

CENTRO UNIVERSITÁRIO UNIFACVEST
JOÃO VITOR PEREIRA DA MAIA

**DESENVOLVIMENTO DE UM DISPOSITIVO
MICROCONTROLADO PARA A GESTÃO DE
OUTSOURCING DE IMPRESSÃO**

LAGES, SC

2023

JOÃO VITOR PEREIRA DA MAIA

**DESENVOLVIMENTO DE UM DISPOSITIVO
MICROCONTROLADO PARA A GESTÃO DE
OUTSOURCING DE IMPRESSÃO**

Trabalho de conclusão de curso apresentado ao Centro Universitário UNIFACVEST como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador (a): Cassandro Albino Devenz

Orientador (a): Jean Carlos Macedo

Orientador (a): Igor Muzeka

LAGES, SC

2023

JOÃO VITOR PEREIRA DA MAIA

**DESENVOLVIMENTO DE UM DISPOSITIVO
MICROCONTROLADO PARA A GESTÃO DE
OUTSOURCING DE IMPRESSÃO**

Trabalho de conclusão de curso apresentado ao Centro Universitário UNIFACVEST como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Aluno: João Vitor Pereira da Maia
Orientador (a): Cassandro Albino Devenz
Orientador (a): Jean Carlos Macedo
Orientador (a): Igor Muzeka

Lages, SC ___/___/2023. Nota _____ (data de aprovação) (assinatura do orientador do trabalho)

RESUMO

O presente trabalho tem como objetivo o desenvolvimento de um dispositivo microcontrolado, capaz de coletar informações de impressoras de forma remota. A prática da terceirização de alguns setores é comum em várias empresas. O intuito do trabalho a seguir é auxiliar a coleta desses dados por parte de empresas que fornecem a terceirização do parque de impressão. A partir desses dados, a empresa que fornece o aluguel das impressoras será capaz de gerar relatórios financeiros sem a necessidade de uma visita física às instalações dos clientes. Para isso foi utilizado um microcontrolador ESP32 e o protocolo SNMP para a obtenção dos dados necessários.

Palavras-chave: Impressora. Microcontrolador. *Outsourcing* de impressão.

ABSTRACT

The following paper aims to develop a microcontroller device capable of collecting information from printers remotely. The practice of outsourcing some sectors is common in several companies, the purpose of the work below is to help the collection of this data by companies that provide outsourcing of the printing park. From this data, the company that rents the printers will be able to generate financial reports without the need for a physical visit to the customers' premises. For this, an ESP32 microcontroller and the SNMP protocol were used to obtain the necessary data.

Keywords: Printing outsourcing, printer, microcontroller

1. INTRODUÇÃO

Os serviços de impressão estão presentes no dia a dia de todos, quando é necessária a assinatura de um contrato, a obtenção de um documento legal como a certidão de nascimento, exames médicos, testes escolares, dentre outros. De acordo com Napoleão Filho, *et al* (2013), no contexto de impressão, o outsourcing se trata do serviço de aluguel, manutenção, reposição de suprimentos e gerenciamento de cópias. “O outsourcing é uma das melhores ideias que permite a muitas empresas se concentrarem no que fazem de melhor e terceirizar o que outras empresas podem fazer melhor, mais rápido, mais barato e com maior qualidade” (SOMJAI, 2017, p. 01).

Para a viabilidade desse modelo de negócio, a contratada deve estabelecer rotinas de captação de informações das impressoras alugadas. Dessa forma, no outsourcing de impressão, se faz necessária a criação de ferramentas automatizadas que realizem o trabalho de captura dos dados necessários para o faturamento do uso das impressoras, a fim de redução de custos.

2. Objetivo Geral

Criar um dispositivo capaz de facilitar a gestão das impressoras por parte do provedor, sem a necessidade do acesso físico ao local e não estar sujeito a remoção ou modificação de comportamento por pessoas não autorizadas, bem como oferecer essas informações em uma interface web de acesso controlado pelo provedor.

3. Objetivos Específicos

- Enumerar de forma automática as impressoras presentes na rede do cliente;
- Capturar através do protocolo SNMP as informações pertinentes ao provedor;
- Exibir essas informações em uma interface web.

4. Justificativa

A área de outsourcing de impressão é importante, tanto na área privada, quanto na área pública. Apesar de nos tempos atuais existirem meios de armazenar e utilizar documentos digitais, ainda são amplamente utilizados documentos físicos. Logo surge a problemática de manter as impressoras em pleno funcionamento, além de garantir a sua manutenção. Apesar de já existir soluções que abstraem a parte da coleta de relatório das impressoras, essas possuem

um problema crônico, necessitam de um computador na mesma rede da impressora. Isso expõe o software aos seguintes problemas:

- Erro humano, através de remoção indevida do software de monitoramento;
- Atualização do sistema operacional, podendo tornar o software obsoleto;
- Antivírus identificando indevidamente o software como malicioso.

5. Referencial Teórico

5.1 SNMP

SNMP (Simple Network Management Protocol) é um protocolo de gerenciamento de dispositivos. Segundo Case *et al.* (1988) a necessidade dele se dá pelo fato de cada dispositivo ter sua própria interface de gerenciamento, seja ela por linha de comando ou interface gráfica. Logo se faz necessário um protocolo universal de gerenciamento. O mesmo se divide entre duas categorias, o “Agente” e o “Gerente”. O “Agente” seria o dispositivo a ser gerenciado, que contém as informações pertinentes ao “Gerente”. Já o “Gerente” é o dispositivo que deseja colher essas informações do “Agente”. O “Agente” pode ser um roteador, impressora, câmeras IP, switches, servidores, por exemplo.

O protocolo se comunica através de uma porta UDP, por padrão 161 e se utiliza do protocolo de codificação ASN.1, é obrigatório que toda implementação do protocolo SNMP contenha todos os cinco PDUs: GetRequest-PDU, GetNextRequest-PDU, GetResponse-PDU, SetRequest-PDU e Trap-PDU (CASE, 1988). O PDU é uma unidade de informação compartilhada entre dois pares (MIT).

ASN.1 é uma notação para descrever tipos e valores abstratos. Por definição, um tipo abstrato é um tipo que não pode ser instanciado. O antagônico de tipo abstrato é um tipo concreto, como um ponto flutuante, que pode ser instanciado sem conhecimento prévio. Grosso modo, essa notação foi criada pela necessidade de sistemas fazerem a troca de informações sem necessitar saber como o sistema no qual você está consumindo foi implementado, como descreve Kaliski (1993).

5.2 PDUs do SNMP

Conforme definido por Carpenter (1987), o PDU é a menor mensagem possível a ser transmitida entre duas entidades.

Como já citado, sistemas que implementem o protocolo SNMP devem conter os cinco PDUs, são esses:

TABELA 1 – Tipos de PDU

Tipo de PDU	Descrição	Fonte
GetRequest	PDU enviado pelo gerente solicitante um ou mais valores do objeto gerenciado	Loiola, 2014, p.19
GetNextRequest	Enviado pelo gerente para receber o próximo OID especificado no PDU, utilizado para percorrer a árvore da MIB	IBM, 2021
Trap	Enviada pelo agente de forma não solicitada para notificar um evento incomum ocorrido na rede	Venâncio Neto, 2001
GetResponse	Enviado pelo agente como resposta a uma solicitação do gerente, podendo ser uma GetRequest, GetNextRequest ou SetRequest	Venâncio Neto, 2001
SetRequest	Enviado pelo gerente para alterar um valor da variável na MIB, definido pelo gerente	IBM, 2021

5.3 ESP32

O ESP32, lançado inicialmente em 2016 é uma unidade microcontroladora integrada com um módulo WiFi e Bluetooth para o desenvolvimento de propósito geral de aplicações. O mesmo foi desenvolvido para atender a demanda de IOT's que necessitem de integração com WiFi, Bluetooth e gerenciamento de energia (Espressif, 2023). Possui dois núcleos do microprocessador Xtensa® 32-bit LX6, de até 240MHz, 520KB de memória SRAM e 4MB de memória flash, para armazenar o software desejado.

5.4 C++

C++ é uma linguagem de programação orientada a objetos fortemente tipada, que é uma evolução da linguagem C (ISO, 2022). As evoluções entre as linguagens se destacam:

- Adição de novos tipos;
- Classes;

- Namespaces;
- Exceções.

O principal motivo para a evolução do C ao C++ foi a sua complexidade. Sistemas grandes em C se tornavam muito difíceis de serem mantidos e até mesmo compreendidos em sua totalidade (RICARTE, 2001). "O objetivo de C ++ é permitir que programadores possam gerenciar e compreender programas maiores e mais complexos." (RICARTE, 2001). Essa linguagem foi escolhida por existir um amplo catálogo de bibliotecas já desenvolvidas para a mesma, além da facilidade de utilização em relação a linguagem C.

5.5 MIB

Para que se entenda o que é uma MIB, deve-se primeiro saber o conceito de objetos gerenciados. "Um objeto gerenciado é a visão abstrata de um recurso real do sistema" (UFRJ, 2023). Logo, se um recurso deve ser gerenciado ele é modelado e se torna um objeto gerenciado. A MIB nada mais é que um conjunto de objetos gerenciados, contendo todos os valores pertinentes a se fazer a gerência de uma rede.

Os objetos são organizados em uma árvore hierárquica e são acessados através de um identificador de objeto (OID), que são representados através de números separados por pontos, exemplo: 1.3.6.1.1.

6. Desenvolvimento

Essa seção se dedica a parte do desenvolvimento da funcionalidade da aplicação utilizando as tecnologias acima mencionadas.

6.1 Client

O Client é a parte mais importante do projeto. Nele vai estar toda a lógica responsável por fazer o mapeamento da rede, identificar as impressoras, coletar os dados desejados e enviar para um servidor de processamento externo. Para o desenvolvimento do client será o utilizado o dispositivo ESP32, aproveitando o seu chip Wifi integrado.

6.2 Mapeamento da rede

Para que o sistema funcione com o menor envolvimento humano, o aparelho deve buscar as impressoras na rede de forma automática. Abaixo segue o código gerado em C++ capaz de realizar essa funcionalidade:

Figura 1 - Busca de Impressoras

```
void DiscoverPrintDevices(void *arg)
{
    while (true)
    {
        IPAddress subnetmask = WiFi.subnetMask();
        IPAddress ipaddress = WiFi.localIP();
        unsigned long first_ip = ntohl(ipaddress & subnetmask);
        unsigned long last_ip = ntohl(ipaddress | ~(subnetmask));
        for (unsigned long ip = first_ip; ip < last_ip; ++ip)
        {
            unsigned long theip = htonl(ip);
            IPAddress deviceIp = inet_addr(inet_ntoa(theip));
            if (Ping.ping(inet_ntoa(theip), 1))
            {
                String sysName = snmp.getString(".1.3.6.1.2.1.1.5.0", 10, deviceIp);
                if (sysName != "")
                {
                    AddPrinterOnList(deviceIp);
                }
            }
        }
    }
}
```

Fonte: Elaboração Própria (2023)

A fim de calcular todos os IPs da rede na qual o dispositivo está conectado, precisamos primeiro saber a máscara de sub rede e o IP atual do positivo, que são armazenados na variável “*subnetmask*” e “*ipaddress*”, respectivamente.

A máscara de sub rede serve para determinar qual faixa de bits representa o host (roteador, nesse caso) e qual representa os dispositivos da rede (FALL *et al*, p39, 2011). Essa determinação se dá a quais bits da máscara são 1 e quais bits da máscara são 0, sendo 1 a faixa reservada ao host, e 0 a faixa reservada aos dispositivos da rede. Em uma rede na qual a máscara é 11111111.11111111.11111111.00000000 (255.255.255.0 em binário), a faixa reservada para os dispositivos é de 0 até 255.

O primeiro IP a ser designado na rede atual é calculado através de uma operação *bitwise* entre o endereço IP e a máscara de sub rede.

Uma operação de *bitwise* serve para comparar a nível binário dois valores. Nesse caso o algoritmo usa a operação *bitwise* AND, representado como “&” na linguagem C++, que irá comparar os dois valores bit a bit, ou seja, o valor de saída será 1 se ambos os bits de cada lado sejam 1, por exemplo, se o endereço IP especificado fosse 192.168.0.155 e a máscara de sub rede fosse 255.255.255.0, a operação seria consumada da seguinte maneira, em verde os valores no qual a operação AND foi positiva:

TABELA 2 – OPERAÇÃO *BITWISE AND*

192.168.0.155	11000000.10101000.00000000.10011011
255.255.255.0	11111111.11111111.11111111.00000000
192.168.0.0	11000000.10101000.00000000.00000000

Logo, o primeiro IP da rede seria “192.168.0.0”. Após o cálculo do primeiro IP, é transformado esse endereço IP em sua representação em decimal utilizando a função “ntohl”, a fim de ser possível percorrer todas as possibilidades de IPs da rede.

Em seguida, é necessário realizar um cálculo similar para obter o último IP possível da rede, feito através da operação *bitwise OR*, representada como “|” na linguagem C++, e a operação *bitwise NOT*, representada como “~” na linguagem C++. A operação OR, irá validar se existe um bit positivo em qualquer lado da comparação e colocá-lo na saída, já o operador NOT irá inverter os bits do valor desejado, logo 11111111.11111111.11111111.00000000 se torna 00000000.00000000.00000000.11111111, resultando no valor 0.0.0.255. Se faz necessário inverter os bits da máscara de sub rede para que a máscara exiba apenas a faixa reservada aos dispositivos da rede, e não a faixa reservada ao gerente da rede. O resultado do cálculo será exibido abaixo, em verde o resultado positivo da operação:

TABELA 3 - OPERAÇÃO *BITWISE OR*

192.168.0.155	11000000.10101000.00000000.10011011
0.0.0.255	00000000.00000000.00000000.11111111
192.168.0.255	11000000.10101000.00000000.11111111

Logo, o último IP da rede seria “192.168.0.255”. Seguindo a lógica, o IP é transformado em sua representação decimal através do método “ntohl”, a fim de definir o fim dos IPs possíveis na rede.

O próximo passo da função é percorrer do primeiro IP ao último IP. Enquanto percorre, a função converte o valor em decimal no formato “char” com a função “inet_ntoa” e logo após converte esse valor em char para um endereço IP, com a função “inet_addr”.

Após isso é necessário checar se existe uma máquina na rede com esse IP designado, nesse passo é utilizado a biblioteca “Ping” para “pingar” o IP esperando uma resposta, caso o IP responda o *ping*, é feito uma tentativa de buscar um OID padrão via SNMP para validar se o dispositivo em questão possui o protocolo SNMP ativo. Caso o tenha, é adicionado em uma lista para que seja capturado os valores essenciais para a gestão.

6.3 Obtenção dos dados e envio

Após o descobrimento dos dispositivos SNMP na rede, é necessário a obtenção dos valores pertinentes ao gerenciamento das impressoras por parte dos provedores. Esses são: Número de série, Modelo, Fabricante e Contador total. Para fim de testes, foram utilizadas impressoras que estão disponíveis na internet, utilizando o mecanismo de busca Censys para encontrar impressoras abertas. Censys é uma empresa de cibersegurança que foca em mapear dispositivos que estão expostos na internet, com o intuito de trazer visibilidade para as empresas em relação aos seus servidores que estão abertos para internet (Censys, 2023). Vale destacar que os dispositivos utilizados durante o teste não sofreram nenhuma alteração, apenas foram coletados os valores necessários, nenhuma informação que não foi descrita no presente artigo foi coletada.

Através de tentativa e erro, o OIDs mais comuns que respondem os valores necessários entre as impressoras são os seguintes:

TABELA 4 - OIDs

OID	Resultado
.1.3.6.1.2.1.25.3.2.1.3.1	Modelo e fabricante da impressora
.1.3.6.1.2.1.43.10.2.1.4.1.1	Contador total de impressões
.1.3.6.1.2.1.43.5.1.1.17.1	Número de série da impressora

A rotina do dispositivo se dará da seguinte maneira:

- 1° Mapear a rede em busca de impressoras;
- 2° Identificar se a impressora possui o protocolo SNMP ativo;
- 3° Requisitar os valores correspondentes dos OIDs acima;
- 4° Enviar os valores para um servidor externo.

Figura 2 – Obtenção de valores

```
if (!printerList->device)
{
    Serial.println("Nenhum dispositivo encontrado ainda, aguardando 1 segundo para checar novamente");
    delay(1000);
    return;
}
DevicesList *printerListCursor = printerList;
Serial.print("Dispositivo ");
Serial.print(printerListCursor->device);
Serial.println(" encontrado");
String printerModel = snmp.getString(".1.3.6.1.2.1.25.3.2.1.3.1", 10, printerListCursor->device);
int totalCounter = snmp.getCounter32(".1.3.6.1.2.1.43.10.2.1.4.1.1", 10, printerListCursor->device);
String serialNumber = snmp.getString(".1.3.6.1.2.1.43.5.1.1.17.1", 10, printerListCursor->device);
String requestBody;
requestBody.concat(printerModelOid);
requestBody.concat("=");
requestBody.concat(printerModel);
requestBody.concat("&");
requestBody.concat(totalCounterOid);
requestBody.concat("=");
requestBody.concat(totalCounter);
requestBody.concat("&ip=");
requestBody.concat(printerListCursor->device.toString());
requestBody.concat("&serialNumber=");
requestBody.concat(serialNumber);
sendToServer(requestBody);
```

Fonte: Elaboração Própria (2023)

Assim que a rotina identifica que foi encontrado uma impressora com o protocolo SNMP ativo, é feita a requisição via SNMP, com os OIDs descritos acima e os envia novamente. A rotina continuará fazendo esses mesmos passos em loop, para que se tenha no servidor sempre a versão mais recente dos dados da impressora.

6.4 Prova de conceito

Durante a prova de conceito, foram utilizados 8 modelos de impressora diferentes, a fim de validar se o algoritmo é capaz de coletar as informações necessárias. Foi utilizado como base de comprovação dos dados a interface Web da impressora.

Para o sucesso da prova de conceito, foram escolhidos modelos de impressora que possuem o número de série e o contador geral de impressão disponíveis em sua interface, dessa forma é possível cruzar o valor obtido pelo algoritmo e o valor reportado pela impressora.

TABELA 5 - Visão geral das impressoras monitoradas pelo dispositivo

Fabricante	Modelo	Contador total	N° de série	Última leitura
RICOH	Aficio MP 301	49984	W915P702735	2023-06-21 23:07:54.150030
Brother	DCP-1610NW series	21827	U63980H1N887083	2023-06-21 23:07:36.561226
Brother	HL-L2390DW	627	U64967B2N362731	2023-06-21 23:06:28.970144
RICOH	IM C300	19708	3922P950766	2023-06-21 23:07:39.404113
HP	LaserJet MFP M234dw	121	VNG3M27998	2023-06-21 23:07:51.486516
Brother	MFC-9340CDW	116600	U63481J4J479892	2023-06-21 23:07:31.953491
KONICA	MINOLTA bizhub C458	430212	A79M011036845	2023-06-21 23:07:35.969092
XEROX	VersaLink B405 DN Multifunction Printer	120413	9HB920639	2023-06-21 23:07:33.214833

TABELA 6 - Informações obtidas na página web de cada impressora

Fabricante	Modelo	Contador total	N° de série
RICOH	Aficio MP 301	49984	W915P702735
Brother	DCP-1610NW series	21827	U63980H1N887083
Brother	HL-L2390DW	627	U64967B2N362731
RICOH	IM C300	19708	3922P950766
HP	LaserJet MFP M234dw	121	VNG3M27998
Brother	MFC-9340CDW	116600	U63481J4J479892
KONICA	MINOLTA bizhub C458	430212	A79M011036845
XEROX	VersaLink B405 DN Multifunction Printer	120413	9HB920639

7. Trabalhos Futuros

No quesito evolução, trabalhos feitos a partir deste podem incluir as seguintes melhorias:

1. Inclusão de um módulo extensor de cartão de memória. É interessante ao provedor ter os dados de histórico de contador mesmo em uma instabilidade na conexão com a internet no qual o dispositivo se encontra;
2. Inclusão de uma arquitetura modular permitindo que um servidor externo instrua o dispositivo em relação a qual OID ser requisitado a impressora, dependendo de cada modelo;
3. Inclusão de uma rotina de atualização do código contido no dispositivo, a fim de poder dar manutenção de forma remota no mesmo;
4. Inclusão de um algoritmo para incluir configurações de ambiente no mesmo. No estado atual, as informações de login na rede WiFi estão escritas de forma permanente no mesmo, seria uma evolução se utilizar do módulo Bluetooth existente no ESP32.

8. Considerações finais

O objetivo do presente trabalho foi validar se é possível obter as informações pertinentes aos provedores de outsourcing de impressão através da coleta automática utilizando o dispositivo ESP32. A tarefa de coletar os dados das impressoras alugadas é crucial para o funcionamento da empresa, uma vez que é necessário fazer a contabilidade para fazer a cobrança dos clientes.

Através da automatização da coleta de dados das impressoras, é possível para a empresa de outsourcing de impressão reduzir o custo da sua operação, uma vez que não será necessário alocar recursos para uma visita física, podendo assim diminuir o preço para o consumidor final, que é o cliente.

Através dessa pesquisa, é possível constatar:

1. O dispositivo é capaz de identificar de forma automática as impressoras em uma rede;
2. O dispositivo é capaz de coletar as informações pertinentes a operação da empresa;
3. O dispositivo é capaz de enviar esses dados para um servidor externo;
4. Com esse dispositivo, a empresa reduz o custo da operação, deixando de lado a necessidade de contratação para uma visita física ao cliente.

9. REFERENCIAS BIBLIOGRÁFICAS

ABATTI, Felipe. **Serviço de outsourcing de impressão e sua viabilidade: um estudo na UTFPR Câmpus Medianeira**. 2019. Dissertação de Mestrado. Universidade Tecnológica Federal do Paraná. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/4001>. Acesso em: 2 de abril de 2023.

CARPENTER, Brian E. **Open systems networking in practice**. CM-P00059871, 1987. Disponível em: <https://cds.cern.ch/record/181665/files/CM-P00059871.pdf>. Acesso em: 10 de abril de 2023.

CASE, Jeffrey D. et al. Rfc1157: Simple network management protocol (snmp). 1990. Disponível em: <https://dl.acm.org/doi/pdf/10.17487/RFC1157>. Acesso em: 26 de dezembro de 2022.

Censys, Censys, 2023. Disponível em: <https://about.censys.io/>. Acesso em: 21 de junho de 2023

Douglas Bruey, Rane . SNMP: Simple? Network Management Protocol. Rane Commercial, 2005. Disponível em: <https://www.ranecommercial.com/legacy/note161.html>. Acesso em: 27 de dezembro de 2022.

ESP32 Series Datasheet, Espressif, 2023. Disponível em:

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 23 de abril de 2023.

ESP32 Technical Reference Manual, Espressif, 2022. Disponível em:

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf. Acesso em: 23 de abril de 2023.

ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems, Espressif, 2023. Disponível em:

<https://www.espressif.com/en/products/socs/esp32>. Acesso em: 23 de abril de 2023.

Especificação do Protocolo. UFRGS, Porto Alegre, s.d. Disponível em:

http://penta.ufrgs.br/gr952/trab1/snmp_especificacao.html. Acesso em: 26 de dezembro de 2022.

FALL, Kevin R.; STEVENS, W. Richard. TCP/IP illustrated, volume 1: The protocols. addison-Wesley, 2011.

Get Started - ESP32 - — ESP-IDF Programming Guide v5.0.1 documentation, Espressif, 2023. Disponível em:

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html>. Acesso em: 23 de abril de 2023.

ISO/IEC 14882:2020(en), Programming languages — C++, ISO, 2020. Disponível em:

<https://www.iso.org/obp/ui/#iso:std:iso-iec:14882:ed-6:v1:en>. Acesso em 24 de abril de 2023.

MADEIRA, John Franklin Loiola. Gerência de redes sensores sem fio com SNMP: uma abordagem com proxy gateway. 2013. Disponível em: http://repositorio.sis.puc-campinas.edu.br/bitstream/handle/123456789/14996/ceatec_ppgrrt_me_John_FLM.pdf?sequence=1&isAllowed=y.

Acesso em: 10 de abril de 2023.

NAPOLEÃO FILHO, Jair et al. Eficiência na Administração Pública: o modelo do outsourcing de impressão na Universidade Federal de Santa Catarina–UFSC. **Universitas: Gestão e TI**, v. 4, n. 2, 2014. Disponível em:

<https://repositorio.ufsc.br/handle/123456789/114795>. Acesso em: 2 de abril de 2023.

ONE, Notation. Information technology–ASN. 1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). **Interfaces**, v. 10, n. 20-X, p. 49, 1998. Disponível em http://penta3.ufrgs.br/normasITU/ORIGINAL/PRE_PUBL/X/X0690E.PDF. Acesso em: 2 de abril de 2023.

Protocol data units (PDUs) - IBM Documentation. IBM, Armonk, 8 de março de 2021. Disponível em:

<https://www.ibm.com/docs/en/ztpf/1.1.0.15?topic=support-protocol-data-units-pdus>. Acesso em: 26 de dezembro de 2022.

RICARTE, Ivan L. M. Origens de C++. 2001. Disponível em:

https://www.dca.fee.unicamp.br/cursos/POO_CPP/node18.html. Acesso em: 24 de abril de 2023.

S/T, UFRJ, 2023. Disponível em: https://www.gta.ufrj.br/grad/04_1/snmp/mib.htm. Acesso em: 12 de abril de 2023.

SOMJAI, Sudawan. Advantages and disadvantages of outsourcing. **The Business and Management Review**, v. 9, n. 1, p. 157-160, 2017. Disponível em:

https://cberuk.com/cdn/conference_proceedings/conference_21121.pdf. Acesso em: 2 de abril de 2023.

VENÂNCIO NETO, Augusto José et al. Implementação de um discriminador de repasse de eventos para o ambiente SNMP. 2001. Disponível em:

<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/82124/188654.pdf>. Acesso em 24 de junho de 2023.