layout(title='Histórico dos Preços da Ação',
yaxis={'title':'Preços'}, xaxis={'title': 'Período'})
fig = go.Figure(data=dados, layout=layout) py.iplot(fig)
```

Atividade

Questão 1

Você já tem duas ótimas visualizações dos dados! Agora é sua vez de explorá-los um pouco mais. Existe algum período que chama a atenção na análise da série temporal? Caso exista, o que poderia explicar esse fato?

Ao analisar o gráfico da série temporal, percebemos uma queda significativa do preço da ação no período que inicia em 19/fev/2020. Esse período foi exatamente o momento em que o planeta inteiro ficou estarrecido com a pandemia do novo coronavírus. Para encontrar o menor valor da ação, você pode criar mais uma célula de código e executar os seguintes comandos:

```
minimo=df['Close'].min()
print(f'mínimo: {minimo}')
```

Assim, você verá que o menor valor da ação do período crucial da pandemia ficou muito abaixo da média dos dados da série. No caso, você verá o seguinte resultado:

Captura de tela da execução do Python no Google Colab exibindo menor valor da ação.

Conclusão

O que você aprendeu neste conteúdo?

Neste conteúdo, você aprendeu:

- A definição de Big Data e sua aplicação prática;
- Os conceitos de IoT e computação distribuída;
- As plataformas em nuvem para aplicações de Big Data;
- Uma aplicação de finanças desenvolvida na nuvem.

Explore +

Acesse o site do Arduino – **Built-in Examples - Arduino Documentation** – e estude os diversos exemplos didáticos de como construir projetos superinteressantes

Em seguida, tente programar esses projetos no site **Tinkercad**.

Acesse o site **Spark Streaming Programming Guide** e aprofunde seu conhecimento sobre processamento de fluxo de dados com base em exemplos práticos.

Referências bibliográficas

GANTZ, J. REINSEL, D. **Extracting value from chaos.** IDC iView, p. 1-12, 2011.

LANEY, D. **3-D data management** : controlling data volume, velocity and variety. META Group Research Note, 2001.

RUSSOM, P. **Big Data Analytics.** TDWI Best Practices Report, Fourth Quarter 2011. TDWI Research, 2011.