

CENTRO UNIVERSITÁRIO UNIFACVEST
CURSO DE CIÊNCIA DA COMPUTAÇÃO
RODRIGO HASSE DA SILVA

**SGPMI – SISTEMA GERENCIADOR DE PERFORMANCE DE
MÁQUINAS INDUSTRIAIS**

LAGES – SC

2020

RODRIGO HASSE DA SILVA

**SGPMI – SISTEMA GERENCIADOR DE PERFORMANCE DE
MÁQUINAS INDUSTRIAIS**

Trabalho de conclusão de curso apresentado à Banca Examinadora da Unifacvest como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação.

Coordenador: Prof. Me Márcio José Sembay

Orientadores: Prof. Esp. Jaison Luiz Granemann Teixeira, Prof. Esp. Jean Carlos Macedo e Prof. Esp. Willen Leollato Carneiro

Lages, SC __/__/2020. Nota _____

Marcio José Sembay

Coordenador do Curso de Ciência da Computação

LAGES – SC

2020

RESUMO

O conceito de Internet das Coisas (IoT), está cada vez mais presente no dia-a-dia das pessoas e empresas, sendo utilizado em muitos tipos de projetos em diversos níveis de complexidade. Desta forma, este estudo tem como objetivo descrever o projeto e o desenvolvimento de um sistema gerenciador de performance de máquinas industriais, sendo baseado em estudos e pesquisas. O trabalho apresenta um levantamento de informações sobre a importância do uso do conceito de internet das coisas na indústria 4.0 e a necessidade da coleta e organização de dados para auxílio nas tomadas de decisões gerenciais para justificar o seu desenvolvimento. O trabalho também apresenta os conceitos de sistemas embarcados e de tempo real utilizados no projeto, assim como a apresentação das ferramentas utilizadas no desenvolvimento do sistema. O projeto tem como objetivo construir um equipamento que consiga capturar os eventos de paradas de máquinas e disponibilizar ao usuário informações como hora inicial e hora final, tempo, usuário e o motivo pelo qual ela ocorreu. Assim, o projeto é constituído por um sistema embarcado utilizando uma plataforma micro controlada, um sistema web onde os dados são apresentados ao usuário para serem interpretados e mantidos e um aplicativo móvel desenvolvido para a plataforma Android, onde os dados das paradas são mostrados em tempo real. Para realizar os objetivos do trabalho foi necessário realizar a pesquisa em artigos, livros, trabalhos acadêmicos e apostila com os conhecimentos sobre Arduino e desenvolvimento de sistemas web, principalmente.

Palavras-chave: Maquinário Industrial, IoT, Aplicativos Móveis.

ABSTRACT

The concept of Internet of Things (Iot) is increasingly present in the daily lives of people and companies, being used in many types of projects at different levels of complexity. Thus, this study aims to design and develop a performance management system for industrial machines, based on studies and research. The study presents a survey of information on the importance of using the internet of things concept in industry 4.0 and the need to collect and organize data to assist in the management decisions to justify its development. The work also presents the concepts of embedded systems and real time used in the project, as well as the presentation of the tools used in the development of the system. The project aims to build equipment that can capture machine downtime events and provide the user with information such as start time and end time, time, user and the reason why it occurred. Thus, the project consists of an embedded system using a micro controlled platform, a web system where data is needed by the user to be interpreted and a mobile application developed for an Android platform where the data of the stops are in real time. To accomplish the objectives of the work it was necessary to carry out research in articles, books, academic papers and handout with the knowledge about Arduino and web systems development mainly..

Keywords: *Industrial Machinery, IoT, Mobile Applications.*

RESUMEN

El concepto de Internet de las Cosas (Iot) está cada vez más presente en el día a día de personas y empresas, siendo utilizado en muchos tipos de proyectos en diferentes niveles de complejidad. Por tanto, este estudio tiene como objetivo diseñar y desarrollar un sistema de gestión del rendimiento para máquinas industriales, basado en estudios e investigaciones. El estudio presenta una encuesta de información sobre la importancia de utilizar el concepto de Internet de las cosas en la industria 4.0 y la necesidad de recopilar y organizar datos para ayudar en las decisiones de gestión para justificar su desarrollo. El trabajo también presenta los conceptos de sistemas embebidos y tiempo real utilizados en el proyecto, así como la presentación de las herramientas utilizadas en el desarrollo del sistema. El proyecto tiene como objetivo construir equipos que puedan capturar eventos de tiempo de inactividad de la máquina y proporcionar al usuario información como la hora de inicio y la hora de finalización, la hora, el usuario y la razón por la que ocurrió. Así, el proyecto consta de un sistema embebido mediante una plataforma microcontrolada, un sistema web donde los datos son necesarios por parte del usuario para ser interpretados y una aplicación móvil desarrollada para una plataforma Android donde los datos de las paradas son en tiempo real. Para lograr los objetivos del trabajo fue necesario realizar investigaciones en artículos, libros, trabajos académicos y folletos con los conocimientos sobre Arduino y desarrollo de sistemas web principalmente.

Palabras clave: Maquinaria industrial, IoT, Aplicaciones móviles.

LISTA DE FIGURAS

Figura 1: Evolução dos dispositivos conectados à internet.....	10
Figura 2: Sistema de tempo real acoplado ao ambiente	12
Figura 3: Interface IDE Arduino	14
Figura 4: Módulo NodeMCU.....	16
Figura 5: IDE do Visual Studio 2019.....	18
Figura 6: Diagrama de Atividade	24
Figura 7: Diagrama de Caso de Uso	25
Figura 8: Diagrama de Classe	26
Figura 9: Tela de Login	27
Figura 10: Tela de Listagem de Máquinas	28
Figura 11: Tela de Login do Sistema Web.....	29
Figura 12: Tela de Consulta de Máquinas.....	29
Figura 13: Tela de Consulta de Paradas	30

LISTA DE ABREVIATURAS E SIGLAS

API	-	Application Programming Interface
CLP	-	Controlador Lógico Programável
HTTP	-	<i>Hypertext Transfer Protocol</i>
IDE	-	<i>Integrated Development Enviroment</i>
IIOT	-	<i>Indutrial Internet of Things</i>
IoT	-	<i>Internet of Things</i>
IP	-	<i>Internet Protocol</i>
LINQ	-	<i>Language-Integrated Query</i>
REST	-	<i>Representational State Transfer</i>
SQL	-	<i>Structured Query Language</i>
TCP	-	<i>Transmission Control Protocol</i>
UML	-	<i>Unified Modeling Language</i>
USB	-	<i>Universal Serial Bus</i>

LISTA DE QUADROS

Quadro 1: Vantagens do ESP8266.....	15
Quadro 2: Cronograma	31

SUMÁRIO

1	INTRODUÇÃO	4
2	JUSTIFICATIVA.....	5
3	OBJETIVOS.....	6
3.1	Objetivos Gerais.....	6
3.2	Objetivos específicos.....	6
4	REFERENCIAL TEÓRICO	7
4.1	Indústria 4.0	8
4.2	Internet das Coisas (IoT)	9
4.3.	Internet das Coisas na Indústria.....	10
4.4	Sistemas de Tempo Real	11
4.5	Sistemas Embarcados.....	12
4.6	API Rest.....	13
4.7	Arduino.....	13
4.8	ESP8266	14
4.9	NODEMCU	15
4.10	Linguagem C#.....	16
4.11	IDE do Visual Studio 2019.....	17
4.12	Xamarin	18
4.13	Vue.js.....	18
5	METODOLOGIA	20
5.1	Documentação	20
5.2	Natureza da Pesquisa.....	20
5.3	Tipo da Pesquisa	20
5.4	Técnicas de Pesquisa.....	20
5.5	Coleta de Dados	21
6	PROJETO.....	22
6.1	Hardware	22
6.2	Desenvolvimento	22
6.3	Pré-requisitos	22
6.4	Diagramas UML	23
6.4.1	Diagrama de Atividades	23
6.4.2	Diagrama de Caso de Uso	24
6.4.3	Diagrama de Classes.....	25

6.5	Telas do Sistema	26
7	CRONOGRAMA	31
8	CONCLUSÃO.....	32
9	PROJETOS FUTUROS.....	33
10	REFERÊNCIAS	34
11	APÊNDICES	36
11.1	APÊNDICE A – Sketch Esp8266	36
11.2	APÊNDICE B – Código Busca Resumo por Motivo.....	38
11.3	APÊNDICE C – Código Tela Consulta de Máquinas para APP Android	39
11.4	APÊNDICE D – Trecho do Código Busca Paradas na Interface Web	41

1. INTRODUÇÃO

Os avanços das tecnologias digitais estão cada vez mais presentes no cotidiano das pessoas, tendo um impacto cada vez maior em todas as estruturas econômicas e sociais. Dessa forma, uma das tecnologias digitais que mais se destaca é a Internet das Coisas, que é responsável por automatizar setores inteiros da economia com base na comunicação máquina a máquina chegando a produção industrial, tendo chamado atenção de governos e da iniciativa privada em todas as partes do mundo (MAGRANI, 2018).

Arduino é uma plataforma que tornou possível o desenvolvimento de sistemas embarcados, com placas de baixo custo e de fácil utilização e integração, sendo que conta com uma grande quantidade de módulos que facilitam a implementação desses sistemas. Entre esses módulos podemos citar sensores, atuadores e módulos de comunicação (OLIVEIRA, 2017).

O C# é uma linguagem orientada a objetos que possibilita o desenvolvimento de aplicativos que rodam no *.Net Framework*. Sendo ela uma linguagem que possibilita a um programador já adaptado a outras linguagens como C, C++ e Java uma boa adaptação em tempo curto, sendo que em alguns casos simplifica a complexidade dessas linguagens (MICROSOFT, 2020)

2. JUSTIFICATIVA

A IIoT (Internet das Coisas Industrial) é responsável por uma mudança significativa na fabricação de produtos, permitindo oportunidades para que as empresas ofereçam novos serviços, otimizem suas produções e permitindo também a coleta de uma grande quantidade de dados, em alta velocidade e com grande eficiência. A IIoT tem melhorado significativamente a conectividade, a eficiência, a escalabilidade, a otimização de tempo e de custos para as indústrias. As redes de dispositivos inteligentes da IIoT possibilitam que as indústrias obtenham dados de toda sua organização, de forma integrada, possibilitando uma visão ampla sobre a gestão da empresa e auxiliando na tomada das melhores decisões (SISINNI et al., 2018).

Atualmente existem vários dispositivos de automação industrial, porém poucas empresas aderem ao seu uso por conta do seu alto custo. Assim, esse estudo apresenta um projeto que tem como objetivo baratear esse custo utilizando um módulo ESP8266, desenvolvida no ambiente Arduino, juntamente com um módulo de sensor detector de tensão que torna possível a coleta de informações da máquina e apresentá-los ao usuário através de um aplicativo Android e uma interface web, ambas conectadas à internet.

3. OBJETIVOS

3.1. Objetivos Gerais

Criar um sistema de monitoramento de paradas de máquinas, que vai possibilitar o acompanhamento em tempo real do estado da máquina (ligado/ desligado), o motivo e o tempo de uma possível parada inesperada, através de um aplicativo *mobile* e uma interface *web*.

3.2. Objetivos específicos

Abaixo serão listadas as principais funcionalidades a serem desenvolvidas neste projeto:

- a) monitorar o estado da máquina;
- b) enviar os dados da mudança do estado da máquina para uma *Web API* que manterá os dados na nuvem;
- c) disponibilizar ao gestor consultas do status das máquinas e o tempo das paradas em tempo real.

4. REFERENCIAL TEÓRICO

O trabalho consiste em um sistema de controle de parada de máquinas industriais, onde será possível ao gestor acompanhar o status e o tempo de paradas de determinada máquina (ligada ou desligada) da sua linha de produção através do seu smartphone e/ou aplicação *web*. Sendo a comunicação sem fio de fundamental importância para que a implementação do conceito de Indústria 4.0 se torne viável. Uma das tecnologias base para este conceito é a Internet das Coisas (STANKOVIC, 2014).

O sistema foi fortemente baseado nos conceitos de IoT, sendo necessário coletar a informação de eventos de paradas em tempo real e disponibilizar esses dados na nuvem. O uso dos conceitos de IoT se explica pelo aumento do número de *smartphones* utilizados mundialmente, sendo estimado que os números ultrapassem os vários bilhões a 1 trilhão de aparelhos conectados, o que altera de forma significativa a forma como monitoramos e otimizamos os ativos e as atividades industriais atingindo todos os setores da indústria, incluindo fabricação e infraestrutura (SCHWAB, 2016).

Desta forma, é utilizado um sistema embarcado, desenvolvido na plataforma da Arduino, em um micro controlador ESP8266 disponível no módulo NodeMCU que atua em conjunto com o rele de *interface Finder 24vac/vdc 1naf 6a* para coleta e disponibilização das informações em tempo real do status da máquina.

Os dados coletados são mantidos na nuvem por uma *API Rest* desenvolvida em C# na ide do *Visual Studio 2019* utilizando banco de dados *SQL Server*. Segundo Wollschlaeger et al. (2017), com a utilização de tecnologias de automação, como redes *Ethernet Industrial Modbus TCP/IP* e o padrão *OPC UA*, e de Tecnologias da Informação, como serviços *Web* no padrão *REST*, e protocolos de IoT, possibilita a disponibilização de serviços em nuvem que podem ser utilizados como serviços de alto nível e aplicações vinculadas a tarefas presentes no contexto de IIoT e I4.0, como monitorar e supervisionar sistemas, controlar e otimizar processos e etc.

O aplicativo do Gestor foi desenvolvido em Xamarin. Para Araújo (2017), o Xamarin surgiu como uma inovação para o desenvolvimento móvel, tornando mais fácil e rápido o desenvolvimento de aplicativos móveis para diversas plataformas. Juntamente com o *Visual Studio* possibilita o desenvolvimento utilizando o C# e plataforma .Net, tornando possível o desenvolvimento de sistemas híbridos para as principais plataformas como *Android*, *IOS* e *Windows Phone*. Sendo assim, o principal diferencial do Xamarin é que o aplicativo final não é um aplicativo híbrido e sim aplicativos nativos gerados a partir de uma

mesma estrutura de código.

4.1. Indústria 4.0

Segundo SENAI et al. (2019), a Indústria está mudando desde a primeira revolução industrial, quando a descoberta das leis da termodinâmica possibilitou o surgimento das primeiras máquinas a vapor, com isso no século XIX vieram as locomotivas a vapor que revolucionaram a forma de transporte de produtos e pessoas. Já a segunda revolução aconteceu em meados dos séculos XIX e XX com o surgimento da eletricidade e eletromagnetismo trazendo inovações importantes como os automóveis, telefones e aparelhos de TV. Já a terceira revolução, iniciada na década de 1970, guiada em grande parte pela evolução no campo das telecomunicações e programação lógica com o surgimento dos CLP (Controlador Lógico Programável) e da internet.

Ainda segundo o autor, em 2013 surge na Alemanha o conceito de indústria 4.0, também conhecida como manufatura avançada, com base em sistemas *cyber* físicos, na internet das coisas e internet dos serviços, possibilitando uma interconectividade entre máquinas, pessoas e processos industriais o que vem de encontro a uma tendência global de inovação tecnológica com produtos inteligentes e acessibilidade a dados em tempo real, derrubando barreiras entre o mundo virtual e o mundo físico.

Para Schwab (2016), a quarta revolução industrial permite a criação de um mundo onde sistemas físicos e virtuais de fabricação trabalham juntos de forma universal e flexível, criando assim o conceito de “fábricas inteligentes”, com produtos personalizáveis e a criação de novos modelos operacionais.

Conforme Morais e Monteiro (2019), as principais oportunidades enxergadas pelos entusiastas da indústria 4.0 são:

- Aumento da competitividade;
- Redução de riscos, integrando as necessidades do consumidor aos processos de produção, melhorando falhas e a qualidade no processo de fabricação;
- Melhorar os setores de pesquisa e desenvolvimento, compras e produção, armazenagem e logística, favorecendo assim setores de venda e serviços;
- Melhorias nos setores de infraestrutura de informação.

4.2. Internet das Coisas (IoT)

Para Magrani (2018), existem divergências para definir um conceito de internet das coisas (*internet of things* – IoT), mas de modo geral, pode-se dizer que se trata de um ambiente onde objetos físicos estão interconectados com a internet, utilizando sensores e outros componentes para criar uma estrutura computacional, tendo como principal objetivo facilitar tarefas e introduzir soluções efetivas a vida das pessoas nos processos do dia a dia.

Já para Schwab (2016), IoT é uma das principais pontes entre aplicações físicas e digitais criadas pela quarta revolução industrial, relacionando coisas (produtos, serviços, lugares etc.) e as pessoas, através de diversas plataformas e tecnologias conectadas.

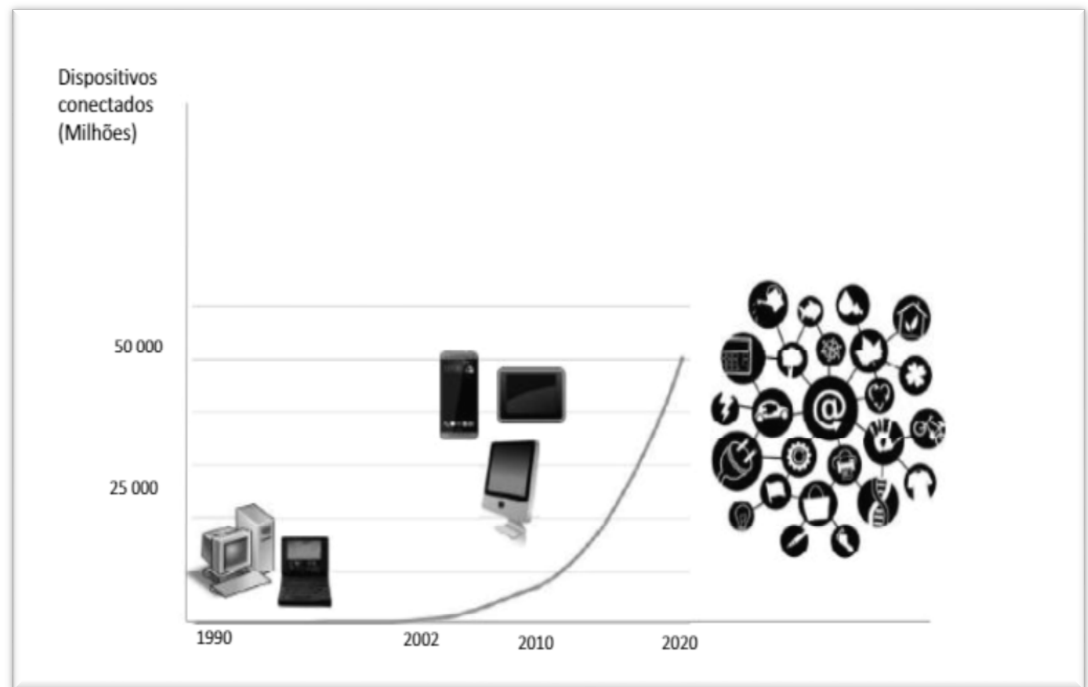
Conforme cita Sinclair (2018), IoT é que uma evolução da internet e que suas subdivisões, terão efeitos significativos em termos de negócio e ditará as bases da economia de resultados. Essa evolução é responsável em fazer a internet chegar aos objetos físicos e transformar todos os setores envolvidos no processo. Desta forma, a IoT se transformará em parte integrante de empreendimentos de negócios, produtos de consumo, comercial, industrial e de infraestrutura.

Ainda conforme Sinclair (2018) graças à internet das coisas, nossas casas deixaram de ser irracionais e passaram a ser automatizadas e depois inteligentes, aumentando assim sua segurança, melhorando a gestão de energia e conforto. Já em nosso corpo utiliza-se produtos IoT para monitoramento constante do organismo e meio ambiente. O autor cita também o aumento da autonomia dos veículos com o uso de IoT, em alguns casos, tornando-os condutores de si mesmos de forma menos suscetível a falhas que um ser humano, o que conforme a estimativa, pouparia trinta mil vidas por ano só nos Estados Unidos.

Segundo Oliveira (2017), o conceito de internet das coisas significa mais que apenas acender lâmpadas ou ligar “coisas” através de um dispositivo móvel, mas torná-las inteligentes, tendo a capacidade de coletar informações dos ambientes ou redes as quais estão conectadas. Sendo assim, a sua implantação está revolucionando a forma como nos relacionamos com os objetos ao nosso redor, transformando segurança, energia, meio ambiente, trânsito, mobilidade e logística.

Já Coelho (2017) destaca a evolução da internet, que evoluiu de um meio de conectar computadores através do envio de mensagens entre eles, para cada vez mais desempenhar diversas funcionalidades, o que acarretou em um aumento no número de dispositivos conectados, tornando possível a revolução do IoT que estamos vivendo, que conecta todo tipo de objeto a rede. Essa evolução é representada na imagem abaixo:

Figura 1 – Evolução dos dispositivos conectados à internet



Fonte: COELHO, 2017, p. 3

4.3. Internet das Coisas na Indústria

Para Sinclair (2018), a indústria também está passando por uma revolução em função da internet das coisas, já que ela possibilita que os fabricantes consigam lançar uma maior quantidade de produtos a um custo mais baixo. Sensores são utilizados na indústria de petróleo e gás para se tornarem mais eficientes na extração, no processamento e distribuição de seus produtos. A mineração tem melhorado a sua segurança e rendimento com a utilização de equipamentos inteligentes que são capazes de trabalhar 24 horas por dia. Já a agricultura tem programado suas safras usando *machine learning* (aprendizado de máquina), aumentando assim a produtividade utilizando dados fornecidos por sensores conectados a terra e plantações, e por serviços de monitoramento do clima.

Segundo Magrani (2018), a internet das coisas tem sido vista com bons olhos pela indústria, podendo vir a se tornar um importante fator para a economia nas próximas décadas. O setor tem a estimativa de crescimento econômico global de US\$ 11 trilhões até 2025, com a previsão de cem bilhões de dispositivos inteligentes conectados até lá.

Ainda segundo Magrani (2018), a indústria brasileira já tem conhecimento do potencial da IoT, sendo o aumento da produtividade dos funcionários e a redução de custos e otimização no uso de seus bens as principais vantagens esperadas pelos executivos

brasileiros.

O Brasil situa-se na posição de número 57 do índice de competitividade mundial (World Competitiveness Yearbook) de 2016.¹⁹² Esse é o principal relatório anual sobre a competitividade dos países publicado pelo *International Institute for Management Development* (IMD) desde 1989. O anuário compara o desempenho de 63 países baseando-se em mais de 340 critérios que medem diferentes aspectos da competitividade.¹⁹³ Quanto ao índice global de inovação, o país está na posição de número 69.¹⁹⁴ Esse indicador mede o nível de inovação de cada país e é resultado de uma colaboração entre a Universidade Cornell, a *Insead* e a Organização Mundial de Propriedade Intelectual (Wipo). O índice global de inovação é parte de uma grande pesquisa que apresenta, por exemplo, os resultados das empresas, bem como a habilidade do governo de encorajar e suportar inovação por meio de políticas públicas. (MAGRANI, 2018, p. 76)

Conforme Silva e Stevan Jr. (2015), é inegável a necessidade de automação dos diversos processos do meio industrial, como por exemplo a velocidade de um motor que necessite ser controlado através do monitoramento de sensores. Porém, em alguns casos esses sensores precisam ser de alto desempenho e alto custo. Entretanto, em alguns casos podemos reduzir radicalmente o custo utilizando sensores micro controlados. Assim, a plataforma Arduino se apresenta como uma solução de desenvolvimento rápida, podendo ser facilmente embarcada e aplicada em processos com complexidade menor, como em pequenas empresas.

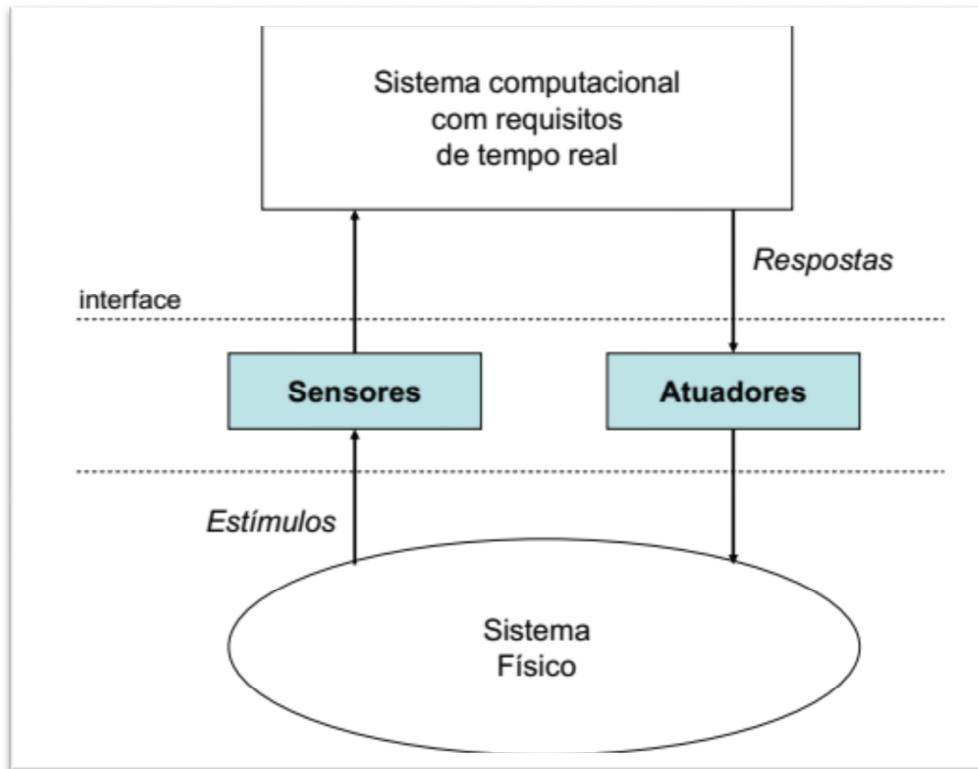
4.4.Sistemas de Tempo Real

Segundo Oliveira (2018), sistemas de tempo real são sistemas que se submetem a requisitos temporais não triviais, executando tarefas com um prazo máximo de execução ou funcionalidade que devem ser executadas em um determinado período. Desta forma, sistemas de tempo real não devem ser precisos apenas no que diz respeito a lógica e matemática, mas também devem obedecer aos requisitos de tempo, sendo esses requisitos temporais definidos em razão do ambiente físico em torno do sistema computacional, não sendo definidos na implementação e sim em fase de especificação.

Ainda segundo Oliveira (2018) um sistema de tempo real deve trabalhar em função do seu ambiente. Normalmente, esse tipo de sistema reage a um estímulo do ambiente, respeitando os requisitos temporais definidos na especificação. Sendo assim, o processamento é ativado através de um gatilho, o qual pode significar a ocorrência de um evento ou a medição de um valor ou ainda simples passagem do tempo. Requisitos de tempo podem

estabelecer prazos máximos para retornos do sistema a determinados estímulos. No caso de um processamento ser terminado fora do prazo, tem-se uma falha no sistema.

Figura 2 – Sistema de tempo real acoplado ao ambiente



Fonte: OLIVEIRA, 2018, p. 15

4.5. Sistemas Embarcados

Para Almeida, Moraes e Seraphim (2016), sistemas embarcados são sistemas micro processados que possuem uma função específicas, estando presentes praticamente em todos os ambientes tendo muitas funcionalidades. Esses sistemas têm a característica de possuir recursos computacionais e de interface bastante limitados, por isso utiliza técnicas e recursos diferentes das usadas na computação comum.

Ainda para Almeida, Moraes e Seraphim (2016), outra característica desses sistemas é que cada micro controlador possui uma arquitetura de hardware e um padrão de periféricos específico. Os autores citam como exemplo ainda um sistema que utilize o mesmo micro controlador, mas com um projeto eletrônico diferente, dessa forma, não sendo possível o reaproveitamento do código.

4.6.API Rest

Conforme Saudate (2014), REST significa *Representational State Transfer* (ou Transferência de Estado Representativo), sendo um modo de desenvolvimento de *web services*, sendo originário da tese de doutorado de Roy Fielding, coautor do protocolo HTTP (*Hypertext Transfer Protocol*), um dos protocolos mais utilizado no mundo. Dessa forma o REST é orientado pelas boas práticas do uso de HTTP entre outros preceitos. Entre essas boas práticas estão:

- Uso adequado dos métodos HTTP;
- Uso adequado de URL's;
- Uso de códigos de status padronizados para representação de sucessos ou falhas;
- Uso adequado de cabeçalhos HTTP;
- Interligações entre vários recursos diferentes.

Para Lecheta, (2015), No REST as requisições são feitas utilizando os métodos do HTTP como o *Post, Get, Delete e Put*, gerando uma padronização de como as chamadas são feitas e as técnicas utilizadas, geralmente o retorno é feito através de XML ou JSON.

Ainda segundo Lecheta (2015), o conceito de REST pode ser definido como um conjunto de técnicas de desenvolvimento de web services fortemente atrelado ao HTTP e a outros padrões de desenvolvimento Web. Já a web services que utilizam o REST chamamos de web services RESTful.

4.7.Arduino

Segundo Oliveira (2017), Arduino é uma plataforma de desenvolvimento de sistemas embarcados de baixo custo e livre, não se tratando de um micro controlador e nem estando vinculada a um fabricante especificamente. Dessa forma, o Arduino tornou possível o desenvolvimento de sistemas embarcados para todos os tipos de desenvolvedores com placas baratas e prontas para utilização e integração, o que antes era uma tarefa cara e complicada.

Ainda segundo Oliveira (2017), a popularização do Arduino trouxe inúmeras vantagens para o seu desenvolvimento. Vários módulos foram sendo desenvolvidos e consumidos em número maior, tornando-se parte de importante na implementação de sistemas embarcados. Entre esses módulos podemos citar: sensores (sensor de corrente elétrica, módulo de sonar, módulo detector de movimento, módulo sensor de umidade do solo, módulo sensor de temperatura e umidade do ar, módulo sensor de gases inflamáveis, módulo sensor

de luminosidade, módulo sensor infravermelho), atuadores (o mais comum é o relé) e módulos de comunicação (WiFi, Ethernet e Bluetooth).

Figura 3 – Interface IDE Arduino



Fonte: OLIVEIRA, 2017, p. 49

4.8.ESP8266

Segundo Oliveira (2017), o micro controlador ESP8266 é produzido pela empresa chinesa Espressif. Trata-se de um micro controlador de 32 bits que tem a capacidade de se comunicar através de uma rede Wifi. Possui memória de 4 MB e tensão nominal de funcionamento de 3,3 volts e no que diz respeito a interfaces de entrada e saída possui 17 interfaces que podem ser configuradas como entradas e saídas digitais.

Ainda segundo o mesmo autor, o micro controlador ESP8266 possui algumas vantagens que favorecem o seu uso em projetos de pequeno porte e baixo custo, conforme detalhado no quadro 1:

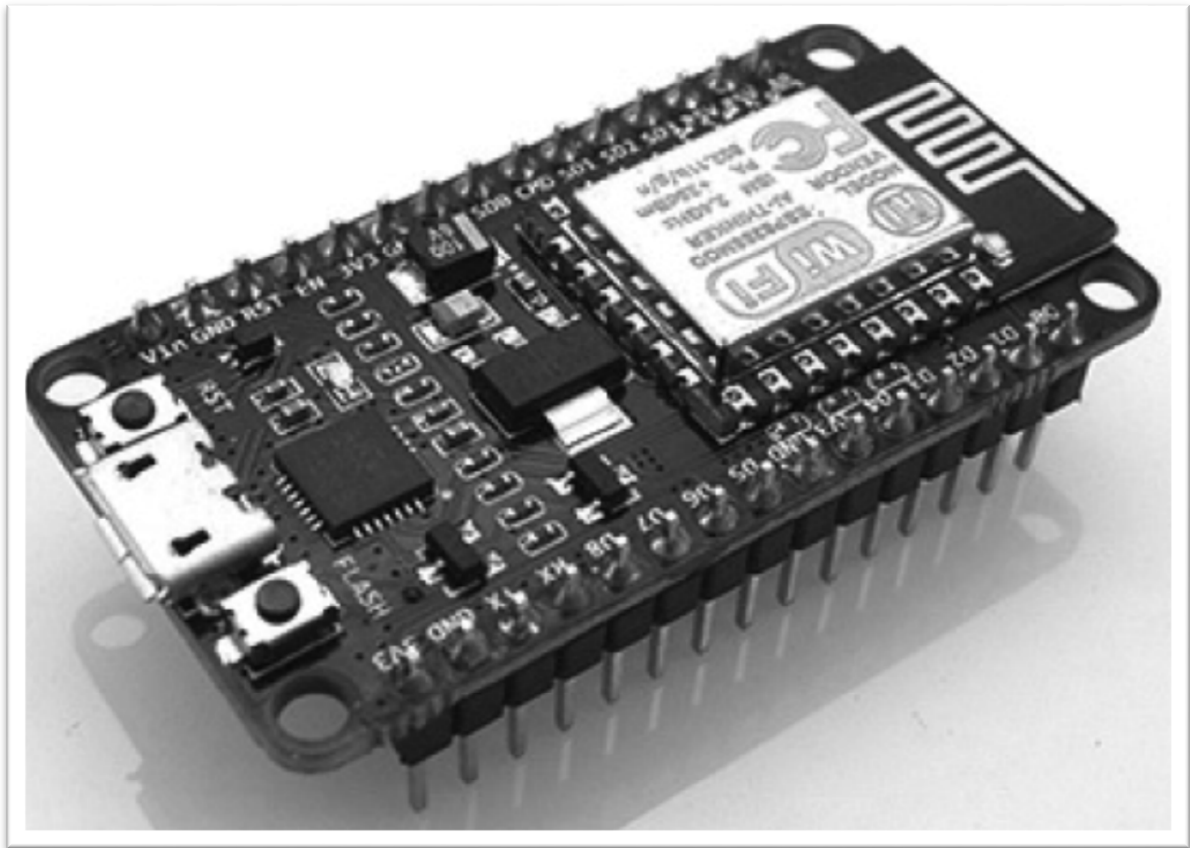
Quadro 1 - Vantagens do ESP8266

Vantagem	Descrição
Preço	Encontrado a um baixo custo, principalmente em sites chineses, e pronto para o uso, o que o torna um produto muito indicado para a internet das coisas.
Ambientes de desenvolvimento	Três ambientes de desenvolvimento estão disponíveis para o ESP8266, sendo o primeiro baseado em linguagem Lua, o segundo se trata do ambiente de desenvolvimento da Arduino herdando todas o conhecimento já desenvolvido para Arduino, juntamente com suas bibliotecas e interfaces. A terceira é um ambiente de desenvolvimento baseado no sistema operacional de tempo real RTOS, que cria aplicações profissionais, explorando toda a capacidade do micro controlador e suas interfaces.
Disponibilidade de módulos periféricos	Todos os módulos desenvolvidos para o Arduino são compatíveis com o ESP8266. Dessa forma, uma grande quantidade de placas está disponível, incluindo sensores, atuadores e ferramentas de comunicação. A facilidade de integração com essas placas e principalmente no ambiente de desenvolvimento do Arduino possibilita que o desenvolvimento de aplicações IoT seja feito de forma rápida e barata.

Fonte: OLIVEIRA, 2017

4.9.NODEMCU

Segundo Oliveira (2017) existem vários módulos no mercado que utilizam o micro controlador ESP8266, um deles é o NodeMCU, que agrega com ele uma plataforma de desenvolvimento LUA, desenvolvida no Brasil por professores da PUC-RIO. Sua composição basicamente se resume a uma ESP-12 com conversor USB-Serial e o regulador de tensão, sendo que sua pinagem facilita a prototipação e montagem de pequenos circuitos.

Figura 4 – Módulo NodeMCU

Fonte: OLIVEIRA, 2017, p. 57

4.10. Linguagem C#

Segundo Microsoft (2020), C# é uma linguagem orientada a objeto e fortemente tipada, permitindo o desenvolvimento de aplicativos robustos e seguros que rodam no .Net Framework. Sendo assim, o C# é utilizado para desenvolver aplicativos Windows, aplicativos Web, componentes distribuídos, aplicativos cliente servidor, aplicativos de banco de dados entre outros.

Ainda segundo a Microsoft (2020), qualquer programador que tenha um bom conhecimento nas linguagens C, C++ e Java é capaz de se adaptar ao C# de forma a programar de forma produtiva em um período de tempo curto, sendo que algumas das complexidades dessas linguagens são simplificadas no C# e ainda fornecendo recursos poderosos como tipos de valor anulados, enumerações, *delegates*, expressões lambda e acesso direto à memória. C# oferece suporte a tipos e métodos genéricos, o que aumenta a segurança e desempenho para os tipos, e iteradores, que permitem aos programadores das classes de coleção definir os comportamentos personalizados da iteração simples de usar pelo código do

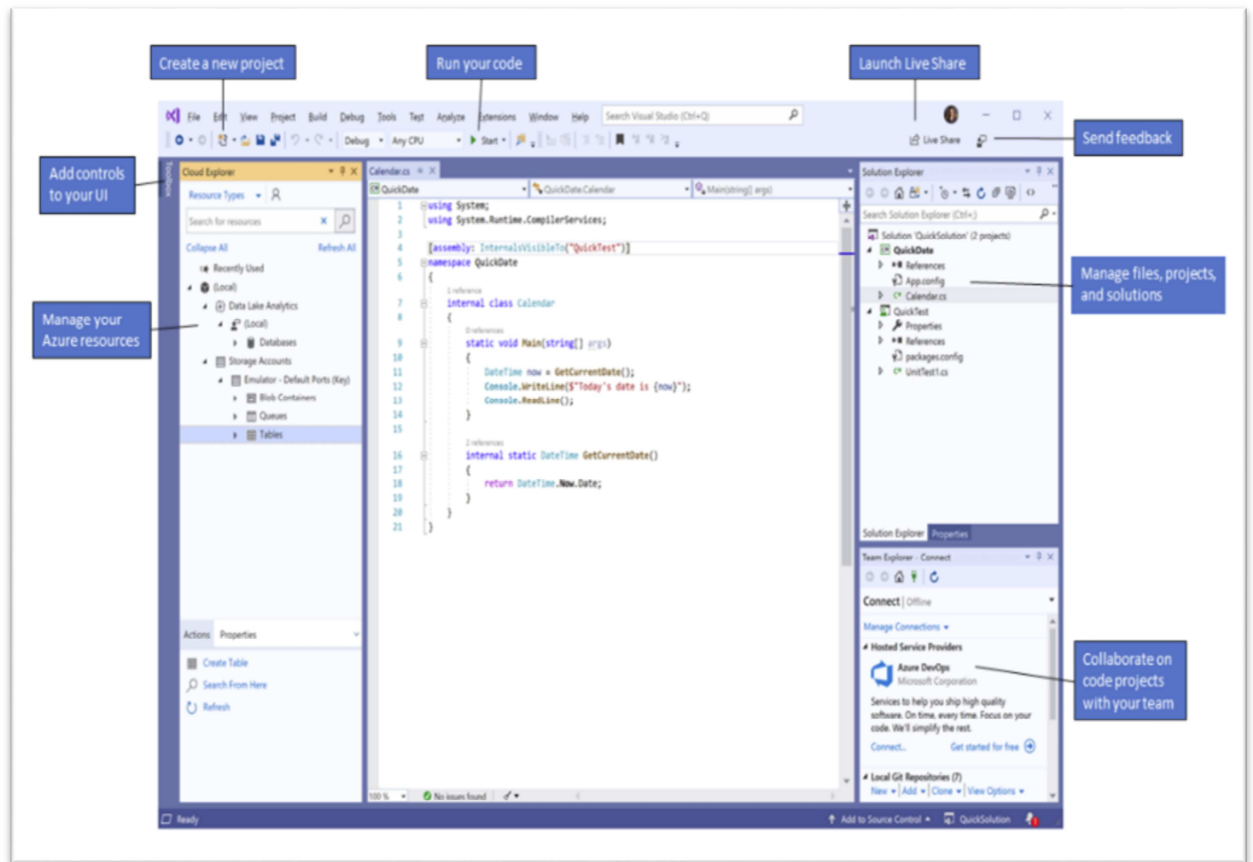
cliente. As expressões LINQ (*Language-Integrated Query*, consulta integrada ao idioma) tornam a consulta fortemente digitada uma construção de idioma de primeira classe.

Por ser uma linguagem orientada a objeto, o C# oferece suporte aos conceitos de encapsulamento, herança e polimorfismo. Todas as variáveis e métodos, incluindo o método *Main*, o ponto de entrada do aplicativo, são encapsulados em definições de classe. Uma classe pode herdar diretamente de uma classe pai, mas pode implementar qualquer quantidade de interfaces. Métodos que substituem métodos virtuais em uma classe pai exigem a palavra-chave *override* como uma forma de evitar uma redefinição acidental. Em C#, um *struct* é como uma classe simplificada; é um tipo alocado na pilha que pode implementar interfaces, mas não oferece suporte a herança. (MICROSOFT, 2020).

4.11. IDE do Visual Studio 2019

Para Microsoft (2020), o ambiente de desenvolvimento integrado (IDE) do Visual Studio é um painel de inicialização que pode ser utilizado para editar, depurar e compilar o código sendo possível em seguida publicar o aplicativo. O IDE é um aplicativo com muitos recursos e pode ser utilizado em diversos ramos do segmento de sistemas. Além do editor e do depurador padrão fornecidos pela maioria dos IDEs, o Visual Studio inclui compiladores, ferramentas de preenchimento de código, designers gráficos e muitos outros recursos para facilitar o processo de desenvolvimento de software.

Figura 5 – IDE do Visual Studio 2019



Fonte: MICROSOFT, 2020

4.12. Xamarin

Segundo Microsoft (2020), o Xamarin é uma plataforma de código aberto e tem como propósito a criação de aplicativos modernos e de alto desempenho para Android, IOS e Windows com o .Net. Assim, o Xamarin é uma camada de abstração que gerencia e comunica o código compartilhado com o código da plataforma adjacente. O Xamarin permite o compartilhamento de praticamente 90% do código entre as diferentes plataformas, sendo essa característica responsável por permitir que os desenvolvedores escrevam toda a regra de negócio em uma linguagem só (reutilizando o código existente), assim possibilitando o desempenho e a aparência nativa de cada plataforma (Android, IOS ou Windows).

4.13. Vue.js

Para Incau (2017), O Vue.js ou simplesmente Vue, é um biblioteca JavaScript para o desenvolvimento de componentes reativos para Web que ganhou uma grande

visibilidade no mercado após ser adotado como padrão pela Laravel, tendo como uma de suas principais características a possibilidade de ser usado em qualquer projeto que possua um *Front-End* independente da linguagem de programação escolhida.

O Vue se diferencia por ser uma biblioteca não intrusiva, ou seja, ele não tem um código que force a seguir o padrão dele, como é o caso do Angular 2 com *TypeScript*. A maioria do código para Vue é escrito em JavaScript puro. Ele possui uma sintaxe muito clara e limpa. Outro grande diferencial é sua flexibilidade. Veja que nos referenciamos ao Vue como uma biblioteca e não um framework, pois ele trabalha com outras bibliotecas que, juntas, formam um framework. O Vue é performático, em sua versão 2.0, utiliza o conceito de Virtual DOM, assim como o seu principal concorrente, o React. (INCAU, 2017, p. 13)

5. METODOLOGIA

5.1.Documentação

Segundo Ruiz (2002), quando estamos tratando de pesquisa bibliográfica, a documentação é o conjunto de conteúdos utilizados para definir ou demonstrar o problema definido como assunto principal.

Este trabalho foi precedido de uma pesquisa com o objetivo de buscar informações que fossem pertinentes no esclarecimento do problema abordado.

5.2.Natureza da Pesquisa

A pesquisa utilizada neste projeto pode ser definida como exploratória, por se tratar de um estudo sobre assuntos relevantes para o tema. O objetivo da pesquisa exploratória baseia-se na caracterização inicial do problema, de sua classificação e sua definição. Dessa forma, faz parte do primeiro estágio de qualquer pesquisa científica, tendo como principal objetivo apenas definir o problema e não o resolver (RUIZ (2002)).

5.3.Tipo da Pesquisa

No processo de coleta de informações a pesquisa pode ser definida como bibliográfica, sendo utilizada com o objetivo de justificar a finalidade deste projeto.

A pesquisa bibliográfica envolve toda bibliografia disponível que tenha relação com o assunto do estudo. Assim, podemos citar: publicações, livros, monografias, pesquisas, etc. (MARCONI, LAKATOS (2003)). As referências bibliográficas utilizadas foram encontradas em bibliotecas físicas e digitais.

5.4.Técnicas de Pesquisa

A técnica de pesquisa aqui utilizada foi a técnica de observação. Observar é estar alerta a um acontecimento ou problema, o capturar e retratar como tal se manifesta (RUIZ, 2002).

A observação facilita o processo do pesquisador de buscar acontecimentos que legitimem o objetivo que está sendo buscado, tornando possível desenvolver uma fonte ampla para a construção de hipóteses.

5.5.Coleta de Dados

Na busca de informações, no que diz respeito a viabilidade e sobre as funcionalidades que deveriam fazer parte do projeto, foram utilizadas a técnica de entrevista.

A entrevista foi feita com Santin de Paula Teixeira, responsável pelo setor de Manutenção na empresa Hercosul, empresa essa que demonstra interesse em um possível uso do projeto no futuro.

Na entrevista, foram tratados de temas relevantes ao assunto e analisadas técnicas que devem ser utilizadas no monitoramento a distância do maquinário e em como o desenvolvimento do projeto poderia ajudar na resolução dos problemas enfrentados pela empresa. Metodologia mostra o caminho a ser percorrido em uma investigação, ou seja, como se responderá aos problemas estabelecidos. Deve estar de acordo com os objetivos específicos, abrangendo a definição de como será feito o trabalho.

6. PROJETO

6.1. Hardware

O Hardware definido para compor esse projeto é composto por um micro controlador ESP8266 presente no módulo NodeMCU, que se destaca no mercado por seu preço acessível, seu ambiente de desenvolvimento e a disponibilidade de módulos e periféricos (OLIVEIRA, 2017). Este micro controlador será responsável por processar as informações vindas do sensor de detecção de tensão e disponibilizar na nuvem esses dados.

6.2. Desenvolvimento

A primeira fase do projeto consiste na integração do hardware ao maquinário que será monitorado. Essa integração se dá conectando um rele acoplador 24vcc ao circuito eletrônico da máquina para que seja possível monitor a presença de tensão e que será monitorada pelo microprocessador.

A segunda fase do processo é a disponibilização da Web Api Rest Full na nuvem para que ela seja capaz de receber as informações oriundas do micro controlador e disponibilizá-los para serem coletados pelo Aplicativo Android e a aplicação web.

Por fim, é necessário a instalação do Aplicativo Android, desenvolvido na plataforma do Xamarin e o, no celular do responsável pelo seu monitoramento. O aplicativo irá comunicar-se com a Web Api Rest Full para manter a lista de máquinas atualizada, sendo que uma mensagem *push* será enviada a cada mudança de estado das máquinas monitoradas.

6.3. Pré-requisitos

Para que o sistema seja executado é necessário que o smartphone do usuário possua o sistema operacional do android na versão 5.0 ou superior com conexão à internet, já que todas as informações são disponibilizadas através de um serviço hospedado na nuvem. Também é necessário que o usuário possua 100 megabytes de espaço em disco no seu dispositivo para que o aplicativo seja instalado e armazene seus dados. Também é necessário que um computador seja utilizado para que seja possível cadastrar os dados necessários para o funcionamento do sistema e para que o operador consiga fazer a justificativa dos motivos das paradas.

6.4. Diagramas UML

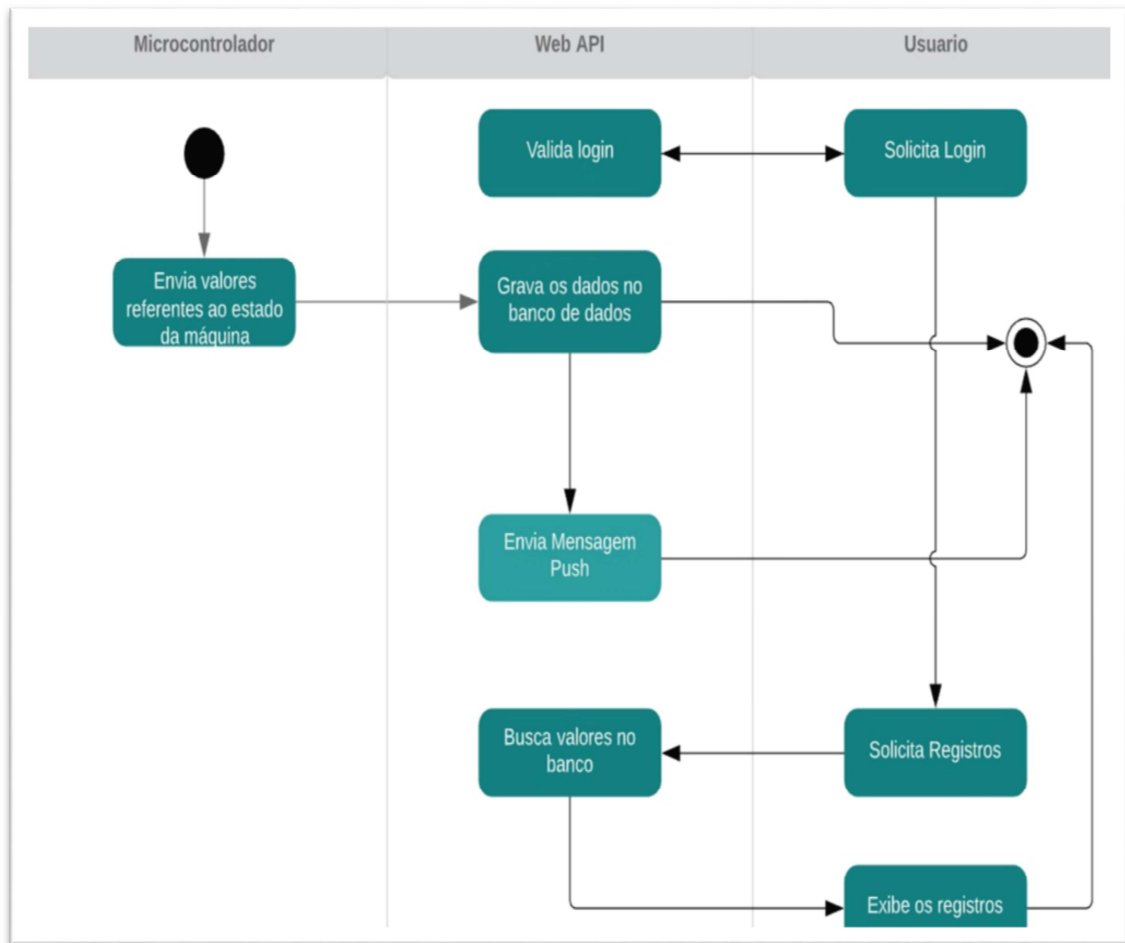
Segundo Pereira (2011), A Linguagem Unificada de Modelagem (em inglês *Unified Modeling Language – UML*) foi criada com o objetivo de estabelecer um padrão a ser utilizado nas especificações das características dos sistemas computacionais. O UML teve seu início em novembro de 1997, tendo um grande impacto e se tornando rapidamente um padrão utilizado por muitas equipes de desenvolvimento nas discussões e especificações de soluções.

A UML também vem sendo usada como especificação com vistas à geração automática de código com base no modelo, de forma modular e com altos índices de reuso. A importância que a UML assumiu no contexto de desenvolvimento de sistemas, fez com que o curso de Análise, Projeto e Gerência de Sistemas – APGS – e o seu sucessor, o de Análise e Projeto de Sistemas – APS –, da PUC-Rio, dedicasse uma parte significativa do programa ao estudo teórico e prático desse assunto, mesclado, como não podia deixar de ser, com as questões relativas às disciplinas de análise e projeto de sistemas e computação (PEREIRA, 2011, p. 17)

6.4.1. Diagrama de Atividades

Para Pereira (2011), o diagrama de atividades representa o fluxo de ações de um processo qualquer, especificando a sua ordem de sua realização em relação ao tempo. Dessa forma, foi criado um diagrama de atividades especificando as ações entre o aplicativo, a Web API e o micro controlador:

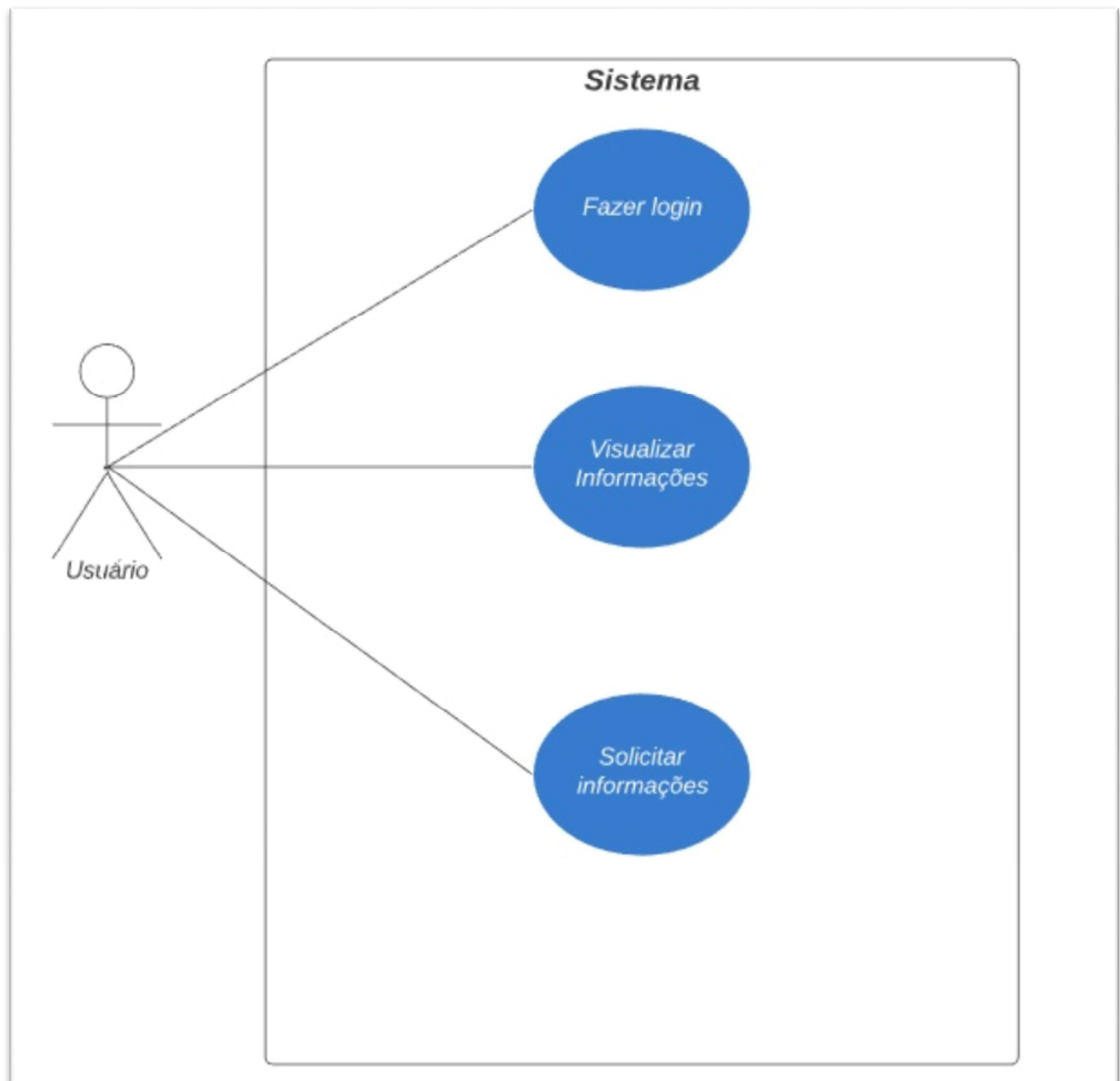
Figura 6 – Diagrama de Atividade



Fonte: o próprio Autor

6.4.2. Diagrama de Caso de Uso

Um diagrama de caso de uso deve representar os requisitos funcionais, ou seja, as funções que devem ser disponibilizadas pelo sistema para que seus objetivos sejam atingidos (PEREIRA, 2011). Assim, são utilizadas figuras que são nomeadas atores, representando os usuários que irão interagir com o sistema. As ações executadas por um ator são ligadas a ele. Abaixo está o diagrama de caso de uso do projeto:

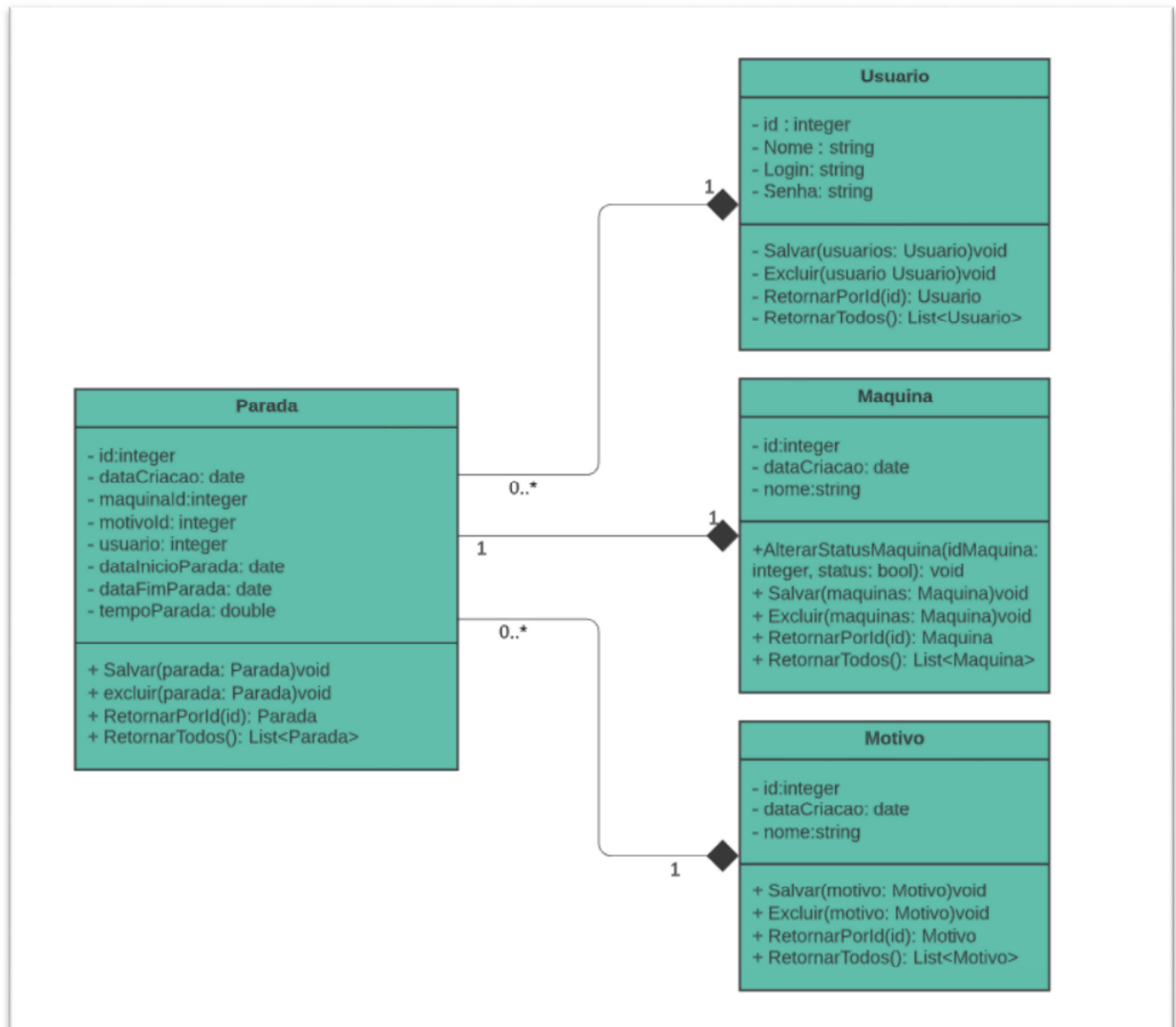
Figura 7 – Diagrama de Caso de Uso

Fonte: o próprio Autor

6.4.3. Diagrama de Classes

No diagrama de classes o objetivo é demonstrar a estrutura lógica de um sistema e o relacionamento entre as classes. Desta forma, também é tarefa desse tipo de diagrama apresentar os atributos e métodos de cada classe, dando uma forma melhor das funções de cada uma, sendo ele um dos mais utilizados no UML (GUEDES, 2018).

Figura 8 – Diagrama de Classe

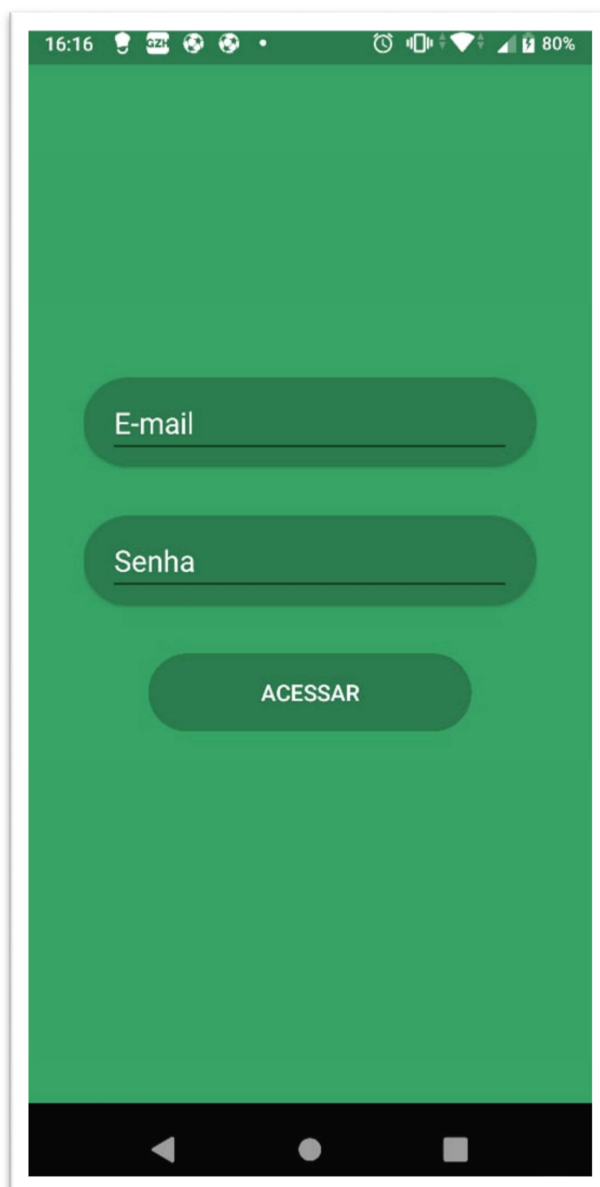


Fonte: o próprio Autor

6.5. Telas do Sistema

Este capítulo tem como objetivo demonstrar o design visual do aplicativo desenvolvido em Xamarin para a plataforma Android e o sistema Web desenvolvido e Vue.js.

Na tela de login, o usuário informará suas credenciais de acesso para que o sistema possa fazer a login e acessar a página com as informações em tempo real das máquinas cadastradas. A tela de login é mostrada na figura 10.

Figura 9 – Tela de Login

Fonte: o próprio Autor

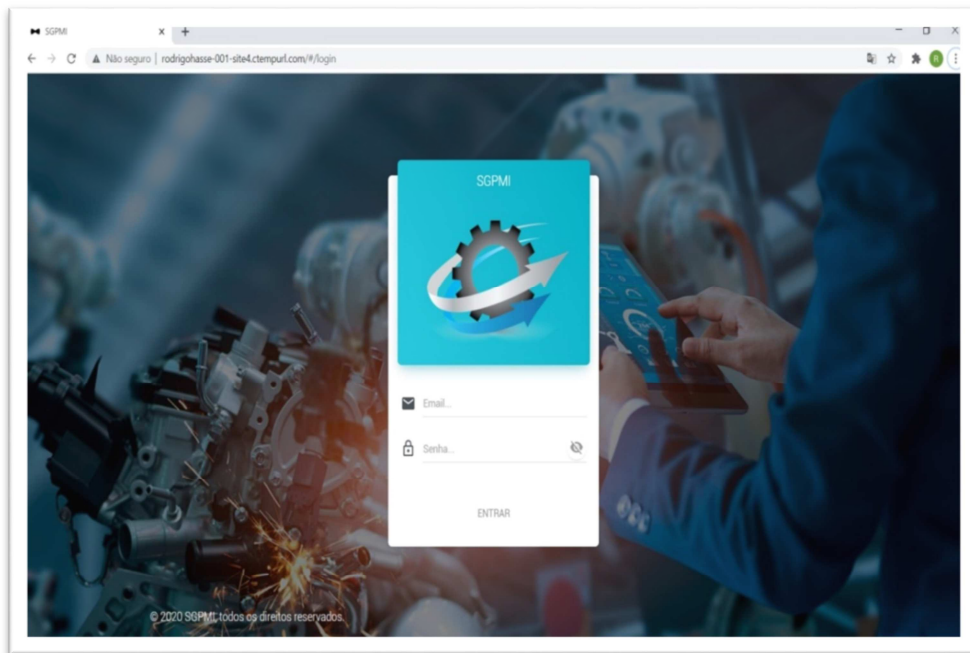
Após o usuário ter feito o login de forma correta, ele será encaminhado para a tela de listagem de máquinas, onde todas as máquinas disponíveis para ele estarão listadas. Cada máquina será representada pelo seu nome e mostrará o estado da máquina em tempo real. Ao estender o *ListView* o componente mostrará as informações das últimas paradas da máquina, o que é demonstrado na figura 11.

Figura 10 – Tela de Listagem de Máquinas

MAQUINA 1	Ligada
20/10/2020 22:18:54	Manutenção planejada
02/11/2020 10:32:04	Teste de produto
02/07/2020 10:38:43	Set up de máquina
02/11/2020 11:53:34	Limpeza de máquina
03/11/2020 22:51:43	Limpeza Final de Turno
04/11/2020 19:28:57	Limpeza Final de Turno
04/11/2020 19:50:11	Manutenção planejada
04/11/2020 21:11:28	Manutenção planejada
05/11/2020 11:09:52	
05/11/2020 13:03:59	

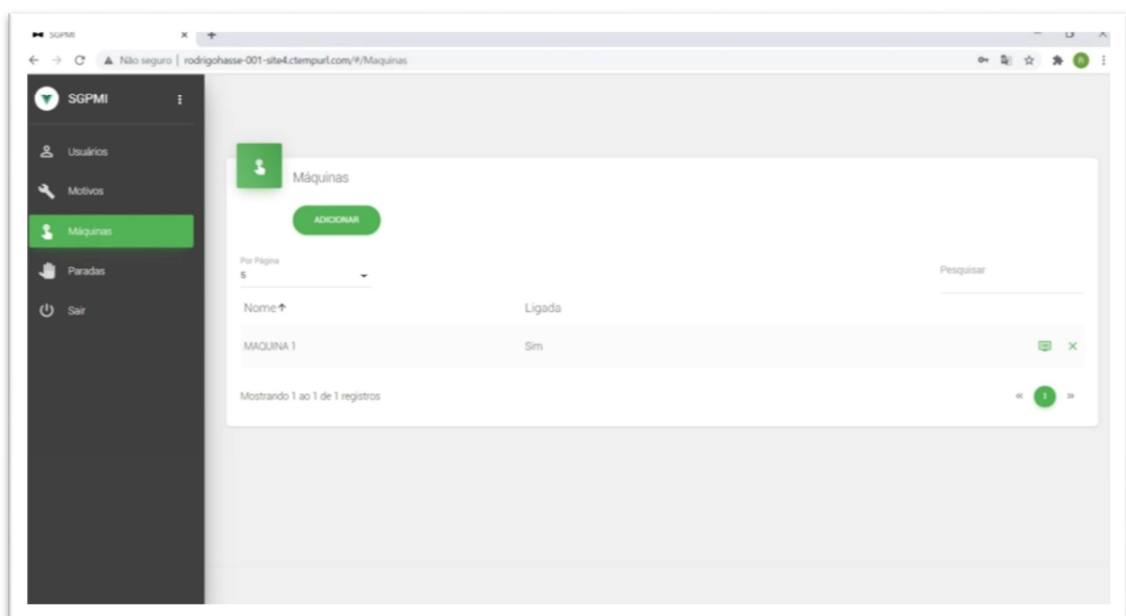
Fonte: o próprio Autor

Na tela de login do Sistema web o usuário deve também informar as credenciais para acesso, as mesmas informadas no aplicativo android, para que tenha acesso aos dados cadastrais do sistema. Após fazer o login ao sistema web o usuário é direcionado para a *dashboard* do sistema com algumas informações gerenciais das paradas das máquinas. A figura 12 apresenta a tela de login do sistema web.

Figura 11 – Tela de Login do Sistema Web

Fonte: o próprio Autor

A tela de consulta tem como objetivo apresentar as máquinas cadastradas no sistema. Nessa tela também é mostrado o estado da maquinas, se está ligada ou desligada, em tempo real.

Figura 12– Tela de Consulta de Máquinas

Fonte: o próprio Autor

Na tela de Consulta de Paradas são listadas todas as paradas registradas dentro o período informado no filtro da tela pelo usuário.

Nessa tela são mostradas as informações de hora e início e fim das paradas além do tempo que a máquina ficou desligada, o nome da máquina, o motivo pelo qual a máquina foi desligada e qual usuário fez a justificativa.

É também através dessa tela que o usuário tem acesso a tela onde é feito a justificativa da parada, clicando no botão existente no registro da parada.

Figura 13 – Tela de Consulta de Paradas

The screenshot displays a web application interface for 'Paradas' (Stops). On the left is a dark sidebar menu with options: 'Usuários', 'Motivos', 'Máquinas', 'Paradas' (highlighted), and 'Sair'. The main content area has a title 'Paradas' and a filter section with 'Data inicial' (05/11/2020) and 'Data final' (08/11/2020), a page size dropdown set to '10', and an 'ATUALIZAR' button. Below is a table with the following data:

Data/Hora Início	Data/Hora Final	Tempo Parada(Min)	Máquina	Motivo	Usuario	
05/11/2020 13:10:19	05/11/2020 13:10:25	0.1	MAQUINA 1	Ajustes Ventosas	Rodrigo Hasse	[icon]
05/11/2020 13:07:14	05/11/2020 13:07:25	0.17	MAQUINA 1			[icon]
05/11/2020 13:04:12	05/11/2020 13:04:16	0.07	MAQUINA 1			[icon]
05/11/2020 13:03:59	05/11/2020 13:04:08	0.14	MAQUINA 1			[icon]
05/11/2020 11:09:52	05/11/2020 11:17:16	7.4	MAQUINA 1			[icon]

Below the table, it says 'Mostrando 1 ao 5 de 5 registros'. At the bottom, a summary bar shows 'Quantidade Total de Paradas' as 5 and 'Total Tempo Paradas(Min)' as 7.88.

Fonte: o próprio Autor

7. CRONOGRAMA

O seguinte cronograma foi utilizado para o desenvolvimento deste trabalho.

Quadro 2 – Cronograma

Atividades	Ago	Set	Out	Nov	Dez
Revisão da Literatura					
Estudo de Técnicas					
Especificações do Protótipo(UML)					
Desenvolvimento Web API					
Desenvolvimento Web					
Desenvolvimento Aplicativo Android					
Desenvolvimento Sistema embarcado					
Ajuste na metodologia					
Entrega do Trabalho					
Defesa do TCC					

Fonte: o próprio Autor

8. CONCLUSÃO

Este trabalho apresentou as principais definições para o desenvolvimento de um protótipo de sistema de gerenciamento de performance de máquinas industriais. O objetivo desse projeto é auxiliar indústrias, por meio de um sistema de gerenciamento, no monitoramento da disponibilidade das máquinas utilizadas no seu processo de trabalho através da coleta das informações de parada.

Com esse protótipo foi possível realizar testes que se mostraram eficazes no que diz respeito a coleta das informações referentes as paradas da máquina e disponibilização destes dados ao usuário, tornando possível uma fácil identificação dos fatores que mais impactam na performance da máquina, permitindo assim que a empresa venha a realizar planos de ações para reduzir o seu impacto na disponibilidade do equipamento.

Desta forma, a utilização desse sistema proporciona uma melhora significativa na qualidade das informações de disponibilidade dos equipamentos que são por ele monitorados, tornando-se assim, uma ferramenta útil no auxílio de tomadas de decisão no que diz respeito a otimização do tempo de disponibilidade em relação ao potencial da máquina.

9. PROJETOS FUTUROS

Como sugestão para melhorias futuras, pode-se mencionar a criação dos módulos de monitoramento de quantidade e a qualidade das peças produzidas pela máquina, para que seja possível ao sistema realizar o cálculo de eficiência global de equipamentos, para medir a produtividade industrial.

Outra sugestão para projetos futuros seria a criação de mais modelos de relatórios e gráficos que venham a ser necessários conforme evolução do sistema. Um dos gráficos sugeridos pode ser o Diagrama de Pareto, que pode vir a ser uma ferramenta importante na análise dos dados relacionados aos motivos das paradas, por se tratar de um gráfico que visa focar no esforço de melhorias.

10. REFERÊNCIAS

ALMEIDA, R. M. A.; MORAES, C. H. V.; SERAPHIM, T. F. P. **Programação de Sistemas Embarcados**. 1. ed. Rio de Janeiro: Elsevier Editora Ltda, 2016.

ARAÚJO, E. C. **Xamarin Forms: Desenvolvimento de Aplicações Móveis Multiplataforma**. [S.l.]: Editora Casa do Código, 2017. v. 1.

COELHO, P. **Internet das coisas: Introdução prática**. 1. ed. Lisboa: FCA – Editora de Informática lda, 2017.

FIRJAN SENAI; FINEP (org.). **Indústria 4.0 no Brasil: oportunidades, perspectivas e desafios**. Rio de Janeiro: [s.n.], 2019.

GUEDES, G. T. **UML 2 – Uma abordagem prática**. 3. ed. São Paulo: Editora Novatec, 2018.

LECHETA, R. R. **Web Services RESTful**. 1. ed. São Paulo: Novatec Editora Ltda, 2015.

MAGRANI, E. **A internet das coisas**. 1. ed. Rio de Janeiro: FGV Editora, 2018.

MARCONI, M. de A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

MASTERWALKER. **Esp8266**. Disponível em: <https://www.masterwalkershop.com.br>. Acesso em: 23/05/2020.

MICROSOFT. **Docs**. 2020. Disponível em: <https://docs.microsoft.com>. Acesso em: 22/03/2020.

MORAIS, R. R.; MONTEIRO, R. **Industria 4.0: Impactos na gestão de operações e logística**. 1. ed. São Paulo: Editora Mackenzie, 2019.

OLIVEIRA, R. S. **Fundamentos dos sistemas de tempo real**. 1. ed. Florianópolis: [s.n.],

2018.

OLIVEIRA, S. **Internet das Coisas com ESP8266**. 1. ed. São Paulo: Novatec Editora Ltda, 2017.

PEREIRA, L. A. de M. **Análise e Modelagem de Sistemas com a UML**. 1. ed. Rio de Janeiro: Edição do Autor, 2011.

RUIZ, J. A. **Metodologia Científica**: Guia para eficiência nos estudos. 5. ed. São Paulo: Atlas, 2002.

SAUDATE, A. **Rest**: Construa Api's inteligentes de maneira simples. 1. ed. [S.l.]: Casa do Código, 2014.

SCHUWAB, K. A **Quarta Revolução Industrial**. 1. ed. [S.l.]: Édípro, 2016.

SILVA, R. A.; STEVAN JR., S. L. **Automação e Instrumentação Industrial com Arduino**: Teoria e Projetos. 1. ed. São Paulo: Editora Érica, 2015.

SINCLAIR, B. **IoT**: Como Usar a Internet Das Coisas Para Alavancar Seus. 1. ed. São Paulo: Autêntica Business, 2018.

SISINNI, E.; SAIFULLAH, A.; HAN, S. **Industrial Internet of Things**: Challenges, Opportunities, and Directions: Transactions on Industrial Informatic. 1. ed. [S.l.]: IEEE, 2018.

STANKOVIC, J. **Research Directions for the Internet of Things**, **IEEE Internet of Things Journal**: IEEE Internet of Things Journal. 1. ed. [S.l.: s.n.], 2014. v. 1.

WOLLSCHLAEGER, M.; SAUTER, T.; JASPERNEITE, J. **The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0**. [S.l.]: IEEE Industrial Electronics Magazine, 2017. v. 17.

INCAU, C. **Vue.js**: Construa Aplicações Incriveis. 1. Ed. Editora Casa do Código, 2017.

11. APÊNDICES

11.1. APÊNDICE A – Sketch Esp8266

Código desenvolvido na IDE do Arduino e compilado para o Esp8266, sendo responsável por criar uma rede local para configuração e monitorar o status da máquina e passar essas informações para a api:

```
void loop() {  
  // WatchDog -----  
  yield();  
  
  // DNS -----  
  dnsServer.processNextRequest();  
  
  // Web -----  
  server.handleClient();  
  
  if(WiFi.status() == WL_CONNECTED)  
  {  
    digitalWrite(4, HIGH);  
  
    // Aguardamos 1 segundo  
    delay(1000);  
  
    // Desligamos o pino 12  
    digitalWrite(4, LOW);  
  
    valor = digitalRead(2);  
  
    if(valor != valor2)  
    {  
      if (valor == 0)  
    {
```

```
        Serial.println("Maquina Ligada");
    http.begin(serverName);
    http.addHeader("Content-Type", "application/json");
    int httpResponseCode = http.POST("{\"MaquinaId\":\"1\",\"Ligada\":\"true\"}");
}
    else
    {
        Serial.println("Maquina Desligada");
        http.begin(serverName);
        http.addHeader("Content-Type", "application/json");
    int httpResponseCode = http.POST("{\"MaquinaId\":\"1\",\"Ligada\":\"false\"}");
    }
}
valor2 = valor;

delay(1000);
}
}
}
```

11.2. APÊNDICE B – Código Busca Resumo por Motivo

Código desenvolvido em C# usando a IDE do Visual Studio, usando o ORM Dapper, que é responsável por fazer a busca no banco dos dados dos dados para geração do resumo de por motivos:

```
public async Task<Option<IEnumerable<ParadasResumoMotivoInputModel>>>
ResumoParadasPorMotivoAsync(FiltroParadasInputModel filtro)
{
    var conexao = ConexaoBanco();

    using (var db = new SqlConnection(conexao))
    {
        await db.OpenAsync();
        var ret = await
db.QueryAsync<ParadasResumoMotivoInputModel>(BuscarResumoParadasPorMotivo(filtro
));

        return ret == null ? Option.None<IEnumerable<ParadasResumoMotivoInputModel>>() :
            Option.Some<IEnumerable<ParadasResumoMotivoInputModel>>(ret);
    }
}

private string BuscarResumoParadasPorMotivo(FiltroParadasInputModel filtro)
{
    var sb = new StringBuilder();
    sb.Clear();
    sb.AppendLine("select sum(par.TempoParada) as Total, count(par.Id) as Quant
,Motivos.Nome as Nome");
    sb.AppendLine("from Paradas par");
    sb.AppendLine("inner join Motivos on Motivos.Id = par.MotivoId");
    sb.AppendLine(Filtrar(filtro));
    sb.AppendLine("group by Motivos.Nome");
    sb.AppendLine("order by Total desc");
}
```

```

return sb.ToString();
}

```

11.3. APÊNDICE C – Código Tela Consulta de Máquinas para APP Android

Código desenvolvido usando o Xamarin Forms que monta a tela onde a listagem de máquinas é mostrada no APP Android:

```

</ContentPage.Resources>
<ContentPage.Content>
    <syncfusion:SfAccordion x:Name="OuterAccordion" ExpandMode="SingleOrNone"
Margin="5" BindableLayout.ItemsSource="{Binding Maquinas}">
    <BindableLayout.ItemTemplate>
        <DataTemplate>
            <syncfusion:AccordionItem x:Name="AccordionItem">
                <syncfusion:AccordionItem.Header>
                    <Grid HeightRequest="50" RowSpacing="0">
                        <StackLayout Orientation="Horizontal">
                            <Label Text="{Binding Nome}" TextColor="#3A3A3A"
FontSize="26" FontFamily="Yu Gothic UI" FontAttributes="Bold"
VerticalTextAlignment="Center" HorizontalOptions="StartAndExpand"
Padding="10,0,0,0"/>
                            <Label Text="Ligada" TextColor="Green" FontSize="26"
FontSize="26" FontFamily="Yu Gothic UI" FontAttributes="Bold" VerticalTextAlignment="Center"
HorizontalOptions="EndAndExpand" Padding="0,0,30,0">
                                <Label.Triggers>
                                    <DataTrigger TargetType="Label" Binding="{Binding Ligada}"
Value="False">
                                        <Setter Property="Text" Value="Desligada" />
                                        <Setter Property="TextColor" Value="Red" />
                                    </DataTrigger>
                                </Label.Triggers>
                            </Label>
                        </StackLayout>
                    </Grid>
                </syncfusion:AccordionItem.Header>
            </syncfusion:AccordionItem>
        </DataTemplate>
    </BindableLayout.ItemTemplate>
</syncfusion:SfAccordion>
</ContentPage.Content>
</ContentPage.Resources>

```

```

</Grid>
</syncfusion:AccordionItem.Header>
<syncfusion:AccordionItem.Content>
    <sfListView:SfListView x:Name="listView" HeightRequest="{Binding
paradas, Converter={StaticResource HeightConverter}, ConverterParameter={x:Reference
listView}}" ItemSize="50" ItemsSource="{Binding paradas}" ItemSpacing="1">
        <sfListView:SfListView.ItemTemplate>
            <DataTemplate>
                <Grid RowSpacing="0">
                    <Label Text="{Binding dataCriacao}" VerticalOptions="Center"
HorizontalOptions="Start" Padding="2"/>
                    <Label Grid.Column="1" Text="{Binding motivoNome}"
VerticalOptions="Center" HorizontalOptions="Start" Padding="2"/>
                </Grid>
            </DataTemplate>
        </sfListView:SfListView.ItemTemplate>
    </sfListView:SfListView>
</syncfusion:AccordionItem.Content>
</syncfusion:AccordionItem>
</DataTemplate>
</BindableLayout.ItemTemplate>
</syncfusion:SfAccordion>
</ContentPage.Content>
</ContentPage>

```

11.4. APÊNDICE D – Trecho do Código Busca Paradas na Interface Web

Código desenvolvido em Vue.js, que é responsável por fazer a requisição para a API e mostra na tela a listagem de paradas:

```
let data = new Date();
let data2 = new Date(data.valueOf() - data.getTimezoneOffset() * 60000);
this.filtro.DataInicial = data2.toISOString().replace(/^\d{3}Z$/, "");
this.filtro.DataFinal = data2.toISOString().replace(/^\d{3}Z$/, "");

let r = await api.cadastros.paradas.retornarVarios(this.filtro);
if (r.sucesso) {
  this.tableData = r.data;
} else {
  // this.$toasted.global.defaultError();
}
```