

Universidade Estácio de Sá

Introdução à Internet das Coisas

Luiz Marcio Faria de Aquino Viana, Pós-D.Sc.
Engenharia de Sistemas e Computação
Área de Arquitetura de Computadores e Sistemas Operacionais
Engenheiro Eletricista
Ênfase em Engenharia de Sistemas e Computação
Referência: <http://lattes.cnpq.br/7394201856453216>
CPF: 024.723.347-10
RG: 08855128-8 IFP-RJ
Registro: 2000103581 CREA-RJ
E-mail Principal: lmarcio@tlmv.com.br
Outro E-mail: luiz.marcio.viana@gmail.com
Outro E-mail: luiz_marcio@hotmail.com
Telefone: +55-21-99983-7207

Conteúdo

1. Conceitos sobre Internet das Coisas

- 1.1. *Local Computation*
- 1.2. *Client-server Computation*
- 1.3. *Distributed Computation*
 - 1.3.1. *Cluster de Servidores*
 - 1.3.2. *Supercomputadores*
 - 1.3.3. *Unidades de Armazenamento com Capacidade de Processamento (CSD)*
 - 1.3.4. *Equipamentos Remotos com Capacidade de Processamento*
 - 1.3.5. *Processamento Distribuído Realizado por Veículos Autônomos*
- 1.4. *Cloud Computing*
 - 1.4.1. *Serviços de Cloud Computing*
 - 1.4.2. *External Datacenters*
 - 1.4.3. *Contratação de External Datacenters*
 - 1.4.4. *Internal Datacenters*
- 1.5. *Edge Computing*
- 1.6. *Fog Computing*
- 1.7. *Internet of Things*

Conteúdo

2. Introdução às Redes de Comunicação

- 2.1. Redes de Comunicação
- 2.2. Redes Cabeadas
 - 2.2.1. Redes por Cabo Metálico
 - 2.2.2. Redes por Fibra Óptica
- 2.3. Satélites de Comunicação
 - 2.3.1. Transmissão por Satélite
- 2.4. Redes sem Fio (DIRECIONAL PONTO-A-PONTO)
- 2.5. Redes sem Fio (OMNI-DIRECIONAL MULTIPONTO)
- 2.6. Redes de Telefonia Celular
- 2.7. Redes de Longo Alcance

Conteúdo

3. Armazenamento e Processamento

- 3.1. Bancos de Dados (OLTP/OLAP)
 - 3.1.1. Bancos de Dados Transacionais (OLTP)
 - 3.1.2. Bancos de Dados Analíticos (OLAP)
 - 3.1.3. Problemas Endereçados
- 3.2. Bancos de Dados (NoSQL)
 - 3.2.1. Introdução aos Bancos de Dados NoSQL
 - 3.2.2. Aplicação dos Bancos de Dados NoSQL
- 3.3. BIG DATA (SPARK + HADOOP)
 - 3.3.1. Introdução ao Sistema de Armazenamento Distribuído HADOOP
 - 3.3.2. Responsabilidade de Cada Nó
 - 3.3.3. Características do Sistema de Arquivos HDFS
 - 3.3.4. Vantagens do Sistema de Arquivos HDFS
 - 3.3.5. Introdução ao Apache Spark
 - 3.3.6. *Resilient Distributed Datasets* (RDD)
 - 3.3.7. API GraphX
 - 3.3.8. API MLLib
- 3.4. WebServices (J2EE + JBOSS EAS)
 - 3.4.1. Servidores de Aplicação
- 3.5. OUTROS: DO-IT-YOURSELF (DIY)

Conteúdo

4. Recursos de Análise dos Dados

4.1. Sistemas Especialistas

4.1.1. Base de Conhecimento

4.1.2. Árvore de Decisão

4.1.3. Regras de Negócio

4.2. Inteligência Computacional

4.2.1. Conceitos de Inteligência Computacional

4.2.2. *Linear Model*

4.2.3. *Multi-layer Perceptron* (MLP)

4.2.4. *Neural Networks* (NN)

4.2.5. Redes Neurais Profundas (*Deep Neural Networks* – DNN)

4.2.6. Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN)

4.2.7. Redes Neurais para Microcontroladores (TinyML)

4.2.8. Processos de Otimização de Redes Neurais (TinyML)

Conteúdo

5. Microcontroladores

5.1. Arduino UNO R3 – ATmega328P

5.1.1. Placa Arduino UNO R3

5.2. ESP32 com OLED – Xtensa LX6

5.2.1. Placa WiFi LoRa 868 MHz ESP32 com OLED

5.3. Outros Microcontroladores

5.3.1. Raspberry PI 4

5.3.2. NVIDIA Jetson Nano

6. Processadores Avançados (SoC)

6.1. *Qualcomm Processors*

6.1.1. Qualcomm APQ8016E

6.1.2. Qualcomm APQ8064E

6.3. Qualcomm (IoT)

6.3.1. Placas Microcontroladoras e Kits de Desenvolvimento da Qualcomm

Conteúdo

7. Exemplos Práticos

7.1. APLICAÇÃO 1: Microcontroladores

7.1.1. Aplicação Usando Microcontroladores com Suporte as Redes WiFi

7.1.2. Objetivo

7.1.3. Modelo (APLICAÇÃO 1)

7.2. APLICAÇÃO 2: Android

7.2.1. Aplicação Usando Smartphones e Tablets com Android

7.2.2. Objetivo

7.2.3. Modelo (APLICAÇÃO 2)

1. Conceitos sobre Internet das Coisas

1.1. *Local Computation*

- O modelo de computação local utiliza os recursos de um único computador na realização de tarefas.
- Este tipo de computação é muito encontrado nas aplicações corporativas usadas nos computadores pessoais, como o Microsoft Word, Microsoft Excel, Microsoft Powerpoint, AutoCAD, Autodesk Revit, Adobe PhotoShop, Games, e etc.
- Devido a necessidade crescente de capacidade de processamento, os computadores tem evoluído constantemente, e tarefas anteriormente complexas são atualmente muito simples de resolver usando o poder de processamento dos computadores atuais.

EVOLUÇÃO DOS COMPUTADORES:



ABACO CHINES

A velocidade de processamento depende do operador!



TK82-C

ZILOG Z80A;
3,25 Mhz; 2 KB RAM.



Color 64

MOTOROLA MC6809E;
1,8 Mhz; 64 KB RAM.



IBM PC

INTEL 8088;
4,77 Mhz; 256 KB RAM.



DELL XPS 8950

INTEL CORE i7-12700;
12-núcleos; 24-threads;
3,6 à 4,9 GHz;
128 GB RAM.

1. Conceitos sobre Internet das Coisas

1.1. *Local Computation*



1981/82 - TK82-C

ZILOG Z80A; 3,25 Mhz; 2 KB RAM.

```
; GALAX-FUNDAMENTOS COMP. DIGITAIS
; AUTOR:
; Luiz Marcio Faria de Aquino Viana
; DATA: 1993/94

; --- rotina de movimento dos silons ; --- continuacao

MovSilon:
LD    IY, y0Fsilon
LD    IX, x0Fsilon
LD    HL, silStat
LD    B, 16

MSil1:
LD    A, (HL)
CP    DEAD
JR    Z, MSil3
JR    CnvSilon
MSil3:      INC    IX
INC    IY
INC    HL
DJNZ  MSil1
RET

CnvSilon:
CALL  ClrImage
LD    A, (cnd1)
CP    LEFT
CALL  Z, CnvLeft
CP    RIGHT
CALL  Z, CnvRight
CALL  PutSilon
JR    MSil3

PutSilon:
PUSH  BC
PUSH  IX
PUSH  IY
LD    B, (IY)
LD    C, (IX)
LD    A, (cndImg)
CP    IMAGE0
JR    NZ, PSil1
LD    DE, silon0
JR    PSil2
PSil1:      LD    DE, silon1
PSil2:      CALL  WriteXY
POP    IY
POP    IX
POP    BC
RET
```



2018/19 - Samsung J4+

Snapdragon 425 Qualcomm MSM8917; 64 bits; Quad Core; 1.4 GHz; 2 GB RAM; GPU Adreno 308; 32 GB Static Memory;

```
/*
 * Chefao1.java
 * emissao: Luiz Marcio Faria de Aquino Viana, 31/12/2011
 * revisoes: ...
 */

package com.tmlsoft.defender;

public class Chefao1 extends GameObject {
    public static int[] IMAGES = { R.drawable.chefao_01, R.drawable.explosion_chefao_01 };

    private int m_direcaoX;
    private int m_direcaoY;
    private int m_hits;

    public Chefao1() {
        super(Defs.OTYPE_CHEFA01, 0, 0, Chefao1.IMAGES, Defs.POINTS_CHEFA01, 0);
        double w = this.m_images.get(this.m_selectedImage).getWidth();
        double h = this.m_images.get(this.m_selectedImage).getHeight();
        m_x = (FrmGame.g_app.getWidth() / 2) - (int)(w / 2);
        m_y = (int)(-h);
        m_numberOfParaquedistas = 100;
        m_hits = 100;
        m_direcaoX = 10;
        m_direcaoY = 10;
    }

    public void move();
    public int crashCheck(GameObject o);
}
```

1. Conceitos sobre Internet das Coisas

1.2. *Client-Server Computation*

O modelo de computação cliente servidor surgiu da necessidade de compartilhar dados corporativos entre um grupo de pessoas em uma empresa.

Existem 2 (dois) modelos de computação cliente servidor:

- (i) Computação cliente servidor com 2 (duas) camadas;
- (ii) Computação cliente servidor multi-camadas;



Computação cliente servidor com 2 (duas) camadas.



Computação cliente servidor com multi-camadas.

1. Conceitos sobre Internet das Coisas

1.3. *Distributed Computation*

O processamento distribuído surgiu da necessidade de distribuir o processamento de algumas aplicações que consomem muito tempo, entre diversos computadores com capacidade computacional elevada, com o objetivo de reduzir o tempo total de processamento da aplicação.

1.3.1. **CLUSTER DE SERVIDORES**

- Os *Clusters* de Servidores são formados por computadores de prateleira (*Off-The-Shelf*), com custo mais baixo, e que distribuem o processamento e armazenamento entre os diversos equipamentos.
- Os servidores do *Cluster* são interligados por uma rede com capacidade de até 10 Gbps.
- Em um Cluster de Servidores, a probabilidade de falha de 1 (um) servidor, ou nó do Cluster, aumenta com o acréscimo de servidores, e desta forma, os dados armazenados são replicados em mais de 1 nó do *Cluster*.

1.3.2. **SUPERCOMPUTADORES**

- Os Supercomputadores atuais são formados por *Cluster* de Servidores de alto desempenho, reduzindo a probabilidade de falha de um dos nós do *Cluster*.
- A interligação dos servidores do *Cluster* ocorre por redes de alto desempenho ou por barramentos especializados e ultrapassam frequentemente os 10 Gbps.
- Exemplos de barramentos usados em Supercomputadores: HPE Slingshot interconnect - 200 Gbps; Tofu interconnect D - 400 Gbps; Quad-rail NVIDIA HDR100 Infiniband - 400 Gbps;

1. Conceitos sobre Internet das Coisas

1.3. *Distributed Computation*



FRONTIER - HPE CRAY EX235A, AMD OPTIMIZED 3RD GENERATION EPYC 64C 2GHZ, AMD INSTINCT MI250X, SLINGSHOT-11

AMD Optimized 3rd Generation EPYC 64C 2 GHz
8.730.112 Núcles de Processamento
Barramento de Interconexão Slingshot-11 - 200 Gbps;



SUPERCOMPUTER FUGAKU - SUPERCOMPUTER FUGAKU, A64FX 48C 2.2GHZ, TOFU INTERCONNECT D

ARM A64FX 48C 2.2GHz
7.630.848 Núcles de Processamento
Tofu interconnect D - 400 Gbps;



Cluster de Servidores de Prateleira (Off-The-Shelf)

Barramento de Interconexão com velocidades de até 10 Gbps;

1. Conceitos sobre Internet das Coisas

1.3. *Distributed Computation*

1.3.3. UNIDADES DE ARMAZENAMENTO COM CAPACIDADE DE PROCESSAMENTO (CSD)

O conceito de *computational storage devices* (CSD) vem sendo amplamente estudado, com a aplicação dos recursos de processamento diretamente na unidade de armazenamento, permitindo que diversos serviços suportados anteriormente no sistema operacional ou na camada da aplicação, estejam disponíveis na unidade de armazenamento, reduzindo o trabalho da CPU e proporcionando maior velocidade na transferência dos dados entre as unidades de armazenamento e memória.



Unidade de armazenamento Newport CSD com 16 TB e com capacidade de processamento embutido (*in-situ processing*). Processador ARM Cortex Série A de 64 bits; 4 núcleos de processamento; 8 KB I-Cache L1 por núcleo; 8 KB D-Cache L1 por núcleo; Linux Ubuntu 16.04.5 LTS; e barramento de dados com 4 Canais PCI-e/NVM-e - 4x 16 Gbps = 64 Gbps.



Servidor x86 de 64 bits com 24 unidades de armazenamento Newport CSD de 16 TB, totalizando 1654 TB de armazenamento em 1U de espaço no seu RACK.

1. Conceitos sobre Internet das Coisas

1.3. *Distributed Computation*

1.3.4. EQUIPAMENTOS REMOTOS COM CAPACIDADE DE PROCESSAMENTO

(a) Smartphones e Tablets

Atualmente, com o aumento da capacidade de processamento dos smartphones, muitas aplicações comerciais usam estes equipamentos para pré-processar os dados de forma distribuída, e transferir o resultado do processamento para um *Cluster* de Servidores. Exemplo: Google Photos, Waze, Games online e outros.

(b) Microcontroladores

Diversos modelos de Microcontroladores atuais possuem capacidade de processamento e armazenamento similar aos computadores de mesa, e muitos estudos tem sido realizados para aproveitar a capacidade de processamento e armazenamento destes equipamentos de forma distribuída.



Aplicações processam os dados de forma distribuída nos smartphones e tablets, e transferem os dados pré-processados para um *Cluster* de Servidores.



Imagem de *Cluster* formado por Microcontroladores modelo Raspberry PI 4.

1. Conceitos sobre Internet das Coisas

1.3. *Distributed Computation*

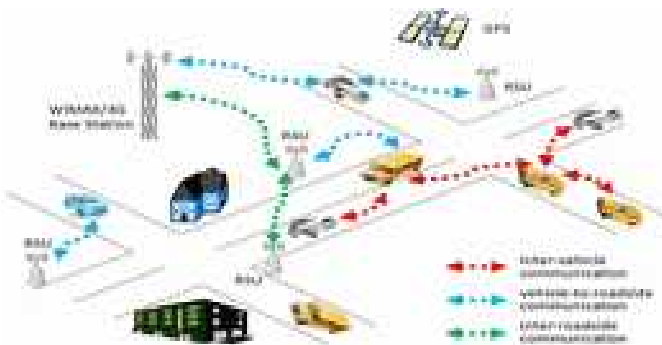
1.3.5. PROCESSAMENTO DISTRIBUÍDO REALIZADO POR VEÍCULOS AUTÔNOMOS

(a) VANET - Vehicular Ad-Hoc Network

Rede de comunicação entre veículos e diversos equipamentos de comunicação instalados nas cidades, que permite o compartilhamento dos dados entre os veículos para proporcionar melhores serviços, como melhorar as informações de tráfego, e permitir a utilização de veículos autônomos em vias urbanas.

(b) Comunicação entre Veículos Aéreos Autônomos

Serviços de aerolevantamento realizados em áreas muito extensas, normalmente são realizados por múltiplos veículos aéreos autônomos, que precisam sincronizar informações sobre o trabalho realizado, permitindo que em caso de falha de um dos equipamentos de aerolevantamento, os outros possam realizar a tarefa com sucesso.



VANET - Vehicular Ad-Hoc Network

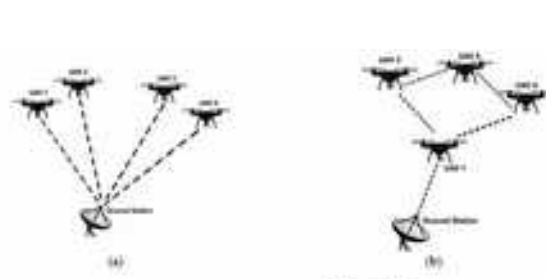


Imagem das topologias de comunicação entre veículos aéreos autônomos.



1. Conceitos sobre Internet das Coisas

1.4. Cloud Computing

O conceito de computação em nuvem, *cloud computing*, permite que os recursos de transferência de dados, armazenamento e processamento, sejam compartilhado por diversas empresas em um serviço de nuvem pública, como Amazon AWS, Microsoft Azure, Oracle Cloud, Google Cloud, e etc, ou por diversos projetos de uma grande empresa que compartilham uma nuvem privada, como ocorre na Petrobrás e outras grandes empresas nacionais e estrangeiras.

1.4.1. SERVIÇOS DE CLOUD COMPUTING

(a) Amazon EC2

- Servidores virtuais na nuvem.
- Plataformas Suportadas: Linux ou Windows. Servidor Windows possui custo maior, devido ao custo de licenciamento.
- Bancos de Dados Suportados: PostgreSQL ou MySQL, que são bancos de dados livres. Outro banco de dados como o Oracle ou SQL Server possui custo maior devido ao custo de licenciamento.

- **Outros:** **MacinCloud** (<https://www.macincloud.com/>)

(b) Amazon Simple Storage Service (S3)

- Armazenamento de dados na nuvem.

(c) Amazon S3 Glacier

- Serviço que permite que backups pouco acessados sejam armazenados com custo reduzido. Excelente opção para armazenamento de dados de fechamento trimestrais, semestrais e anuais das empresas.

Imagem servidor virtual na nuvem - Locaweb

* A vantagem da Locaweb sobre os concorrentes estrangeiros que cobram em Dólar (US\$), é a previsibilidade dos custos do servidor que é precificado em Reais (R\$).

1. Conceitos sobre Internet das Coisas

1.4. *Cloud Computing*

1.4.2. EXTERNAL DATACENTERS

O uso de Datacenters externos, é um recurso bastante explorado por diversas grandes empresas, como Netflix, DPASCHOAL e DataSafer.

Estas empresas usam Datacenters como os da Acenty (<https://ascenty.com/>), para instalar servidores e aproveitar a infraestrutura destas unidades, que inclui segurança no acesso aos servidores, redundância na comunicação de dados e de energia elétrica, permitindo o funcionamento dos servidores sem interrupção (24x7).



ASCENTY (<https://ascenty.com/>)

Datacenter Rio de Janeiro 1

Endereço:

Av. Coronel Phidias Távora 513

Pavuna - Rio de Janeiro - RJ

Área: 7.000 m²

Capacidade Energia: 10 MW

Capacidade de Energia do Gerador: 21 MVA

Redundância de UPS: Tribus System - DRUPS (N+1)

Redundância de Resfriamento: N+1 Chilled Water Plant

Carga Máxima do Piso Elevado: 1.504 Kg/m²

1. Conceitos sobre Internet das Coisas

1.4. *Cloud Computing*

1.4.3. CONTRATAÇÃO DE EXTERNAL DATACENTERS

Existem 3 (três) modelos de contratação de espaço nos Datacenters da Ascenty (<https://ascenty.com>).



(a) RACK: Modelo de contratação onde o cliente instala servidores em 1 ou mais RACKs que ficam em áreas comuns do Datacenter.



(b) CAGE: Modelo de contratação onde o cliente aluga uma área fechada e instala 1 ou mais RACKs.

* Este modelo de contratação, assegura maior segurança aos equipamentos devido a cerca transparente e resistente que contorna a área do cliente, o controle no acesso à CAGE do cliente, e o monitoramento 24 horas por dia com câmeras de vigilância.



(c) DEDICATED AREA: Modelo de contratação onde o cliente aluga um andar inteiro da unidade, e instala 1 ou mais RACKs e as mesas de apoio das equipes de suporte e manutenção própria do cliente.

* Este modelo de contratação, assegura maior segurança aos equipamentos devido ao controle no acesso à DEDICATED AREA do cliente, ao monitoramento 24 horas por dia com câmeras de vigilância, e a presença da equipe própria do cliente responsável pelo suporte e manutenção de seus servidores no local.

1. Conceitos sobre Internet das Coisas

1.4. *Cloud Computing*

1.4.4. INTERNAL DATACENTERS

(a) Empresas de Pequeno e Médio Porte

Empresas de diversos portes possuem áreas reservadas para os servidores. Nestas áreas podem ser instalados servidores sobre bancadas ou em RACKs, e normalmente existe espaço para as ações dos operadores responsáveis pelo suporte e manutenção dos servidores.

(b) Empresas de Grande Porte e Governo

Grandes empresas nacionais e estrangeiras, como a Petrobrás, possuem Datacenters próprios em diversos locais do país. A Petrobrás possui 3 (três) grandes Datacenters no Rio de Janeiro, São Paulo e Salvador, e outros menores espalhados por diversas localidades do país.

A Prefeitura da Cidade do Rio de Janeiro, possui um grande Datacenter localizado no Edifício Teleporto - Cidade Nova - Rio de Janeiro - RJ.



Datacenter da Prefeitura da Cidade do Rio de Janeiro, localizado no Edifício Teleporto na Cidade Nova.

1. Conceitos sobre Internet das Coisas

1.5. *Edge Computing*

O crescimento na capacidade de processamento e armazenamento dos Microcontroladores atuais, como as placas Raspberry PI 4 e NVIDIA Jetson Nano, favorecem a implementação de soluções de computação na borda, edge computing, permitindo que parte do processamento e classificação dos dados seja realizado pelos equipamentos remotos.

Estudos mostram que o tráfego total de dados entre os equipamentos remotos e o Datacenter, pode ser reduzido em até 20 vezes com o uso de computação na borda (*edge computing*).

(a) Smartphones e Tablets

Vários serviços disponíveis para Smartphones e Tablets, como o Waze e o Google Photos, usam dados dos usuários, como informação de localização (GPS) e imagens (Fotos), para aperfeiçoar modelos de roteirização (Waze) ou modelos de redes neurais (Google Photos).

Atualmente, existem algoritmos distribuídos para treinamento de modelos de redes neurais, estes algoritmos são denominados Aprendizado Federado (*Federated Learning*), e não há necessidade de transferir imagens para aperfeiçoar os serviços.

Porém, o Google Photos também é um serviço de armazenamento de imagens (Fotos) na nuvem e certamente devem analisar as imagens de um subconjunto de usuários.

(b) Raspberry PI 4 e NVIDIA Jetson Nano (Microcontroladores)

Os microcontroladores Raspberry PI 4 e NVIDIA Jetson Nano possuem grande capacidade de processamento e são indicados para soluções que exigem o pré-processamento de imagens oriundas de câmeras de vigilâncias, e imagens geradas por aerolevantamento com drones.



Raspberry Pi 4 - Processador Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64bit SoC - Clock 1.5 Ghz - Memória RAM: 4GB DDR4



Jetson Nano - Arquitetura NVIDIA Maxwell com 128 NVIDIA CUDA cores - Processador Quad-core ARM Cortex-A57 MPCore - 4 GB 64-bit LPDDR4 - 16 GB eMMC 5.1 Flash

1. Conceitos sobre Internet das Coisas

1.6. Fog Computing

Fog Computing é o termo usado para caracterizar os *Clusters* de Servidores de armazenamento intermediários, que ficam localizados próximo aos equipamentos remotos.

Estes equipamentos concentram os dados de diversos dispositivos e transferem os dados para Clusters de Servidores na nuvem para consolidação dos dados.

CASO 1: PLATAFORMAS DE PETRÓLEO

(a) **EDGE COMPUTING**

Em uma plataforma de petróleo são gerados grandes volumes de dados de telemetria de cada equipamento.

Estes dados são coletados por sensores e transferidos para Microcontroladores localizados próximos aos equipamentos.

Os Microcontroladores acumulam os dados e no caso de um evento de tempo, falha de equipamento, ou limite de valores, os dados são transferidos para um *Cluster* de Servidores localizados na plataforma.

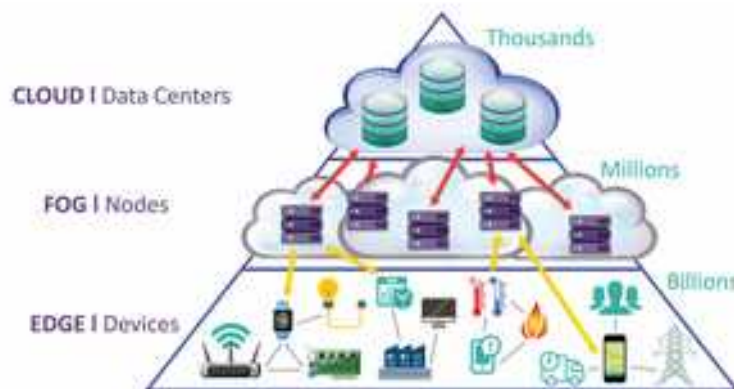
Caso esteja programado alguma ação automática de correção, a ação pode ser realizada imediatamente pelos Microcontroladores,

(b) **FOG COMPUTING**

O Cluster de Servidores armazena os eventos registrados pelos Microcontroladores, e um sistema de gerenciamento e manutenção da operação, consula os dados do *Cluster* de Servidores periodicamente e sinaliza ao operador a ocorrência de eventos, e caso seja necessário o operador pode comandar as ações de correção de um evento a partir do sistema de gerenciamento e manutenção da operação.

(c) **CLOUD COMPUTING**

Periódicamente, todos os eventos registrados no Cluster de Servidores são enviados para o Datacenter da empresa, onde sistemas corporativos avaliam métricas sobre a operação e produção de cada plataforma de petróleo.



1. Concentração da hierarquia da informação em função de cada camada na nuvem. Na extremidade inferior bilhões de equipamentos remotos (*EDGE Devices*) pré-classificam os dados e transferem os dados para servidores próximos (*FOG Nodes*).

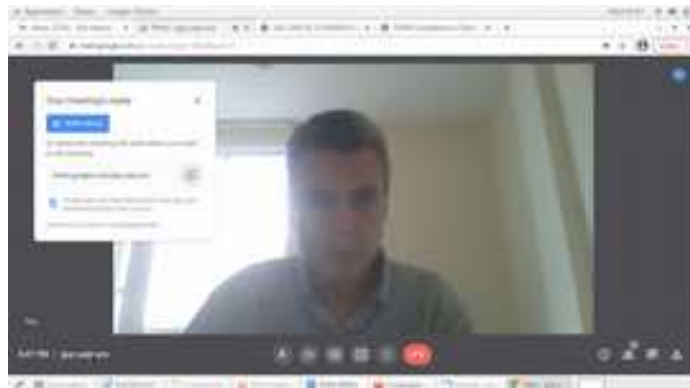
2. Em seguida, milhões de nós de processamento presentes na camada intermediária (*FOG Nodes*), concentra a informação e periodicamente envia os dados para os servidores na nuvem (*CLOUD Data Centers*).

3. Aplicações comerciais da empresa processam e fornecem relatórios sobre os dados concentrados nos milhares de servidores na nuvem (*CLOUD Data Centers*).

1. Conceitos sobre Internet das Coisas

1.7. Internet of Things

- Atualmente, os dispositivos portáteis do tipo smartphones geram um grande volume de informações que são coletadas automaticamente pelas aplicações.
- Informações de localização (GPS) e imagens (Fotos), geradas por smartphones e tablets são utilizadas para aprimorar modelos de roteirização e modelos de inteligência computacional.
- Esta quantidade gigantesca de informações é enviada automaticamente a diversos servidores, para armazenar vídeos, fotos, audios, mensagens, informações de navegação, informações de usabilidade, erros do sistema, dados de posicionamento geográfico, dados da rede celular, dados das redes WiFi e Bluetooth que estes equipamentos identificam a todo instante, e tudo que a turma do FBI, CIA, NSA, Polícia Federal, Polícia Militar, Polícia Civil, Exército, Marinha e Aeronáutica, e os fabricantes de serviços em nuvem, softwares e hardwares imaginarem que será útil hoje ou no futuro.
- Este gigantesco conjunto de informações são armazenados, jamais apagados, e analisados constantemente para fornecer um **“PERFIL DOS USUÁRIOS”** e prover um serviço melhor.



Google Meeting - Serviço de Web Meeting muito usado durante a COVID-19

Prof. Luiz Marcio Faria de Aquino Viana, Pós-D.Sc., pronto para uma reunião online com a equipe de desenvolvedores (ADN) da Autodesk Inc, durante a COVID-19 em 2021-22.

2. Introdução às Redes de Comunicação

2.1. Redes de Comunicação

- A necessidade de comunicação é fundamental em qualquer sociedade.

"No Brasil, o primeiro cabo submarino fez parte da primeira linha telegráfica brasileira. Foi inaugurado em 1857 e interligava a Praia da Saúde no Rio de Janeiro com a cidade de Petrópolis. A linha tinha extensão total de 50km, sendo 15km em cabo submarino. (<https://www.teleco.com.br/>)"

"A Advanced Research Projects Agency Network (ARPANET; Rede da Agência para Projetos de Pesquisa Avançada) foi uma rede de computadores construída em 1969 para transmissão de dados militares sigilosos e interligação dos departamentos de pesquisa nos Estados Unidos, inicialmente financiada pela então Agência de Projetos de Pesquisa Avançada (ARPA, atual DARPA) do Departamento de Defesa dos Estados Unidos. (<https://pt.wikipedia.org/wiki/ARPANET>)"

* NOTAS:

(a) Em 1987, com 16 anos, cheguei a ter contato com um dos primeiros celulares da motorola, mas ainda NÃO tinha visão do que estava por vir.

(b) Em 1988-89, com 17-18 anos, tive a oportunidade de visitar o IBGE - Instituto Brasileiro de Geografia e Estatística, e a diretora em exercício me informou que as Telecomunicações iriam mudar o mundo.

(c) Entre 1992-94, tive os meus primeiros contatos com a World Wide Web (WWW) na UERJ - Universidade do Estado do Rio de Janeiro.

- Usava serviços como Gopher, Telnet, FTP e Mosaic.

- Usava também Linux, PostgreSQL, Apache HTTP Server, HTML, CGI, e PHP.

(d) Somente entre 1995-2001, quando comecei a desenvolver aplicações de Web Mapping, e após adquirir um Palm Pilot e em seguida ganhar um HP iPACK da empresa Autodesk Inc, é que comecei a ter uma visão maior das possibilidades no mundo das telecomunicações com a World Wide Web (WWW) e com os dispositivos móveis (Microcontroladores, Smartphones e Tablets).

* SEGUE ABAIXO IMAGENS DE ALGUNS DISPOSITIVOS COM OS QUAIS JÁ DESENVOLVI APLICAÇÕES DESDE 1995



MOTOROLA
MODELO 1987
SOMENTE VOZ!



Palm Pilot
PalmOS; C/C++



HP iPACK
Windows CE;
C/C++



NOKIA T68
Symbian;
Serial AT
Commands



NEXTEL
MOTOROLA i870
J2ME; GPS e Câmera



TREO
PalmOS; C/C++



MOTOROLA Q9
Windows Mobile;
C/C++



SAMSUNG J4+
Android; Java



iPhone
iOS; IDE GRATUITO: Xamarin (CSHARP);
IDE LICENCIADO: C/C++; Swift

2. Introdução às Redes de Comunicação

2.2. Redes Cabeadas

2.2.1. REDES POR CABO METÁLICO

- Ao transferir os dados entre computadores, usamos diversos meios de transferência, como os cabos metálicos do tipo coaxial ou par trançado, capazes de atender pequenas distâncias, e com capacidade de transmissão limitadas em 10 Mbps (coaxial), 100 Mbps ou 1 Gbps (par trançado).
- Em áreas menores, como em um RACK, podemos alcançar velocidades de transmissão de 10 Gbps à 40 Gbps, usando ainda soluções de redes Ethernet.
- Normalmente, as redes de cabo metálico são aplicadas na distribuição de rede de dados em pequenas localidades, como em uma unidade empresarial ou residência.



CABO COAXIAL REDE 10 BASE-2

Ligação de passagem; Ligação final com terminador;
Máx: 10 Mbps



CABO CAT5/CAT5e REDE 100 BASE-TX

Máx: 100 Mbps



CABO CAT6 REDE 1000 BASE-TX

Máx: 1 Gbps



CABO CAT8 REDE 40G BASE-T

Máx: 40 Gbps

2. Introdução às Redes de Comunicação

2.2. Redes Cabeadas

2.2.1. REDES POR CABO METÁLICO

I. CODIFICAÇÃO DE DADOS E TRANSMISSÃO POR CABO METÁLICO

- A transmissão por cabo metálico, é realizada através da codificação de 1 byte de dados, 8 bits, em um sinal elétrico de período T e frequência ($f = 1/T$).
- A composição deste sinal usa um conceito matemático apresentado por Jean Fourier (Francês), que demonstrou que qualquer função periódica pode ser decomposta em um conjunto de funções senoidais periódicas: AS COMPONENTES HARMÔNICAS.

(a) Imagens da aproximação da composição de um Sinal (em Vermelho) por suas Harmônicas (em Azul).

(b) Observe que independentemente da frequência, 60 Hz para Rede Elétrica ou de 38.400 Hz para Rede de Dados, a quantidade de harmônicas necessárias para formar o sinal codificado de 8 bits é sempre a mesma. O aumento da frequência, aumenta a quantidade de dados trafegados pelo meio, mas a característica do sinal será sempre o mesmo. Este fato está visível na equação ($= 2 \times \pi \times N$).

$$g(t) = \frac{1}{2}C + \sum_{n=1}^{\infty} a_n \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cos(2\pi n f t)$$

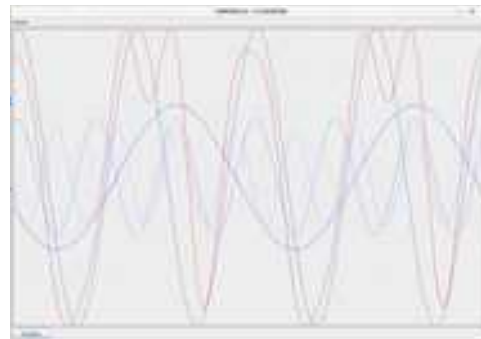
$$a_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi n f t) dt$$

$$b_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi n f t) dt$$

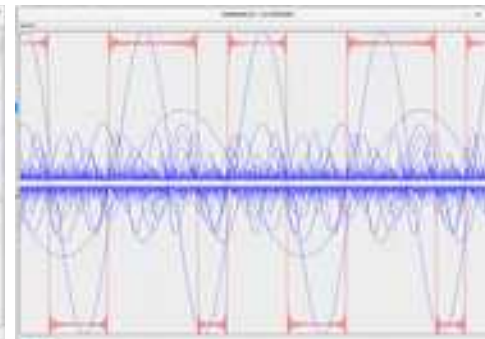
$$C = \frac{2}{T} \int_0^T g(t) dt$$

$$a_n = \frac{1}{\pi n} [\cos(\pi n / 4) - \cos(3\pi n / 4) + \cos(5\pi n / 4) - \cos(7\pi n / 4)]$$
$$b_n = \frac{1}{\pi n} [\sin(3\pi n / 4) - \sin(\pi n / 4) + \sin(5\pi n / 4) - \sin(7\pi n / 4)]$$
$$C = 3/8$$

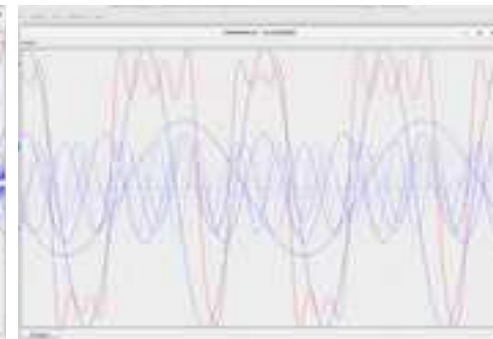
Valores das constantes A_n , B_n e C , calculados para a transmissão do carácter ASCII "b" (=62 HEX), com representação binária "0110 0010".



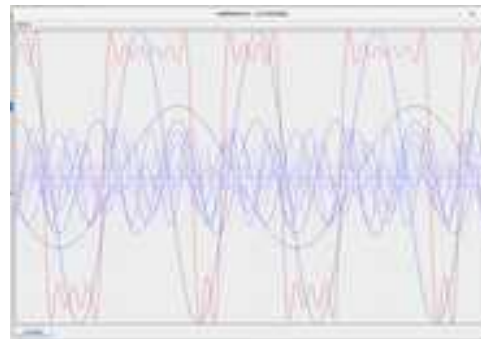
5 HARMÔNICAS



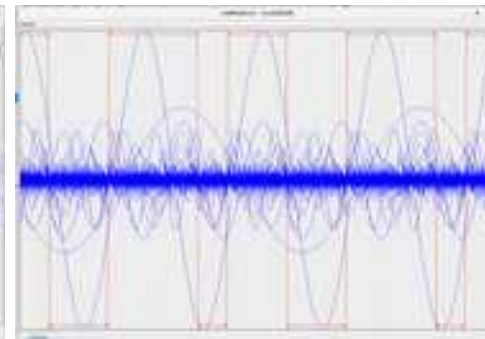
125 HARMÔNICAS



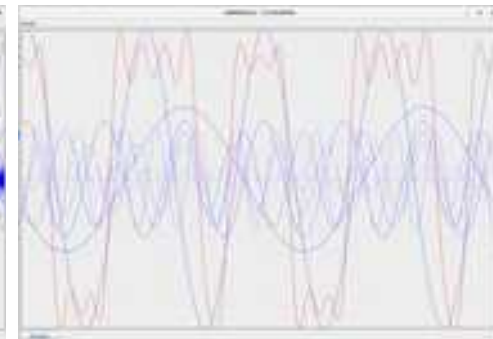
10 HARMÔNICAS (FREQ = 60 Hz)



15 HARMÔNICAS



1250 HARMÔNICAS



10 HARMÔNICAS (FREQ = 38.400 Hz)

2. Introdução às Redes de Comunicação

2.2. Redes Cabeadas

2.2.2. REDES POR FIBRA OPTICA

- As redes por fibra optica, podem cobrir grandes distâncias e possuem capacidade ilimitada de transferência de dados. Porém, a sua instalação e manutenção é mais cara e de difícil execução, porque muitas vezes necessitam de obras em vias públicas para passagem da fibra optica para atendimento dos consumidores.
- O atendimento por fibra optica nas cidades Brasileiras evolue de forma muito lenta pelas operadoras de telecomunicações. Por ser o meio de comunicação com maior capacidade de transmissão de dados e cobertura de atendimento, é o meio de comunicação preferido para interligar centros de comunicação de dados e grandes consumidores comerciais e residenciais.
- As operadoras Telefônica/Vivo, Claro/NET/EMBRATEL, TIM/Intelig e Oi Telecom, fornecem serviços de comunicação de dados usando fibra optica em algumas regiões das grandes cidades.

* NOTA:

- (i) Entre 2003-2004, tive a oportunidade de participar do projeto de levantamento de dados de mercado, GeoCorp, que foi promovido pela Oi Telecom.
- (ii) O projeto GeoCorp realizou um levantamento dos consumidores por área de atendimento, e forneceu a base para o plano de expansão da infraestrutura de rede cabeada da empresa.
- (iii) Recursos usados na solução:
Autodesk AutoCAD Map, Autodesk MapGuide 6.5, Autodesk OnSite View (Free) e HP iPACK.
- (iv) Aplicações desenvolvidas para Autodesk AutoCAD Map, Autodesk MapGuide 6.5 e Autodesk OnSite View (Free) pelo Prof. Luiz Marcio Faria de Aquino Viana, Pós-D.Sc, através de contrato de prestação de serviço com a empresa GRID LAB, localizada na Penha - Rio de Janeiro - RJ.

* **IMPORTANTE:** Na época, também era funcionário de carteira assinada da empresa PARS Produtos de Processamento de Dados LTDA, e trabalhava 8 horas na empresa e 24 horas para mim, como todos os empreendedores deve fazer!



Mapeamento de
redes
subterrâneas
com Georadar



Abertura de
valas Método
Destrutivo



Redes
subterrâneas
Método MND



Fusão de fibra
óptica

SEMATEL - CONSTRUÇÃO DE REDES DE FIBRA ÓPTICA
<http://sematel-rj.com.br/construcao-de-redes-de-fibra-optica/>

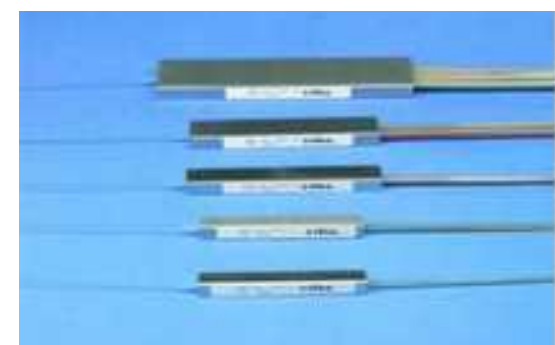
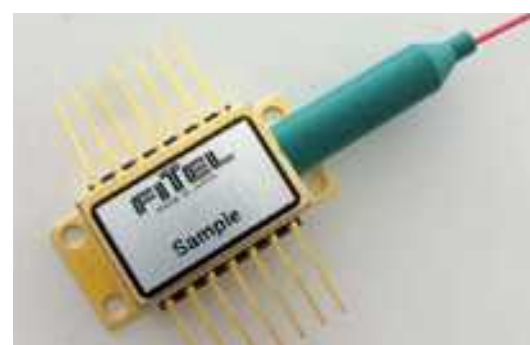
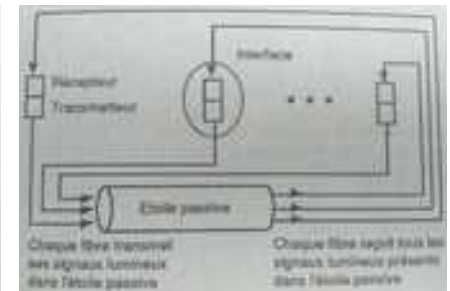
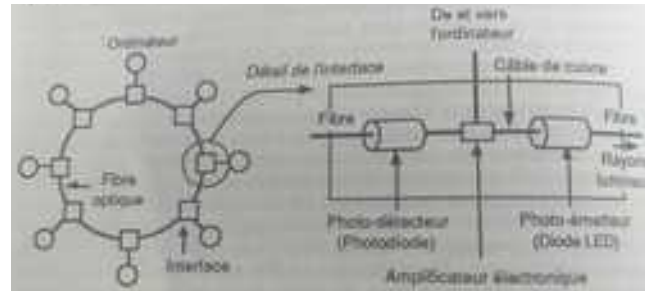
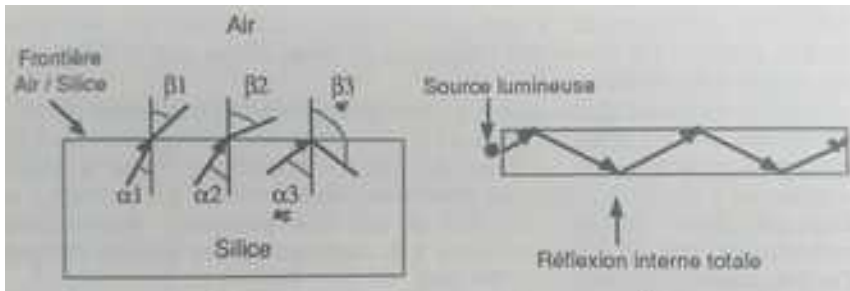
2. Introdução às Redes de Comunicação

2.2. Redes Cabeadas

2.2.2. REDES POR FIBRA OPTICA

I. TRANSMISSÃO DE DADOS POR FIBRA OPTICA

- A transmissão por fibra óptica é realizada por um feixe de luz que percorre um cabo optico. O feixe luminoso é gerado por um foto emissor em uma extremidade da fibra óptica e recebida por um foto receptor na outra extremidade. A existência de luz em um período de tempo representa o bit 1 e a ausência de luz representa o bit 0.
- O conceito da propagação da luz pela fibra optica segue o mesmo princípio da refração da luz quando atravessa diferentes meios físicos, como o ar e o vidro. Dependendo do ângulo de incidência do feixe luminoso, parte da luz é refletida e outra parte é refratada. Quando o ângulo de incidência é maior que um valor determinado, toda a luz é refletida.
- Por usar a faixa de frequência do espectro luminoso para propagação dos dados, a capacidade de transferência de dados por fibra óptica é ilimitada, e os dados podem ser trafegados por kilometros com taxa de transferência de Giga bits/s.

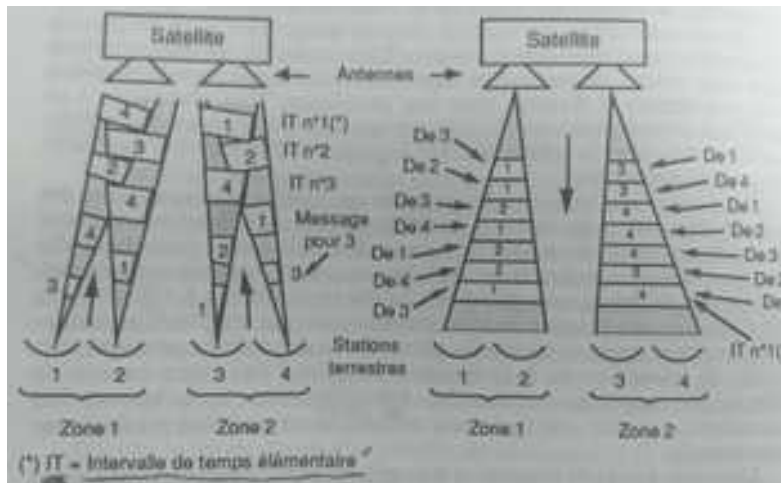


2. Introdução às Redes de Comunicação

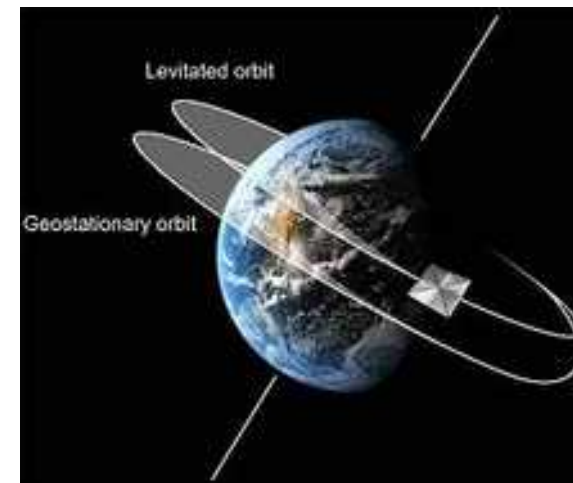
2.3. Satélites de Comunicação

2.3.1. TRANSMISSÃO POR SATÉLITE

- A comunicação por satélite possui uma capacidade de transmissão de dados muito grande, e consegue alcançar diferentes receptores simultaneamente.
- Para permitir a comunicação bi-direcional de dados entre as estações terrestres e os satélites, os sistemas de comunicação por satélite dividem o tempo de transmissão entre as N estações terrestres que estão se comunicando em intervalos de tempo.
- Nos sistemas de telecomunicações usamos de preferência satélites geo-estacionários. Os satélites em órbita geo-estacionária ficam próximos a linha do equador. Para evitar a interferência na comunicação dos dados, cada satélite fica a 4° de distância um do outro. Desta forma, existem somente 90 ($360 / 4 = 90$) satélites geo-estacionários em órbita ao redor da terra.



Transmissão de dados por satélite, com a divisão do tempo de transmissão entre as estações terrestres.



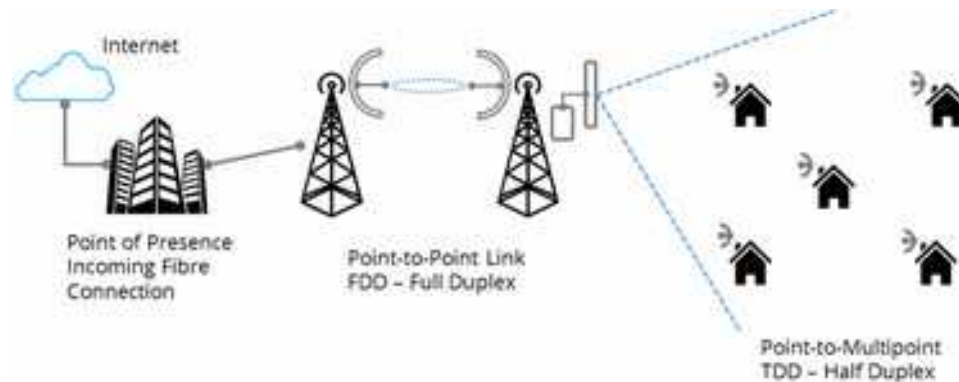
Os satélites em órbita geo-estacionária ficam em uma órbita próxima à linha do equador. Existem no máximo 90 satélites geo-estacionários ao redor da terra em um determinado momento.

2. Introdução às Redes de Comunicação

2.4. Redes sem Fio (PONTO-A-PONTO)

REDES SEM FIO USANDO TRANSMISSÃO DIRECIONAL PONTO-A-PONTO

- As redes sem fio usando transmissão direcional ponto-a-ponto, podem atender com capacidade de transferência de dados de até 40 Gbps. Neste processo de transmissão são utilizadas antenas direcionais com grande precisão.
- A melhoria nos equipamentos de transmissão por rádio, fornece ao sistemas de transmissão direcional ponto-a-ponto uma alta confiabilidade, sendo este o meio preferido de comunicação para atender centros de comunicação, unidades consumidoras e Datacenters localizados em regiões que não possuem atendimento por fibra optica.



Transmissão Direcional Point-To-Point com capacidade de até 40 Gbps de transferência.

2. Introdução às Redes de Comunicação

2.5. Redes sem Fio (MULTIPONTO)

REDES SEM FIO USANDO TRANSMISSÃO OMNI-DIRECIONAL MULTIPONTO

- As redes omni-direcional multiponto são frequentes utilizadas em instalações de unidades comerciais e residenciais. Entre elas podemos destacar as redes sem fio WiFi e Bluetooth, as redes de telefonia Celular, e as redes que cobrem grandes distâncias com baixo consumo de energia, como a rede LoRaWan.



WiFi Communication



5G Communication



Cisco Catalyst 5G Indoor Gateway

<https://blogs.cisco.com/networking/new-catalyst-products-bring-5g-and-accelerated-sase-to-the-wan-edge>



Bluetooth Communication



LoRaWan Communication



Cisco Wireless Gateway for LoRaWan

<https://www.cisco.com/c/en/us/products/collateral/se/inter-net-of-things/datasheet-c78-737307.html>

2. Introdução às Redes de Comunicação

2.6. Redes de Telefonia Celular

REDES DE TELEFONIA CELULAR PÚBLICAS (OUTDOOR) E PRIVADAS (INDOOR/OUTDOOR)

- As redes de telefonia celular possuem capacidade de transferência limitada e com o aumento da frequência a cobertura de atendimento reduz, deste modo, a nova geração da rede de telefonia Celular 5G, necessitará de um maior número de estações radio base para atender uma mesma região que possui cobertura com as redes atuais 4G e 4.5G.
- Estas redes podem ser públicas, como as redes das operadoras de telecomunicações, Telefônica / Vivo, Claro / NET / EMBRATEL, TIM / Intelig e Oi Telecom, ou privadas, como uma rede particular em uma unidade comercial e industrial de grande dimensões.



5G Communication



Cisco Catalyst 5G Indoor Gateway
<https://blogs.cisco.com/networking/new-catalyst-products-bring-5g-and-accelerated-sase-to-the-wan-edge>



Zyxel NR7101 5G NR Outdoor Router
<https://www.zyxel.com/global/en>

2. Introdução às Redes de Comunicação

2.7. Redes de Longo Alcance

REDES LORAWAN

- As redes LoRaWan, usam uma frequência baixa, e desta forma conseguem alcançar longas distâncias com baixo consumo de energia.
- Diferente do que ocorre nas redes WiFi, Bluetooth e de telefonia Celular, os equipamentos que usam as redes LoRaWan, não efetuam uma conexão direta com a estação radio base, e a transmissão dos dados pode ser recebida por mais de uma estação, sendo de responsabilidade do serviço filtrar a informação duplicada no data center.
- Estas redes possuem forte abrangência nas grandes cidades, e podem ser utilizadas de forma bastante útil em grandes áreas industriais, nas áreas rurais e em grandes fazendas.
- Atualmente, muitas soluções de monitoramento e gerenciamento de sensores remotos que transmitem pouco volume de dados, como equipamentos de localização pessoal, equipamentos de medição de consumo de energia e de consumo de água usam redes LoRaWan.



LoRaWan Communication



Cisco Wireless Gateway for LoRaWan

<https://www.cisco.com/c/en/us/products/collateral/se/inter-net-of-things/datasheet-c78-737307.html>

3. Armazenamento e Processamento

3.1. Bancos de Dados (OLTP/OLAP)

3.1.1. Bancos de Dados Transacionais (OLTP)

- Os sistemas de banco de dados transacionais (OLTP) são otimizados para o processamento de transações, porque realizam diversas análises sobre a estrutura das tabelas, estrutura dos índices e sequencia de processamento em cada transação, para otimizar o acesso aos dados.
- Os dados armazenados nos sistemas de banco de dados OLTP estão organizados em tabelas normalizadas para evitar a redundância dos dados e priorizar as operações de inserção, atualização e consulta dos dados.
- Os sistemas de bancos de dados OLTP são usados nas transações online de uma organização.
- Exemplos de bancos de dados OLTP: IBM Db2, Oracle, Microsoft SQL Server, PostgreSQL e MySQL

3.1.2. Bancos de Dados Analíticos (OLAP)

- Os sistemas de banco de dados analíticos (OLAP) são bastante eficientes no processamento de consultas, porque a estrutura de armazenamento permite a redundância dos dados para otimizar as operações de consulta.
- Os sistemas de banco de dados OLAP utilizam um subconjunto dos dados extraídos do banco de dados OLTP, os dados são extraídos periodicamente em um certo momento (Data e Horário de Corte), que pode ser D-1, ou de madrugada e no meio do dia.
- A determinação da Data e Horário de Corte, dependerá da prioridade de ter as consultas analíticas disponíveis para a equipe gerencial da empresa.

3.1.3. Problemas endereçados

- Sistemas de banco de dados OLTP e OLAP são bastante eficientes no processamento de transações e consultas analíticas, porque realizam diversas análises que otimizam estes tipos de processamento.
- No entanto precisam realizar um grande esforço em cada operação, e muitos sistemas que realizam operações simples de leitura e escrita com base no par chave e valor podem ser substituídos por bancos de dados NoSQL que são mais eficientes.



1. Imagem de sistemas de ERP e CRM que operam de forma online em uma organização, armazenando dados em bancos de dados transacionais (OLTP).
2. Processos de extração e transformação de dados (ETL), executados periodicamente, e armazenando dados consolidados em um banco de dados analítico (OLAP).
3. Processamento das consultas gerenciais em uma organização em um banco de dados analítico (OLAP), otimizados para processos de *Business Intelligence (BI)* e *Data Mining*.

3. Armazenamento e Processamento

3.1. Bancos de Dados (OLTP/OLAP)

HIPÓTESE 1: Usar um Único Sistema para Processamento OLTP e OLAP

- Os bancos de dados relacionais tradicionais, como o IBM Db2, Oracle e Microsoft SQL Server, possuem suporte para os modelos OLTP e OLAP em um único sistema de banco de dados, e usando um único tipo de organização de dados, denominado *row-store*. Entretanto eles não são eficientes para ambos os tipos de processamento.

Facebook_Friends		
Name	City	Age
Matt	Los Angeles	27
Dave	San Francisco	30
Tim	Oakland	33

Matt	Los Angeles	27	Dave	San Francisco	30	Tim	Oakland	33
------	-------------	----	------	---------------	----	-----	---------	----

Jen	Vancouver	30
-----	-----------	----

Matt	Los Angeles	27	Dave	San Francisco	30	Tim	Oakland	33	Jen	Vancouver	30
------	-------------	----	------	---------------	----	-----	---------	----	-----	-----------	----

Disk 1		
Name	City	Age
Matt	Los Angeles	27

Disk 2		
Name	City	Age
Dave	San Francisco	30

Disk 3		
Name	City	Age
Tim	Oakland	33

Row Oriented Database

<https://dataschool.com/data-modeling-101/row-vs-column-oriented-databases/>

3. Armazenamento e Processamento

3.1. Bancos de Dados (OLTP/OLAP)

HIPÓTESE 2: Usar uma Organização de Dados Separada para Processamento OLTP e OLAP

- SAP HANA ou Oracle TimesTen tem sistemas de banco de dados otimizados para processamento do tipo *columnar* em memória que oferece melhor desempenho para processamento OLAP. Entretanto, eles utilizam diferentes organizações de dados para o modelo transacional (*row-wise*) e analítico (*columnar*).

Name	City	Age
Matt	Los Angeles	27
Dave	San Francisco	30
Tim	Oakland	33

Matt	Dave	Tim	Los Angeles	San Francisco	Oakland	27	30	33
------	------	-----	-------------	---------------	---------	----	----	----

Jen	Vancouver	30
-----	-----------	----

Matt	Dave	Tim	Jen	Los Angeles	San Francisco	Oakland	Vancouver	27	30	33	30
------	------	-----	-----	-------------	---------------	---------	-----------	----	----	----	----

Disk 1		
Name		
Matt	Dave	Tim

Disk 2		
City		
Los Angeles	San Francisco	Oakland

Disk 3		
Age		
27	30	33

Column Oriented Database

<https://dataschool.com/data-modeling-101/row-vs-column-oriented-databases/>

3. Armazenamento e Processamento

3.2. Bancos de Dados (NoSQL)

3.2.1. Introdução aos Bancos de Dados NoSQL

- Os bancos de dados NoSQL se destacam porque podem processar um grande volume de dados.
- Este modelo de banco de dados também é conhecido com banco de dados do tipo chave-valor.
- Os objetos armazenados possuem um campo chave para a busca e um campo valor que pode armazenar qualquer tipo de informação.
- Bancos de dados NoSQL possuem facilidade de replicação de dados entre servidores.
 - (a) A réplica dos dados consiste na cópia de um objeto entre diversos nós de processamento.
 - (b) A atualização de um objeto consiste na atualização do objeto local e da atualização de suas réplicas.
 - (c) Não existe um controle de transação, e a atualização dos dados é realizada levando em consideração um *Timestamp Global* usado pelo *Cluster* de Servidores.
- Este modelo de banco de dados é muito usados por sistemas distribuídos que precisam de redundância no armazenamento dos dados, para garantir a segurança da informação e permitir o processamento das consulta de forma otimizada. Isto é, se M servidores possuem os mesmos objetos chave-valor armazenados, então a consulta aos dados pode ser distribuída por N processos independentes ($N \leq M$).
- Existem algumas variações de bancos de dados NoSQL:
 - (a) Chave-valor – armazenam qualquer tipo de dado no formato chave-valor.
 - (b) Documento – armazenam documentos usando uma chave de acesso.
 - (c) Colunar – armazenam uma lista de valores usando uma chave de acesso.
 - (d) Graph – armazenam qualquer tipo de dado no formato chave-valor em uma estrutura de grapho.

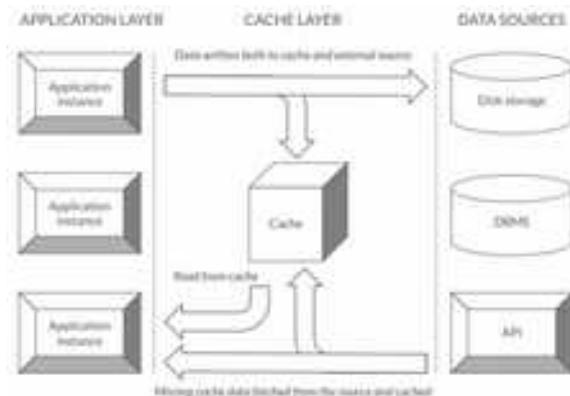
3. Armazenamento e Processamento

3.2. Bancos de Dados (NoSQL)

3.2.2. Aplicação dos Bancos de Dados NoSQL

- Os bancos de dados NoSQL são bastante usados para armazenar dados de sessão de usuários, que precisam estar disponível em memória para otimizar o acesso aos dados e reduzir o número de consultas ao banco de dados transacional.
- Aplicações que possuem um grande número de dispositivos e usuários, tais como (a) redes sociais; (b) sistemas de mensageria; (c) sistemas de telemetria e outros, usam banco de dados NoSQL para armazenar os dados de sessão dos usuários que são lidos e atualizados com muita frequência.
- Os dados armazenados em bancos de dados NoSQL ficam normalmente em memória.
- Os dados contidos no banco de dados NoSQL em memória são periodicamente armazenados em disco em um banco de dados transacional, como o IBM Db2, Oracle, Microsoft SQL Server, PostgreSQL e MySQL.
- Os bancos de dados NoSQL são usados com *cache* de dados dos bancos de dados transacionais, permitindo que um grande número de dispositivos e usuários acessem o sistema, e exigindo menos recurso de processamento do banco de dados transacional.

IMPORTANTE: Os bancos de dados transacionais (OLTP) suportam centenas (ou milhares) de operações por segundo, enquanto um banco de dados NoSQL em memória suporta milhões de operações por segundo usando um servidor bastante simples.



1. Imagem de aplicação com grande volume de dispositivos e usuários, como uma rede social, que efetua inicialmente uma consulta ao banco de dados NoSQL em memória.
2. Caso o dado não esteja disponível, ele é carregado do banco de dados OLTP, ou de um arquivo de dados externo, ou de um Web Service, para o banco de dados NoSQL em memória.
3. Caso o dado esteja disponível no banco de dados NoSQL, ele é usado pela aplicação e periodicamente a aplicação atualiza o seu valor no banco de dados OLTP ou no arquivo de dados externo.

3. Armazenamento e Processamento

3.3. BIG DATA (SPARK + HADOOP)

3.3.1. INTRODUÇÃO AO SISTEMA DE ARMAZENAMENTO DISTRIBUÍDO HADOOP

- O Hadoop é um projeto da Apache (hadoop.apache.org) em que o Yahoo! Contribuiu com 80% do desenvolvimento do núcleo do sistema.
- O HDFS é o componente do Hadoop responsável pelo armazenamento de arquivos distribuídos.
- O sistema de arquivos distribuídos HDFS usado no Hadoop, foi desenvolvido com o objetivo de otimizar o armazenamento de grandes volumes de dados de forma eficiente e com redundância, para possibilitar o acesso eficiente aos dados armazenados e garantir a disponibilidade dos dados mesmo que haja falha de algum dos nós.

3.3.2. RESPONSABILIDADE DE CADA NÓ

(a) NAME NODE

- Responsável pelo armazenamento e pela hierarquia de diretórios.
- Os arquivos e diretórios são representados no *Name Node* como inodes que registram os atributos, as permissões e a data de criação e modificação de cada arquivo.
- Os arquivos são divididos em blocos de 128 MB que são armazenados de forma distribuída e com replicação independente entre os diversos nós de dados do Hadoop. Se for necessário, o tamanho do bloco de armazenamento pode ser alterado para cada arquivo.

(b) DATA NODE

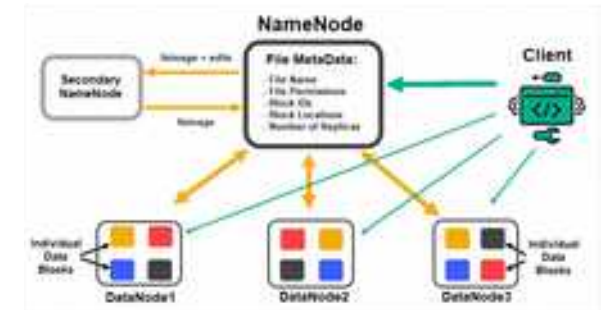
- O *DataNode* é responsável por armazenar os blocos de dados dos arquivos em um sistema Hadoop, e cada bloco de dados armazenado possui redundância e são replicados em diversos *Data Node* do sistema.
- Durante a inicialização é realizado um Handshake que verifica o identificador e a versão do software instalado em cada *Data Node*, e caso ocorra alguma diferença no registro do *Data Node*, ele é desligado do sistema (*shutdown*).
- Qualquer *DataNode* inicializado recentemente, e que ainda não possui um identificador, pode ser incluído em um sistema Hadoop.
- O *NameNode* não chama diretamente o *DataNode*, ele usa a resposta da mensagem de Heartbeat para receber informações sobre o *DataNode*.

LISTA DE COMANDOS ENVIADOS ATRAVÉS DO HEARTBEAT

(a) Replicação de blocos entre nós do sistema Hadoop; (b) Remoção das réplicas do bloco local; (c) Registro e desligamento (*shutdown*) dos nós; (d) Envio de relatório sobre os blocos armazenados.

PROCESSO DE LEITURA DOS ARQUIVOS

(a) Inicialmente o cliente solicita ao *Name Node* a localização do arquivo, e recebe a informação sobre o *Data Node* mais próximo e disponível para fornecer o arquivo; (b) Cada arquivo armazenado no sistema de arquivo HDFS possui dois documentos: um que contém os dados e outro com os metadados do arquivo que descreve o tipo do arquivo; (c) Cada *Name Node* possui uma lista completa dos arquivos armazenados no sistema Hadoop.



Estrutura do sistema de armazenamento distribuído Hadoop

3. Armazenamento e Processamento

3.3. BIG DATA (SPARK + HADOOP)

3.3.3. CARACTERÍSTICAS DO SISTEMA DE ARQUIVOS HDFS

(a) IMAGEM E JOURNAL

- Registros persistentes armazenados no disco, são registrados inicialmente em um arquivo de log que antecipa a gravação dos dados.
- Em caso de falha e reinicialização do nó de dados, é efetuado uma verificação dos registros de log com os dados armazenados, permitindo a recuperação do último estado estável do sistema.

(b) CHECKPOINT NODE

- O *Checkpoint Node* registra periodicamente pontos de verificação do sistema, para proteger os arquivos e os metadados armazenados.
- Periodicamente o *Checkpoint Node* combina o *checkpoint* atual do sistema com os registros do log para criar um novo *checkpoint* e reinicializar o registro dos nós.

(c) BACKUP NODE

- De forma similar ao *Checkpoint Node*, o *Backup Node* é capaz de registrar pontos de verificação do sistema Hadoop, além de manter uma imagem atualizada do sistema em memória.
- A função do *Backup Node* é servir como um *Name Node* em uma consulta, e armazena a lista de localização dos blocos e o metadado para atender situações de falhas dos *Name Nodes*.

(d) UPGRADES, FILE SYSTEM SNAPSHOTS

- Os *snapshots* criam uma imagem em disco do estado atual do sistema, permitindo a atualização dos softwares dos nós.

(e) HDFS CLIENT

- O cliente de acesso ao sistema de arquivo HDFS usa uma aplicação remota para acessar o sistema e efetuar as operações básicas.

(f) FILE I/O OPERATIONS

- No processo de leitura e escrita de arquivos no sistema do Hadoop, os arquivos armazenados nunca são apagados, isto permite que o sistema Hadoop implemente um estado que permite uma única operação de escrita com múltiplas operações de leitura.
- Quando um cliente solicita o acesso de escrita de um arquivo, nenhum outro cliente poderá escrever no arquivo.
- Durante o período em que o arquivo está aberto para escrita, um mecanismo de *Heartbeat* verifica o estado atual do cliente para manter o arquivo aberto.
- Após o fechamento do arquivo o acesso de escrita ao arquivo é liberado para os outros clientes.

3. Armazenamento e Processamento

3.3. BIG DATA (SPARK + HADOOP)

3.3.4. VANTAGENS DO SISTEMA DE ARQUIVOS HDFS

- O sistema de arquivos distribuídos HDFS apresentou excelentes resultados em termos de durabilidade dos dados, desempenho no processamento de grandes volumes de dados e na taxa de operações executadas por segundo (Throughput).

(a) DURABILIDADE DOS DADOS

- A replicação dos blocos de dados entre diversos nós é a forma mais robusta de armazenar os dados e evitar a perda de dados devido a ocorrência de falhas do sistema.
- No sistema de arquivos HDFS, a probabilidade de perda de dados no período de 1 ano é inferior a 0,005, considerando que 0,8% dos nós falham a cada mês.
- Existem também as falhas “correlacionadas” que são provocadas por defeitos em equipamentos que atendem a um conjunto de nós, como os defeitos nos switches de rede.
- No sistema de arquivos HDFS, mesmo na ocorrência destas falhas o sistema pode ser recuperado com a simples substituição do equipamento.

(b) NECESSIDADES COMUNS

- O sistema de armazenamento distribuído HDFS usado no Hadoop, possui também diversas outras aplicações. Cada nova aplicação que usa este sistema necessita de diferentes recursos para lidar com os dados armazenados.
- O HDFS não possui identificação de arquivos disponíveis para execução, uma sinalização bastante simples e comum aos sistemas Unix (POSIX). Porém, no HDFS todos os arquivos são armazenados e tratados como dados comuns. Algumas aplicações que usam o sistema de armazenamento HDFS precisam lidar também com arquivos executáveis.

3. Armazenamento e Processamento

3.3. BIG DATA (SPARK + HADOOP)

3.3.5. INTRODUÇÃO AO APACHE SPARK

- O Apache Spark é uma ferramenta para análise dos dados armazenados no Hadoop.
- O Apache Spark distribui uma tarefa entre diversos nós do Hadoop, para que o processamento dos dados seja executado usando os dados armazenados localmente em cada nó.
- O Hadoop MapReduce realiza operações de seleção dos dados (map) e consolidação da informação (reduce), e o Apache Spark é uma solução que amplia os recursos de processamento do MapReduce.

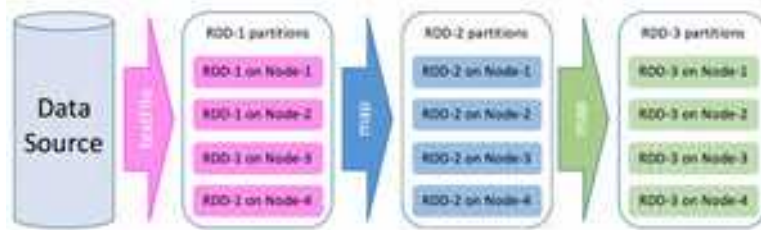
3.3.6. RESILIENT DISTRIBUTED DATASETS (RDD)

- Para processamento dos dados, o Apache Spark, organiza os dados armazenados em cada *Data Node* do Hadoop em uma estrutura denominada *Resilient Distributed Datasets* (RDD).
- As operações de mapeamento dos dados executadas pelas tarefas disparadas pelo Apache Spark, criam diferentes conjuntos de RDDs, que são denominados *Partitions*.
- Após uma operação de mapeamento, é criado uma nova *Partition*, formada por um subconjunto dos dados contidos nos RDDs da *Partition* original.

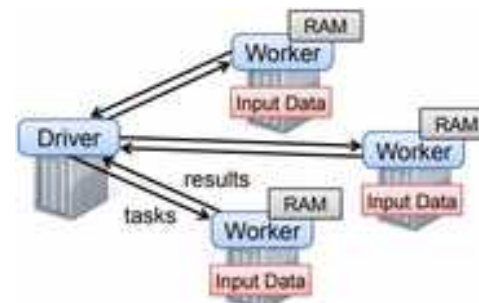
EXTENSÕES DO APACHE SPARK: API GraphX + API MLlib

(a) API GraphX: Biblioteca do Apache Spark, que permite o processamento otimizado dos dados armazenados nos nós usando uma estrutura de grapho.

(b) API MLlib: Biblioteca do Apache Spark, que permite o processamento otimizado dos dados armazenados nos nós, usando recursos de *Machine Learning*.



Criação das *Partitions* com os conjuntos de objetos selecionados em cada nó, reunidos em estruturas denominadas *Resilient Distributed Datasets* (RDD).



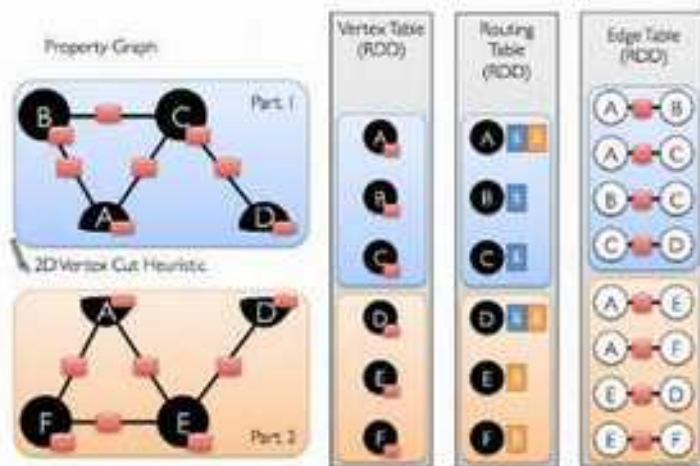
Distribuição das tarefas realizada pelo Apache Spark para processamento nos *Data Nodes*.

3. Armazenamento e Processamento

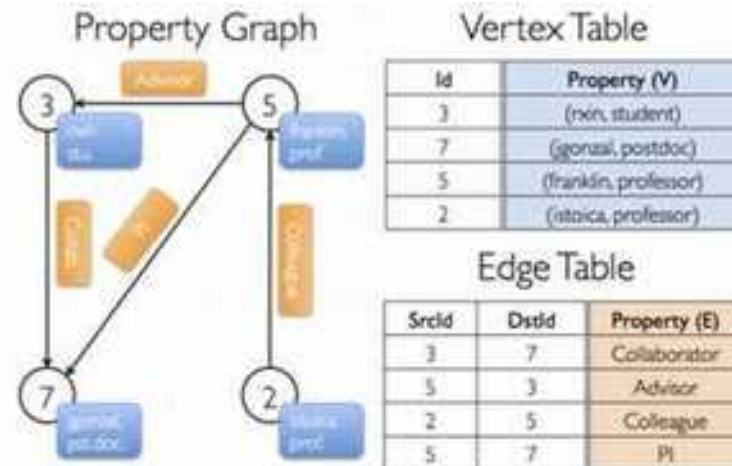
3.3. BIG DATA (SPARK + HADOOP)

3.3.7. API GRAPHX

- A biblioteca GraphX é um componente do Apache Spark que estende o conceito de *Resilient Distributed Datasets* (RDD), para facilitar o processamento de estruturas de dados representadas no formato de grafo.
- A GraphX permite a criação de grafos com atributos associados a cada nó e aresta do grafo.
- Para permitir o armazenamento de estruturas em formato de grafo de forma distribuída, a biblioteca GraphX usa uma estrutura de armazenamento, onde cada vértice mantém a relação de conectividade.
- Este processo de armazenamento, permite que a estrutura de dados em uso seja distribuída por diversas máquinas do cluster.



Representação otimizada do grafo, onde cada vértice mantém a relação de conectividade.



Grafo com vértices e arestas, contendo propriedades.

3. Armazenamento e Processamento

3.3. BIG DATA (SPARK + HADOOP)

3.3.8. API MLLIB

- A biblioteca Machine Learning Library (MLlib), para aprendizado de máquina, provê ferramentas para processamento de dados usando Apache Spark.
- A MLlib possui ferramentas para realização de operações usando álgebra linear e estatística, algoritmos de aprendizado de máquina, ferramentas para transformação e simplificação de dados, além de ferramentas para construção, avaliação e armazenamento de modelos de aprendizado de máquina.

ESTRUTURAS DE DADOS USADAS PELA MLLIB

(a) DataFrame: A biblioteca MLlib usa as estruturas de *Data Frame* sobre RDD, que estão definidas no Spark SQL como representação de *Data Sets* para *Machine Learning*, que podem armazenar estruturas de diferentes tipos complexos de dados, tais como, textos e elementos vetoriais.

(b) Transformador: Os algoritmos de transformação, podem converter um *Data Frame* de um formato para outro. Por exemplo, efetuar a transformação de um *Data Frame* de elementos vetoriais em um de preditores.

(c) Estimador: Algoritmos estimadores, são algoritmos que ajustam um *Data Frame* para realizar uma transformação. Por exemplo, um algoritmo de aprendizado de máquina pode ser usado como estimador, treinar usando os dados de um *Data Frame*, e gerar um modelo que poderá ser usado na classificação de dados.

(d) Pipeline: Um *Pipeline*, é uma sequência de operações com transformadores e estimadores que realizam o processamento automatizado dos dados usando a MLlib.

3. Armazenamento e Processamento

3.4. Web Services / Python + JBOSS EAS

3.4.1. SERVIDORES DE APLICAÇÃO

- Os servidores de aplicação surgiram para prover serviços de publicação de páginas HTML (*Hiper Text Markup Language*) no início da *World Wide Web* (WWW).

(a) Apache HTTP Server

- O primeiro servidor de aplicação usado por mim foi o servidor Apache HTTP Server (em 1993).

- O Apache HTTP Server provia suporte para serviços de publicação de páginas HTML e suporte para processamento CGI (*Common Gateway Interface*).

- A interface CGI permitiu o surgimento de diversas ferramentas de processamento de páginas HTML, e entre elas podemos destacar o PHP e o Adobe Coldfusion.

- Posteriormente a equipe de manutenção do Apache HTTP Server começou a introduzir diversos módulos no servidor de aplicação, tais como o módulo SSL, Proxy, Proxy Reverso, Monitoramento e etc.

(b) Microsoft Internet Information Server (IIS)

- O Microsoft Internet Information Server (IIS) foi incorporado inicialmente ao Microsoft Windows NT Server e ao Microsoft Windows NT Workstation.

- A partir do Microsoft Windows 2000, o Microsoft Internet Information Server (IIS) passou a ser incluído em todas as versões seguintes do Microsoft Windows.

- O Microsoft Internet Information Server (IIS) possui suporte para serviços de publicação de páginas HTML, CGI, ISAPI, ASP, ASP .Net, PHP, Adobe Coldfusion e outros.

(c) Apache TomCat

- O Apache TomCat é um servidor de aplicação desenvolvido 100% em Java, e usado por desenvolvedores Java para publicação de páginas HTML usando Servlets, JSP, JSF, e um subconjunto dos recursos da especificação J2EE.

- O Apache TomCat é um servidor de aplicação Java simples e NÃO implementa toda a especificação J2EE.

(d) JBOSS Enterprise Application Server (EAS)

- O JBOSS Enterprise Application Server (EAS) é um servidor de aplicação desenvolvido 100% em Java, que implementa a especificação J2EE.

3. Armazenamento e Processamento

3.5. OUTROS: DO-IT-YOURSELF (DIY)

(a) C/C++

- Linguagem de programação que surgiu na década de 1970, sendo muito usada no desenvolvimento de sistemas operacionais.
- Esta linguagem de programação, foi usada no desenvolvimento dos servidores Apache HTTP Server e Microsoft Internet Information Server (IIS).
- A linguagem de programação C/C++ também foi usada no desenvolvimento do PHP, Adobe Coldfusion, Python, Java VM e .Net.

IMPORTANTE: PARA CONSTRUIR UMA NOVA ARQUITETURA DE COMUNICAÇÃO EM QUALQUER NÍVEL DEVEMOS USAR C/C++

(b) Python

- A linguagem Python é uma linguagem de programação interpretada, que surgiu também na década de 1970, é uma linguagem de programação não tipada, sendo muito usada para desenvolvimento de pequenos serviços (*scripts*).
- Atualmente, a linguagem Python ganhou muita força ao ser adotada pelo pessoal que trabalha com Inteligência Computacional.
- Esta linguagem de programação permite a incorporação de bibliotecas desenvolvidas em C/C++.
- Existem diversas bibliotecas desenvolvidas em C/C++ que foram empacotadas para serem usadas em aplicações Python, como Keras e Tensorflow.

IMPORTANTE: EXISTEM BIBLIOTECAS PARA CRIAÇÃO DE SERVIÇOS HTTP/HTTPS USANDO PYTHON

(c) Java

- A linguagem Java, surgiu na década de 1990, sendo uma linguagem de programação compilada, cujo código binário é executado pela máquina virtual Java VM.
- Existem máquinas virtuais para diversas plataformas: Linux, FreeBSD, Oracle Solaris, Microsoft Windows, MAC OS, Mainframe, Celulares, Televisores, Smartphones, Tablets, Microcontroladores, e etc.
- Pela facilidade de portabilidade das aplicações Java, muitos servidores de aplicação foram escritos 100% em Java.

IMPORTANTE: A LINGUAGEM DE PROGRAMAÇÃO JAVA POSSUI TODOS OS RECURSOS PARA DESENVOLVIMENTO DE UMA NOVA ARQUITETURA DE COMUNICAÇÃO, EM QUALQUER NÍVEL, SOBRE UMA MÁQUINA JAVA VM QUE PODE SER EXECUTADA EM QUALQUER PLATAFORMA.

(d) Plataforma .Net

- A Plataforma .Net, surgiu no início da década 2000, sendo uma linguagem de programação compilada, cujo código binário é executado pela máquina virtual .Net.
- Existem máquinas virtuais .Net para diversas plataformas devido ao trabalho constante da equipe do projeto Mono.

IMPORTANTE: A PLATAFORMA .NET POSSUI TODOS OS RECURSOS PARA DESENVOLVIMENTO DE UMA NOVA ARQUITETURA DE COMUNICAÇÃO, EM QUALQUER NÍVEL, SOBRE UMA MÁQUINA VIRTUAL .NET QUE PODE SER EXECUTADA EM QUALQUER PLATAFORMA.

3. Armazenamento e Processamento

3.5. OUTROS: DO-IT-YOURSELF (DIY)

(e) OUTROS: DO-IT-YOURSELF

- O surgimento de novas tecnologias muitas vezes geram demandas por novas soluções de comunicação.
- Para estes casos, a única forma é desenvolver uma nova estrutura de comunicação é realizando o desenvolvendo em todas as camadas – Do-It-Yourself (DIY).

EXEMPLO DO SERVIDOR DE BANCO DE DADOS DISTRIBUÍDO: HORUS IMAGE SERVER v2.0

- Para elaboração dos experimentos da minha pesquisa de Pós-Doutoramento, foi desenvolvido um servidor de banco de dados distribuído que coordena o processo de armazenamento, classificação, pesquisa e processamento de imagens armazenadas nas unidades CSDs (*computational storage devices*).
- Soluções de geoprocessamento acessam um grandes volume de dados compostos por múltiplos arquivos de imagens de grandes dimensões, com mais de 400 MB, e que precisam ser frequentemente processadas para conversão de formatos, alteração do sistema de coordenadas e projeção, criação de cache com diferentes níveis de precisão, comparação e identificação de mudanças, reconhecimento de padrões e etc.
- Deste modo, distribuindo o processamento entre as unidades de armazenamento, obtemos maior agilidade nas execução destas tarefas. Além de efetuar o treinamento dos modelos de Redes Neurais Profundas, *Deep Neural Networks* (DNN), e a classificação das imagens, usando um processo em *background* nas unidades CSDs.
- Os resultados iniciais apresentaram um ganho em *speedup* de até 2,41 (241%), na execução do sistema de banco de dados distribuído, Horus Image Server v2.0, em um ambiente simulado com 4 unidades de armazenamento do tipo CSDs, com capacidade de processamento embutido, in-situ processing, quando comparado com o ambiente de referência, que utiliza 1 unidade de armazenamento do tipo SSD, tradicional e sem capacidade de processamento embutido.



Unidade de armazenamento Newport CSD com 16 TB e com capacidade de processamento embutido (*in-situ processing*).



Servidor x86 de 64 bits com 24 unidades de armazenamento Newport CSD de 16 TB, totalizando 1654 TB de armazenamento.

4. Recursos de Análise dos Dados

4.1. Sistemas Especialistas

- Os sistemas especialistas são construídos a partir da experiência adquirida por profissionais especializados em uma área.
- Estes sistemas são construídos a partir de uma base de conhecimento que é convertida em regras de negócio.

4.1.1. BASE DE CONHECIMENTO

- Representam a experiência do especialista em uma área.

4.1.2. ARVORE DE DECISÃO

- Representa a relação entre a experiência do especialista e a ocorrência de um evento.

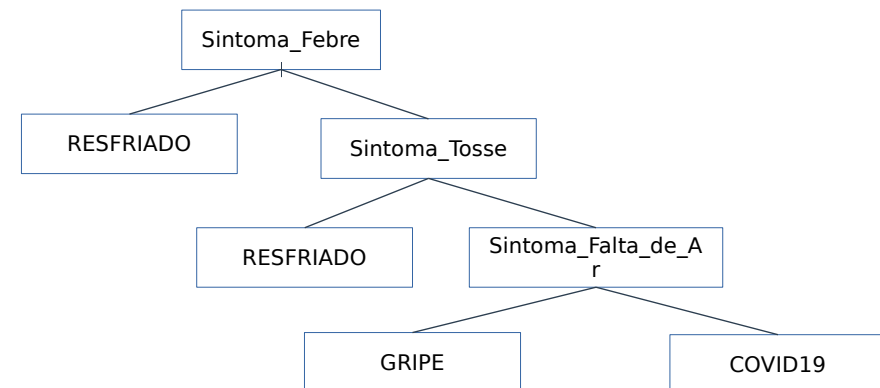
4.1.3. REGRAS DE NEGÓCIO

- Representa o conjunto de regras que processam os dados de acordo com a experiência do especialista.

EXEMPLO 1: SISTEMA ESPECIALISTA DE DIAGNOSTICO DE COVID-19

SINTOMAS	COVID-19 <small>(Os sintomas são de início e súbito)</small>	RESFRIADO <small>(Os sintomas são de início gradual e prolongado)</small>	GRIPE <small>(Os sintomas são de início súbito)</small>
Febre	Comum	Raro	Comum
Falta de ar	Às vezes	Raro	Raro
Tosse	Comum <small>(Característica comum)</small>	Leve	Comum <small>(Característica comum)</small>
Espirros	Raro	Comum	Raro
Dores no corpo e mal-estar	Raro	Comum	Raro
Coriza ou nariz entupido	Raro	Comum	Às vezes
Dor de garganta	Às vezes	Comum	Às vezes
Diarreia	Raro	Raro	Às vezes <small>(Não é comum)</small>
Dor de cabeça	Às vezes	Raro	Comum
Cansaço	Às vezes	Às vezes	Comum

BASE DO CONHECIMENTO do sistema especialista de diagnostico COVID-19



ARVORE DE DECISÃO do sistema especialista de diagnostico COVID-19

SE Sintoma_Febre = True E Sintoma_Tosse = True ENTÃO:
 SE Sintoma_Falta_de_Ar = True ENTÃO: Diagnostico = 'COVID19';
 SENÃO: Diagnostico = 'GRIPE';
 SENÃO: Diagnostico = 'RESFRIADO';

REGRAS DE NEGÓCIO do sistema especialista de diagnostico COVID-19

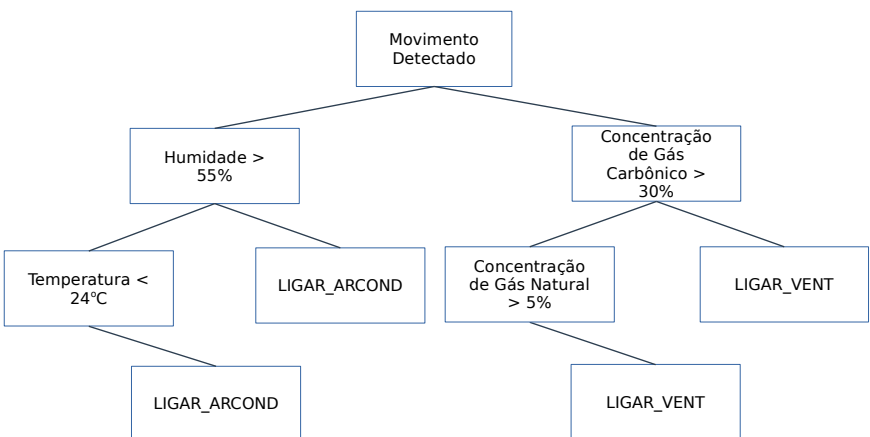
4. Recursos de Análise dos Dados

4.1. Sistemas Especialistas

EXEMPLO 2: SISTEMA ESPECIALISTA PARA RESFRIGERAÇÃO E VENTILAÇÃO DE 'KITNETES' DE ATÉ 28 M²

Timestamp	Equipamento	Luz	Movimento Detectado	Nível de Fumaça	Concentração de Gás Carbônico	Concentração de Gás Natural	Humidade	Temperatura
1594512094.39	h8-27-ep-bf-9d-51	falso	falso	0.0204	0.0090	0.0077	51.0000	22.7000
1594512094.74	00-0f-00-70-91-0a	falso	falso	0.0133	0.0028	0.0051	78.0000	19.7000
1594512098.07	h8-27-ep-bf-9d-51	falso	falso	0.0205	0.0090	0.0077	50.9000	22.8000
1594512098.59	1c-bf-cc-15-ec-4d	true	true	0.0188	0.4300	0.0070	78.8000	27.0000
1594512101.76	h8-27-ep-bf-9d-51	falso	falso	0.0204	0.0090	0.0077	50.9000	22.8000
1594512104.47	1c-bf-cc-15-ec-4d	true	true	0.0188	0.4500	0.0070	77.8000	27.0000
1594512105.45	h8-27-ep-bf-9d-51	falso	falso	0.0205	0.0090	0.0077	50.9000	22.8000
1594512106.87	00-0f-00-70-91-0a	falso	falso	0.0138	0.0029	0.0052	78.0000	19.7000
1594512108.28	1c-bf-cc-15-ec-4d	true	true	0.0100	0.1900	0.0070	77.9000	32.0000
1594512109.14	h8-27-ep-bf-9d-51	falso	falso	0.0205	0.0090	0.0077	50.9000	22.8000
1594512112.80	h8-27-ep-bf-9d-51	falso	falso	0.0204	0.0090	0.0077	50.9000	22.8000
1594512115.29	1c-bf-cc-15-ec-4d	true	true	0.0188	0.1700	0.0070	78.0000	27.0000
1594512116.50	h8-27-ep-bf-9d-51	falso	falso	0.0205	0.0090	0.0077	50.9000	22.8000
1594512118.10	1c-bf-cc-15-ec-4d	true	true	0.0188	0.0045	0.0071	78.0000	27.0000
1594512120.18	h8-27-ep-bf-9d-51	falso	falso	0.0204	0.0090	0.0077	50.9000	22.8000
1594512122.79	00-0f-00-70-91-0a	falso	falso	0.0135	0.0029	0.0052	78.8000	19.7000
1594512123.87	h8-27-ep-bf-9d-51	falso	falso	0.0205	0.0090	0.0077	50.9000	22.8000
1594512127.16	h8-27-ep-bf-9d-51	falso	falso	0.0204	0.0090	0.0077	50.9000	22.8000
1594512129.37	1c-bf-cc-15-ec-4d	true	true	0.0187	0.0044	0.0071	77.9000	27.0000
1594512131.25	h8-27-ep-bf-9d-51	falso	falso	0.0204	0.0090	0.0077	50.9000	22.8000
Mínimo:				0.0133	0.0028	0.0051	50.9000	19.7000
Máximo:				0.0100	0.1700	0.3400	78.0000	32.0000
Média:				0.0488	0.1026	0.0237	62.7150	23.7486

BASE DO CONHECIMENTO do sistema especialista para ventilação de ambientes



ARVORE DE DECISÃO do sistema especialista de diagnostico COVID-19

SE Movimento_Detectado = True ENTÃO:
 SE Concentracao_de_Gás_Carbonico > 30% ENTÃO: Ação = 'LIGAR_VENT'
 SENÃO:
 SE Concentracao_de_Gás_Natural > 30% ENTÃO: Ação = 'LIGAR_VENT'
 FIM SE
 SENÃO:
 SE Humidade > 55% ENTÃO: Ação = 'LIGAR_ARCOND'
 SENÃO:
 SE Temperatura < 24 ENTÃO: Ação = 'LIGAR_ARCOND'
 FIM SE
 FIM SE

REGRAS DE NEGÓCIO do sistema especialista de diagnostico COVID-19

4. Recursos de Análise dos Dados

4.2. Inteligência Computacional

4.2.1. CONCEITOS DE INTELIGÊNCIA COMPUTACIONAL

- Podemos definir Inteligência Computacional como a capacidade de aprendizado e classificação de informação com base na avaliação de resultados anteriores.

- A Inteligência Computacional é usada em situações onde não existe uma expressão matemática direta para resolver um problema, sendo necessário construir um modelo computacional com base nos resultados obtidos do processamento de um subconjunto dos dados.

IMPORTANTE:

(a) Sempre que for possível ver inteiramente os dados, será possível encontrar uma função matemática capaz de representá-los.

(b) Usamos Inteligência Computacional, quando não temos acesso a todos os dados, mas somente a um subconjunto deles. Este subconjunto de dados será usado como base de treinamento e teste do modelo de Inteligência Computacional.



MAQUINA DE CONTAGEM DE MOEDAS

1. UMA EMPRESA DECIDE FABRICAR MÁQUINAS DE CONTAGEM DE MOEDAS.
2. A MÁQUINA RECEBE EM UM CONJUNTO DE MOEDAS DE DIFERENTES VALORES: R\$ 1,00; R\$ 0,50; R\$ 0,25; R\$ 0,10; e R\$ 0,05.
3. E A MÁQUINA SEPARA AS MOEDAS PELO SEU VALOR.
4. A EMPRESA PRETENDE SELECIONAR AS MOEDAS USANDO A DIMENSÃO E O PESO.
5. A PARTIR DE UM CONJUNTO DE EXPERIMENTOS DEFINE UMA TABELA DE VALORES MÍNIMOS E MÁXIMOS PARA A DIMENSÃO E O PESO DE CADA MOEDA.
6. COM BASE NA TABELA DE DIMENSÕES E PESO, A MÁQUINA SEPARA AS MOEDAS.

CONCLUSÃO: A MÁQUINA DE CONTAGEM DE MOEDAS NÃO UTILIZA UM SISTEMA DE INTELIGÊNCIA COMPUTACIONAL, PORQUE OS VALORES DE DIMENSÕES E PESO DE CADA TIPO DE MOEDA SEGUE UMA FORMULAÇÃO MATEMÁTICA.

Exemplo de sistema que NÃO utiliza Inteligência Computacional

ANÁLISE DE CONCESSÃO DE CRÉDITO PARA PESSOA FÍSICA

1. UM BANCO CRIA UM SISTEMA AUTOMATIZADO PARA CONCESSÃO DE CRÉDITO.
2. O BANCO POSSUI UM SISTEMA DE BANCO DE DADOS QUE ARMAZENA DIVERSOS DADOS DE SEUS CLIENTES.
 - (a) RENDA FAMILIAR;
 - (b) NÍVEL DE ENDIVIDAMENTO;
 - (c) CADASTRO NEGATIVO;
 - (d) PATRIMÔNIO;
 - (e) INVESTIMENTOS;
 - (f) SALDO EM CONTA CORRENTE;
3. O SISTEMA DE CONCESSÃO DE CRÉDITO UTILIZA UMA FUNÇÃO CONSTRUÍDA A PARTIR DE UM SUBCONJUNTO DAS INFORMAÇÕES CONTIDAS NO BANCO DE DADOS PARA CLASSIFICAR OS CLIENTES:
 - (a) CONCEDER_CREDITO = 'Sim';
 - (b) CONCEDER_CREDITO = 'Não';
4. PERIODICAMENTE, O BANCO AJUSTA A FUNÇÃO A PARTIR DE UM NOVO SUBCONJUNTO DE INFORMAÇÕES CONTIDAS NO BANCO DE DADOS.

CONCLUSÃO: O SISTEMA DE CONCESSÃO DE CRÉDITO PARA PESSOA FÍSICA É UM SISTEMA DE INTELIGÊNCIA COMPUTACIONAL, PORQUE NÃO EXISTE UMA FÓRMULA MATEMÁTICA CAPAZ DE PREVER OS VALORES FUTUROS, E PERIODICAMENTE O BANCO PRECISA AJUSTAR A FUNÇÃO MATEMÁTICA EM FUNÇÃO DE MUDANÇAS EM CENÁRIOS ECONÔMICOS.

Exemplo de sistema que utiliza Inteligência Computacional

4. Recursos de Análise dos Dados

4.2. Inteligência Computacional

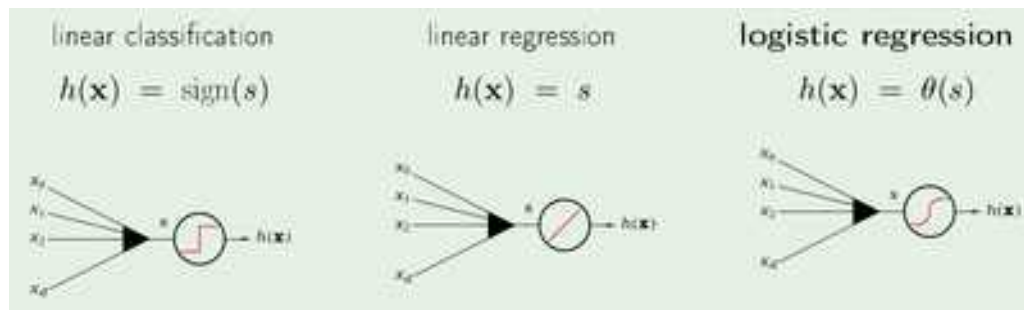
4.2.2. Linear Model

- Os modelos lineares de aprendizado de máquina (*linear models*), podem ser classificados em três grupos:

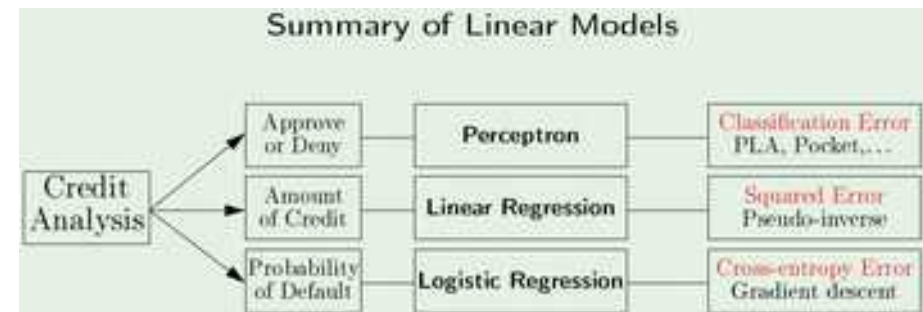
(a) Perceptron: O modelo *Perceptron* é usado em situações onde precisamos classificar diretamente um objeto. Por exemplo, podemos usar um modelo perceptron para aprovar ou negar crédito a uma pessoa.

(b) Linear Regression: O modelo de regressão linear, é usado em situações onde precisamos encontrar um valor apropriado em função de dados históricos. Por exemplo, podemos usar um modelo de regressão linear para determinar qual o limite de crédito a ser concedido a uma pessoa.

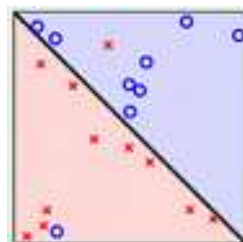
(c) Logistic Regression: O modelo de regressão logística, é usado em situações onde precisamos encontrar um valor capaz de fornecer uma ideia de probabilidade de ocorrência de um evento. Por exemplo, podemos usar um modelo de regressão logística para determinar se uma determinada pessoa pode ou não ficar inadimplente após concedermos um empréstimo.



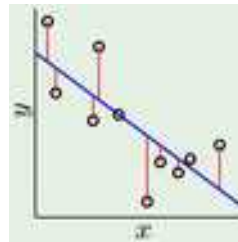
Exemplo de sistema que NÃO utiliza Inteligência Computacional



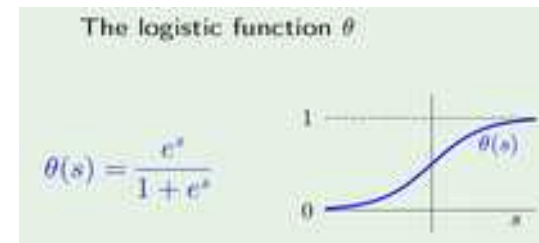
Exemplo de sistema que NÃO utiliza Inteligência Computacional



(a) Perceptron: Função linear que separa os dados seguindo uma classificação.



(b) Linear Regression: Função linear que se aproxima aos valores dos dados.



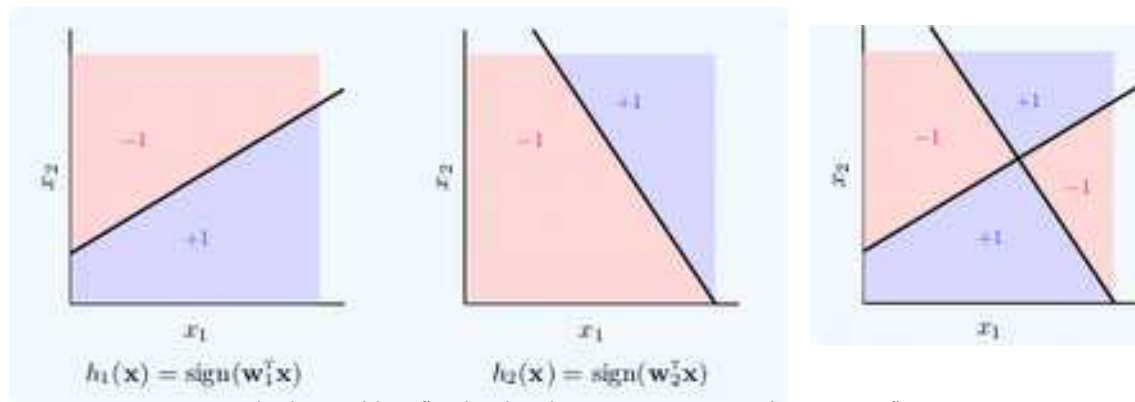
(c) Logistic Regression: Função linear que retorna um valor capaz de fornecer uma ideia de probabilidade de ocorrência de um evento.

4. Recursos de Análise dos Dados

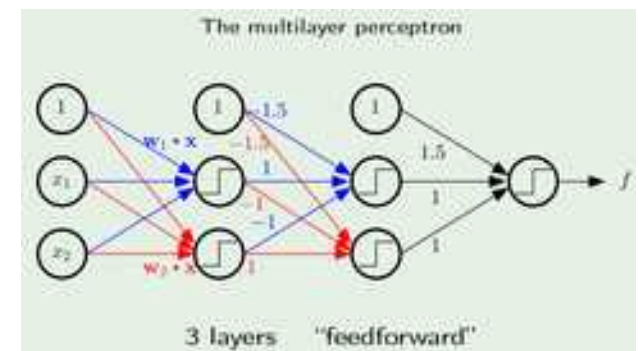
4.2. Inteligência Computacional

4.2.3. Multi-layer Perceptron (MLP)

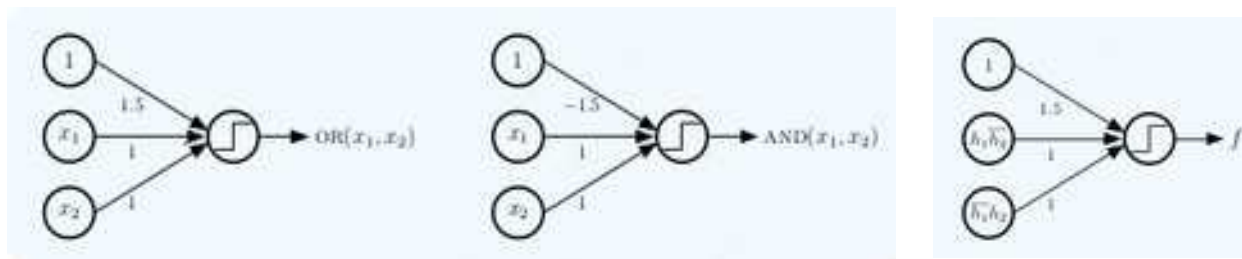
- A utilização de 1 (um) único *Perceptron*, muitas vezes não é suficiente para classificar de forma eficiente um conjunto de dados, e nestes casos usamos um *Multi-layer Perceptron* (MLP).
- O *Multi-layer Perceptron*, realiza a combinação entre 2 (dois) ou mais *Perceptrons*.
- Alguns exemplos de combinações simples de *Perceptrons*: NOT, XOR, OR, e AND.



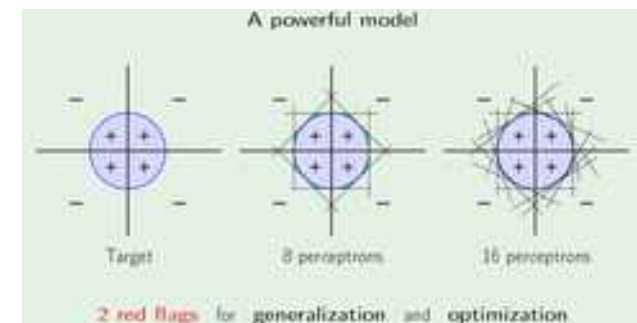
Exemplo de combinação simples de *Perceptrons* usando a operação XOR.



Exemplo de Multi-layer Perceptron com 3 camadas de operações.



Exemplo de combinação de *Perceptrons* usando as operações OR e AND, e operações mais complexas.



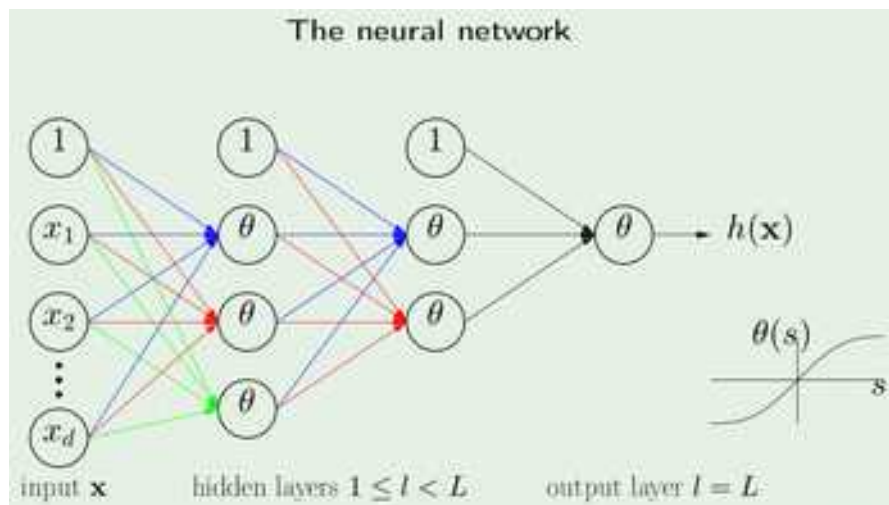
Diferença na classificação dos dados em um *Multi-layer Perceptron* com 8 e 16 *Perceptrons*.

4. Recursos de Análise dos Dados

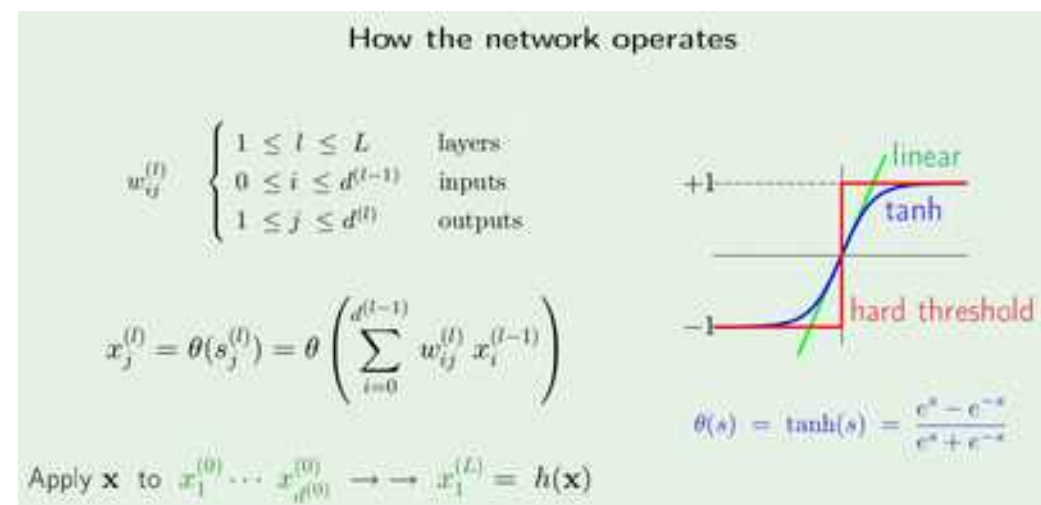
4.2. Inteligência Computacional

4.2.4. Neural Networks (NN)

- Os modelos de Redes Neurais são recursos mais poderosos que o modelo *Multi-layer Perceptron*, porque cada nó de uma rede neural representa uma função de transformação capaz de se ajustar de forma muito precisa aos dados.
- Os algoritmos de treinamento de redes neurais efetuam várias iterações procurando ajustar cada vês mais os parâmetros aos dados de entrada. Esta operação é muito demorada porque é necessário recalculr todos os parâmetros em cada iteração.



Modelo de Rede Neural com 1 camada de entrada, 2 camadas ocultas e uma camada de saída.



Exemplo de processo de treinamento da Rede Neural.

4. Recursos de Análise dos Dados

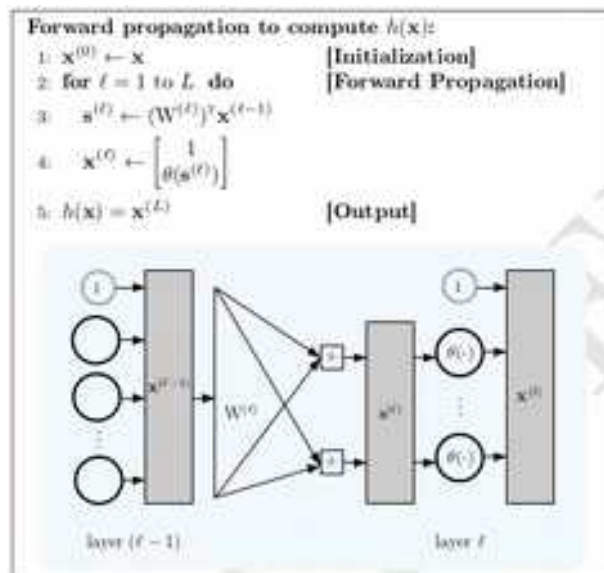
4.2. Inteligência Computacional

4.2.4. Neural Networks (NN)

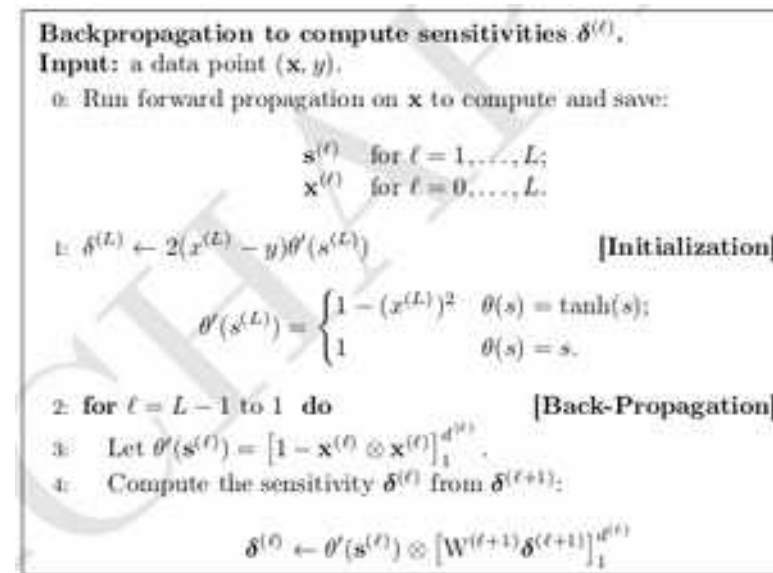
I. Algoritmos de Treinamento de Redes Neurais

- Existem diversos algoritmos de treinamento de Redes Neurais, sendo alguns dos principais:

- (a) Forward Propagation; e
- (b) Backpropagation;



Exemplo de algoritmo de treinamento de Redes Neurais, que utiliza propagação dos dados para frente.



Exemplo de algoritmo de treinamento de Redes Neurais, que utiliza além da propagação dos dados para frente, a correção dos parâmetros com processamento retroativo.

4. Recursos de Análise dos Dados

4.2. Inteligência Computacional

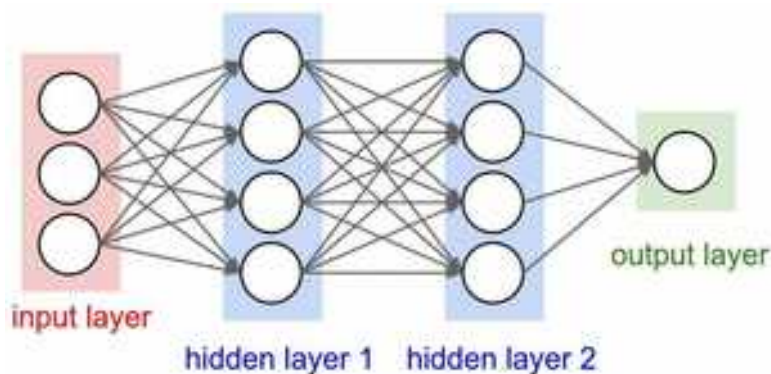
4.2.5. Redes Neurais Profundas (*Deep Neural Networks* - DNN)

- Redes neurais profundas (*Deep Neural Networks* - DNN), são modelos de redes neurais fortemente conectadas que apresentam uma camada de entrada, duas ou mais camadas ocultas e uma camada de saída.

4.2.6. Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN)

- Redes neurais convolucionais (*Convolutional Neural Networks* - CNN), são modelos de redes neurais profundas que são largamente utilizadas na classificação de imagens e áudios.

- Este modelo de rede neural procura separar em cada camada as características que definem as imagens, permitindo a identificação dos dados.



Exemplo de Rede Neural Profunda (DNN)



Exemplo de Rede Neural Convolucional (CNN)

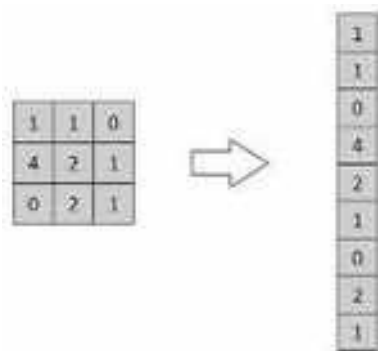
4. Recursos de Análise dos Dados

4.2. Inteligência Computacional

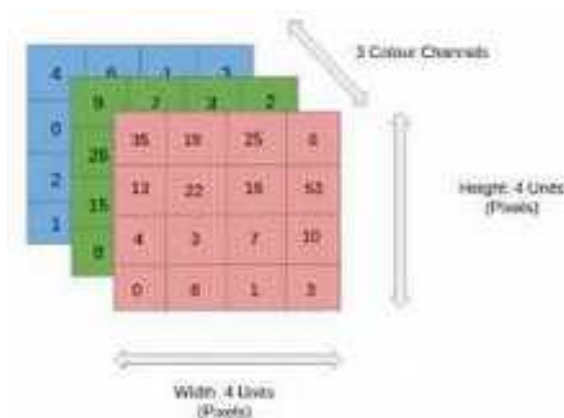
4.2.6. Redes Neurais Convolucionais (*Convolutional Neural Networks - CNN*)

(a) Conceitos Básicos

- Ao analisarmos imagens de 256 cores usando redes neurais, podemos arrumar os elementos da imagem em um vetor.
- Desta forma, uma imagem de 28 x 28 pixels e 256 cores, seria representada por um vetor com 784 bytes, e utilizaríamos na camada inicial da rede neural 784 entradas.
- Quando estudamos imagens com maior quantidade de cores, como imagens RGB, o tamanho do vetor necessário para representar a imagem é de 2.352 bytes.
- Além disso, ao representar uma imagem por um vetor de dados, perdemos as características associadas a cada profundidade de cor, gerando um modelo com menor taxa de assertividade.
- Desta forma, sugerimos as redes neurais convolucionais, que separam os dados de acordo com a profundidade de cor aplicada as imagens, assegurando um melhor desempenho na classificação das imagens.



Organização dos pixels da imagem em vetor.



Organização dos pixels da imagem por canal de cor.

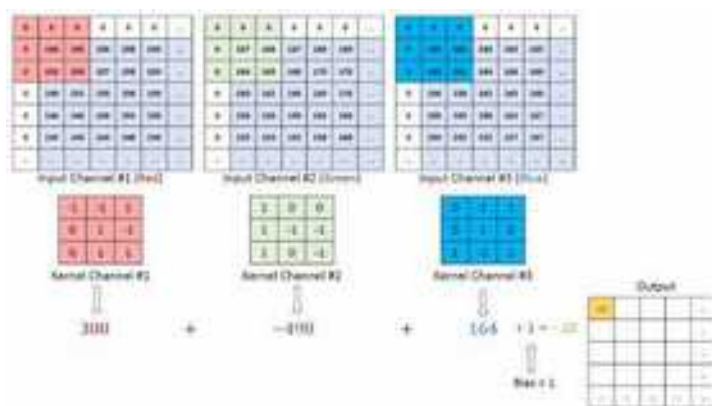
4. Recursos de Análise dos Dados

4.2. Inteligência Computacional

4.2.6. Redes Neurais Convolucionais (*Convolutional Neural Networks - CNN*)

(a) Camadas Convolucionais (*Kernel*)

- Cada camada convolucional do modelo de redes neurais implementa uma função de filtro denominada *Kernel*, que procura extrair os detalhes da imagem, e reduzir a quantidade de informações para um número específico de elementos.
- O filtro é aplicado em cada canal de cor, e o resultado obtido da camada de convolução é uma matriz com um número reduzido de elementos.
- No processo de convolução, as operações mais comuns aplicadas pelos filtros, são operações de identificação de valores médios ou máximos da camada de entrada.
- A análise das imagens para construção de um modelo de redes neurais convolucionais (CNN), é um processo que demanda uma grande capacidade de processamento, que não existe nos microcontroladores atuais.
- Este processo é extremamente demorado, e a taxa de erro do modelo reduz com o aumento no número de execuções do algoritmo, processo denominado Épocas de treinamento do algoritmo (*Epoch*).



Processo de filtro aplicado pela camada de convolução.

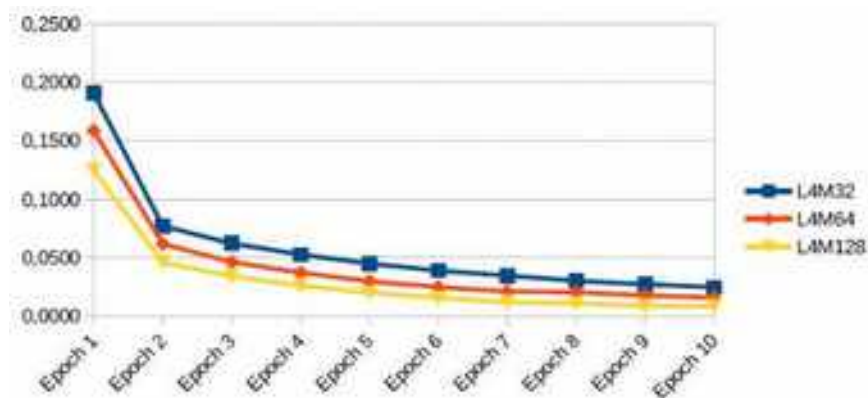


Gráfico de redução das taxas de erro, obtido usando um modelo com 4 camadas de convolução.

4. Recursos de Análise dos Dados

4.2. Inteligência Computacional

4.2.7. REDES NEURAIIS PARA MICROCONTROLADORES (TinyML)

- As redes neurais convolucionais usadas na classificação de elementos nas imagens ocupam grande espaço em memória e para usar estes modelos em microcontroladores é necessário efetuar a otimização do modelo de redes neurais.

- O processo de otimização das Redes Neurais Profundas (DNN) para permitir o seu uso em microcontroladores é denominado *Tiny Machine Learning* (TinyML), e envolve algumas etapas:

ETAPA 1: Identificação dos termos do modelo de Redes Neurais (RN) que possui maior importancia na classificação dos dados, processo denominado de identificação e aprimoramento (*knowledge distillation*).

ETAPA 2: Poda do modelo (prunning), isto é, a eliminação dos termos de menor importância na classificação dos dados.

ETAPA 3: Normalização dos parâmetros (*quantization*), permitindo a representação dos termos por valores inteiros de 8, 16 ou 32 bits. A utilização de valores inteiros no lugar dos valores de ponto flutuante, otimiza os cálculos durante o processo da classificação dos dados.

ETAPA 4: Agrupamos dos padrões de bits iguais (*encoding*), reduzindo o tamanho do modelo representado em memória. Deste modo, padrões de bits iguais são representados por seu código interno, obtido da tabela de padrões de bits previamente identificados no modelo.

ETAPA 5: Descrição do modelo em uma linguagem de alto nível (*compilation*) como C/C++, Java ou Python, para permitir a inserção do código da aplicação que irá ser executada no microcontrolador.

4. Recursos de Análise dos Dados

4.2. Inteligência Computacional

4.2.8. PROCESSOS DE OTIMIZAÇÃO DE REDES NEURAIS (TinyML)

- Descreveremos a seguir o processo de otimização de Redes Neurais Profundas (DNN), detalhando as 5 (cinco) etapas deste processo: *knowledge distillation*, *prunning*, *quantization*, *encoding* e *compilation*.

(a) APRIMORAMENTO DA REDE NEURAL (*Knowledge Distillation*)

- Após o treinamento de uma rede neural, seus parâmetros são calculados, e para o aprimoramento do modelo da rede neural, cada termo é analisado para verificar a contribuição dele no resultado da classificação de uma base de teste.

- Caso o termo não contribua de forma significativa no resultado final, ele é marcado como um termo possível de ser removido do modelo.

- A remoção de termos do modelo de redes neurais que não influenciam muito no resultado final da classificação de uma base de teste reduz a quantidade de termos calculados e otimiza o processamento do modelo.

(b) SIMPLIFICAÇÃO DO MODELO (*Prunning*)

- Após a fase de identificação dos termos que podem ser removidos do modelo de redes neurais, *knowledge distillation*, efetuamos a remoção destes termos, e geramos um novo modelo simplificado (*prunning*).

- O modelo simplificado, possui uma quantidade menor de termos que precisam ser calculados, e otimiza o processamento do modelo de redes neurais.

(c) NORMALIZAÇÃO DOS PARÂMETROS (*Quantization*)

- O processo de normalização dos parâmetros, faz com que os valores dos parâmetros sejam convertidos em valores inteiros de 8, 16 ou 32 bits, com o objetivo de otimizar o cálculo usando números inteiros no lugar do cálculo com valores de ponto flutuante.

NOTA: (a) Em uma Placa Arduino UNO R3, que usa o processador ATmega328P, uma operação de multiplicação usando registradores com números inteiros de 8 bits, e terminando em um valor inteiro de 16 bits (*Unsignet Integer Multiply* - MUL), consome 1 ciclo de processamento. (b) Enquanto que uma operação de multiplicação usando registradores com números em ponto flutuante de 8 bits, e terminando em um valor em ponto flutuante de 16 bits (*Fractional Unsignet Integer Multiply* - FMUL), consome 2 ciclos de processamento.

(d) COMPACTAÇÃO DO MODELO (*Encoding*)

- O processo de compactação do modelo de redes neurais (*encoding*), procura identificar a ocorrência de sequências de bits iguais no vetor de parâmetros do modelo, para substituir o padrão de bits por seu código interno, obtido da tabela de padrões de bits previamente identificados no modelo, reduzindo o tamanho dos dados armazenados.

(e) CONVERSÃO DO MODELO (*Compilation*)

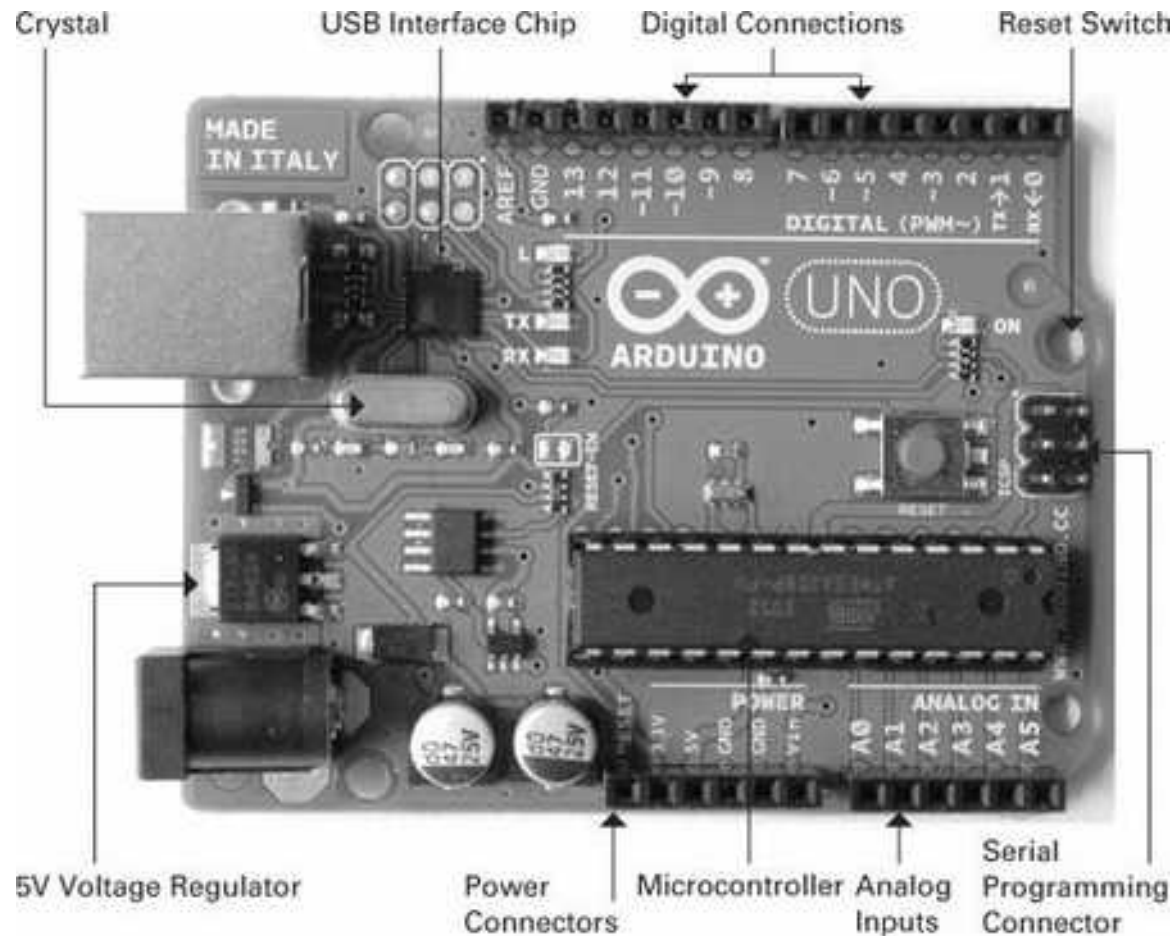
- O processo de conversão do modelo de redes neurais (*compiling*), converte o modelo de redes neurais em uma linguagem de alto nível como C/C++, Java ou Python, com o objetivo de permitir a inserção do modelo no código da aplicação que irá executar no microcontrolador.

5. Microcontroladores

5.1. Arduino UNO R3 - ATmega328P

5.1.1. Placa Arduino UNO R3

- A Placa Arduino UNO R3 com microprocessador ATmega328P de 8 bits, 16 MHz, 32 KB de memória Flash, 1 KB de memória EEPROM e 2 KB de memória RAM.



5. Microcontroladores

5.2. ESP32 com OLED - Xtensa LX6

5.2.1. Placa WiFi LoRa 868 MHz ESP32 com OLED

- A placa ESP32 com OLED, utiliza o microprocessador Tensilica Xtensa LX6 de 32 bits, 2 núcleos, 240 MHz, 4 MB de memória Flash interna, 1 KB de memória eFuse, 16 KB de memória SRAM temporária, 520 KB de memória SRAM para processamento, e com suporte para até 16 MB de memória Flash externa.



5. Microcontroladores

5.3. Outros Microcontroladores

5.3.1. Raspberry Pi 4

- Atualmente, muito investimento tem sido aplicado na implementação de soluções de Inteligência Computacional.
- A utilização das placas Raspberry Pi 4 e NVIDIA Jetson Nano, favorecem a implementação de soluções de computação na borda, *Edge Computing*, possibilitando que parte do processo de classificação dos dados seja realizado nos equipamentos remotos e depois consolidado, usando o conceito de *Distributed Deep Neural Networks* (DDNN) para reduzir o tráfego dos dados durante o treinamento do modelo de redes neurais.

5.3.2. NVIDIA Jetson Nano

- Atualmente, muito investimento tem sido aplicado na implementação de soluções de Inteligência Computacional.
- A utilização das placas Raspberry Pi 4 e NVIDIA Jetson Nano, favorecem a implementação de soluções de computação na borda, *Edge Computing*, possibilitando que parte do processo de classificação dos dados seja realizado nos equipamentos remotos e depois consolidado, usando o conceito de *Distributed Deep Neural Networks* (DDNN) para reduzir o tráfego dos dados durante o treinamento do modelo de redes neurais.



Raspberry Pi 4

Processador Broadcom BCM2711

Quad-core Cortex-A72 (ARM v8) 64 bit SoC Clock 1.5 Ghz

Memória RAM: 4GB DDR4



NVIDIA Jetson Nano

Arquitetura NVIDIA Maxwell com 128 NVIDIA CUDA cores

Processador Quad-core ARM Cortex-A57 MPCore 64 bit

4 GB LPDDR4 - 16 GB eMMC 5.1 Flash

6. Processadores Avançador (SoC)

6.1. Qualcomm Processors

6.1. Qualcomm APQ8016E

- O processador Qualcomm APQ8016E é um modelo de Systems-on-a-Chip (SoC), muito utilizado em dispositivos móveis.

RECURSOS: (a) Designed up to 1.2 GHz Quad ARM Cortex-A53 64-bit application processors with 512 kB L2 cache; (b) 28nm LP process for lower active power dissipation, and faster peak CPU performance; (c) Single-channel, non-PoP high-speed memory – LPDDR2/LPDDR3 SDRAM up to 533 MHz clock rate; (d) QDSP6 v5 processor (vocoder) up to 691 MHz; (e) Three dual-voltage UIM ports; (f) Qualcomm Adreno™ A306 3D graphics core.

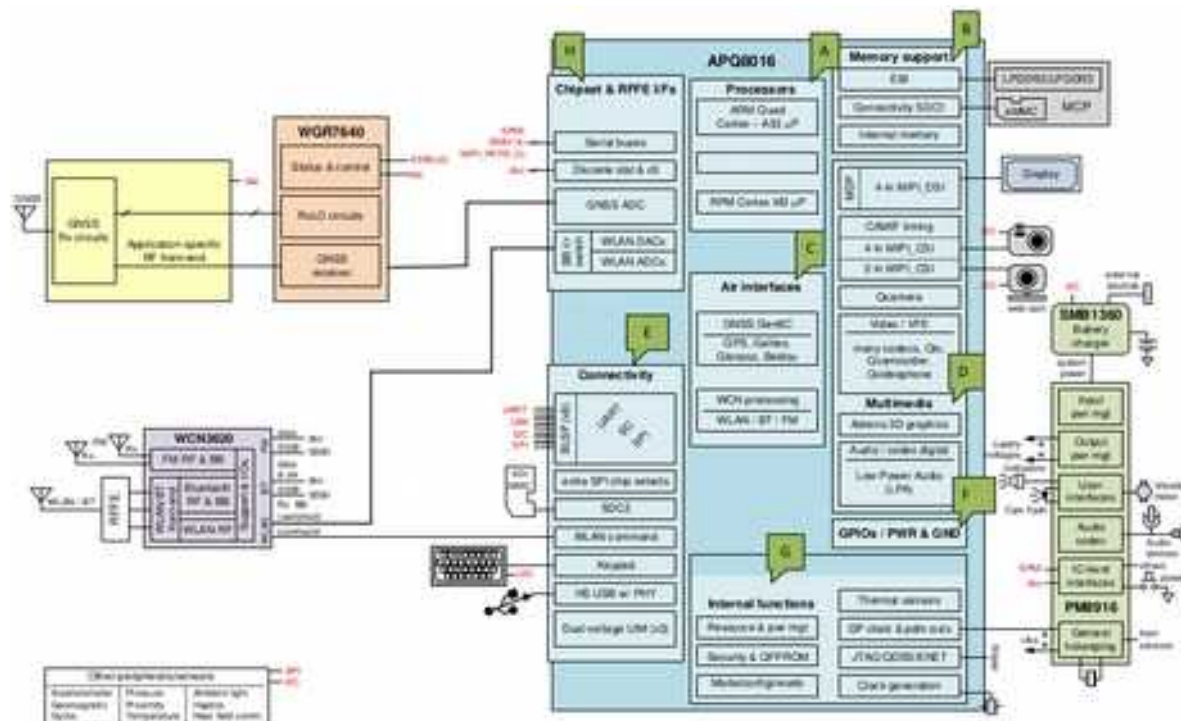


Figure 1-1 APQ8016E functional block diagram and example application

6. Processadores Avançados (SoC)

6.3. Qualcomm (IoT)

6.3.1. Placas Microcontroladoras e Kits de Desenvolvimento da Qualcomm

- A Qualcomm possui uma extensa linha de processadores e microcontroladores, voltados para a elaboração de soluções de IoT para os segmentos: Robótica, Câmeras de Segurança, Maquinas Automáticas de Venda, Terminais Caixas com Operador, e para dispositivos portáteis.



7. Exemplos Práticos

7.1. APLICAÇÃO 1: Microcontroladores

7.1.1. APLICAÇÃO USANDO MICROCONTROLADORES COM SUPORTE AS REDES WIFI

Autor: Prof. Luiz Marcio Faria de Aquino Viana, Pós-D.Sc.

7.1.2. Objetivo

- O objetivo desta aplicação, é apresentar aos alunos um sistema especialista que utiliza a Placa WiFi LoRa 868 MHz ESP32 com OLED para coletar informações do ambiente através de sensores de iluminação, movimento, fumaça, concentração de gás carbônico, concentração de gás natural, temperatura e humidade do ar.
- Estas informações são utilizadas pelo sistema especialista para automatizar o acionamento da ventilação, ar condicionado, e do desumidificador de ar, que estão instalados no ambiente.
- Na ocorrência de um evento os dados coletados são enviados ao servidor local usando um serviço RESTfull.
- No servidor local, os dados são armazenados em uma base de dados NoSQL e podem ser observados pelo operador através de uma interface Web.
- O operador pode a qualquer momento acionar a ventilação, o ar condicionado e o desumidificador de ar, através da interface Web.

Lista de eventos: (I) elevado índice de fumaça, (ii) elevada concentração de gás carbônico, (iii) elevada concentração de gás natural, (iv) alta temperatura e (v) alto índice de humidade do ar.

7.1.3. Modelo (APLICAÇÃO 1)

ServerMon

Pacotes:

- br.com.tlmv.servermon.data
- br.com.tlmv.servermon.datecreator
- br.com.tlmv.servermon.nosql
- br.com.tlmv.servermon.nosql.query
- br.com.tlmv.servermon.rest
- br.com.tlmv.servermon.rest.diy
- br.com.tlmv.servermon.tests
- br.com.tlmv.servermon.utils

DeviceMon

Pacotes:

- All
- ResourceBr
- AppContext
- AppConfig
- AppDefs
- AppError
- SvcCom
- SvcSim
- OLEDUtil

7. Exemplos Práticos

7.2. APLICAÇÃO 2: Android

7.2.1. APLICAÇÃO USANDO SMARTPHONES E TABLETS COM ANDROID

Autor: Prof. Luiz Marcio Faria de Aquino Viana, Pós-D.Sc.

7.2.2. Objetivo

- O objetivo desta aplicação, é apresentar aos alunos um sistema de localização indoor.
- A aplicação efetua periodicamente uma varredura das redes WiFi próximas ao aparelho celular, e envia os dados para um serviço RESTfull.
- O valor do nível de sinal em dBm (logarítmico), fornecido pelo sistema Android, é convertido para um valor em mW (linear).
- No servidor local, os dados são armazenados em uma base de dados NoSQL e podem ser observados pelo operador através de uma interface Web.
- Através da interface Web, o operador obtém a distância aproximada dos smartphones e tablets registrados no sistema.

7.2.3. Modelo (APLICAÇÃO 2)

ServerMon

Pacotes:

- br.com.tlmv.servermon
- br.com.tlmv.servermon.data
- br.com.tlmv.servermon.datecreator
- br.com.tlmv.servermon.nosql
- br.com.tlmv.servermon.nosql.query
- br.com.tlmv.servermon.rest
- br.com.tlmv.servermon.rest.diy
- br.com.tlmv.servermon.tests
- br.com.tlmv.servermon.utils

DeviceMon

Pacotes:

- br.com.tlmv.sensormovapp
- br.com.tlmv.sensormovapp.adapter
- br.com.tlmv.sensormovapp.events
- br.com.tlmv.sensormovapp.form
- br.com.tlmv.sensormovapp.util
- br.com.tlmv.sensormovapp.vo

Dúvidas

