Sumário

[**OBJETIVOS** 3](#_Toc18766701)

[**JUSTIFICATIVA** 4](#_Toc18766702)

[**1 INTRODUÇÃO** 5](#_Toc18766703)

[**1.1** **O Motor de Indução Trifásico** 7](#_Toc18766704)

[1.1.3 Rotor 10](#_Toc18766705)

[1.1.4 Categorias 11](#_Toc18766706)

[1.1.4 Falts Induction Motor 13](#_Toc18766707)

[**2 MANUTENÇÃO, Estado da Arte** 16](#_Toc18766708)

[**2.1 A Manutenção** 16](#_Toc18766709)

[**2.2 Estratégias e Tipos de manutenção** 16](#_Toc18766710)

[**2.3 Os sistemas de gestão da manutenção informatizados** 16](#_Toc18766711)

[CAPITULO 3 17](#_Toc18766712)

[**3 CONCEITOS DA COMPUTAÇÃO** 17](#_Toc18766713)

[**3.1 Computação na Nuvem** 17](#_Toc18766714)

[**3.2 IBM Cloud** 20](#_Toc18766715)

[3.2 Cloud Services 21](#_Toc18766716)

[**3.5 Aprendizado de Máquina** 21](#_Toc18766717)

[3.5.1 Aprendizado supervisionado 24](#_Toc18766718)

[3.5.2 Aprendizagem não supervisionada 25](#_Toc18766719)

[3.5.3 Aprendizagem por reforço 25](#_Toc18766720)

[**3.6 Watson Studio** 26](#_Toc18766721)

[6.2.1 Arquitetura deWatson Studio ® 27](#_Toc18766722)

[3.6.1 subtópicos Watson Studio 30](#_Toc18766723)

[**4 MÁQUINAS DE VETORES DE SUPORTE (silva vinicius)** 30](#_Toc18766724)

[**4.1 Teoria do aprendizado estatístico** 31](#_Toc18766725)

[4.1.1 Minimização do risco empírico 31](#_Toc18766726)

[4.1.2 Minimização do risco estrutural 31](#_Toc18766727)

[**4.2 Fundamentação teórica das SVMs** 31](#_Toc18766728)

[4.2.1 Hiperplano de separação ótimo 32](#_Toc18766729)

[**4.3Margens rígidas** 32](#_Toc18766730)

[**4.4 Margens flexíveis** 32](#_Toc18766731)

[**4.5 SVMs não lineares** 32](#_Toc18766732)

[**4.6 SVM para multiclasses** 32](#_Toc18766733)

[**4.7 Considerações Da aplicação da ferramenta** 32](#_Toc18766734)

[**5 MATERIAIS e MONTAGEM** 32](#_Toc18766735)

[**5.1 System On Chip Esp 8266** 32](#_Toc18766736)

[5.1.1Comunicação SPI 34](#_Toc18766737)

[**5.2 Accelerometer Mems Adxl 345** 36](#_Toc18766738)

[5.2.1 Spi No Adxl345 43](#_Toc18766739)

[**5.3 Sct-013** 44](#_Toc18766740)

[6 METODOLOGIA 45](#_Toc18766741)

[**6.1 Desenvolvimento de sistema para aquisição de dados do ADXL345 E SCT-013** 45](#_Toc18766742)

[**6.2 Instâncianto De Serviços na Cloud** 46](#_Toc18766743)

[**6.3 Utilização De Watson Studio**®. 46](#_Toc18766744)

[**6.4 SPSS MODELER** 46](#_Toc18766745)

[6.4.2 Utilização de Blocos 47](#_Toc18766746)

[6.4.3 Type 47](#_Toc18766747)

[6.4.4 Svm Node 47](#_Toc18766748)

[6.4.5 Table 47](#_Toc18766749)

[6.4.6 Analyses 47](#_Toc18766750)

[**6.6** **Discussão dos resultados obtidos no estudo do modelo físico do sistema de monitoramento de aceleração de vibração e corrente, para aplicação na IBM® Cloud** 47](#_Toc18766751)

[**6.7 Geração Do Modelo Preditivo** 47](#_Toc18766752)

[**6.8 Discussões Sobre O Modelo** 47](#_Toc18766753)

[**7 Conclusão e Trabalhos Futuros** 47](#_Toc18766754)

[7.1 Conclusões 47](#_Toc18766755)

[7.2 Trabalhos Futuros 47](#_Toc18766756)

[**9 APÊNDICE** 47](#_Toc18766757)

[**8 REFERÊNCIAS** 51](#_Toc18766758)

# **OBJETIVOS**

**Objetivos Gerais**

Este trabalho tem como objetivo adquirir dados de condições de falhas externas e condições normais de um motor de indução em laboratório utilizando um sistema de aquisição embarcado e condições normais através da simulação de um motor de indução afim de implementar na nuvem uma técnica de inteligência artificial que busca correlacionar informações através dos dados à falhas de desbalanceamento de massa no eixo, desnível no solo e funcionamento com somente duas fases.

**Objetivos Específicos**

* Implementar um software no ESP8266 capaz de coordenar sensores e captar dados através de protocolos, para enviar os dados para a nuvem, onde acontecerão as análises.
* Desenvolvimento de um sistema capaz de monitorar vibração e correntes para máquinas rotativas de baixo custo, objetivando identificar diferentes condições de funcionamento.
* Obter dados para alimentar análises preditivas do condicionamento de funcionamento do motor de indução trifásica.
* Utilizar a infraestrutura disponível gratuitamente de análises estatísticas da Cloud para aplicação de MVS e gerar os modelos preditivos adequados às situações propostas.

# **JUSTIFICATIVA**

O desenvolvimento de sistemas embarcados vem beneficiando o controle de manutenção na atualidade, de modo a aumentar a disponibilidade de operação das máquinas e controlar melhor as intervenções de manutenção. Porém há de se notar que bons hardwares de sistemas disponíveis no mercado possuem um elevado custo de implementação, não sendo viáveis para uma difusão em larga escala. Este estudo propõe uma alternativa de baixo custo para análise de falhas críticas em motor de indução trifásica usando um sistema de captação de dados composto por um Systen on Chip – ESP8266, sensor acelerômetro MEMS e sensor de corrente SCT-013, para gerar o banco de dados necessários para as implementações e análises em IBM Cloud®.

# **1 INTRODUÇÃO**

O motor de indução é um dos principais componentes utilizados na indústria. Isso se dá pelo fato da sua alta performance, baixo custo e confiabilidade para gerar energia mecânica a partir da energia elétrica. Tais motores são utilizados em várias áreas da indústria devido sua flexibilidade de aplicação, é encontrado desde aplicações domésticas até aplicações de alta potência e prioridade na indústria. Entretanto, apesar de sua alta confiabilidade, o motor de indução pode ser exposto á diferentes condições de falhas e tais falhas podem levar o motor a um colapso e consequentemente à uma parada não planejada na produção que pode ser bastante disruptivo para o processo devido à escala industrial de produtividade.

A partir disso, é importante que a manutenção esteja continuamente monitorando o motor para que falhas sejam detectadas ainda em seus estados iniciais evitando um colapso. Essa detecção de falhas com antecedência permite que as ações corretivas sejam executadas o quanto antes. Dentre essas falhas externas as principais são sobrecarga, falha em uma única fase, tensão de alimentação desbalanceada, rotor travado, reversão de fase e desnível do motor.

A primeira solução encontrada por engenheiros de manutenção para o monitoramento de motores de indução foi a utilização de relés eletromecânicos que protegiam o motor contra falhas (Elmore, 2004). Entretanto, essa solução se mostrou não eficiente devido a características do relé como lentidão na operação, significativo consumo de energia e necessitam de manutenção periódica devido às partes mecânicas envolvidas.

Com o surgimento da tecnologia dos semicondutores a manutenção preditiva obteve uma melhoria na capacidade de proteger um motor de indução. A partir desse momento, os relés eletromecânicos foram substituídos por relés de estado sólido que eram mais eficientes nos quesitos citados acima e ainda menor custo de fabricação e confiabilidade. O desenvolvimento da tecnologia de microprocessadores no final dos anos 1970 permitiu que os mesmos fossem utilizados nos relés para proteção de motores de indução (IEEE, 1997) e a partir desse momento a lógica para a proteção passou a ser implementada através de softwares.

Os recentes desenvolvimentos nos softwares baseados em sistemas inteligentes fizeram com que engenheiros passassem a utiliza-los no diagnóstico de falhas de componentes de sistemas elétricos de potência como o motor de indução. (Kezunovic, 1997). Os sistemas baseados em técnicas de inteligência artificial substituem um professional ao prover as informações necessárias sobre a performance do sistema. Algumas dessas técnicas são Redes Neurais Artificiais, Lógica Fuzzy e Máquina de Vetores de Suporte.

As Redes Neurais Artificiais oferecem características sobre o processo interno de predição e de detecção de falhas, e ainda, não possuem algoritmos para treino que maximizam o processo de generalização de forma sistemática o que pode levar à sobreajuste do modelo em relação aos dados em estudo.

Por outro lado, é possível utilizar a Lógica Fuzzy para a detecção antecipada de condições de falha e conseguir interpretar os resultados com uma base teórica. Entretanto a Lógica Fuzzy requer um professional habituado às definições de suas normas e ao processamento dos dados de entrada.

As Máquinas de Vetores de Suporte-MVS vem sendo mais utilizadas para compor sistemas de detecção de falha devido à sua boa capacidade de generalização e sua alta taxa de acerto.

Existem estudos propostos que obtiveram bons resultados com o objetivo de fazer o diagnóstico de um motor de indução. Podem ser citados trabalhos como o de Kolla e Altmann, onde uma Rede Neural Artificial foi treinada com sinais de corrente e de voltagem para detectar falhas em motores de indução. RNN foi utilizada para o mesmo propósito, mas em um motor de indução simulado, por Kolla e Varatharasa. Somente falhas elétricas foram analisadas.

Baccarini utilizou Maquinas de Vetores de Suporte treinadas com sinais de vibração afim de detectar falhas mecânicas. Desta vez somente falhas mecânicas foram analisadas.

Widodo e Yang, através de uma pesquisa apresentada sobre a detecção de falhas em máquinas utilizando Maquinas de Vetores de Suporte, concluíram que MVS é a técnica mais promissora para diagnóstico de falha e que necessita de mais trabalhos focados na pesquisa de MVS como ferramenta para o monitoramento da condição e o diagnóstico de falhas em motores de indução. Brun e Ernst, também encontraram bons resultados em algumas aplicações de MVS.

A maior eficiência na fase de treino obtida com o modelo MVS em relação ao modelo de RNN foi encontrada por (E. Avci & D. Avci,2009). A fase de treino consiste na correlação entre medidas obtidas na aquisição de dados e as correspondentes falhas, e em termos práticos, dependendo da variação da severidade de falha, se não provido de suficientes padrões de falha o modelo poderá não correlacionar de maneira certa um dado de aquisição à uma falha especifica gerando um falso diagnostico. MVS é regido por teorias de aprendizado estatístico onde o modelo tenta desenvolver um sistema de classificação confiável que seja capaz de treinar em menos tempo.

Este trabalho propõe uma aplicação prática de MVS à identificação de falhas mecânicas e elétricas externas à um motor de indução trifásico implementada na infraestrutura de algoritmos na nuvem. A metodologia proposta busca captar as falhas através de um acelerômetro para medir vibrações e um sensor de corrente não invasivo para captar a corrente no estator e ainda, através da simulação de um motor de indução em diferentes níveis de severidade de falha. SVM será implementado na IBM Cloud através de uma interface

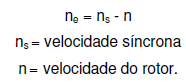
## **O Motor de Indução Trifásico**

O fluxo magnético girante aparece no estator devido as correntes alternadas circulantes nas bobinas do estator. Este fluxo magnético do estator se desloca em relação ao rotor, cortando as barras do rotor induzindo tensões (Lei de Faraday e Lenz) que farão circular correntes também alternadas no rotor. Como as correntes do rotor têm polaridades contrárias do estator (Lei de Faraday e Lenz), cria-se também no rotor um campo magnético girante que será atraído e arrastado pelo campo girante do estator. [99]

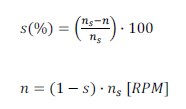
Desenvolve-se assim um conjugado mecânico no rotor levando o mesmo a girar. A velocidade do rotor (n) é sempre menor que a velocidade do campo girante do estator (ns), também chamada velocidade síncrona. Se o rotor fosse levado até a velocidade síncrona (n = ns), não haveria mais velocidade relativa entre os campos girantes do estator e do rotor e consequentemente a tensão induzida cessaria, não haveria mais corrente no rotor, o conjugado mecânico diminuiria e o rotor automaticamente perderia velocidade (nr<ns), então, novamente o rotor iria adquirir o conjugado. [99]et al

A operação do motor girando sem carga denomina-se operação em vazio. A medida que se coloca carga no eixo a tendência da velocidade é diminuir para compensar o conjugado resistente da carga. A operação do motor com carga é denominada operação em regime permanente.

A diferença entre a velocidade síncrona e a velocidade do rotor é chamada de velocidade de escorregamento (ne):



Assim, o *escorregamento s* é definido por:



Sendo que s para motores de indução de gaiola é de 2 a 5%. A velocidade síncrona (ns) é dado por:



onde:

*f* = freqüência em hertz

p = número de pólos.

Podemos variar a velocidade ns e, consequentemente, n variando-se o número de pólos **p** (alterando-se construtivamente as bobinas do estator).

1.1.2 Características Construtivas e Princípios de Funcionamento

Com base em [4], a Figura 1 mostra a estrutura de motor de indução, que compreende:

As máquinas elétricas rotativas são constituídas de duas partes:

a) o estator que é a parte fixa;

b) o rotor que é a parte móvel.

É construído com chapas de material magnético e recebe o enrolamento de campo, cujas espiras são colocadas em ranhuras, como mostra a Figura 2.

Aí se situa o enrolamento de campo, que pode ser mono ou trifásico. A maneira como esse enrolamento é construído determina o número de pólos do motor, entre outras características operacionais. Suas pontas (terminais) são estendidas até uma caixa de terminais, onde pode ser feita a conexão com a rede elétrica de alimentação. [99]



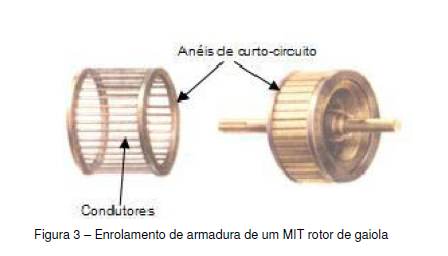
Figura 2 – Enrolamento de campo de um motor de indução: (a) execução dos enrolamentos; (b) núcleo

### 1.1.3 Rotor

O rotor do motor de indução pode ser de 2 tipos:

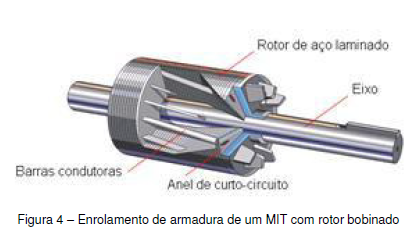
a) Rotor em gaiola de esquilo ou rotor em curto: Os condutores (ou bobinas) são

constituídos por barras de cobre ou alumínio colocadas em ranhuras.



Nas duas extremidades das barras existem 2 anéis curto-circuitando todas as barras. Esta estrutura é semelhante a uma gaiola de esquilo (EUA, "*Squirrel Cage*"). Conforme já dito anteriormente é o tipo de rotor mais empregado (mais barato e não requer manutenção elétrica).

b) Rotor bobinado ou rotor de anéis: A construção de um rotor bobinado é muito mais cara que um rotor em gaiola, e é executada quando se deseja a variação da velocidade da máquina. Um motor de rotor bobinado é possível controlar-se através da corrente que circula no rotor além da velocidade conjugado do mesmo.

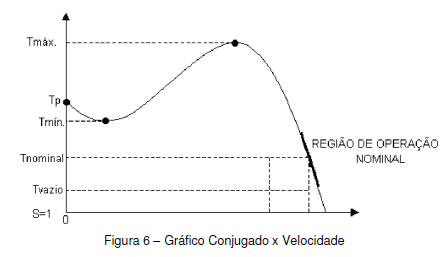


### 1.1.4 Categorias

O motor de indução tem um conjugado nulo à velocidade síncrona (n=ns), s=0 e T=0.

A medida que é aumentada a carga no eixo do motor, a sua velocidade diminui até um ponto onde o conjugado desenvolvido é máximo.

Qualquer acréscimo de carga além desse ponto (Tmáx ® Região de operação instável) faz com a velocidade caia bruscamente, podendo algumas situações travar o rotor.



Sendo:

Tp = conjugado de partida: é o conjugado com o motor travado, ou torque desenvolvido na partida do motor.

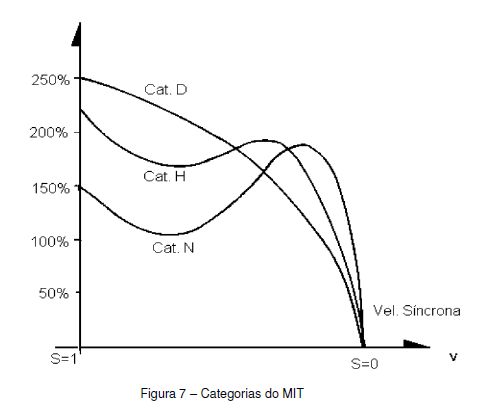
Tmín = conjugado mínimo: é o menor valor de conjugado obtido desde velocidade zero até a velocidade correspondente ao conjugado máximo.

Tmáx = conjugado máximo: é o máximo valor de conjugado que o rotor pode desenvolver *sem travar o eixo*.

Tnominal = conjugado nominal: é o conjugado que o motor fornece com carga nominal no eixo. Normalmente Tnominal ocorre com S entre 2 e 5%.

Tvazio = conjugado para o motor operando sem carga. Representa o conjugado sem carga no eixo.

[99]De acordo com as formas construtivas do rotor de motores de indução de gaiola podem apresentar diferentes características de conjugado e corrente de partida conforme o gráfico:



a) Categoria N: conjugado de partida Tp normal; corrente de partida Ip normal (6 a 7 x Inominal); Escorregamento baixo (2% £ S £ 5%). Nesta categoria se enquadram a maioria dos motores que acionam cargas normais tais como: bombas centrífugas, máquinas operatrizes. O rotor possui gaiola única. [99]

b) Categoria H: conjugado de partida Tp alto; corrente de partida Ip normal; escorregamento baixo. São motores adequados para cargas com elevada inércia, como: peneiras e transportadoras - carregadoras. O rotor é de Dupla

Gaiola.

c) Categoria D: Tp alto; Ip normal; Salto (maior que 5%), motores para cargas que apresentam picos intermitentes, tais como: prensas excêntricas, tesouras e elevadores.

### 1.1.4 Falts Induction Motor

As condições diversas as quais está sujeito o motor de indução, o conduz a diferentes maneiras de falhas que se propagam no motor podem ser internas ou externas. Neste trabalho serão abordadas falhas externas, que podem se classificar da seguinte forma:

1. Sobrecarga

2. Fase única

3. Tensão de alimentação desequilibrada

4. Rotor bloqueado

5. Reversão de fase

6. Falta de aterramento

7. Sobretensão

Uma breve descrição dessas dessas falhas e suas características é fornecida abaixo. A proteção desses motores é uma tarefa importante e desafiadora para para o âmbito da manutenção. A utilização de relés de proteção foi amplamente difundida, objetivando monitorar essas falhas e desconectar o motor em caso de falha. Porém o avanço das aplicações de monitoramento e diagnóstico de falha vem se modificando de acordo com a evolução das aplicações.

Sobrecarga

A falha de sobrecarga ocorre quando o torque mecânico excede o ponto limite em aplicação de carga mecânica ao motor acima da classificação de carga total. Sobrecarga causa aumento em correntes de fase, superaquecendo a máquina. Em um sistema tradicional de proteção de relé, a sobrecorrente desarma o motor quando encontram sobrecarga de corrente na linha.

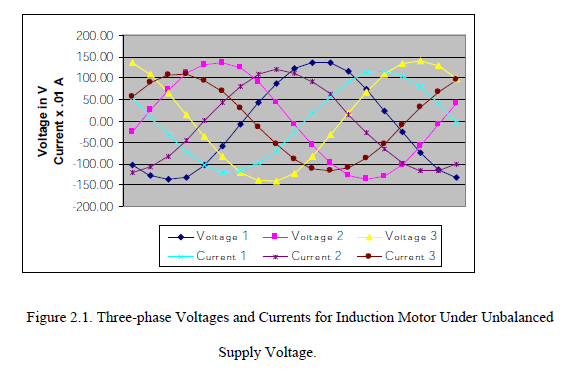
Fase única:

A fase única é um dos casos desequilibrados do motor. Ocorre quando um dos três linhas estão abertas. Mais corrente flui através das outras duas linhas e mais calor é gerado no enrolamento do estator. Nos sistemas de proteção tradicionais, um relé de unidade de disparo instantâneo de alto nível é usado (Elmore, 2004). A fase única também gera corrente de sequência negativa. Um negativo o relé de seqüência também pode ser usado para proteger contra essa falha.

Tensão de alimentação desequilibrada:

Existem muitas causas de tensões de alimentação desbalanceadas, como carga desbalanceada, transformadores a e configuração desigual de distribuição das fases. Esta condição leva à redução da eficiência do motor, aumenta a temperatura do motor e a corrente de carga total desequilibrada excessiva (Sudha & Anbalagan, 2009). Tensões e correntes trifásicas durante uma alimentação desequilibrada são mostradas na Figura 2.1.

O projeto de proteção deve detectar a condição de sobrecorrente durante a alimentação desbalanceada (Elmore, 2004).



Rotor bloqueado

O rotor bloqueado ocorre quando a tensão é aplicada a um motor não rotativo. O estator a corrente pode ser quase seis vezes o seu valor nominal durante essa condição (Sudha & Anbalagan, 2009).

Existem muitas causas para que essa falha ocorra, por exemplo, se o eixo do rotor estiver conectado a carregar o motor pode experimentar condições bloqueadas do rotor. O rotor bloqueado causa alta corrente que leva ao aquecimento do rotor. Portanto, a condição de rotor bloqueado não pode ser suportada por muito tempo.

A duração permitida do motor sobrecarregado sob a condição de rotor bloqueado depende da tensão aplicada aos terminais do motor. (Elmore, 2004).

Reversão de Fase

A inversão de fase ocorre quando qualquer uma das duas fases é revertida da sequência normal, o que leva o motor a girar na direção oposta. Quando o motor começa a girar no direção oposta, pode causar danos intensos. Portanto, essa condição deve ser corrigida imediatamente. Relés de fase reversa e de seqüência negativa são usados para a proteção (Elmore, 2004).

Falta de aterramento

As falhas de aterramento ocorrem quando qualquer uma das fases toca o solo. São mais freqüente em motores do que qualquer outro sistema de energia, devido à sua condição violenta e frequente começa. Os efeitos dessa falha são intensos, como causar riscos à segurança humana e interferência nas telecomunicações. Pode ser detectado medindo a corrente de fuga ao solo (Elmore, 2004).

Sob tensão

A falha de subtensão está reduzindo a tensão de alimentação nas três fases, por percentual, o que faz com que o motor atinja a velocidade nominal no tempo especificado, aumenta a atual e superaquece a máquina. Relés de proteção de baixa tensão são usados em sistemas tradicionais.

Entretanto, para evitar desligamentos indesejados do relé devido a quedas momentâneas de tensão, o os contatos precisam de um mecanismo de atraso que retarde a proteção de subtensão por um período de tempo.

Esse mecanismo adicional precisa de dispositivos de alta sensibilidade e envolve calibrações (Elmore, 2004).

# **2 MANUTENÇÃO, Estado da Arte**

## **2.1 A Manutenção**

## **2.2 Estratégias e Tipos de manutenção**

## **2.3 Os sistemas de gestão da manutenção informatizados**

# CAPITULO 3

# **3 CONCEITOS DA COMPUTAÇÃO**

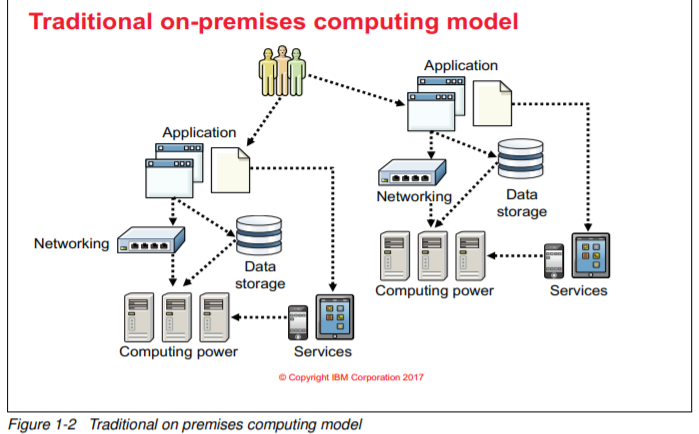
## **3.1 Computação na Nuvem**

O termo nuvem é usado como um termo genérico/metafórico frente a um conjunto virtualizado de hardware e recursos, tornando-se uma abstração para a infraestrutura complexa que que ela possui. Uma boa definição de computação em nuvem vem do Instituto Nacional de Padrões e Tecnologia (NIST). A definição do NIST essencialmente diz que:

“A computação em nuvem é um modelo para permitir acesso conveniente e sob demanda da rede a um conjunto compartilhado de recursos de computação configuráveis que podem ser rapidamente provisionados e lançado com esforço mínimo de gerenciamento ou interação com o provedor de serviços ".

Exemplos de recursos de computação incluem:

* Redes
* Servidores
* Armazenamento
* Aplicações
* Serviços



A computação na nuvem como modelo de implantação está substituindo uma abordagem mais antiga, na qual cada aplicativo com o qual um usuário interage teve seus próprios serviços personalizados, rede, dados armazenamento e poder de computação.

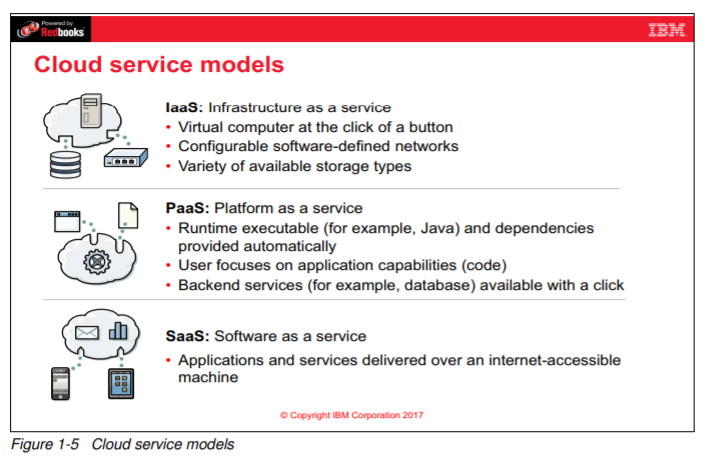
A capacidade de reutilizar e adaptar novamente o hardware rapidamente, além de hospedar vários aplicativos e sistemas em um único conjunto de hardware de maneira isolada, são algumas das principais características que impulsionam a adoção da computação em nuvem.

Na abordagem antiga, a equipe de TI necessitaria gerenciar todos os processos, desde o hardware, até alterações mais recentes do software. Esse modelo não escala tão bem quanto os negócios e as empresas atuais organizações exigem.

Uma contribuição importante para o crescimento da computação em nuvem acontece através da necessidade atual da velocidade da demanda de entrega de serviços, que devem ser entregues rapidamente. Os desenvolvedores são pressionados a colocar seu produto no mercado assim que possível. Necessitando de feedbacks rapidamente e, em seguida, repetem a ideia de tornar o produto melhor e mais rápido.

A nuvem torna os recursos de hardware prontamente disponíveis e rápidos de configurar, desta forma reduzindo o tempo necessário para que seja mostrada uma versão funcional de produtos. Além disso, permite a reutilização dos mesmos recursos para vários projetos sucessivos, o que é mais econômico. Um outro fator que contribui para o crescimento da computação em nuvem é que os desenvolvedores podem usar diferentes linguagens de programação.

Descrição dos Modelos de serviço:



- Infraestrutura como Serviço (IaaS), um conjunto de ativos físicos, como servidores, dispositivos de rede e discos de armazenamento, são oferecidos como dedicados e particulares acessível aos consumidores. Os serviços neste modelo suportam a infraestrutura de aplicativos.

- Plataforma como serviço (PaaS) é um modelo de serviço em nuvem no qual a estrutura de aplicativos e o tempo de execução é uma entidade virtualizada, de autoatendimento. O objetivo do PaaS é permitir o desenvolvedor ou equipe para se concentrar nas funções, código e dados de negócios do aplicativo, em vez de se preocupar com infraestrutura.

- Comumente utilizada por usuários da web o serviço de Software como Serviço (SaaS), embora possam não sabe disso. Os aplicativos no modelo SaaS são fornecidos sob demanda aos usuários através do Internet, em oposição aos aplicativos de desktop. Exemplos de aplicativos SaaS incluem Salesforce.com, Google Apps e Facebook.

## **3.2 IBM Cloud**

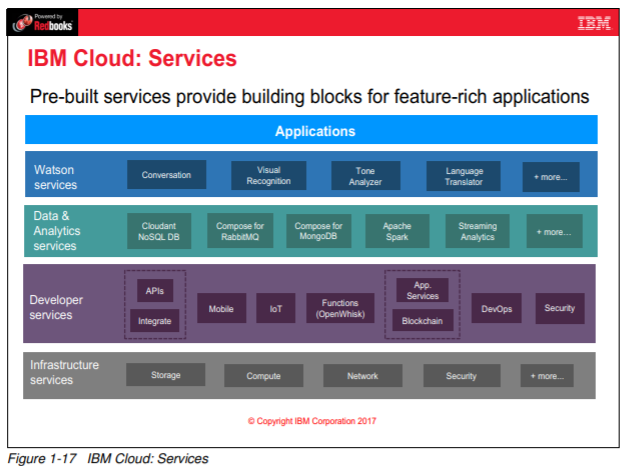
A IBM Cloud é uma plataforma de computação em nuvem aberta que combina a plataforma como serviço (PaaS) com infraestrutura como serviço (IaaS) e inclui um catálogo de diversos serviços, que podem ser usados para criar e implantar rapidamente aplicativos ou infraestruturas diversas.

Como PaaS, fornece aos desenvolvedores acesso ao software IBM para integração, segurança e outras funções-chave. Os tipos de aplicativos podem variar de web, dispositivos móveis, big data e dispositivos inteligentes à Internet das coisas.

Como IaaS, ele permite que os desenvolvedores controlem detalhadamente a infraestrutura na qual seus aplicativos são implantados. Os desenvolvedores podem implantar servidores de alto desempenho, servidores virtuais e contêineres, utilizando a infraestrutura e sistemas de hardware nos locais dos data centers da IBM Cloud em todo o mundo.

### 3.2.1IBM Cloud Services

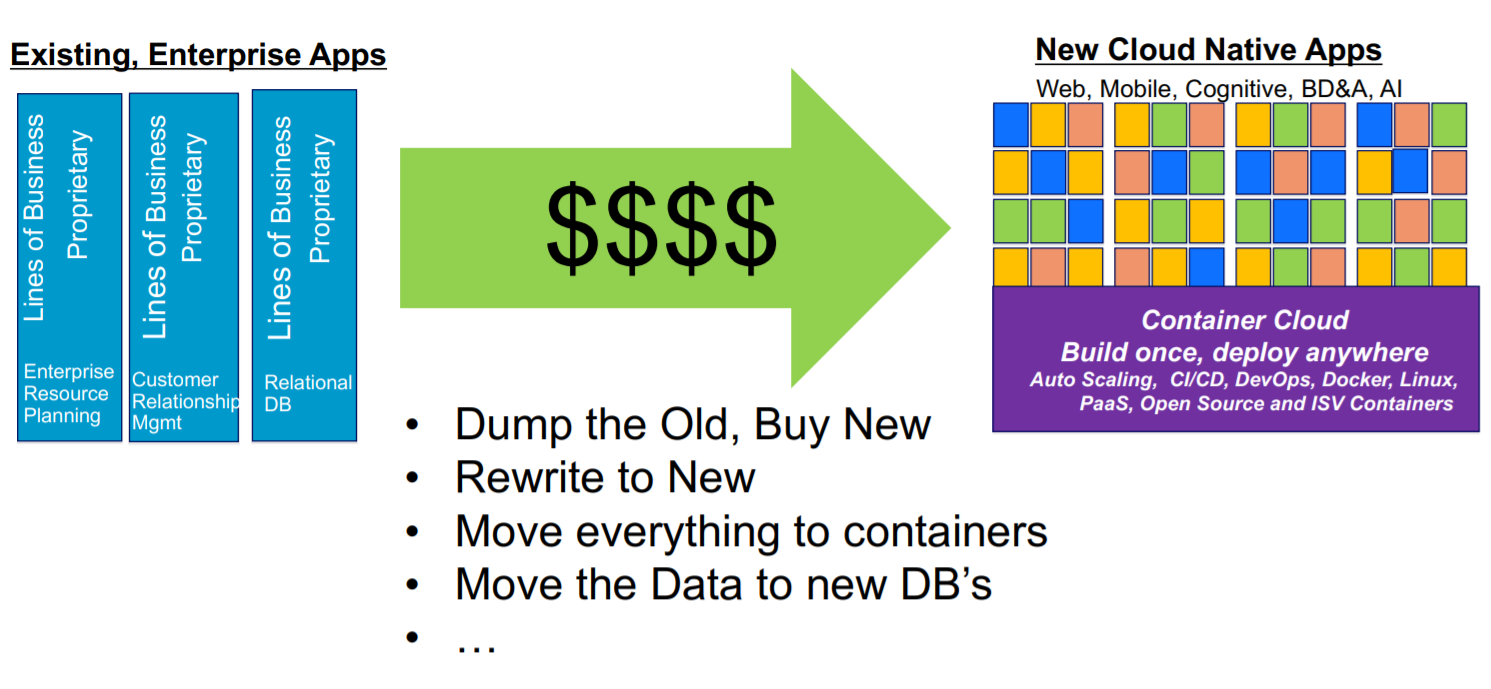
Através da imagem 87 é possível observar a gama de divisões dentre aplicações em IBM Cloud:



O IBM Cloud fornece uma ampla gama de serviços pré-criados (da IBM e de terceiros) que pode ser usado frente a diferentes necessidades de tratamento estatístico.

Além disso, pode-de citar a criação, gerenciamento, e execução de APIS, uso da infraestrutura de back-end para criar e testar aplicações. Comunicando-se com dispositivos, sensores e gateways conectados caso sejam implementados. Várias opções de serviços pré desenvolvidos são oferecidos, como Blockchain. Message Hub, WebSphere Application Server, Business Rules, Watson Studio entre outros serviços na nuvem totalizando mais de 190 serviços disponibilizados em IBM Cloud

Recomendações comuns para transformação atual das aplicações, na fig 60



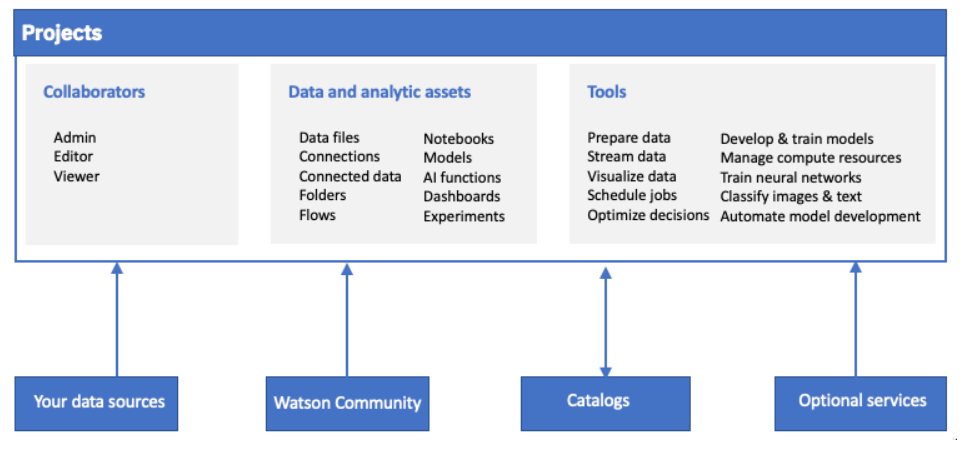
A figura 60 ilustra bem o que vem acontecendo com todas as aplicações que buscam inovar e fazer parte dos avanços em quesitos de desenvolvimento junto a investimento na migração para a computação na nuvem, como já mostrado anteriormente.

Dentre a vasta biblioteca de desenvolvimento oferecida pela IBM Cloud, foi necessário provisionar os serviços de Watson Studio, Cloud Object Storage e Watson Machine Learn, descritos a seguir.

Como proprietário ou administrador da conta, é possível configurar uma instância do Cloud Object Storage para iniciar projetos e catálogos em uma página Delegação de Armazenamento já em Watson(IBM DOCS):

O Watson Studio, fornece o ambiente e as ferramentas para analises estatísticas em conjunto de dados. Oferecendo funções de visualização, limpeza e modelagem dados, podendo ainda criar e treinar modelos de aprendizado de máquina.

Esta ilustração mostra como a arquitetura do Watson Studio é centralizada em torno de um projeto.



Os colaboradores tem acesso as entradas de dados em Data Assets, podendo upar ou carrega-los automaticamente, tendo disposição posterior de uma importante ferramenta na aplicação, Data Refinery(anexo X), onde podem ser criados fluxos de tratamentos dos dados incialmente dispostos, podendo retirar espaços vazios, ou realizar visualizações gráficas dos dados de entrada.

Tendo opções de criação, treinamento e testes de modelos de aprendizado de máquina e aprendizado profundo (deep learn).

Sendo possível também a realização de experimentos com modelos de aprendizado profundo em paralelo com redes neurais. O Watson Studio também pode trazer dados e ativos analíticos da Comunidade IBM Watson, para avaliar testes como exemplos.

É possível encontrar ativos nos catálogos e os adicionar a qualquer projeto. Ao concluir o desenvolvimento de modelos em um projeto, a documentação de IBM CLOUD indica que deve-se promover os modelos no espaço de implantação associado ao projeto. Tornando possível configurar modelos no espaço de implantação.

Algumas ferramentas exigem complementos, havendo necessidade de fazer provisionamento das ferramentas: IBM Cloud Object Stourage e IBM Watson Machine Learn, para dar os subsídios necessários para o upload de dados e análises estatísticas. A seleção de ferramentas complementares se deu através de considerações sobre tipos de dados, tipos de tarefas e quanto de automação pode ser implementado através de editores como *Jupyter, RStudio IDE ou SPSS MODELER*.

Estas ferramentais diferenciam-se por serem melhor aplicadas a contextos de dados diferentes *como :* [*dados tabulares ou relacionais*](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/tools.html#tab)*,* [*dados textuais*](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/tools.html#text)*,* [*dados de imagem*](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/tools.html#image)

Uma tabela disposta pela documentação de IBM Cloud mostra as tarefas que são executadas pelas ferramentas, para análises de dados tabulares ou relacionais(dados do projeto).

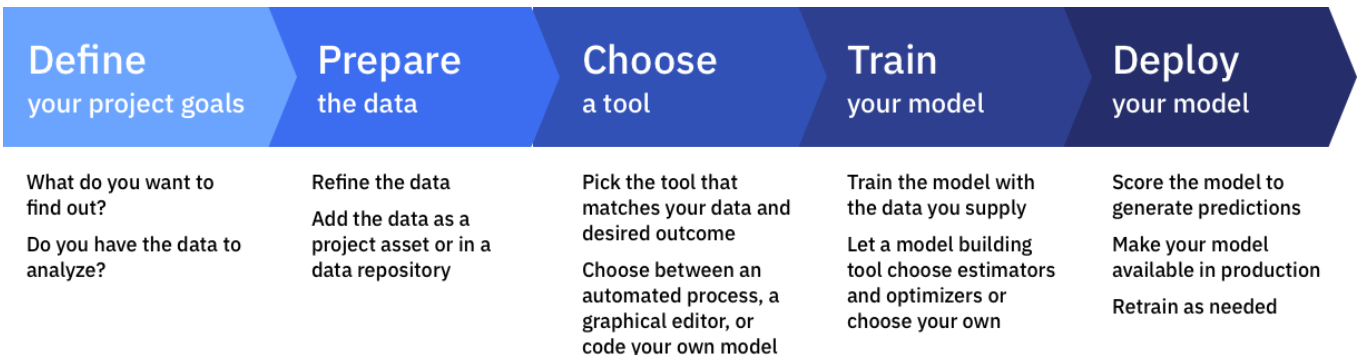


O ambiente de Watson Studio conta com ferramentas gráficas para projetar, treinar, implantar e gerenciar modelos com seus serviços do Watson Aprendizagem de Máquina:

* As experiências da AutoAI processam automaticamente seus dados, selecionam o melhor estimador para os dados e, em seguida, geram pipelines candidatos a modelos para você revisar e comparar. Implante o pipeline com melhor desempenho como modelo de aprendizado de máquina.
* A Rede Neural Sintetizada sintetiza automaticamente uma rede neural e a treina nos seus dados de treinamento sem que você precise projetar ou construir qualquer coisa manualmente.
* O modelador Spark MLlib apresenta uma visualização gráfica do seu modelo enquanto você o constrói combinando nós que representam nós de algoritmo.
* O modelador SPSS apresenta uma visualização gráfica do seu modelo enquanto você o constrói combinando nós que representam objetos ou ações.
* O modelador de rede neural apresenta uma visualização gráfica do seu modelo enquanto você o constrói combinando nós de rede neural).
* Os notebooks fornecem um ambiente de programação interativo para trabalhar com dados, modelos de teste e prototipagem rápida.
* O criador de experiências automatiza a execução de centenas de execuções de treinamento enquanto rastreia e armazena resultados.
* O construtor de modelo Decision Optimization o orienta na construção e resolução de modelos prescritivos.

Através de Watson Aprendizado de Máquina, construído em uma plataforma escalável de código aberto baseada nos componentes Kubernetes e Docker, você pode construir modelos analíticos e redes neurais, treinadas com seus próprios dados, que podem ser implementadas para uso em aplicativos.

Este gráfico ilustra um processo típico para um modelo de aprendizado de máquina.



Dividindo em etapas como definições, preparação, escolha de ferramentas, treinamento e implementação do modelo desenvolvido.

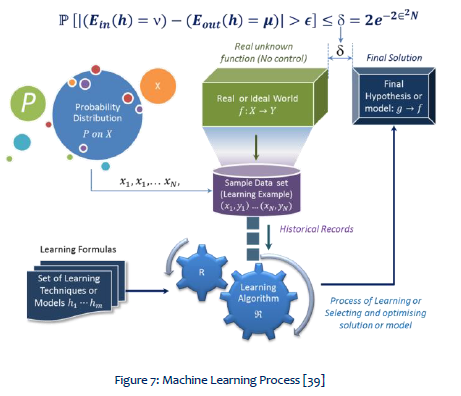
ENTRAR EM SPSS MODELER

## **3.3 Aprendizado de Máquina**

A essência do Aprendizado de Máquina(AM) é um processo automático de reconhecimento de padrões. O principal objetivo do aprendizado de máquina é criar sistemas que possam executar ou exceder a competência no nível humano ao lidar com muitas tarefas ou problemas complexos. O aprendizado de máquina faz parte da Inteligência Artificial (IA). Durante o início da era de pesquisa da IA, o objetivo era construir robôs e simular atividades humanas. Posteriormente, a aplicação da IA foi generalizada para resolver problemas gerais por uma máquina. A solução popular era alimentar um computador com algoritmos (ou uma sequência de instruções) para transformar os dados de entrada em respostas.

No entanto, para muitos problemas, não podemos encontrar facilmente algoritmos adequados, por exemplo, o reconhecimento da caligrafia humana. Não sabemos como transformar a entrada da letra de escrita manual na saída da letra reconhecida padrão. Uma alternativa é aprender com os dados. Isso significa que, com uma tentativa, há um grande erro, mas se pudermos agregar muitas tentativas, o erro será reduzido para um nível ou convergência aceitável. A Figura 7 ilustra um exemplo típico de processo de aprendizado de máquina ou aprendizado de dados.

Desde o final dos anos 90 o volume de dados se tornou cada vez maior. Uma questão lógica é como lidar com esses grandes volumes de dados e como encontrar padrões úteis ou significativos a partir de um volume maior de dados. Isso leva à “descoberta de conhecimento no banco de dados” (ou KDD- knowledge discovery in database), que também é chamada de mineração de dados. Em outras palavras, queremos cavar no banco de dados e descobrir o significado ou conhecimento para a tomada de decisão. Larose et al. [47] definiram o termo como "o processo de descobrir padrões e tendências úteis em grandes conjuntos de dados". Para descobrir padrões significativos de um conjunto massivo de dados, a estatística é a ferramenta vital para agregar valor à amostragem, modelagem, análise, interpretação e apresentação de dados, assim como Jiawei Han et al. [48] indicaram: “A mineração de dados tem uma conexão inerente às estatísticas”. Isso leva à convergência do sistema de mineração de dados e do sistema especialista difuso sob o grande guarda-chuva do aprendizado de máquina. Do ponto de vista da evolução do aprendizado de máquina, a teoria estatística ou a modelagem de probabilidade mudaram a disciplina de IA de sistemas especialistas baseados em regras ou aprendizado de esquema na gravação, para uma metodologia de esquema na leitura ou orientada a dados, que é resolver o problema de incerteza com probabilidade dos parâmetros de um modelo. Nessa perspectiva, as estatísticas foram incorporadas ao aprendizado de máquina.



Desde a década de 1950, tem havido muitas definições funcionais de AM. Diferentes autores enfatizariam diferentes aspectos do aprendizado de máquina, como processo, aplicativo e utilidade. Por exemplo, a definição de Arthur Samuel enfatizou o "aprendizado automático" de AM. Tom M. Mitchell descreveu todos os componentes do processo de AM [50]. Kevin P. Murphy [51] e Christopher M. Bishop [52], por outro lado, enfatizaram a função do reconhecimento de padrões. Noam Nisan e Shimon Schocken [53] argumentaram que AM poderia transformar pensamentos abstratos em operação física. No resumo de mais de 30 definições, podemos encontrar alguns dos ingredientes essenciais e comuns dessas definições de AM:

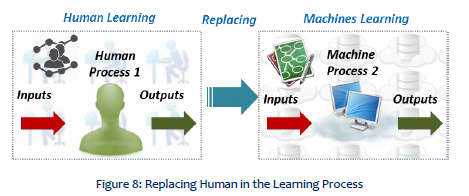
• Treinar a máquina para aprender automaticamente e melhorar os resultados à medida que obtém mais dados.

• Descobrir ou reconhecer padrões com dados de entrada.

• Executar predições através de entradas desconhecidas.

• A máquina irá adquirir conhecimento diretamente dos dados e resolverá problemas propostos.

De acordo com esses elementos, podemos descobrir que, fundamentalmente, o AM é "uma consequência da interseção entre ciência da computação e estatística, visa aprender automaticamente a reconhecer padrões complexos e a tomar decisões inteligentes com base em conjuntos de dados existentes" [54]. O objetivo final do AM é construir sistemas com nível de competência humana (veja a Figura 8) na execução de tarefas complexas.



O aprendizado de máquina permite que os modelos treinem em conjuntos de dados antes de serem implantados. Alguns modelos de aprendizado de máquina são online e contínuos. Esse processo iterativo de modelos online leva a uma melhoria nos tipos de associações feitas entre elementos de dados. Devido à sua complexidade e tamanho, esses padrões e associações poderiam ter sido facilmente ignorados pela observação humana. Depois que um modelo é treinado, ele pode ser usado em tempo real para aprender com os dados. As melhorias na precisão são resultado do processo de treinamento e automação que fazem parte do aprendizado de máquina [ibm knowledge].

O aprendizado de máquina oferece um valor potencial para as empresas que tentam alavancar o big data e as ajuda a entender melhor as mudanças no comportamento, preferências ou satisfação do cliente. Buscando padrões e anomalias ocultos nos dados que podem ajudar ou prejudicar os processos.

Técnicas de aprendizado de máquina são necessárias para melhorar a precisão dos modelos preditivos. Dependendo da natureza do problema de negócios que está sendo resolvido, existem diferentes abordagens baseadas no tipo e no volume dos dados. No tópico seguinte, será discutido as categorias de aprendizado de máquina.

### 3.5.1 Aprendizado supervisionado

O aprendizado supervisionado começa com um conjunto estabelecido de dados e um certo entendimento de como esses dados são classificados. O aprendizado supervisionado tem como objetivo encontrar padrões nos dados que podem ser aplicados a um processo de análise. Esses dados rotularam recursos que definem o significado dos dados. Por exemplo, você pode criar um aplicativo de aprendizado de máquina que distinga entre milhões de animais, com base em imagens e descrições escritas.

### 3.5.2 Aprendizagem não supervisionada

O aprendizado não supervisionado é usado quando o problema exige uma enorme quantidade de dados não rotulados. Por exemplo, aplicativos de mídia social, como Twitter, Instagram e Snapchat, todos têm grandes quantidades de dados não rotulados. Compreender o significado por trás desses dados requer algoritmos que classificam os dados com base nos padrões ou clusters encontrados. O aprendizado não supervisionado conduz um processo iterativo, analisando dados sem intervenção humana. É usado com a tecnologia de detecção de spam de e-mail. Existem muitas variáveis em emails legítimos e de spam para um analista marcar emails em massa não solicitados. Em vez disso, os classificadores de aprendizado de máquina, com base em cluster e associação, são aplicados para identificar emails indesejados.

### 3.5.3 Aprendizagem por reforço

O aprendizado por reforço é um modelo de aprendizado comportamental. O algoritmo recebe feedback da análise de dados, guiando o usuário para o melhor resultado. O aprendizado por reforço difere de outros tipos de aprendizado supervisionado, porque o sistema não é treinado com o conjunto de dados de amostra. Em vez disso, o sistema aprende por tentativa e erro. Portanto, uma sequência de decisões bem-sucedidas resultará no reforço do processo, porque melhor resolve o problema em questão.

3.5.4 Aprendizagem profunda

O aprendizado profundo é um método específico de aprendizado de máquina que incorpora redes neurais em camadas sucessivas para aprender com os dados de maneira iterativa. O aprendizado profundo é especialmente útil quando você está tentando aprender padrões a partir de dados não estruturados. As redes neurais complexas de aprendizado profundo são projetadas para imitar como o cérebro humano funciona, para que os computadores possam ser treinados para lidar com abstrações e problemas mal definidos. A criança média de cinco anos de idade pode reconhecer facilmente a diferença entre o rosto do professor e o rosto. da guarda de cruzamento. Por outro lado, o computador deve trabalhar bastante para descobrir quem é quem. Redes neurais e aprendizado profundo são frequentemente usados em aplicativos de reconhecimento de imagem, fala e visão computacional.

# **4 MÁQUINAS DE VETORES DE SUPORTE (silva vinicius)**

As Máquinas de Vetores de Suporte (SVM - do inglês Support Vector Machines) constituem uma técnica fundamentada pela teoria do aprendizado estatístico ou teoria VC (Vapnik- Chervonenkis), desenvolvida por Vapnik (1995), a partir de estudos iniciados por Vapnik e Chervonenkis (1971). Essa teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com boa generalização, sendo generalização, definida como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu.

A formulação apresentada por Vapnik (1995), é baseada no princípio de minimização do risco estrutural (SRM, do inglês Structural Risk Minimization). No ano seguinte Vapnik et al. (1996) mostraram que este princípio é superior ao princípio de minimização do risco empírico (ERM, do inglês Empirical Risk Minimization), este último utilizado no projeto de redes neurais artificiais, o que atribui a SVM melhor capacidade de generalização, (LIMA, 2004).

O SRM envolve a minimização de um limite superior para o erro de generalização, enquanto que ERM envolve a minimização do erro sobre os dados de treinamento. Assim, modelos de aprendizado de máquina baseados no princípio SRM tendem a apresentar uma maior habilidade de generalização frente a dados não observados, sendo este um dos principais propósitos do aprendizado estatístico.

## **4.1 Teoria do aprendizado estatístico**

### 4.1.1 Minimização do risco empírico

### 4.1.2 Minimização do risco estrutural

## **4.2 Fundamentação teórica das SVMs**

### 4.2.1 Hiperplano de separação ótimo

## **4.3Margens rígidas**

## **4.4 Margens flexíveis**

## **4.5 SVMs não lineares**

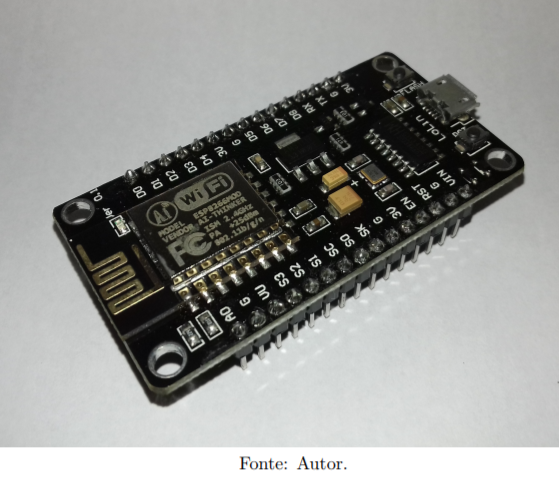
## **4.6 SVM para multiclasses**

## **4.7 Considerações Da aplicação da ferramenta**

# **5 MATERIAIS e MONTAGEM**

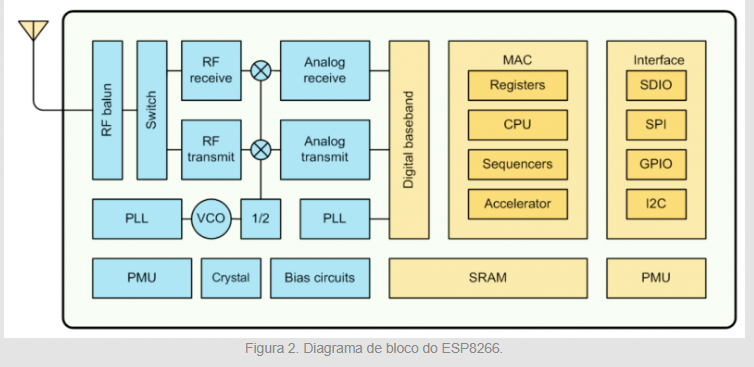
## **5.1 System On Chip Esp 8266**

ESP8266



O ESP8266 é um microcontrolador produzido pela empresa Espressif Systems. Uma característica relevante deste microcontrolador é que ele é capaz de fazer comunicação via Wi-Fi e tem um baixo custo, algo em torno de 3 usd. Além da comunicação por Wi-Fi (seguindo o protocolo IEEE 802.11 bgn), este microcontrolador conta com 16 portas GPIO, comunicação I 2C, UART, SPI, um módulo ADC, um CPU da Tensilica L106 de 32-bit com a arquitetura Xtensa, clock de 80MHz podendo chegar a 160MHz, sendo amplamente utilizado em projetos IoT.

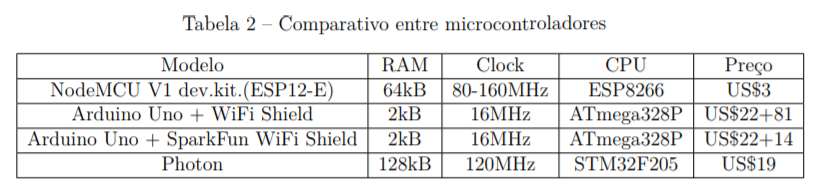
É um conjunto de alto desempenho, alta interação wireless, projetado para espaços pequenos com restrição de consumo de energia para plataformas móveis. Ele fornece a capacidade de incorporar Wi-Fi dentro de outros sistemas, podendo funcionar como aplicativo independente, com menor custo e com um mínimo de espaço. O diagrama de blocos do ESP8266 é ilustrado na Figura 2.



ESP8266 contém todas as ferramentas para se estabelecer uma comunicação Wi-Fi, como; conectores para antenas ou antena integrada, RF balun (dispositivo que casa impedâncias), amplificadores, filtros. O microcontrolador pode rodar aplicações que usam Wi-Fi e também pode servir de adaptador Wi-Fi para outro microcontrolador, utilizando uma comunicação como I2C,SPI ou UART. A parte de RF do microcontrolador é formada pelos seguintes principais blocos: receptor, transmissor, gerador de de clock de alta precisão, relógio de tempo real, reguladores e gerenciador de energia.

O receptor converte o sinal RF para um sinal em quadratura e em banda base, posteriormente este sinal é digitalizado por 2 conversores A/D de alta precisão e alta velocidade. Para contornar os problemas gerados pelas condições adversas do canal de comunicação, são integrados no ESP8266 filtros RF, controle automático de ganho (AGC) e circuitos que cancelam o offset do nível DC. O transmissor converte o sinal em quadratura para um sinal em banda passante de 2,4GHz, e utiliza um amplificador CMOS para ligar a antena. Todos os componentes utilizados pelo gerador de clock estão integrados no microcontrolador.

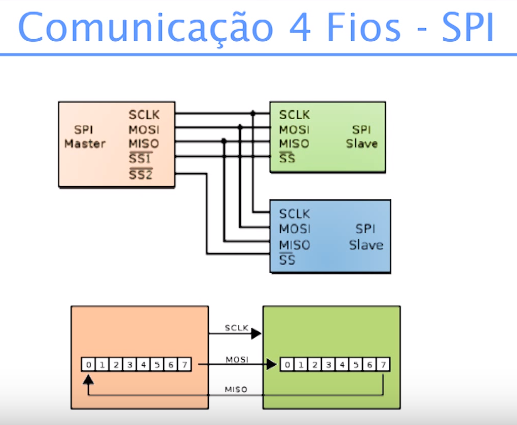
O principal motivo da escolha deste microcontrolador foi seu preço, na Tabela x é possível verificar algumas especificações e o preço de mircrocontroladores.



### 5.1.1Comunicação SPI

Is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers. With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices. It is a full duplex connection, which means that the data is sent and received simultaneously. The maximum baud rate is higher than that in the I2C communication system.

A comunicação SPI possui algumas características básicas. Primeiramente os sinais de comunicação possuem uma direção fixa e definida. Isso significa que sempre existem dois transistores definindo o estado de um pino (Push-Pull). Essa característica é uma das grandes diferenças entre outras comunicações seriais como I2C e OneWire, que possuem um mesmo barramento de dados para os sinais de entrada e saída através do esquema de dreno-aberto (Pull-Up).



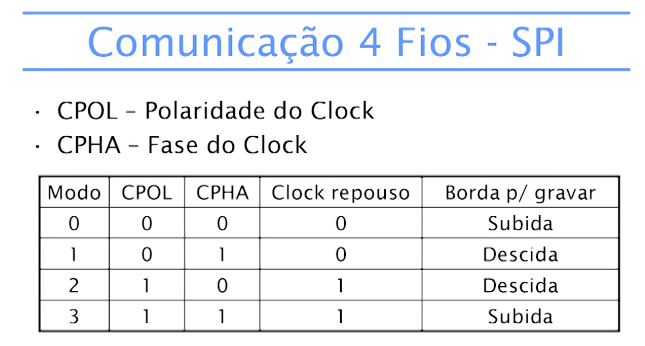
Typically there are three lines common to all the devices:

* **MISO** (Master In Slave Out) – The Slave line for sending data to the master,
* **MOSI** (Master Out Slave In) – The Master line for sending data to the peripherals,
* **SCK** (Serial Clock) – The clock pulses which synchronize data transmission generated by the master

and one line specific for every device:

* **SS** (Slave Select) – the pin on each device that the master can use to enable and disable specific devices.

Além do clock e sinal de seleção, possui dois sinais de dados, tem potencial de fazer comunicação simultânea: basicamente a medida que os bits estão saindo do mestre(out), eles podem fazer o caminho inverso ao mesmo tempo. O mais comum é um trânsito de uma via somente. Um exemplo comum, para usar cartão SD.



Pode ser trabalho diversas polaridades do clock, descrevendo, se no repouso do clock está low/high, além de saber qual a borda do clock que o dado está disponível. Normalmente o clock ta em zero, enquanto ele passa de 0 para 1, significa que tenho um dado para processar.

Micro wire, trata somente da comunicação em modo 0 em half duplex , enquanto somente um dado transita com dados.

When a device’s Slave Select pin is low, it communicates with the master. When it’s high, it ignores the master. This allows you to have multiple SPI devices sharing the same MISO, MOSI, and CLK lines.

To write code for a new SPI device need to note a few things:

* What is the maximum SPI speed your device can use? This is controlled by the first parameter in SPISettings. If you are using a chip rated at 15 MHz, use 15000000. Arduino will automatically use the best speed that is equal to or less than the number you use with SPISettings.
* Is data shifted in Most Significant Bit (MSB) or Least Significant Bit (LSB) first? This is controlled by second SPISettings parameter, either MSBFIRST or LSBFIRST. Most SPI chips use MSB first data order.
* Is the data clock idle when high or low? Are samples on the rising or falling edge of clock pulses? These modes are controlled by the third parameter in SPISettings.

The SPI standard is loose and each device implements it a little differently. This means you have to pay special attention to the device’s datasheet when writing your code.

## **5.2 Accelerometer Mems Adxl 345**

Basicamente, todos os tipos de acelerômetros traduzem o sinal externo de aceleração em um deslocamento correspondente de sua massa móvel, também conhecida como massa inercial ou de prova. Este deslocamento pode ser detectado através de diferentes esquemas de medição, sendo que os mais comuns são: capacitivo, piezoelétrico, piezoresistivo, ressonante e óptico (YADI; AYAZI; NAJAFI, 1998; KRISHNAN et al., 2007).

As principais especificações que devem ser consideradas na seleção de um acelerômetro são discutidas em seguida.

• Sensibilidade: fator de escala de um sensor, medida em termos de mudança na saída para uma alteração no sinal de entrada. É uma referência à habilidade do sensor em detectar movimento e é normalmente especificada em mV/g;

• Largura de Banda (“Bandwidth”): é a faixa de frequências para a qual o sensor é aplicável. É normalmente especificado em Hertz(Hz), sendo tipicamente limitada a 1/5 da primeira frequência de ressonância (HARRIS; CREDE, 1976);

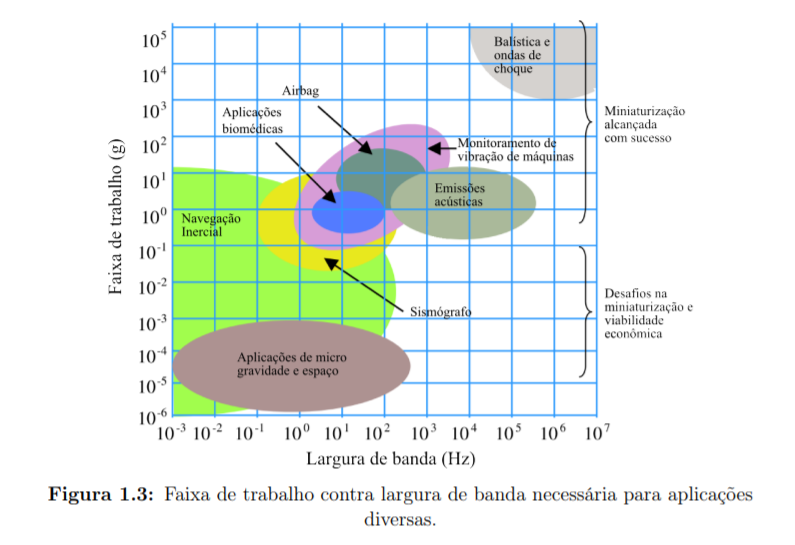
•Estabilidade: define o quão constante é o sinal de saída em condições de entrada constantes, sendo a alteração na saída chamada de deriva (“drift”). Acelerômetros para uso em aplicações de alto desempenho (aeroespaciais), devem apresentar alta estabilidade e por isso são muito mais caros que os utilizados em produtos de consumo, como videogames e celulares. O gráfico da Fig. 1.1, adaptado de Jean-Michel (2004), ilustra a importância desta característica na definição do preço de um sensor;

•Resolução: menor nível de aceleração detectável pelo sensor e é limitada pelo nível de ruído do sensor. É normalmente especificada em mili-g (mg) ou micro-g (µg);

•Alcance Dinâmico (“Dynamic Range”): é a faixa de valores de acelerações que podem ser medidas pelo sensor. O limite inferior é determinado pela resolução do dispositivo e o superior pela sua saturação, que é o ponto a partir do qual o sensor perde a linearidade na sua resposta;

•Confiabilidade: descreve a probabilidade do dispositivo desempenhar adequadamente suas funções, durante um período de tempo especificado e dentro de condições operacionais pré-estabelecidas;

• Custo: tem importância menor para dispositivos de alto desempenho, no entanto, é talvez a característica mais importante em aplicações de consumo.



As origens Do uso da tecnologia do sistema micro-eletromecânico (MEMS) é datada de abril de 1954, quando um artigo de Smith (1954), depois do Bell Telephone Laboratories, foi publicado na Physical Review. Quando é descrita nas referências bibliográficas pela primeira vez certos efeitos sensíveis ao estresse no silício e no germânio, denominados piezoresistência. Em meados da década de 1950, os pesquisadores começaram a investigar se as mesmas tecnologias que produziram o transistor, que posteriormente revolucionou a incipiente indústria eletrônica, poderia ser aplicada aos sensores.

Mostrando que os sensores eletromecânicos volumosos anteriormente bastante utilizados podem ser substituídos por dispositivos pequenos e robustos da mesma maneira que o transistor substituiu a válvula termiônica. O artigo de Smith foi seguido no ano seguinte pela provavelmente a primeira publicação a considerar essa possibilidade (Paul e Pearson, 1955) e, no início dos anos 1960, a série de artigos do Honeywell Research Center e do Bell Labs descreveu a primeira pressão do diafragma de silício. sensores e extensômetros (Pfann e Thurston, 1961; Tufte et al., 1962). O interesse na tecnologia de sensores de silício cresceu dramaticamente e, no final dos anos 60, vários pioneiros americanos comercializavam os primeiros sensores de pressão de silício. Esses padrões eram brutos para os padrões atuais, mas no início dos anos 70 os desenvolvimentos em micromáquina, como era então chamado, e as melhorias no processamento de silício levaram a sensores com geometrias de que produziram desempenho superior.

Os Acelerômetros do tipo MEMS (Micro-ElectroMechanical Systems) detiveram grande impacto comercial com aplicação nos mais diversos campos. Sua criação foi motivada além de tudo pelo avanço da indústria para atender a diferentes tecnologias. Os acelerômetros comerciais são baseados na medição das componentes cartesianas do vetor de aceleração gravitacional, comumente encontrados no mercado oferecendo boa resposta à aceleração dinâmica resultante de movimento, baixa demanda de energia e baixa tensão de excitação. Suas dimensões permitem a integração estrutural ao conjunto do dispositivo a ser monitorado.

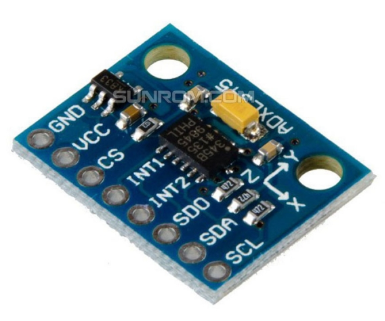
O acelerômetro piezoelétrico é o mais recomendado para aplicações em Manutenção Preditiva (MARTINS, 2000), pois a resposta de frequência estende-se até dezenas de kHz, deslocamento e velocidade podem ser obtidos a partir da integração elétrica da aceleração e, medição de transientes é geralmente melhor relatada por aceleração do que deslocamento ou velocidade. São também comercializados os do tipo IntegratedCircuitPiezoelectric, ICP, que possuem incorporado o circuito integrado de condicionamento do sinal. Este tipo de transdutor necessita, para isso, ser alimentado com uma corrente externa fornecida por outro dispositivo conhecido como SignalConditioner (figura 11). Em seguida, o sinal é enviado para o filtro antialiasing incorporado no CAD. Entretanto, o uso de acelerômetros é limitado devido ao relativo alto custo deste transdutor.



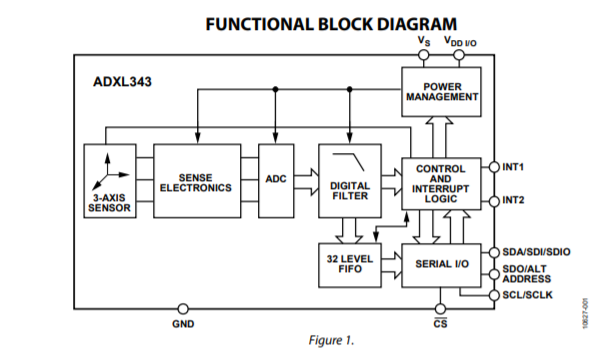
Neste trabalho foi analisado o comportamento do acelerômetro MEMS ADXL345 da AnalogDevice e um sensor sct013.

ADXL 345

O ADXL343 é um MEMS versátil de 3 eixos, saída digital e e facilmente configurado para faixas de trabalho diferentes. Faixas de medição e largura de banda selecionáveis além de detecção de movimento integrada e configurável o tornam adequado para a detecção aceleração em diversas aplicações. Robustez para 10.000 g de choque e uma ampla faixa de temperatura (-40 ° C a + 85 ° C) permitir o uso do acelerômetro mesmo em ambientes agressivos.



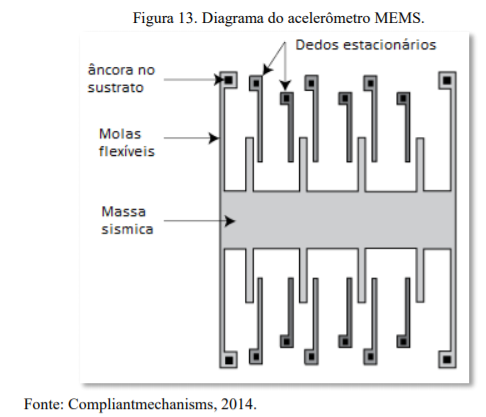
O ADXL345 mede a aceleração com alta resolução (13 bits) medição até ± 16 g. Os dados da saída digital são formatados de16 bits complementam e são acessíveis através de um SPI ou I2C. Ele pode medir a aceleração estática da gravidade em aplicações com sensor de inclinação, bem como a aceleração dinâmica resultante do movimento ou choque. Sua alta resolução (3,9 mg / LSB) permite a medição inclinação varia menos de 1,0 °.

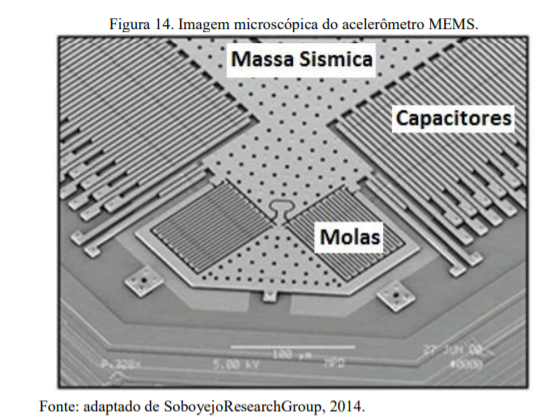


O ADXL343 possui uma faixa de medição selecionável de ± 2 g, ± 4 g, ± 8 g, ou ± 16 g. Ele mede a aceleração dinâmica resultante de movimento ou choque e aceleração estática, como a gravidade, que permite que o dispositivo seja usado como um sensor de inclinação.

O sensor é uma estrutura micro-usinada em superfície de polissilício construído em cima de uma pastilha de silicone. Molas de polissilício suspendem a estrutura sobre a superfície da bolacha e fornecer uma resistência contra forças devido à aceleração aplicada.

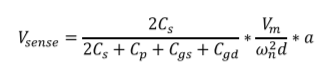
Agindo como um transdutor o qual permite transformar energia mecânica em energia elétrica, sendo sua fonte de informação a aceleração do sistema. É formado por três estruturas fundamentais, a massa sísmica, uma região de molas e as estruturas ou dedos capacitivos os dedos capacitivos se encontram nos dois lados da massa sísmica, e tem a mesma capacitância na posição de equilíbrio mecânico. Ao se aplicar uma aceleração na massa sísmica que conectada fisicamente com as placas dos condensadores, altera a distância ou a superfície entre os dedos capacitivos, alterando a capacitância do condensador. Essa capacitância é proporcional à aceleração aplicada no sistema (TEZ; AKIN, 2013).





A deflexão da estrutura é medida usando capacitores diferenciais que consistem em placas fixas independentes e placas fixadas a massa em movimento. A aceleração desvia a massa de prova e desequilibra o capacitor diferencial, resultando em uma saída do sensor cuja amplitude é proporcional à aceleração. Demodulação sensível à fase é usado para determinar a magnitude e a polaridade da aceleração.

O acelerômetro capacitivo MEMS é governado pela seguinte equação:

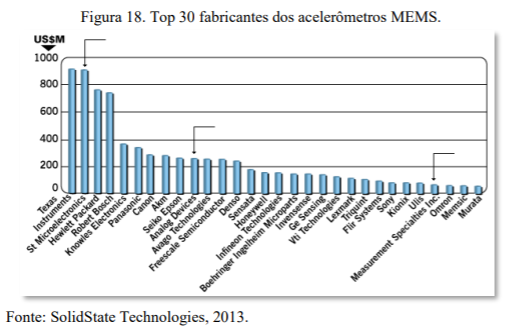


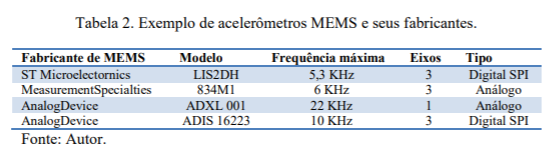
Onde Vsense é a tensão de saída, Cs é capacitância nos dedos capacitivos,Cp é a capacitância parasita, Cgs e Cgd são duas capacitâncias do transistor MOS, Vm é amplitude do sinal modulado, d é a distância entre os dedos capacitivos do sensor, wn é a frequência de ressonância mecânica do transdutor e, a é aceleração a ser medida.

Um acelerômetro típico MEMS tem frequência de ressonância 6 kHz, 1,5 µm na separação dos dedos capacitivos e uma sensibilidade de capacitância de 0,4 fF/g. Em um sinal de modulação de um 1 V de amplitude, a sensibilidade da tensão global é de cerca de 1 mV/g (JIANGFENG, 2004).

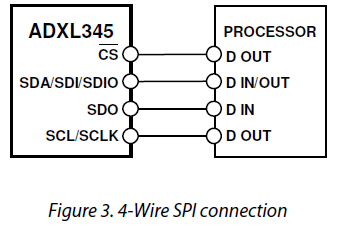
No caso da Manutenção Preditiva, as vibrações das máquinas rotativas geram sinais estacionários, que normalmente resultam da superposição de sinais senoidais com amplitudes e frequências diferentes. As características do acelerômetro que tem relação com esses dados são:

* Faixa de frequência (Bandwith) de medição.
* O limite de amplitude (Acceleration Range) que especifica o valor máximo de aceleração que pode ser medido com precisão, em “g”.





### 5.2.1 Spi No Adxl345



For SPI, either 3-wire or 4-wire configuration is possible, as shown in the connection diagrams in Figure 3 and Figure 4. Clearing the SPI bit in the **DATA\_FORMAT** register selects 4-wire mode while setting the SPI bit selects 3-wire mode. The maximum SPI clock speed is 5 MHz.

CS is the serial port enable line, and is controlled by the SPI master. It must go low at the start of transmissions and back high at the end as shown in Figure 5. SCLK is the serial port clock and is supplied by the SPI master. It is stopped high when CS is high, during period of no transmission. SDI and SDO are the serial data in and out respectively. Data should be sampled at the rising edge of SCLK.

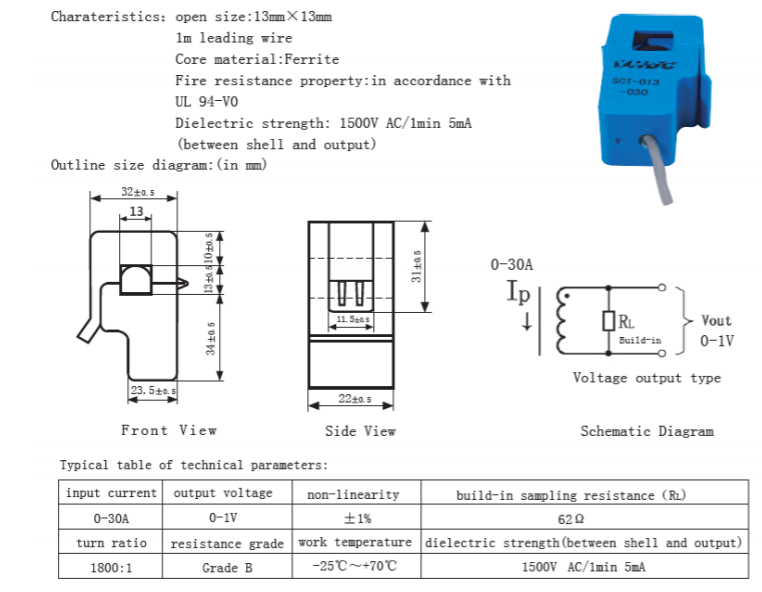
To read or write multiple bytes in a single transmission, the

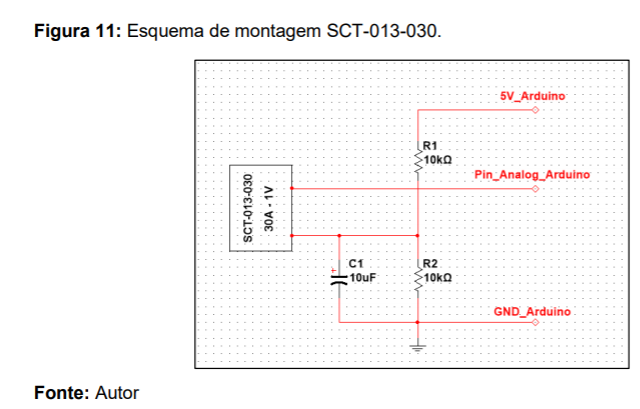
Multi-Byte bit, located after the R/W bit in the first byte transfer, must be set. After the register addressing and the first byte of data, continued clock pulses will cause the ADXL345 to point to the next register for read or write. Continued clock pulses will continue to shift the register that is pointed to until the clock pulses are ceased and CS is de-asserted. To perform reads or writes on different, non-sequential registers, CS must be de-asserted between transmissions and the new register must be addressed separately.

## **5.3 Sct-013**

Sensor SCT 013 (PODE IR NO APENDICE TB )

Sensor de corrente SCT-013-000 é uma ótima opção onde pode-se verificar correntes de até 100A e que não seja invasivo. Muito usado em projetos de automação residencial como medidores de corrente elétrica, proteção de motores AC/DC, iluminação e dentre outros.





## 6 METODOLOGIA

## **6.1 Desenvolvimento de sistema para aquisição de dados do ADXL345 E SCT-013**

O desenvolvimento do sistema objetivando a aquisição do sinal de corrente e vibração. Se se deu através da leitura dos registros do ADXL345 e a partir da leitura dos valores obtidos da porta analógica do ESP8266, na qual o SCT-013 é conectado. A implementação do sistema foi desenvolvida através da IDE do Arduino, que utiliza linguagem C/C ++, proporcionando facilidade de uso e implementação.

Após a concretização do sistema de monitoramento de vibração e corrente (APENDICE X), foram realizados os primeiros testes para avaliar o correto funcionamento do sistema. Segue a foto da visualização dos resultados obtidos da captação de sinais dos sensores ADXL345+SCT-13 controlados do ESP 8266, tendo a visualização dos resultados no Serial da IDE(I[ntegrated Development Environment](https://en.wikipedia.org/wiki/Integrated_development_environment)) do Arduino.

FOTO

Após a implementação da rotina no sistema, foram realizados testes para obter dados das seguintes condições de funcionamento:

* + 1. Normal.
    2. Desbalanceamento do rotor.
    3. Fase única.
    4. Falhas no solo.

O Instituto Florence de Ensino disponibilizou suas instalações laboratoriais para que pudessem ser realizados os testes. A obtenção de dados para alimentar as etapas subsequentes do desenvolvimento do projeto. Foto x e y mostram as instalações de onde ocorreram os testes.

FOTO FOTO

Para armazenar os dados em um arquivo no formato CSV, foi implementada uma rotina no software Processing (anexo 888), em períodos de tempo determinados (10 segundos), e organizados em colunas em uma planilha.

A disposição das leituras do sistema de monitoramento de vibração e corrente gravou os dados em formato CSV, de acordo com a foto 81

Foto 81

A aquisição dos dados nas condições anteriormente citadas foi realizada. E desta forma foi obtido o material fundamental para dar sequência à etapa de implementações em computação na nuvem.

## **6.2 Instâncianto De Serviços na Cloud**

Explicar a seleção de ferramentas disponibilizadas na infraestrutura da clouds IBM

## **6.3 Utilização De Watson Studio**®.

## **6.4 SPSS MODELER**

Seleção de blocos e geração

### 6.4.2 Utilização de Blocos

### 6.4.3 Type

### 6.4.4 Svm Node

### 6.4.5 Table

### 6.4.6 Analyses

## **6.6** **Discussão dos resultados obtidos no estudo do modelo físico do sistema de monitoramento de aceleração de vibração e corrente, para aplicação na IBM® Cloud**

## **6.7 Geração Do Modelo Preditivo**

## **6.8 Discussões Sobre O Modelo**

# **7 Conclusão e Trabalhos Futuros**

### 7.1 Conclusões

### 7.2 Trabalhos Futuros

# **9 APÊNDICE**

APENDICE 0 PROCESSING

(várias fontes , editar)

Processing é uma biblioteca gráfica de [código aberto](https://en.wikipedia.org/wiki/Open-source_software) e [um ambiente de desenvolvimento integrado](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) criado para comunidades de artes eletrônicas, [novas mídias](https://en.wikipedia.org/wiki/New_media_art) e [design visual](https://en.wikipedia.org/wiki/Visual_design), com o objetivo de ensinar a não programadores os fundamentos da [programação](https://en.wikipedia.org/wiki/Computer_programming) de [computadores](https://en.wikipedia.org/wiki/Computer_programming) em um contexto visual.

O processamento usa a [linguagem Java](https://en.wikipedia.org/wiki/Java_(programming_language)) , com simplificações adicionais, como classes adicionais e funções e operações matemáticas aliadas. Além disso, ele também possui uma interface gráfica do usuário para simplificar o estágio de compilação e execução.

A linguagem Processing e o IDE foram os precursores de outros projetos, incluindo [Arduino](https://en.wikipedia.org/wiki/Arduino) , [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) e p5.js

Além disso, a ideia da comunicação entre Processing e Arduino é basicamente assim: o Processing pode ser utilizado de duas formas, fazendo o Arduino executar determinada tarefa ou atuando como uma representação gráfica e visual daquilo que está acontecendo no seu protótipo em protoboard.

Quando o Processing atua como um agente que vai “comandar” o Arduino, então temos que fazer com que o Arduino apenas realize a leitura da porta serial (onde é feita a comunicação entre eles).

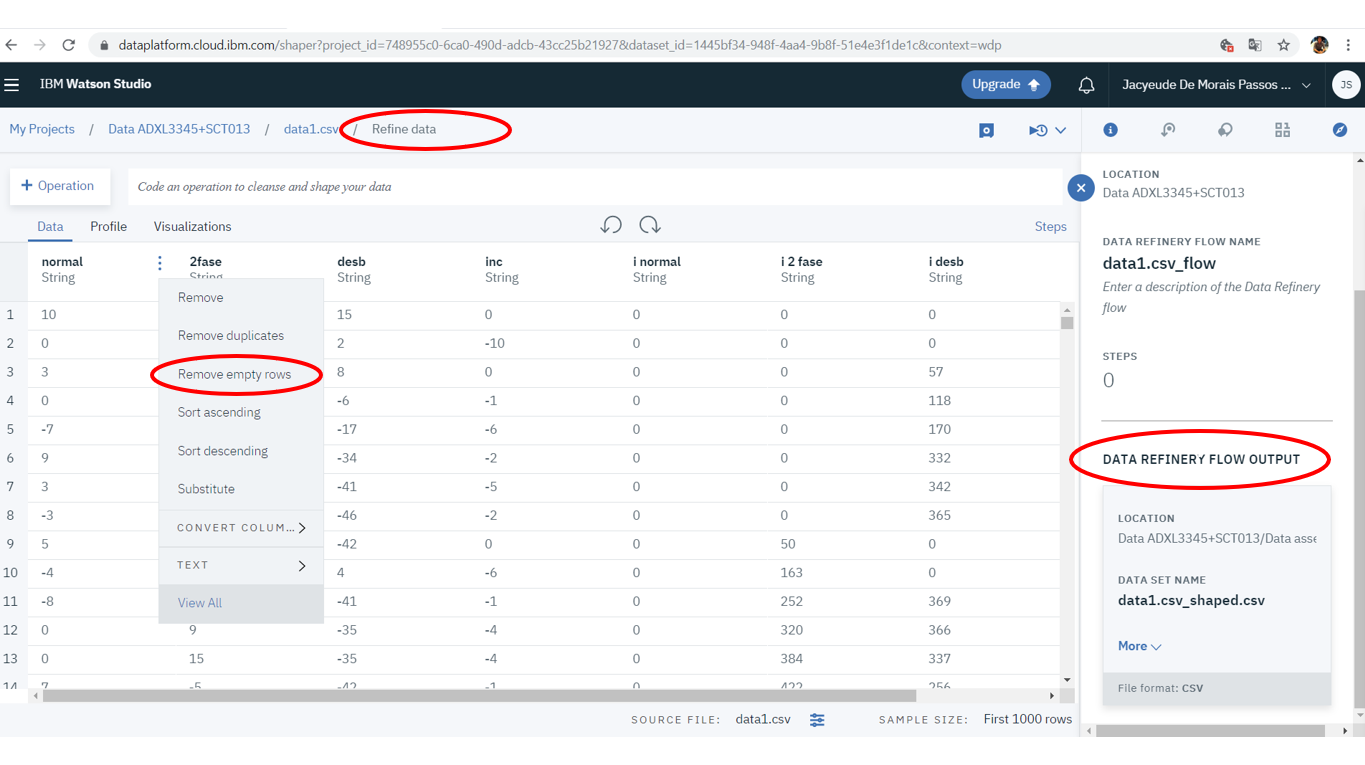
Caso o Arduino receba algum valor, enviado pelo processing, ele realiza determinada função, ou seja, ele só fará uma ação X, ligar um led, por exemplo, se, o Processing mandar o valor correto, caso contrário isso não ocorrerá.

Agora quando o processing atua como uma representação visual daquilo que está ocorrendo na protoboard, temos basicamente o contrário. Agora é o Arduino que vai enviar pela serial o que está acontecendo na protoboard e a partir desse dado enviado, uma imagem (ou uma janela de cores) é mostrada no processing

APÊNDICES 95

**Refinaria de Dados**

Através da Refinaria de Dados(Data Refinery) é possível limpar e modelar dados tabulares com um editor de fluxo gráfico, havendo possibilidade também de usar operações, funções e operadores lógicos da biblioteca *dplyr R(..)*, para executar modificações iniciais inerentes a organização e disposição dos dados importados. Ao limpar os dados*(figura XZ)*, você corrige ou remove dados incorretos, incompletos, formatados ou duplicados incorretamente. Podendo personalizar os dados, filtrando, classificando, combinando ou removendo colunas e executando operações. Na figura XZ é possível notar a seleção da opção “Remove empty rows”, para remover os espaços vazios da coluna selecionada.



É possível criar um fluxo da refinaria de dados como um conjunto de operações ordenadas, para ser realizado sob demanda quando houverem novas importações de dados. A Refinaria inclui uma interface gráfica para criar um perfil de seus dados, para validá-los e possui opção de mais de 20 gráficos personalizáveis que oferecem perspectiva e insights sobre databases importadas.

6.4.1 Especificação do formato de dados

Há de ser notado que há uma limitação quanto a especificidade de formatos que podem ser importados, sendo eles:

- Avro, CSV, JSON, Parquet ou dados de texto

- Tabelas em fontes de dados relacionais

A Refinaria de dados opera em um subconjunto de linhas de amostra no conjunto de dados, realizando importação de arquivos de grande tamanho e volume de dados. No entanto, quando você executa uma tarefa de fluxo da refinaria de dados, todo o conjunto de dados do arquivo é processado. De acordo com IBM Cloud Knowledge Center.

6.4.2 Validação de dados

Após adicionados ao Data Refinery, é possível validar os dados. Normalmente, isso deve ser feito em vários pontos do processo de refinamento.

Para validar seus dados, o IBM Cloud Knowledge Center, da os seguintes passos:

1. Na refinaria de dados, clique na guia perfil.
2. Revise as métricas para cada coluna.
3. Tome as ações apropriadas, frente a frequência e estatística dos dados.
4. Salvar o fluxo gerado na Refinaria de Dados.

Sobre frequência e distribuição estatística, tem-se no IBM Cloud Knowledge que:

* Frequência é o número de vezes que um valor ou um valor em um intervalo especificado, ocorre. Cada distribuição de frequência mostra a contagem de valores exclusivos em uma coluna. Revisar a distribuição de frequência para encontrar anomalias nos dados é preciso.

Nas colunas, é possível personalizar o número de posições (agrupamentos) que é desejado para visualizar.

* Estatísticas são uma coleção de dados quantitativos. As estatísticas para cada coluna mostram o mínimo, o máximo, a média e o número de valores exclusivos nessa coluna.

Dependendo do tipo de dados de uma coluna, as estatísticas para cada coluna pode variar ligeiramente. Por exemplo, as estatísticas para uma coluna do tipo *Integer*(inteiro) de dados têm valores mínimo, máximo e médio, enquanto as estatísticas para uma coluna do tipo de dados *String (corda)*possuem valores de comprimento mínimo, comprimento máximo e comprimento médio.

# **8 REFERÊNCIAS**

[99] Lorena Teixeira Marques**. Análise Computacional Do Motor De Indução Trifásico: Regime Transitório E Permanente.** UNIVERSIDADE DE SÃO PAULO,São Carlos 2009

[88]Vinícius Augusto Diniz Silva. **Detecção de falhas em motores elétricos**

**através das máquinas de vetores de suporte**. UNIVERSIDADE ESTADUAL DE CAMPINAS, Campinas, 2012.

[40] James Surowiecki, The Wisdom of Crowds, why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations, Anchor Books, 2004, pp 66-83.

[47]Daniel Larose and Chantal Larose, **Discovering Knowledge in Data: An Introduction to Data Mining**, John Wiley & Sons, 2014.

[48] Jiawei Han, Micheline Kamber and Jian Pei, **Data Mining Concepts and Techniques 3rd Edition**, Elsevier Inc., 2012, pp 23

[49] Ian H. Witten and Eibe Frank, **Data Mining Practical Machine Learning Tools and Techniques**, 3rd Edition, Elsevier Inc. 2011.

[50] Tom M. Mitchell, **Machine Learning**, McGraw-Hill Science, 1997, pp 2

[51] Kevin P. Murphy, **Machine Learning, A Probabilistic Perspective**, Kevin P. Murphy, The MIT Press, 2012.

[52] Christopher M. Bishop, **Pattern Recognition and Machine Learning**, Springer, 2006.

[53] Noam Nisan and Shimon Schocken, **The Element of Computing Systems Building a Modern Computer from First Principles**, MIT press 2005, pp 57-58.

[54] Xin Liu et al, **Computational Trust Models and Machine Learning**, CRC Press, 2015.

[55] Peter Harrington, **Machine learning in Action**, Manning Publications, 2012.

[56] Sameer Wadkar, Madhu Siddalingaiah, **Pro Apache Hadoop**, 2nd Edition, Apress, 2014.

[74]. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, **Mastering Cloud Computing**, Morgan Kaufmann, USA, May 2013.

[76] Caesar Wu, Rajkumar Buyya, **Cloud Data Centers and Cost Modeling**, Morgan Kaufmann, 2015.

[797] Rajkumar Buyya e Kotagiri Ramamohanarao ,**Big Data Analytics = Machine Learning + Cloud Computing** , University of Melbourne January 2016

[9999]<https://introduceti.com.br/blog/iaas-paas-saas-conheca-os-modelos-fundamentais-de-cloud-computing/>