

CENTRO UNIVERSITÁRIO UNIFACVEST
CIÊNCIA DA COMPUTAÇÃO
ELCIO HINTZE

**SYS MONITOR: MONITORAMENTO AUTOMATIZADO COMO
INSTRUMENTO DE PREVENÇÃO DE PROBLEMAS.**

LAGES – SC

2021

ELCIO HINTZE

**SYS MONITOR: MONITORAMENTO AUTOMATIZADO COMO
INSTRUMENTO DE PREVENÇÃO DE PROBLEMAS.**

Projeto apresentado à Banca Examinadora do
Trabalho de Conclusão de Curso II de Ciência
da Computação para análise e aprovação.

Orientador: Prof.Esp. Marcio José Sembay.

LAGES – SC

2021

ELCIO HINTZE

**SYS MONITOR: MONITORAMENTO AUTOMATIZADO COMO
INSTRUMENTO DE PREVENÇÃO DE PROBLEMAS.**

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.

Orientador: Prof.Esp. Marcio José Sembay.

Lages, SC __/__/2021.

Nota _____

Marcio José Sembay (Coordenador do curso de graduação)

LAGES – SC

2021

Dedico este trabalho ao meu Pai Ivo, minha mãe Margarete, minhas irmãs Elenice e Elaine e a minha Namorada Paloma, pessoas que sempre me apoiaram e fizeram o possível por mim na busca dos meus objetivos. Agradeço principalmente a Deus pela minha vida e pelas pessoas que conheci durante minha jornada.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, pelas pessoas que eu tenho ao meu lado e pelas que conheci no decorrer da minha jornada dentro da Unifacvest.

A minha família em especial aos meus pais, Ivo Jair Hintze e Maria Margarete da Silva

Hintze, por sempre me apoiarem e insistirem no meu sucesso até mesmo quando pensei em desistir.

Minha namorada Paloma Fernandes de Lima, pelo apoio, parceria, assistência e principalmente por sempre estar ao meu lado.

Ao meu cunhado Bruno pelo conhecimento transmitido sobre a linguagem C#.

Aos meus orientadores por todo auxílio prestado nas dúvidas ou revisões.

Para finalizar, agradeço a todas as pessoas que de uma forma direta ou indireta auxiliaram neste projeto.

“O Próprio Senhor irá à sua frente e estará com você; ele nunca o deixará, nunca o abandonará. Não tenha medo! Não desanime!” (Deuteronômio 31:8)

RESUMO

O presente estudo nos chama a voltar o olhar para a área automatizada acerca do monitoramento de serviços, em razão da sua utilização em grande escala, visto que, a cada dia que passa os sistemas se encontram integrados uns com os outros. Desta maneira, na atualidade, tudo está associado a um computador, por este motivo, é inegável deixar de falar sobre a saúde de servidores, fazendo-se refletir gastos, mão-de-obra e demanda de serviço intensificado, visto que a cada evolução da tecnologia sentimos a expansão da área da informática que implica significadamente na vida do indivíduo. Sendo assim, objetivou-se elaborar um sistema que facilita o monitoramento de ambiente de processamento, visando ser econômico financeiramente. Esta ferramenta, foi desenvolvida utilizando as linguagens de programação CSharp e Java, contando com sistema de armazenamento SQL Server. Já a parte visual, conta com codificação realizado através da IDE Visual Studio Communit e Android Studio, sendo o sistema operacional alvo, neste caso o Windows em conjunto com o Android.

Palavras-chave: Monitoramento de serviços. Computador. Serviços. Linguagens. Protótipos.

ABSTRACT

The present study calls us to look again at the automated area regarding the monitoring of services, due to their use on a large scale, since, with each passing day, the systems are integrated with each other. Thus, nowadays, everything is associated with a computer, for this reason, it is undeniable to stop talking about the health of servers, making expenses, labor and the demand for intensified service reflect, since with each evolution of technology we feel the expansion of the area of information technology that significantly implies in the life of the individual. Thus, the objective was to develop a system that facilitates the monitoring of the processing environment, aiming to be financially economical. This tool was developed using the programming languages CSharp and Java, with a SQL Server storage system. The visual part, on the other hand, is coded using the IDE Visual Studio Communit and Android Studio, being the target operating system, in this case Windows in conjunction with Android.

Keywords: Service monitoring. Computer. Services. Languages. Prototypes.

LISTA DE FIGURAS

Figura 1: Quantidade de emissões NF-e, nos últimos 3 meses do ano de 2018.....	18
Figura 2: Gráfico demonstrando a utilização dos sistemas operacionais em desktops domésticos	23
Figura 3: Demonstrativo sobre a preferência por Android.	24
Figura 4: Linha do tempo Windows	26
Figura 5: Tela sobre serviço Windows	28
Figura 6: Pilares do princípio da abstração da POO	29
Figura 7: Exemplos de dispositivos que foram desenvolvidos em linguagem Java ...	31
Figura 8: Criação de método público CSharp.	33
Figura 9: Esquema representando o serviço de uma aplicação API	34
Figura 10: Interface Visual Studio Community.	35
Figura 11: Interface Android Studio.	36
Figura 12: Interface Balsamiq Wireframes	37
Figura 13: Tela inicial	41
Figura 14: Configuração banco de dados	42
Figura 15: Configuração servidor E-mail	42
Figura 16: Configuração notificações	43
Figura 17: Configuração monitoramento processos	43
Figura 18: Configuração monitoramento serviço	44
Figura 19: Histórico de notificações	44

LISTA DE QUADROS

Quadro 1: Cronograma.....	40
----------------------------------	----

LISTA DE SIGLAS

TI	Tecnologia da Informação
IDE	Integrated Development Environment
NF-e	Nota fiscal eletrônica
OHA	Open Handset Alliance
MS DOS	Microsoft Disk Operating System
MB	Megabyte
KB	Kilobyte
GUI	Interface Grafica do Utilizador
DOS	Disk Operating System
LANS	Local Area Network
POO	Programação Orientada a Objetos
API	Application Programming Interface
URL	Uniform Resource Locator
WCF	Windows Communication Foudation
HTTP	Hypertext transfer protocol
TCP	Trasmission Control protocol
MQ	Message Queue
SOAP	Simple Object Acess Protocol
XML	eXtensible Markup Language

SUMÁRIO

1	INTRODUÇÃO.....	13
2	OBJETIVOS.....	15
	2.1 Objetivos Gerais.....	15
	2.2 Objetivos Específicos.....	15
3	JUSTIFICATIVA.....	16
4	REVISÃO DE LITERATURA.....	17
	4.1 A inclusão da informática no operacional das empresas.....	17
	4.2 A importância do setor de TI na qualidade final do serviço.....	19
	4.3 Gerenciamento de desempenho.....	19
	4.4 Gerenciamento de falhas.....	20
	4.5 Verificação através de temporizadores.....	21
	4.6 Sistemas Operacionais.....	22
	4.7 Android.....	23
	4.8 Windows.....	25
	4.9 Serviço Windows.....	27
	4.10 Programação Orientada a Objetos.....	28
	4.11 Linguagem de programação JAVA.....	30
	4.12 Linguagem de programação C# (CSharp).....	32
	4.13 API.....	33
	4.14 Visual Studio Community.....	34
	4.15 Android Studio.....	35
	4.16 Balsamiq Wireframes.....	36
	4.17 Windows Forms.....	37
5	METODOLOGIA.....	39
	5.1 Documentação.....	39
	5.2 Natureza de pesquisa.....	39
	5.3 Método de pesquisa.....	39
	5.4 Técnicas de pesquisa.....	39
6	CRONOGRAMA.....	40
7	PROJETO.....	41
	7.1 SYS Monitor.....	41
8	LIMITAÇÕES DA PESQUISA.....	45
9	RESULTADOS.....	46
10	REFERENCIAS BIBLIOGRAFICAS.....	47

1 INTRODUÇÃO

Considerando a elaboração da presente obra, destacamos a importância de um olhar voltado para área automatizada sobre o monitoramento de sistemas, devido a sua utilização em abundante escala, e dado que a cada dia mais os sistemas se encontram integrados uns com os outros, desta maneira percebemos que tudo está relacionado a um processo operacional envolvendo sistemas computacionais. Eminentemente, na atualidade, tudo está relacionado a um computador, desde a simples ação de realizar o pagamento de uma conta em uma agencia lotérica, onde conforme destacado nesta situação, podemos determinar a importância deste estudo sobre este tema específico, pois quem nunca, ao depender de alguma agencia de pagamento, simplesmente teve que adiar seus vencimentos devido a avaria de um sistema fora de operação? Isto ocorre por inúmeros fatores, entre eles, uma queda de rede, falta de espaço no disco de armazenamento ou consumo excessivo de processamento, por exemplo.

É crucial ressaltar que o monitoramento e acompanhamento da saúde do servidor de processamento e hospedagem se torna muito importante, porém dispor de mão de obra para realizar este monitoramento se torna inviável ou traz consigo uma carga de gastos de alta relevância, e este valor tende a aumentar na medida que a demanda de serviço vai se intensificando, pois novos recursos humanos serão necessários.

A evolução da tecnologia juntamente com a expansão da área da informática trouxe consigo uma mudança significativa na forma como o ser humano se relaciona pessoalmente e profissionalmente, na visão de GARNER (2001, p. 1), destacamos que:

O crescimento de dispositivos relacionados à computação não se limitou a computadores pessoais ou estações de trabalho. O número de dispositivos de computação pessoal cresceu substancialmente tanto no tipo quanto no formato. Pequenos computadores portáteis carregam uma grande quantidade de informações pessoais, de documentos e de contato, além de serem sofisticados o suficiente para permitir que um usuário envie fax, envie e-mails e se comunique de outras maneiras sem fio. Mesmo telefones celulares avançados carregam memória e capacidade de processamento suficientes para armazenar informações de contato, navegar na Web e fornecer mensagens de texto. Juntamente com o crescimento da sofisticação desses dispositivos, a necessidade de transferir informações entre eles cresceu significativamente também.

Portanto, em concordância com a empresa NDD, foi autorizado que se realizasse a implantação deste sistema em um ambiente de testes para que fosse possível

acompanhar esse processo de forma prática, então, elaborou-se um sistema que facilita o monitoramento destes ambientes de processamento, necessitando, apenas de uma pessoa para manuseá-la, evitando desta maneira, altos custos com mão de obra e assim alterando a forma como o orçamento da corporação é investido, o dinheiro que antes era gasto para contornar a falha agora é utilizado em um serviço de prevenção de falhas, fornecendo assim um monitoramento, via software de alta qualidade, visando assim, auxiliar os times de TI das corporações na gestão dos serviços e servidores, por consequência permitindo reduzir ou até mesmo evitar paradas inesperadas do sistema, assim como oportunizar a melhor distribuição dos recursos humanos ou a redução do quadro de membros da equipe.

O desenvolvimento deste sistema foi realizado utilizando a linguagem de programação CSharp e o sistema de armazenamento SQL Server, onde toda a parte visual do sistema, assim como a codificação do mesmo foi realizado através da IDE Visual Studio Community, a escolha por esta linguagem e IDE se deu devido ao sistema operacional alvo, neste caso Windows nas suas diferentes versões, desktop e servidor, pois conforme a visão do Strapazzon (2018), “O Windows se trata de um software proprietário que atualmente engloba uma grande parte do mercado, pois é um software antigo e familiar que facilita a utilização dos usuários”, o que torna a mais rápida a aprendizagem e a familiarização com o sistema desenvolvido, ideal para usuários inexperientes e empresas de pequeno porte.

2 OBJETIVOS

2.1 Objetivo Geral

Foi desenvolvido um sistema de monitoramento de servidores para empresas de pequeno porte, a ser ofertado de forma gratuita, que utiliza da inteligência artificial para detectar possíveis pontos problemáticos ou falhas, permitindo que os profissionais que gerenciam estes servidores possam trabalhar de forma preventiva e com isso proporcionar uma redução na taxa de incidentes ou indisponibilidades dos serviços.

2.2 Objetivos Específicos

- A) Sistema objetivo e configurável, que permita ao usuário definir diretrizes e propriedades do monitoramento de acordo com a sua necessidade operacional;
- B) Portal de acompanhamento com dashboard's, contendo gráficos objetivos de fácil entendimento, que permite ao usuário acompanhar a saúde do sistema;
- C) Aplicativo Android para acompanhamento das notificações e seus diferentes níveis

3 JUSTIFICATIVA

A evolução tecnológica transformou a sociedade humana, nos dias de hoje consumimos uma quantidade elevada de serviços eletrônicos, juntamente com a demanda de oferta e procura, surge a necessidade de entregar serviços de qualidade e com disponibilidade frequente. A disponibilidade de um sistema ou serviço, e parte da qualidade passam logicamente pela saúde do servidor de hospedagem e processamento, ou seja, realizar um acompanhamento de perto pode evitar a ocorrência de muitos problemas, reduzindo assim interrupções e prejuízos.

Pensando assim, no decorrer de 5 anos de atuação laboral na área de suporte da empresa NDD, onde atuava diretamente com o monitoramento e gestão dos servidores responsáveis por processar todo o faturamento de clientes, como Ipiranga, ADM do Brasil e Havan, empresas de grande porte e com alto faturamento, foi possível constatar o quanto é crítico a parada do sistema, e algumas vezes foi presenciado o acontecimento de paradas no faturamento ocasionadas por motivos muitas vezes simples, como por exemplo, falta de espaço em disco, ou alto consumo de memória. Estas paradas podem acarretar em prejuízos enormes para as empresas, seja em seu faturamento ou no desgaste relacional com o consumidor resultando na perda de confiança com o serviço prestado. Levando isto em consideração, entende-se que toda venda exige a emissão de nota fiscal eletrônica, e sem um sistema de emissão funcional se torna impossível gerar tal documento e isso impossibilita a continuidade de tal operação, ocasionando em atrasos nas entregas e dependendo do tipo de mercadoria, podendo inclusive, resultar na perda da mesma. Estes itens citados poderiam possivelmente serem evitados com o monitoramento destes servidores e serviços.

No entanto deslocar um recurso humano para tal atividade pode não ser muito atrativo, uma vez que traria gastos com a mão de obra e tornaria o monitoramento vulnerável a falhas, levando em consideração que o fator humano está envolvido no processo. Acatando a necessidade de dispor de serviços de qualidade e com alto nível de disponibilidade, assim como a redução de custos, a criação de novas ferramentas para execução automatizada desta atividade de monitoramento se torna essencial para um resultado satisfatório.

4 REVISÃO DE LITERATURA

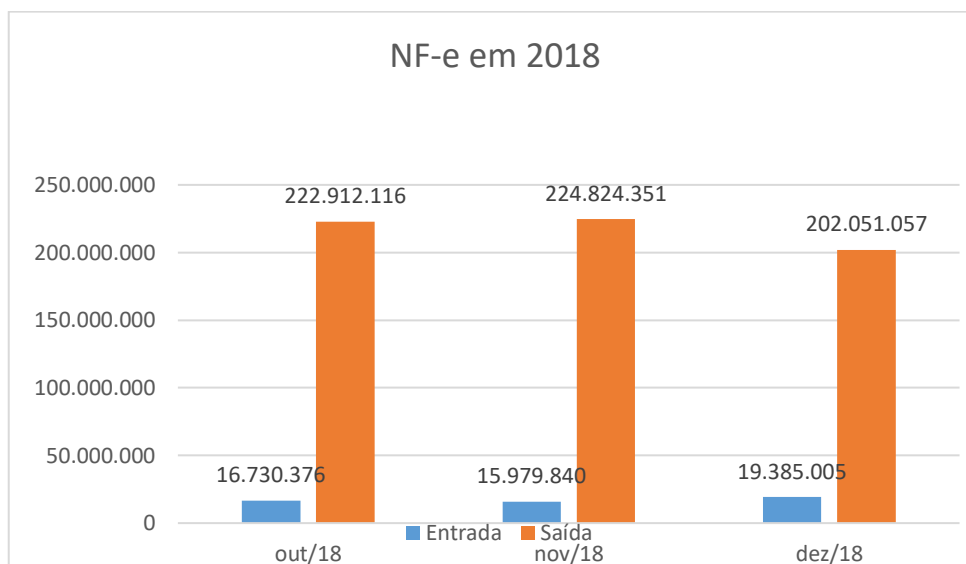
Essa revisão de literatura foi dividida em itens que promovem todo escopo e entendimento do tema proposto.

4.1 A inclusão da informática no operacional das empresas

A introdução dos computadores na vida das pessoas juntamente com o surgimento e a evolução da internet mudaram o mundo que vivemos. Segundo Ribeiro (2011), destaca que “Se hoje vivemos num período que pode ser chamado de Sociedade da Informação, isso só é possível pela oportunidade de acesso à rede, proporcionando comunicação em tempo real e em escala global.” Essa mudança global causada pela informatização, afetou os modos de vida do homem moderno, desde a comunicação até a forma de trabalho.

Processos que antes exigiam muita mão de obra ou espaço físico para alocar o histórico de processamento foram substituídos por processos mais ágeis e eficientes, e isso só foi possível graças a era digital.

Um exemplo é a emissão de notas fiscais, documento contábil fiscal emitido por pessoas jurídicas para registrar a venda ou tráfego de mercadorias. O sistema de NF-e começou a ser implantado no ano de 2008, seu objetivo é substituir a emissão manual das notas fiscais.

Figura 1: Quantidade de emissões NF-e, nos últimos 3 meses do ano de 2018.

Fonte: Elaborado pelo autor, baseado nas estatísticas em arquivos, disponibilizado no portal NF-e.

O Projeto da Nota Fiscal Eletrônica (NF-e) tem como objetivo implantar um modelo nacional de documento fiscal eletrônico para substituir a sistemática atual de emissão do documento fiscal em papel, simplificando as obrigações acessórias dos contribuintes e permitindo o acompanhamento em tempo real das operações comerciais pelo Fisco. (MÜLLER, 2019)

Podemos conceituar a NF-e como sendo um documento de existência exclusivamente digital emitido e armazenado eletronicamente, com um intuito de documentar uma operação de circulação de mercadorias ou prestação de serviços, cuja validade jurídica é garantida pela assinatura digital do emitente e a Autorização de Uso fornecida pela administração tributária. (SECRETÁRIA DA FAZENDA VERSÃO 4.0.1, 2009)

A NF-e trouxe consigo diversos facilitadores e benefícios para todos os envolvidos na circulação de mercadorias ou serviços. Conforme Müller (2019), o Projeto NF-e traz benefícios a todos os envolvidos, direta ou indiretamente.

Como benefícios para os vendedores os chamados emissores podemos destacar a redução de custos de envio e armazenagem de documentos fiscais assim como a quantidade de impressões. Já para os compradores popularmente conhecidos como destinatário da operação podemos destacar o planejamento de logística de entrega pela

recepção antecipada da informação da NF-e, a redução de erros de escrituração devido a erros de digitação de notas Fiscais. Para os administradores tributários os ganhos são na redução das sonegações, além de possibilitar um melhor compartilhamento de informações entre fiscos. Todos estes ganhos destacados acima também refletem em benefícios para a sociedade, pois com o processo eletrônico há uma redução absurda no consumo de papel, reduzindo assim o impacto em termos ecológicos.

4.2 A importância do setor de TI na qualidade final do serviço

Para muitas, se não todas empresas e setores de todo tipo de atividade, onde o uso de sistemas computacionais é imprescindível para o sucesso do negócio, uma falha que interrompa seu funcionamento ou que cause a perda de dados importantes pode provocar prejuízos incalculáveis ou até mesmo a falência dessas empresas.

Nestes ambientes, onde os sistemas computacionais e recursos tecnológicos são importantíssimos para atividade, fim da organização, os processos podem ser chamados de processos críticos, pois possuem características particulares e com elevadas exigências em termos de confiança e disponibilidade (NEIL 1996).

Para evitar qualquer tipo de transtorno, a maioria das empresas utilizam de uma equipe de TI para monitorar e realizar a manutenção destes ambientes, visando evitar a paralisação de serviços computacionais por algum tipo de falha, sobrecarga, e inclusive a perda de dados e informações importantes.

O uso de ferramentas com recursos tecnológicos tem suma importância neste processo de monitoramento e gestão de servidores, pois permite um monitoramento constante, além de evitar possíveis erros humanos neste processo.

4.3 Gerenciamento de desempenho

Quando falamos em gerenciamento de desempenho devemos basear-se em monitoramento de atividades, assim como, ajustes e trocas. Temos alguns pontos que são alusivos á esse processo de gerenciamento de desempenho, tais como: nível de capacidade de utilização; tempo de resposta; nível de taxa de dados processados, entre outros itens.

Para que possamos zelar por esta questão, o gerente deve realizar um check - list inicial de conjuntos a serem expostos ao monitoramento, desta maneira, determinar níveis de desempenho. Sendo assim, coletando e analisando as informações obtidas, o gerente de infra designado vai ficando cada vez mais apto a reconhecer situações que indicam um problema e por consequência afetam o desempenho de seus processos.

Sendo assim, podemos perceber que a formação de estatísticas de desempenho se faz útil no planejamento, gerenciamento e preservação de grandes redes. Tais informação são importantes para que seja reconhecido gargalos, antes que os mesmos venham causar complicações para o usuário final. Ações assertivas podem ser aplicadas, como por exemplo, apontar a necessidade de expansão de linhas para determinada área entre outros (SPECIALSKI, s.d).

4.4 Gerenciamento de falhas

Uma parte significativa do processo de gerenciamento de redes e sistemas de computação baseia-se na aquisição de informações relevantes sobre a rede e demais recursos do sistema de computação, sendo as mais importantes aquelas relativas a erros, falhas e outras condições excepcionais. Os dados devem ser armazenados em forma bruta, sendo importante definir os valores aceitáveis como limiar de tolerância que, quando ultrapassados, determinam uma sinalização para pedir intervenção de um operador, ou o início de uma operação corretiva automatizada.

A definição de gerência de falhas apresentada em (KUROSE, 2005) é o conjunto de atividades de detecção, isolamento e correção das operações anormais em uma rede. Em outras palavras, a gerência de falhas tem a responsabilidade de monitorar os estados dos recursos, dar manutenção a cada um dos objetos gerenciados, e tomar decisões para restabelecer as unidades do sistema que venham a dar problemas.

As informações que são coletadas sobre os vários recursos da rede podem ser usadas em conjunto com um mapa desta rede, para indicar quais elementos estão funcionando, quais estão em mau funcionamento, e quais não estão funcionados. Opcionalmente, pode-se gerar um registro das ocorrências na rede, um diagnóstico das falhas ocorridas e uma relação dos resultados deste diagnóstico com as ações posteriores a serem tomadas para o reparo dos objetos que geraram as falhas. O ideal é que as falhas

sejam detectadas antes que os efeitos prejudiciais, decorrentes destas, possam vir a acontecer. Pode-se conseguir este ideal através do monitoramento das taxas de erros do sistema e da evolução do nível de severidade gerado pelos alarmes (função de relatório alarme), que permitem a emissão das notificações de alarme ao gerente, podendo assim definir as ações necessárias para corrigir o problema e evitar as situações mais críticas.

A gerência de falhas, então, requer constante observação do funcionamento dos dispositivos que a compõem de forma que se possa identificar rapidamente o problema e resolvê-lo sem grandes prejuízos.

4.5 Verificação através de temporizadores

Verificação por temporização é eficaz na detecção de erros. Neste mecanismo, a execução de uma operação não deve exceder um tempo máximo pré-determinado, caso contrário um timeout (tempo ultrapassado) é sinalizado para indicar que um defeito ocorreu. Esta técnica é normalmente associada com o uso de temporizadores, que podem ser implementados tanto em hardware quanto em software. Quando um temporizador de hardware é utilizado, ele é chamado de temporizador “cão de guarda” (watchdog timer) e tem que ser periodicamente reinicializado pelo programa. Caso este temporizador não seja reinicializado devido a um problema qualquer no hardware ou software, o sistema irá levantar uma exceção.

Esta forma de redundância temporal é usada em conjunto com outras formas, para que possa cobrir uma porcentagem maior de falhas, pois embora possa indicar a presença de falhas, não existe como garantir que elas não aconteçam (ANDERSON 1981). O grande diferencial desta técnica é seu custo baixo. As principais vantagens são: o desempenho do sistema não é afetado significativamente, pois a checagem é feita de forma concorrente, e o processo de checagem é totalmente independente do processo checado.

4.6 Sistema Operacional

Ao falar sobre sistemas operacionais devemos pensar ligeiramente em um conjunto de sistemas que nos ofertam uma interface entre o computador e o usuário. Assim que a máquina é ligada, primordialmente este recurso já está sendo executado e

a partir disto, não deixa de operar enquanto ocorre a execução de outros programas, como se estivesse supervisionando todo o processo computacional. Na concepção de Maziero (2019), os sistemas operacionais são uma camada de software que opera entre o hardware e os programas de aplicativos voltados ao usuário final, refere-se a uma estrutura de software abrangente, várias vezes complexa, que incorpora aspectos de baixo nível, como por exemplo, programas utilitários e a própria interface gráfica.

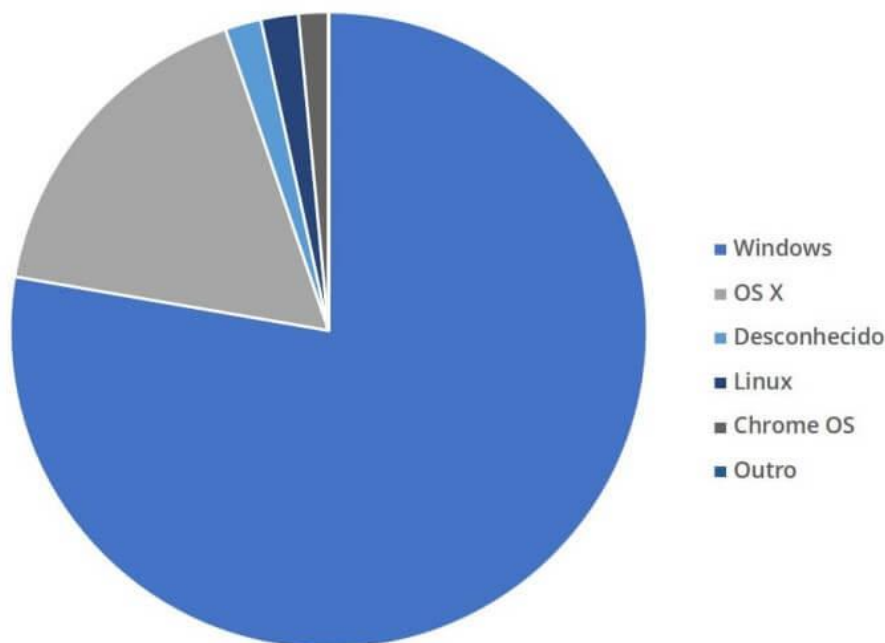
Em concordância com o tema, os autores Silberschatz, Galvin e Gagne, 2000, p.22 portanto supracitados por Coutinho (2010, p.15), destacamos que:

Um sistema operacional é um programa que atua como intermediário entre o usuário e o hardware de um computador. O propósito de um sistema operacional é propiciar um ambiente no qual o usuário possa executar outros programas de forma conveniente, por esconder detalhes internos de funcionamento e eficiência, por procurar gerenciar de forma justa os recursos do sistema.

Portanto, podemos afirmar que o sistema operacional é um programa de grande complexidade que se torna responsável por todo o funcionamento de uma máquina, desde de o software (parte lógica) até o hardware (parte física), que compõe o computador, todos os processos estão ligados a uma programação complexa que comanda todas as funções que um usuário impõe na máquina. Na atualidade, podemos citar aqui os sistemas operacionais mais conhecidos na atualidade e normalmente utilizados em computadores domésticos que são: Windows, Linux e o Mac OS X. A seguir faremos uma breve explicação sobre cada sistema:

- Microsoft Windows Vista e XP – Sistema operacional para estações de trabalho, sejam elas domésticas ou em ambiente corporativo.
- Linux (CentOS, Fedora, Debian, Suse, Slackware, Kurumin) – Sistema operacional utilizado tanto em servidores quanto estações de trabalho. Possui uma ampla disponibilidade de aplicativos compatíveis, como servidores de rede, aplicativos multimídia, entre outros.
- Mac OS X: Esses computadores rodam em um sistema operacional gráfico proprietário, exclusivo para eles, produzido pela própria Apple.

Figura 2: Gráfico demonstrando a utilização dos sistemas operacionais em desktops domésticos.



Fonte: <https://www.gamerinfo.com.br/2020/05/05/melhor-sistema-operacional-para-pc-fraco/>

Na figura 2, podemos observar os sistemas operacionais mais utilizados a nível de mundo.

4.7 Android

Na atualidade, ninguém vive longe de um celular, este pequeno aparelho se tornou indispensável, seja para mandar um e-mail, tirar uma foto, acompanhar as redes sociais, conversar com amigos, etc. Pesquisas apontam que cerca de mais de 3 bilhões de pessoas possuem um telefone celular, e com esse sucesso todo o mercado de aplicativos virou uma febre rendendo assim bilhões todos os anos. (LECHETA, 2016).

Na visão de Lecheta (2016), nos traz um panorama de como o mercado está se compondo frente a busca por aparelhos de fácil acesso.

Nesse mercado competitivo, temos vários lados da moeda. Os usuários comuns buscam um celular com um visual elegante, moderno, de fácil navegação, assim como uma infinidade de aplicativos e recursos. Tanto as empresas quanto os desenvolvedores buscam uma plataforma moderna e ágil para desenvolver aplicativos. Os fabricantes (LG, Motorola, Samsung, HTC, Intel, Sony etc.) precisam de uma plataforma robusta e rica em funcionalidades para lançar no mercado os seus produtos. É aqui que o Android se encaixa, pois ele é perfeito para todos os casos.

Na concepção de Pereira (2009), o Android faz-se uma plataforma de tecnologia móvel, abrangendo um conjunto de programas para celulares, já contendo um sistema operacional. Seguindo a mesma linha de raciocínio, o Android foi criado para permitir que desenvolvedores criassem aplicações móveis para que então pudesse usufruir ao máximo tudo aquilo que um aparelho portátil é capaz de nos proporcionar, a principal proposta é ser verdadeiramente aberto, como por exemplo, uma aplicação pode ser recorrer a qualquer uma das funcionalidades que esses aparelhos apresentam, tais como, realizar chamadas, enviar mensagens de textos e até mesmo fazer uso da câmera, desta maneira permitindo que os desenvolvedores façam ajustes e aprimorem cada vez mais estas funcionalidades. Por possuir uma sistemas open source, permite que seja sempre adaptado conforme surge a necessidade.

Figura 3: Demonstrativo sobre a preferência por Android.



Fonte: <https://www.tudocelular.com/google/noticias/n147787/evolucao-so-mobile-nos-ultimos-20-anos.html>

Na figura 3, podemos ter um breve conhecimento sobre como o Android se coloca como preferência em grande maioria no mercado.

O sistema operacional Android é pertencente da Google e na atualidade é líder mundial neste seguimento, apesar disso, o sucesso do Android não se deve apenas a Google, mas sim de toda a junção, dos gigantes envolvidos neste processo de desenvolvimento, tais como fabricantes de celulares e operadoras. Este grupo que contribui no desenvolvimento desta plataforma é chamado de OHA (Open Handset Alliance), contando com nomes de peso dentro deste ramo, tais como, Intel, Samsung,

LG, Motorola, ASUS, Acer, Dell, etc. Ainda que a Google seja a representante da maior parte da força do Android, essa dimensão alcançada não seria possível sem a cooperação de outras aliadas potências do mercado móvel.

4.8 Windows

A palavra Windows derivada do inglês, que possui o significado de janelas, de acordo com a fala de Mascarenhas, Sousa e Costa (2016), foi no ano de 1975 em que tudo se iniciou, Bill Gates e Paul Allen, fundaram a Microsoft; já em 2003, a empresa havia se tornado uma corporação global, gerando mais de 50 mil empregos.

Ainda seguindo a mesma linha de pensamento, em 1981, a Microsoft lançou o MS-DOS 1.0 – “Microsoft Disk Operating System” era um sistema operacional de 16 bits que suportava 1MB de memória principal, lembrando que este era um valor alto referente aos padrões da época. Lançado 2 anos mais tarde, nasceu a versão MS-DOS 2.0, que tinha suporte para um disco rígido de 10 MB e disquetes flexíveis de 360 KB. As próximas versões deste sistema seguiam aprimorando sua capacidade de aumentar sua capacidade de disco e junto com isso suportar um número cada vez mais maior de dispositivos periféricos. Em 1985, a Microsoft lançou o Windows 1.0, que era um sistema operacional baseado em interface gráfica de usuário (GUI). Logo surgiu a versão 2.0, que trouxe consigo uma proteção para programas DOS, embora seu desempenho se mostrasse mais lento, ele inibia que o programa sobrescrevesse o espaço de memória de outro programa. Já em 1990, foi lançado o Windows 3.0, logo em seguida, Windows 3.1 e também Windows For Workgroups 3.1, que apresentava suporte à rede, em especial redes locais (LANs) que já estavam se popularizando naquela época.

Ainda falando sobre a evolução que este sistema operacional teve, em 1993, lançou-se no mercado o Windows NT 3.1, que apresentava uma nova tecnologia sendo assim se tornando uma linha de sistema operacional corporativo, embasado em um código-base dissociado do código básico da sua linha de consumo. Esses mesmos continuaram separados até se fusionaram e assim apresentado o Windows XP, em 2001. Os autores a seguir, nos falam mais detalhes sobre o lançamento de 1993, Windows NT 3.1

Como o Windows NT 3.1 foi desenvolvido para usuários empresariais, seu foco principal estava na segurança e na estabilidade. Esse Windows introduziu o Sistema de Arquivos da Nova Tecnologia (New Technology File System ou NTFS), que é mais seguro e mais eficiente do que os então populares sistemas de arquivos, Windows 3.1

FAT e IBM OS/2 HPFS. Além disso, o sistema operacional de 32 bits protege sua memória contra acesso direto por aplicações de usuário. O núcleo do NT executa em seu próprio espaço de memória protegido. Essa camada extra de segurança teve um preço, muitos jogos existentes que utilizavam recursos gráficos intensamente não podiam executar em Windows NT porque precisavam acessar a memória diretamente para otimizar desempenho. Ele não tinha grande parte do suporte de multimídia que tornou o Windows 3.1 popular entre os usuários. (Mascarenhas, Sousa e Costa, 2016, p. 12,13.)

Portanto, segundo a linha de raciocínio do mesmo autor, após o lançamento do Windows XP, em 2003, estreou o Windows Server 2003, em 2007, foi a vez do Windows Vista, em seguida Windows Server 2008, Windows 7 e Windows 8 em 2012, e desde de 2015 temos a versão Windows 10.

Figura 4: Linha do tempo Windows.



Fonte: <https://www.tecmundo.com.br/windows-10/64136-windows-1-windows-10-29-anos-evolucao-do-so-microsoft.htm>

Conforme visto na figura 4, temos a possibilidade de visualizar em formato de linha do tempo a evolução que o sistema operacional Windows comportou no decorrer dos anos.

4.9 Serviço Windows

Conforme estudos, o site da Microsoft nos traz a definição de serviço de Windows como sendo serviços que permitem criar aplicativos executáveis de longa efetivação que são processados em suas próprias sessões do Windows. Este tipo de serviço pode ser inicializado automaticamente quando o computador é inicializado,

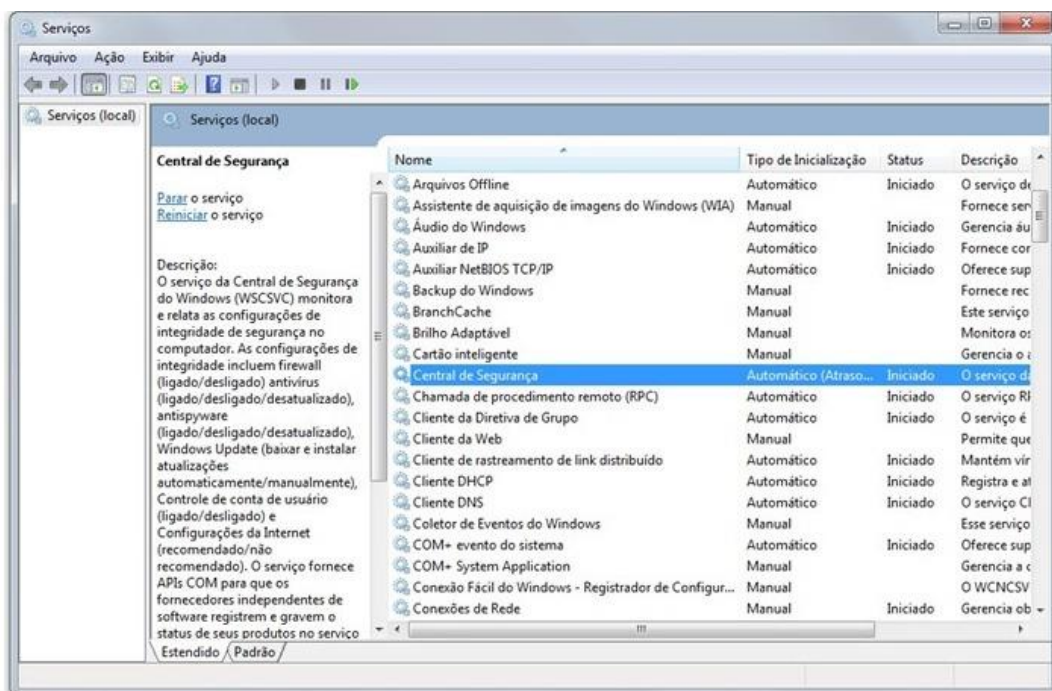
podem ser colocados em pausa ou reiniciado sem afetar a interface do usuário. Este mecanismo nos dá autonomia para serem usados em um servidor ou sempre que necessitar de uma funcionalidade de longa execução que não venha a afetar usuários

Você pode criar serviços facilmente, criando um aplicativo que é instalado como um serviço. Por exemplo, considere que você queira monitorar dados do contador de desempenho e reagir a valores de limite. Você pode escrever um aplicativo de Serviço Windows que escute os dados do contador de desempenho, implante o aplicativo e comece a coleta e a análise de dados. (MICROSOFT, 2017).

Com o auxílio do Microsoft Visual Studio, podemos criar um projeto, instituir um código que vai controlar os comandos que podem ser enviados aos serviços, e quais ações podem ser realizadas quando esses comandos forem recebidos. Dentre os comandos que podem ser enviados a um serviço, estão, iniciar, pausar, retomar e parar serviço. Tendo também a possibilidade em executar comandos personalizados.

Depois de criar e compilar o aplicativo, você pode instalá-lo executando o utilitário de linha de comando InstallUtil.exe e passando o caminho para o arquivo executável do serviço. Você pode usar o Gerenciador de Controle de Serviços para iniciar, parar, pausar, retomar e configurar o serviço. (MICROSOFT, 2017).

Figura 5: Tela sobre serviço Windows.



Fonte: <https://www.tecmundo.com.br/aumentar-desempenho/20343-28-servicos-desnecessarios-do-windows-que-voce-pode-desligar.htm>

Na figura 5, podemos visualizar uma tela sobre o serviço Windows, que devemos ter um certo cuidado ao manuseá-lo, pois caso, seja realizada uma alteração em algum serviço errado poderá afetar a sua rotina de serviços.

Ainda com embasamento no site da própria Microsoft, temos a possibilidade de criar dois tipos de serviços no Visual Studio, usando o .NET Framework. Os serviços que são únicos em um processo, recebem o tipo Win32OwnProcess. Já os serviços que compartilham dos seus processos com outro serviço, recebem o tipo Win32ShareProcess.

4.10 Programação Orientada a Objetos

Desde a invenção dos computadores, a programação tem percorrido um caminho radical para sua evolução. A principal causa para tais mudanças, vem com o objetivo de acomodar a ampliação da complexidade dos programas. Por volta dos anos 60, surgiu a programação estruturada, utilizando este tipo de linguagem, foi possível produzir, um programa razoavelmente complexo. Escritos de maneira consideravelmente fácil. A cada passo obtido na evolução da programação, técnicas foram criadas e aprimoradas para que então programadores pudessem tratar dos sistemas. (CARVALHO; TEIXEIRA, 2012)

A programação orientada a objetos foi criada á partir dos melhores princípios da então programação estruturada que veio de encontro com novos conceitos que levaram a programação a um novo caminho, desta maneira, permitindo que certo problema seja facilmente analisados em subgrupos. (CARVALHO; TEIXEIRA, 2012).

A programação orientada a objetos (POO), se apoiam em quatro pilares do princípio da abstração, são eles: Encapsulamento, herança, abstração e polimorfismo. A obra de Ricarte (2001), nos traz uma definição sobre os pilares a qual a programação orientada a objetos se baseia, através faremos um breve relato sobre a função de cada item:

Figura 6: Pilares do princípio da abstração da POO.



Fonte: <https://homepages.dcc.ufmg.br/~rimsa/documents/decom009/lessons/Aula07.pdf>

Na figura 6, apresentamos de forma mais concreta os pilares que dão suporte ao princípio da abstração, em seguida, faremos uma definição breve sobre a função de cada um deles.

- Encapsulamento, diz respeito a um grupo de propriedades, métodos e outros membros relacionados é referido como uma única unidade ou objeto.
- Herança, este princípio descreve a capacidade de gerar novas classes, sustentadas por uma classe existente.
- Abstração, carrega o significado de ocultar os detalhes desnecessários.
- Polimorfismo, nos traz o sentido de que você pode ter várias classes que possuem a possibilidade de serem usadas de forma intercambiável, ainda que cada classe implemente os mesmos métodos de maneira diferente.

A pesar disso, vale destacar, que este conceito de orientação a objeto, necessita mais do intelecto do programador do que da linguagem de programação que será utilizada. Algumas vantagens são obtidas ao fazer utilização da POO, tais como, redução no custo de manutenção, pois se faz permitir modificar apenas os objetos que requer alterações; também outro ganho é o aumento na reutilização de códigos, pois caso existam dois

objetos bastante semelhantes, com diferenças mínimas, se tem a possibilidade de escrever os métodos apenas uma única vez e utiliza-lo para os dois objetos então.

4.11 Linguagem de programação JAVA

Em concordância com a ideia de Claro e Sobral (2008), podemos afirmar que Java é uma linguagem usada na programação orientada a objetos (POO), criada pela Sun Microsystems, fazendo-se muito parecida com a C++, porem subtraindo os pontos considerados mais complexos, dentre elas, ponteiros e herança múltipla. Este tipo e linguagem é facilmente encontrado em notebook, consoles de games e muito comum em telefones celulares.

Java é uma linguagem de programação muito conveniente para o desenvolvimento de software que funcione em conjunto com a Internet. Ela também é uma linguagem de programação orientada a objetos que utiliza uma metodologia que está se tornando cada vez mais útil no mundo do design de software. Além disso, ela é uma linguagem multiplataforma, o que significa que seus programas podem ser criados para executar do mesmo modo no Microsoft Windows, Apple Macintosh e na maioria das versões de UNIX, incluindo a Solaris. A linguagem Java vai além da área de trabalho, sendo executada em dispositivos como televisões, relógios de pulso e telefones celulares. (LEMAY E CADENHEAD, 2005).

Inspirados nas palavras de Deitei (2003), faremos um breve aparato da história sobre Java. Em 1991, a Sun Microsystems, financiou uma pesquisa com o codinome Green, em seguida este mesmo projeto resultou no desenrolar de uma linguagem baseada em C e C++ que foi chamada de Oak em homenagem a uma arvore que ficava de frente para a janela no seu escritório na Sun, porém mais tarde, descobriram que já havia uma linguagem de computador batizada com este mesmo nome. Então ao visitarem uma cafeteria que possuía o nome de Java, lhe foi sugerido e assim este nome se fixou.

As revisões bibliográficas ainda apontam, que o grupo Green vivenciava algumas dificuldades durante alguns períodos, contudo, em meados de 1993, a World Wide Web se popularizou e então os integrantes da Sun perceberam o verdadeiro potencial de utilizar da linguagem Java para desenvolver páginas Web.

Ainda na visão de Deitei (2003), nos relata que em maio de 1995, o grupo Sun anunciou formalmente Java em um vídeo conferencia e este anuncio gerou uma relevância imediata por conta do interesse pela World Wide Web

Java é agora utilizada para criar páginas da Web com conteúdo interativo e dinâmico, para desenvolver aplicativos corporativos de grande porte, para aprimorar a funcionalidade de servidores da World Wide Web (os computadores que fornecem o conteúdo que vemos em nossos navegadores da Web), fornece aplicativos para dispositivos destinados ao consumidor final (como telefones celulares, pagers e assistentes pessoais digitais) e para muitas outras finalidades.

Figura 7: Exemplos de dispositivos que foram desenvolvidos em linguagem Java.



Fonte: <https://pt.slideshare.net/claudineym/javafx-mobile>

Como podemos observar na imagem 7, a linguagem está muito presente em nosso dia a dia, fazendo com que tenhamos mais conforto com essas contribuições.

Para o próximo item, vamos descrever com um pouco mais de detalhes sobre como o projeto foi planejado para que sua execução fosse realizada.

4.12 Linguagem de programação C# (CSharp)

O C# (CSharp) é uma linguagem de programação orientada a objetos criada pela Microsoft, faz parte da sua plataforma .Net. A companhia baseou CSharpe na linguagem C++ e Java. A criação da linguagem, embora tenha sido feita por vários

desenvolvedores, é atribuída principalmente a Anders_Hejlsberg, hoje um Distinguished Engineer na Microsoft.

Quando comparada com C e C++, a linguagem é restrita e melhorada de várias formas incluindo: ponteiros e aritmética sem checagem só podem ser utilizados em uma modalidade especial chamada modo inseguro (unsafe mode).

Normalmente os acessos a objetos são realizados através de referências seguras, as quais não podem ser invalidadas e normalmente as operações aritméticas são checadas contra sobrecarga (overflow). Os Objetos não são liberados explicitamente, mas através de um processo de coleta de lixo (garbage collector) quando não há referências aos mesmos, prevenindo assim referências inválidas. Destruítores não existem, o equivalente mais próximo é a interface Disposable, que juntamente com a construção using block permitem que recursos alocados por um objeto sejam liberados prontamente.

Figura 8: Criação de método público CSharp.

```
namespace SystemMonitoring.Domain.Features.Services
{
    5 referências
    public enum Intervalo
    {
        [Description("1 Minuto")]
        um = 1,
        [Description("2 Minutos")]
        dois = 2,
        [Description("5 Minutos")]
        cinco = 5,
        [Description("10 Minutos")]
        dez = 10
    }

    0 referências
    public static class ExtensionMethods
    {
        1 referência
        public static string GetDescription(this Enum value)
        {
            Type type = value.GetType();
            string name = Enum.GetName(type, value);
            if (name != null)
            {
                FieldInfo field = type.GetField(name);
                if (field != null)
                {
                    [DescriptionAttribute attr =
                    Attribute.GetCustomAttribute(field,
                    typeof(DescriptionAttribute)) as DescriptionAttribute;
                    if (attr != null)
                    {
                        return attr.Description;
                    }
                }
            }
            return null;
        }
    }
}
```

Fonte: Autoria própria.

Na imagem 8, podemos contemplar a criação de um método público em CSharp.

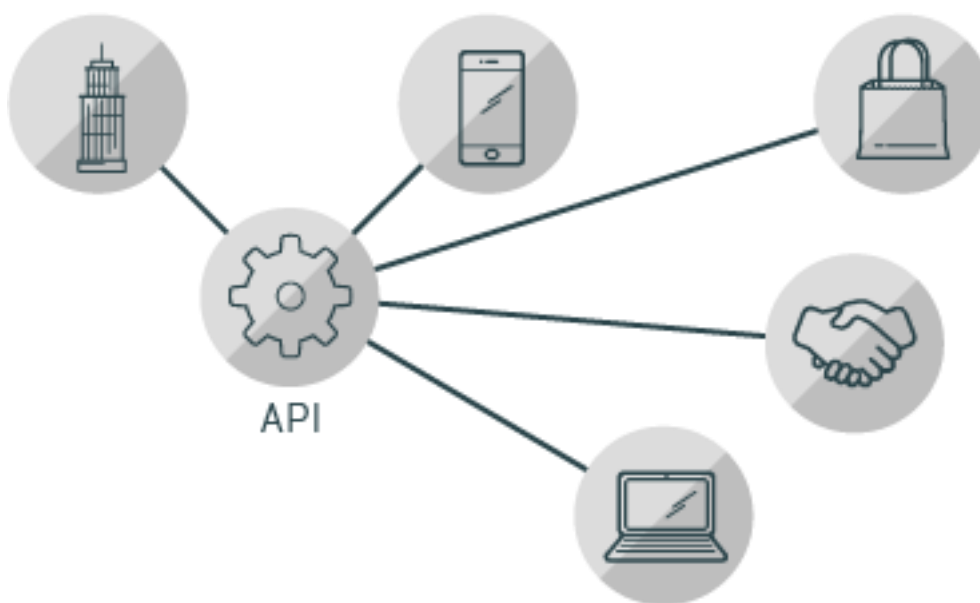
Também existem finalizadores, mas como no Java, não é permitida herança múltipla, mas uma classe pode implementar várias interfaces abstratas. O objetivo principal é simplificar a implementação do ambiente de execução.

4.13 API

Para falar sobre APIs, inicialmente devemos reconhecer que toda comunicação desta interface é feita via web, isto significa que, tudo é feito através de uma requisição a uma URL, que por sua vez traz respostas. Atualmente serviços de internet são bem populares, WebServices foi desenvolvido e consumido durante muito tempo por ser uma tecnologia independente de plataforma. Mais à frente, a Microsoft criou o Remoting e por fim o famoso WCF que englobou (HTTP, TCP, MQ), essas duas tecnologias são baseadas em SOAP que por sua vez é fundamentado em XML e busca padronizar o conteúdo que é trafegado entre as pontas. No entanto, a SOAP apresentava um certo problema, porque suas plataformas não conseguiam acompanhar sua evolução, devido esta causa, começou a aparecer alternativas de soluções mais simples.

Uma solução alternativa ao SOAP e altamente adotada é o REST (Representational State Transfer), baseado totalmente em HTTP e seus recursos. Para o melhor entendimento sobre ASP.Net Web API e REST sugiro conhecer um pouco sobre o protocolo HTTP. (PIRES, 2013).

Figura 9: Esquema representando o serviço de uma aplicação API.



Fonte: <https://www.redhat.com/pt-br/topics/api>

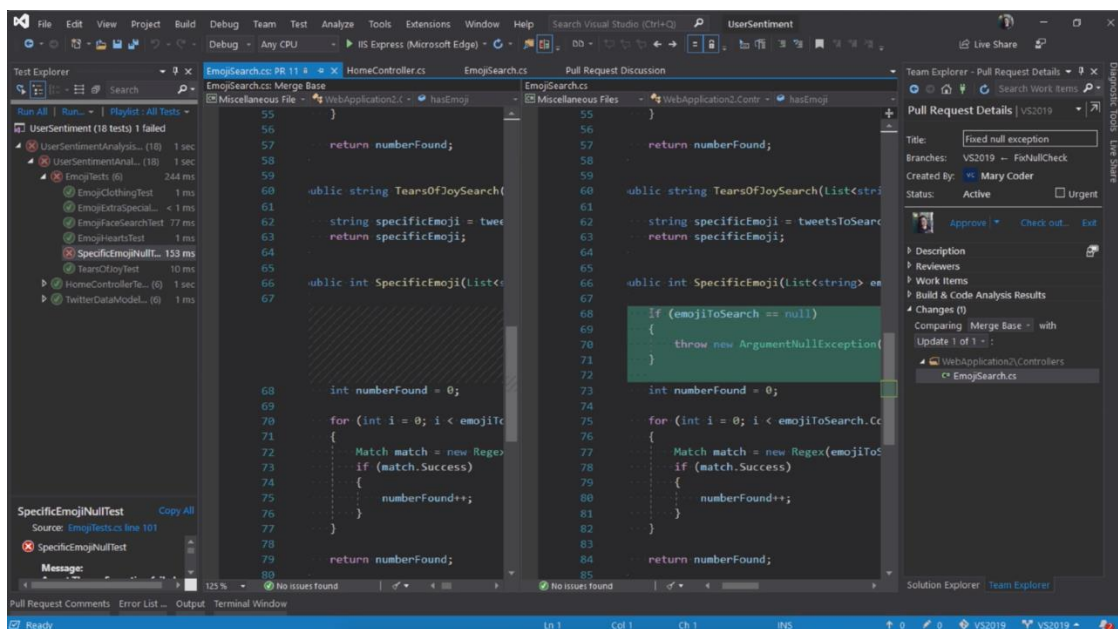
Conforme visto na figura 9, podemos perceber uma integração de serviços e aplicações, desta maneira tendo a possibilidade de ser particular, de parceria ou até mesmo pública.

Grandiosas empresas, tais como a Google, Facebook, LinkedIn, Netflix entre outras já disponibilizam APIs Web para serem consumidas, todas baseadas em HTTP com REST.

4.14 Visual Studio Community

Um IDE (ambiente de desenvolvimento integrado) é um programa cheio de recursos que visam facilitar o ofício de um programador, visto que esta ferramenta tem uma gama de itens que vem para somar do desenvolvimento de algum mecanismo, além do editor e depurador.

Figura 10: Interface Visual Studio Community.



Fonte: <https://www.windowmode.com/wp-content/uploads/2019/05/VS-2019-code-review.png>

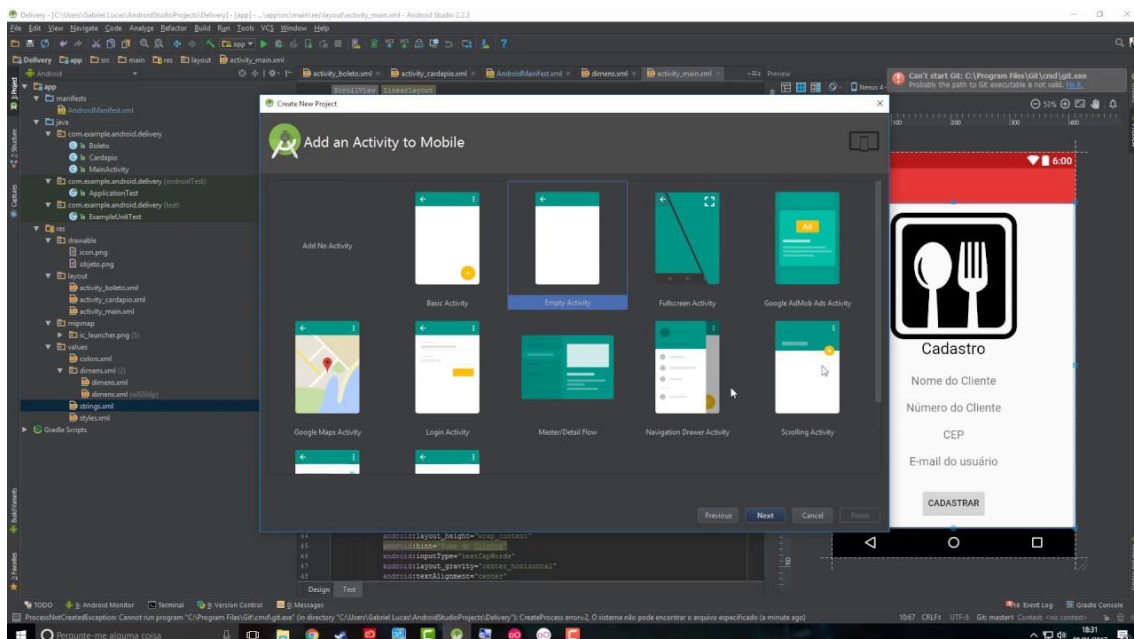
O Visual Studio ainda conta com compiladores, ferramentas de preenchimento de código, designers gráficos, e muitos outros recursos para promover o desenvolvimento de um software (MICROSOFT, 2019).

4.15 Android Studio

Assim como o Visual Studio visto anteriormente, o Android Studio, da mesma forma, é um ambiente de desenvolvimento integrado (IDE), porem este ambiente é voltado para desenvolver app's para Android que é fundamentado no IntelliJ IDEA. Além da sua função de editar códigos e das ferramentas de desenvolvedor avançadas, o Android Studio, disponibiliza mais recursos que auxiliam no rendimento de compilações de apps Android, o site Developers nos apontam essas funcionalidades cooperativas:

- Um sistema de compilação flexível baseado em Gradle
- Um emulador rápido com inúmeros recursos
- Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android
- A aplicação de alterações para enviar alterações de código e recursos ao aplicativo em execução sem reiniciar o aplicativo
- Modelos de código e integração com GitHub para ajudar a criar recursos comuns de apps e importar exemplos de código
- Frameworks e ferramentas de teste cheios de possibilidades
- Ferramentas de lint para detectar problemas de desempenho, usabilidade, compatibilidade com versões, entre outros
- Compatibilidade com C++ e NDK
- Compatibilidade integrada com o Google Cloud Platform, facilitando a integração do Google Cloud Messaging e do App Engine.

Figura 11: Interface Android Studio.



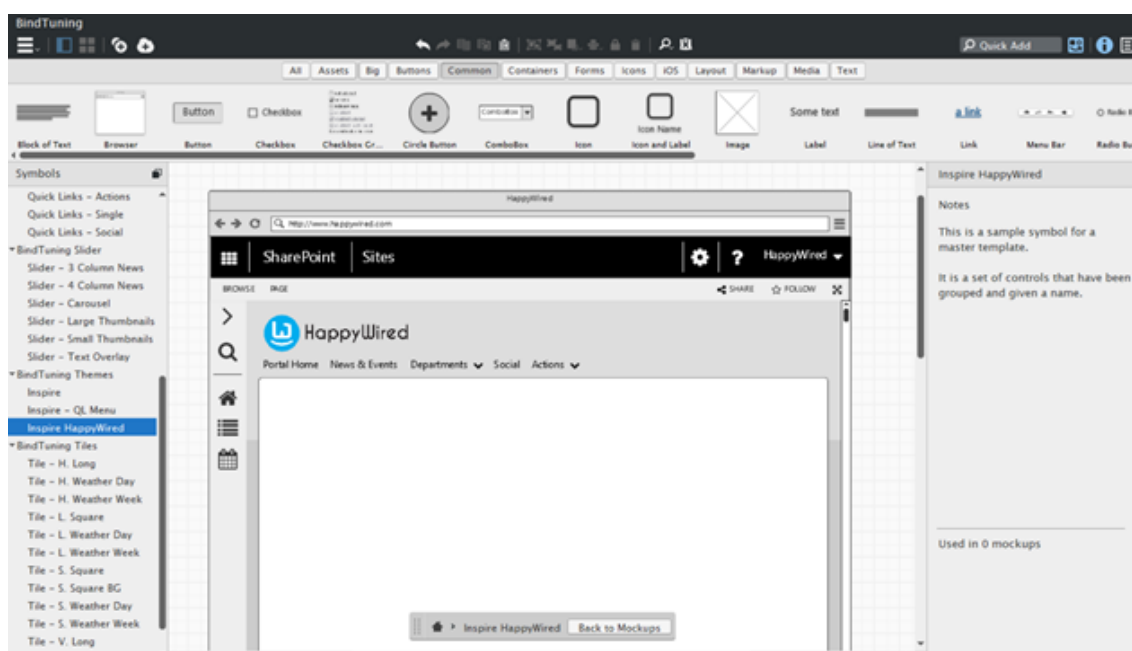
Fonte: <https://i.ytimg.com/vi/y4vI0Kn4yqU/maxresdefault.jpg>

Cada projeto realizado no Android Studio possui módulos que contém arquivos de código-fonte e de recursos dentre eles: módulos de apps Android; módulos de biblioteca; módulos do Google app Engine.

4.16 Balsamiq Wireframes

Esta ferramenta elaborar protótipos ou modelos tais como tela de sistema desktop, web ou mobile. Esta aplicação, foi criada baseando-se na linguagem de programação ActionScript que utiliza adobe AIR. (HEBERLE, 2016)

Figura 12: Interface Balsamiq Wireframes.



Fonte: https://www.happywired.com/wpcontent/uploads/2017/09/092117_1739_Wireframing2.png

Já por sua vez, na concepção de Santos (2006), protótipos, são compreendidos como modelos funcionais servindo de simulador de aparência e software a ser desenvolvido posteriormente. Por meio do protótipo, é possível, avaliar, interagir, fazer alterações e aprovar características do software, estas avaliações podem ser realizadas, tanto pelos futuros usuários quanto pelos próprios desenvolvedores.

4.17 Windows Forms

Windows Forms se trata de uma tecnologia Smart Client adequado para o .NET Framework, formando assim, um conjunto de bibliotecas regulares que simplificam as tarefas comuns do aplicativo, como por exemplo, ler e gravar no sistema de arquivos. Ao fazer uso desta ferramenta, um formulário, se torna uma superfície visual.

Segundo a Microsoft (2017), o Windows Forms é uma tecnologia de clientes inteligente ¹ para o .NET Framework, refere-se a uma coleção de bibliotecas gerenciadas que facilitam as tarefas comuns de aplicativos tais como, leitura e gravação no sistema de arquivos. Ao fazer uso deste ambiente de desenvolvimento, você tem a possibilidade de criar aplicativos cliente inteligente que exibem informações, solicitam entrada de usuário, e fazem comunicações com computadores remotos em uma rede.

O Windows Forms também traz consigo, uma diversidade de controles que podem contribuir aos formulários, tais como, controles que exibem caixas de texto, botões de opções, entre outros. Essa ferramenta conta também, com controles desenvolvidos a partir de interface de usuário que emulam recursos em aplicativos de alta tecnologia, como por exemplo, o Microsoft Office.

¹ *Cientes inteligentes* são aplicativos graficamente ricos que são fáceis de implantar e atualizar, podem funcionar quando eles estão conectados ou desconectados da Internet e podem acessar recursos no computador local de maneira mais segura do que os aplicativos tradicionais baseados no Windows. (MICROSOFT, 2017).

5 METODOLOGIA

5.1 Documentação

A Pesquisa bibliográfica ou documental tem sua finalidade ligada ao posicionamento do pesquisador quanto ao assunto abordado de maneira que entre em contato com todo o conteúdo que exista sobre determinado assunto, englobando por completo a bibliografia e documentos públicos (MARCONI; LAKATOS, 2001).

A utilização desses documentos possui suma importância nas explicações e pode ser considerada uma grande fonte de informações, tendo em vista o esclarecimento mais amplo, bem como a afirmação concisa de determinado tema.

5.2 Natureza da pesquisa

A pesquisa será de natureza qualitativa, conforme destaca Gibbs (2009):

Análise de dados qualitativos oferece uma base para analisar todos os tipos de dados qualitativos verbais, como entrevistas, grupos focais e biografias. Examina as várias abordagens e questões práticas relacionadas ao entendimento dos dados qualitativos. Destaca o uso da informática na pesquisa qualitativa e auxilia na elaboração descritiva dos dados.

5.3 Método da pesquisa

O método a ser usado nessa pesquisa será o relato de experiência, pois visa a identificação e análise de um determinado problema, busca de soluções ou implementação de soluções na prática, ou ainda, identificação e análise dos resultados da implementação de uma determinada solução.

5.4 Técnicas da pesquisa

Como base desta pesquisa, será utilizada a técnica de observação, visto que a ideia principal deste trabalho é identificar problemas recorrentes e comuns dos servidores e com isso apresentar soluções

6 CRONOGRAMA

O seguinte cronograma foi utilizado para desenvolvimento deste trabalho.

Quadro 1: Cronograma

Atividade	Mar	Abril	Maio	Jun.	Jul
Revisão de literatura	X	X			
Estudo de técnicas		X	X		
Especificações do protótipo			X	X	
Ajustes metodologia				X	
Entrega do TCC à coordenação					X
Defesa TCC					X

Fonte: Arquivo do Autor.

7 PROJETO

No presente artigo, elaboramos e destacamos a importância dos monitoramentos de sistemas, viabilizando o quanto ele se faz presente em nosso dia a dia, um exemplo disto é a importância de um pagamento de conta através de uma agência bancária. A partir disto, se torna importante zelar pela saúde do monitoramento para que desta maneira não haja imprevistos, evitando assim as falhas repentinas que podem ocorrer. Desta forma, na elaboração de um sistema que visa o auxílio no monitoramento, obtivemos a prevenção de paradas ocasionados por sobrecarga com relação a consumo excessivo de processamento. O software desenvolvido neste trabalho foi implantado no ambiente Pollux da empresa NDD, este ambiente é responsável por simular o processamento Sefaz e nele são realizados diversos testes das aplicações da empresa, dentre os testes realizados estão os de benchmark, testes estes que demandam de bastante recurso de processamento do ambiente.

Para que fosse possível alcançar esse objetivo, foi desenvolvido um sistema baseado na linguagem de programação CSharpe e sistema de armazenamento SQL Server, onde toda a parte visual do sistema foi realizada através da IDE Visual Studio Communit, visando ser aplicado no sistema operacional Windows, se tornando ideal para usuários inexperientes e empresas de pequeno porte.

7.1 SYS Monitor

Tela inicial exibindo a sidebar e todos os grupos de configuração. (Figura 13).

Figura 13: Tela Inicial



Fonte: Autoria própria

Tela de configuração do banco de dados. Esta tela permite ao usuário informar os dados de acesso do banco de dados SQL, o qual será utilizado para salvar todas as configurações e rotinas do sistema. (Figura 14).

Figura 14: Configuração Banco de Dados

SYS Monitor

Configuração Banco de dados

Endereço (Host)
127.0.0.1

Nome do banco
DBSystemMonitoring

Usuário
sa

Senha
.....

Autenticação Windows

Testar Conexão

Salvar

Fonte: Autoria própria

Tela de configuração de e-mail. Esta tela permite ao usuário definir os dados do servidor e conta de e-mail utilizados nos envios das notificações do sistema. (Figura 15).

Figura 15: Configuração Servidor de E-mail.

SYS Monitor

Configuração Servidor de E-mail

Host
smtp.gmail.com

Protocolo
SMTP

Porta
587

E-mail de envio

Endereço de e-mail
sysmonitor@gmail.com

Senha
.....

Salvar

Fonte: Autoria própria

Tela de configuração de notificações. Esta tela permite ao usuário definir os destinatários interessados nas notificações, o título dos e-mails enviados e também a geração do token de autenticação utilizado no aplicativo de notificações. (Figura 16).

Figura 16: Configurações de Notificações

SYS Monitor

Configurações de Notificações

Notificações via E-mail

Endereço de e-mail
alio.hintze95@gmail.com

Título do e-mail
SYS Monitor

Notificações via APP

TOKEN
A8403E51-FBF9-46AE-987A-A243FD189C96

Salvar

Fonte: Autoria própria

Tela de configuração de monitoramento de recursos. Esta tela permite ao usuário definir quais recursos serão monitorados, como por exemplo espaço em disco e consumo de CPU (Figura 17).

Figura 17: Configuração Monitoramento de Recursos.

SYS Monitor

Configuração Monitoramento de Recursos

Monitorar repartição C:
Define o espaço mínimo (GB)
_10_GB

Monitorar memória RAM
Define o consumo limite (%)
_30

Monitorar CPU
Define o consumo limite (%)
_30

Monitorar CPU (Temperatura °C)
Define a temperatura limite (°C)
_30

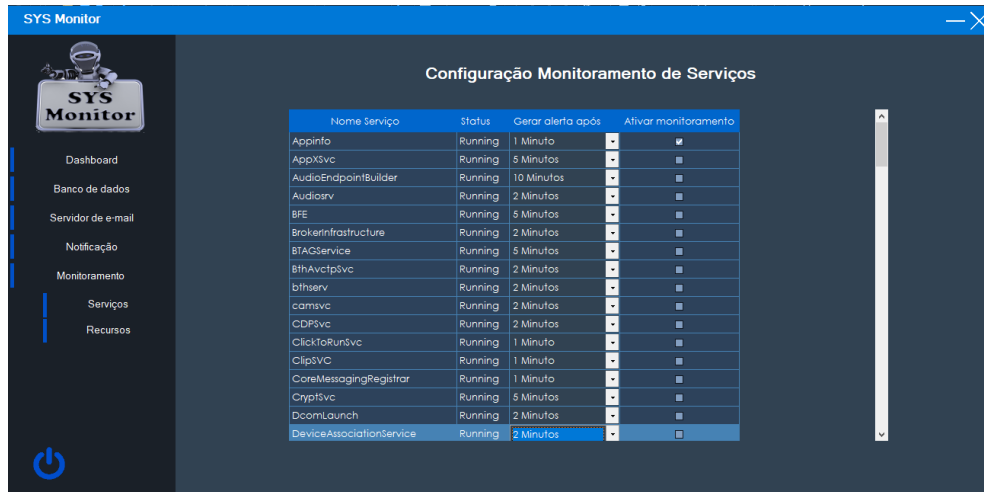
Verificar recurso a cada:
_5_minutos

Salvar

Fonte: Autoria própria

Tela de configuração de monitoramento de serviços. Esta tela permite ao usuário definir quais serviços Windows serão monitorados e também o tempo que o sistema deve aguardar antes de gerar as notificações (Figura 18).

Figura 18: Configuração Monitoramento de Serviços

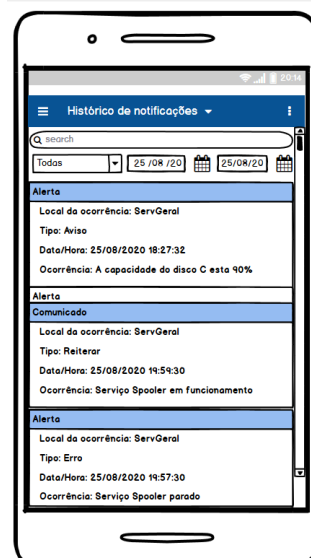


Nome Serviço	Status	Gerar alerta após	Ativar monitoramento
Appinfo	Running	1 Minuto	<input checked="" type="checkbox"/>
AppXSvc	Running	5 Minutos	<input type="checkbox"/>
AudioEndpointBuilder	Running	10 Minutos	<input type="checkbox"/>
Audiosrv	Running	2 Minutos	<input type="checkbox"/>
BFE	Running	5 Minutos	<input type="checkbox"/>
BrokerInfrastructure	Running	2 Minutos	<input type="checkbox"/>
BTAGService	Running	5 Minutos	<input type="checkbox"/>
BthAvctpSvc	Running	2 Minutos	<input type="checkbox"/>
bthserv	Running	2 Minutos	<input type="checkbox"/>
camsvc	Running	2 Minutos	<input type="checkbox"/>
CDPSvc	Running	2 Minutos	<input type="checkbox"/>
ClickToRunSvc	Running	1 Minuto	<input type="checkbox"/>
ClipSVC	Running	1 Minuto	<input type="checkbox"/>
CoreMessagingRegistrar	Running	1 Minuto	<input type="checkbox"/>
CryptSvc	Running	5 Minutos	<input type="checkbox"/>
DcomLaunch	Running	2 Minutos	<input type="checkbox"/>
DeviceAssociationService	Running	2 Minutos	<input type="checkbox"/>

Fonte: Autoria própria

Tela de histórico de notificações. Esta tela tem como objetivo facilitar a visualização das ocorrências do sistema, através dela será possível visualizar todos os alertas, avisos e informações ou filtrar por algo específico (Figura 19).

Figura 19: Histórico de Notificações



Fonte: Autoria própria

8 LIMITAÇÕES DA PESQUISA

O projeto SYS Monitor foi desenvolvido para ser utilizados por empresas de pequeno à médio porte, que pretendem monitorar seus ambientes, mas não podem ou não pretendem custear o uso de softwares mais complexos e já consagrados no mercado.

9 RESULTADOS

O software desenvolvido neste trabalho foi implantado no ambiente Pollux da empresa NDD, este ambiente possui um comportamento semelhante ao ambiente de processamento da SEFAZ, o mesmo é utilizado pelo time de testes para homologar as funcionalidades dos produtos da empresa. A utilização da solução neste ambiente proporcionou aos responsáveis pelo ambiente, identificar momentos de gargalos ou de insuficiência de recursos e trata-los antes que os mesmos comprometessem o correto funcionamento do ambiente

10 REFERÊNCIAS

ANDERSON, T.; LEE, P. A. **Fault Tolerance Principles and Practice**. Englewood Cliffs, Prenticehall. 1981. Data de acesso: 25 de nov. 2019.

CADENHEAD, R; LEMAY, L. **Aprenda Java em 21 dias 2**. Rio de janeiro: editora campus 1ª edição, 2003.

CARVALHO, V. A; TEIXEIRA, G. F. **Programação orientada a objetos: curso técnico em informática**. Colatina: IFES, 2012. Disponível em: http://redeotec.mec.gov.br/images/stories/pdf/eixo_infor_comun/tec_inf/081112_prog_obj.pdf . Acesso em: 30/08/2020.

CLARO, D. B; SOBRAL, J. B. M. **Programação em Java**. Florianópolis, 2008.

COUTINHO, B. C. **Sistemas operacionais: curso técnico em informática**. Colatina: 2010. Disponível em: http://proedu.rnp.br/bitstream/handle/123456789/711/Sistemas_Operacionais_web.pdf?sequence=3&isAllowed=y . acesso em: 25/08/2020

DAVID L. Multer; ROBERT E. Garner; LEIGHTON A. Ridgard; LIAM J. Stannard; DONALD W. Cash, **Management Server For Synchronization System**, 2004. Disponível em: <https://patents.google.com/patent/US6757696B2/en>. Data de acesso: 26 de nov. 2019

DEITEI, H. M; DEITEI, P. J. **Java, como programa**. Porto alegre: editora Bookman, 4ª edição, 2003.

FONSECA, G. **Internet – Uma Visão Do “Antes” E “Depois”**. 2011. Disponível em: <https://www.profissionaisti.com.br/2011/11/internet-uma-visao-do-antes-e-depois/>. Data de acesso: 26 de nov. 2019

GIBBS, Graham. **Análise de dados qualitativos**. Porto Alegre: Artmed, 2009. (Coleção pesquisa qualitativa). Disponível em: <http://www.ufrgs.br/bibicbs/livros-novos/gibbsanalise-de-dados-qualitativos>. Data de acesso: 25 de nov. de 2019.

HEBERLE, Moises. **Ferramenta para prototipação de interfaces e apoio ao mapeamento de requisitos de sistema.** Disponível em: <https://www.univates.br/bdu/bitstream/10737/1394/1/2016MoisesHeberle.pdf> . Acesso em: 25/08/2020.

J. F. KUROSE, K. W. ROSS, “**Redes De Computadores E A Internet**”, Addison Wesley, 5a Ed., 2010. Disponível em: https://www.academia.edu/22391313/Redes_de_Computadores_e_a_Internet_Kurose_5_ed. Data de acesso: 27 de nov. de 2019.

LECHETA, Ricardo R. **Android Essencial: edição resumida do Google Android.** São Paulo: Novatec editora Ltda., 2016.

LIMA, T. **O guia completo de APIS RESTFUL.** Disponível em: <https://www.linkapi.solutions/uploads/2019/10/LinkApi-eBook-APIs-RESTful.pdf> . Acesso em: 25/08/2020.

MARCONI, M. de A.; LAKATOS, E. M. **Técnicas de pesquisa: planejamento e execução de pesquisas, amostragens e técnicas de pesquisa, elaboração, análise e interpretação de dados.** São Paulo: Atlas, 2001. Disponível em: https://docente.ifrn.edu.br/olivianeta/disciplinas/copy_of_historia-i/historia-ii/china-eindia. Data de acesso: 28 de nov. de 2019.

MASCARENHAS, G. N; SOUSA, L. M; COSTA, L. K. **Linux X Windows.** Disponível em: <http://projetointegrador.go.senac.br/~172M154200225/trabalhos/03/SO.pdf> . Acesso em: 26/08/2020.

MAZIERO, C. A. **Sistemas operacionais: conceitos e mecanismos.** Curitiba: 2019. Disponível em: <http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-livro.pdf> . Acesso em: 25/08/2020.

MICROSOFT. **Bem-vindo ao IDE do Visual Studio.** Disponível em: <https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019> . Acesso em: 29/08/2020.

MICROSOFT. **Introdução a aplicativos do Serviço Windows**. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/framework/windows-services/introduction-to-windows-service-applications> . Acesso em: 28/08/2020.

MICROSOFT. **Programação orientada a objeto (C#)**. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/framework/winforms/windows-forms-overview> . Acesso em: 29/08/2020.

MICROSOFT. **Visão geral de Windows Forms**. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/framework/winforms/windows-forms-overview> . Acesso em: 29/08/2020

NEIL, S.; **Safetycritical Computer Systems**. Addison Wesley, 1996. Disponível em: <https://dl.acm.org/citation.cfm?id=524721>. Data de acesso: 29 de nov. de 2019.

PAULINO, B.C.G; VIANA, H.C. **Introdução ao ambiente Windows**. Manaus: centro de educação tecnológica do amazonas, 2010.

PEREIRA, L. C. O. SILVA, M. L. **Android para desenvolvedores**. Rio de Janeiro: editora Brasport, 2009

PIRES, Eduardo. **ASP.NET WEB API -meu primeiro serviço REST**. Disponível em: <https://www.eduardopires.net.br/2013/07/asp-net-web-api-meu-primeiro-servico-rest/> . Acesso em: 28/08/2020.

Portal da NF-e 2019 - Nota Fiscal Eletrônica. Disponível em: <http://www.nfe.fazenda.gov.br/portal/perguntasFrequentes.aspx?tipoConteudo=E4+tmY+ODf4=>. Data de acesso: 30 de nov. de 2019.

RICARTE, Ivan Luiz Marques. **Programação Orientada a Objetos: Uma Abordagem com Java**. Campinas, 2001.

SANTOS, R. L. G. **Usabilidade de interfaces para sistemas de recuperação de informação na web**. Disponível em: https://www.maxwell.vrac.puc-rio.br/9731/9731_11.PDF . Acesso em: 27/08/2020.

SPECIALSKI, Elizabeth Sueli. **Gerência de Redes de Computadores e de Telecomunicações**, s.d. Disponível em: <http://cassio.orgfree.com/disciplinas/gredes/ApostilaGerenciamento.pdf>. Acesso em: 25/11/2020.

STRAPAZZON, M. Q. **Diferenças de Sistemas Operacionais: Windows e Linux.**

Disponível em: <https://www.studocu.com/pt-br/document/pontificia-universidade-catolica-do-rio-grande-do-sul/sistemas-operacionais/outro/artigo-sobre-as-diferencas-de-windows-e-linux/4946430/view> . Acesso em: 29/08/2020.

TAKAHASHI, T; LIESENBERG, H. K. E. **Programação orientada a objetos.**

Disponível em: http://www.pedrocarvalho.com.br/PDF/TUTORIAL_ANALISE_O_OBJETO.pdf .

Acesso em: 26/08/2020.