

# Complemento OPC-UA: Desenvolvimento de um Cliente OPC- UA em Visual Basic

Documento realizado por:

Daniel Jesus Camarneiro (daniel.camarneiro@ua.pt)

## Índice

|   |    |
|---|----|
| Introdução.....                                   | 1  |
| Trabalho de aula .....                            | 2  |
| Interações entre equipamentos em OPC-UA.....      | 4  |
| Desenvolvimentos da aplicação.....                | 6  |
| Criação e configuração do template.....           | 6  |
| Criação da interface gráfica (Front end) .....    | 8  |
| Desenvolvimento do código (Back end).....         | 9  |
| Configuração do IP do computador (Opcional) ..... | 10 |
| Referências.....                                  | 15 |



## Índice de figuras

|   |    |
|---|----|
| Fig. 1: Exemplo de arquitetura OPC-UA.....  | 1  |
| Fig. 2: Arquitetura para trabalho de aula.....  | 2  |
| Fig. 3: Exemplo de interface gráfica do programa .....                                      | 2  |
| Fig. 4: Gráfico UML da sequência de mensagens no protocolo OPC-UA na leitura de variáveis.. | 4  |
| Fig. 5: Gráfico UML da sequência de mensagens no protocolo OPC-UA na escrita de variáveis.. | 4  |
| Fig. 6: Página de rosto do Visual Studio.....   | 6  |
| Fig. 7: Seleção do template .....   | 6  |
| Fig. 8: Finalização da criação do projeto.....  | 7  |
| Fig. 9: Gestão das bibliotecas do projeto.....  | 7  |
| Fig. 10: Instalação da biblioteca OPC-UA.....   | 8  |
| Fig. 11: Exemplo de interface gráfica para o programa.....                                  | 8  |
| Fig. 12: Definições do Sistema Operativo Windows.....                                       | 11 |
| Fig. 13: Definições das placas de rede.....   | 11 |
| Fig. 14: seleção do adaptador ethernet.....   | 12 |
| Fig. 15: Menu das propriedades do adaptador ethernet .....                                  | 12 |
| Fig. 16: Modificação das definições IPv4 .....  | 13 |
| Fig. 17: Designação de um IP fixo.....  | 13 |

## Introdução

Neste documento será desenvolvida uma introdução ao protocolo de comunicação OPC-UA e a elaboração de um programa em Visual Basic para comunicar com um módulo OPC-UA. O módulo utilizado neste exemplo será o equipamento o IBH-Link UA [1], onde encontra-se em execução o Servidor OPC-UA.

A Fig. 1 representa uma arquitetura exemplo. No computador 1 existe o programa UaExpert [2], um cliente OPC UA ready-to-use que servirá de referência para o programa a desenvolver em Visual Basic do computador 2, permitindo assim explorar as propriedades do Servidor OPC-UA. Por fim, no IBH Link UA e no PLC encontram-se os programas desenvolvidos no TIA Portal, tanto para a configuração do Servidor OPC-UA como o programa do PLC.

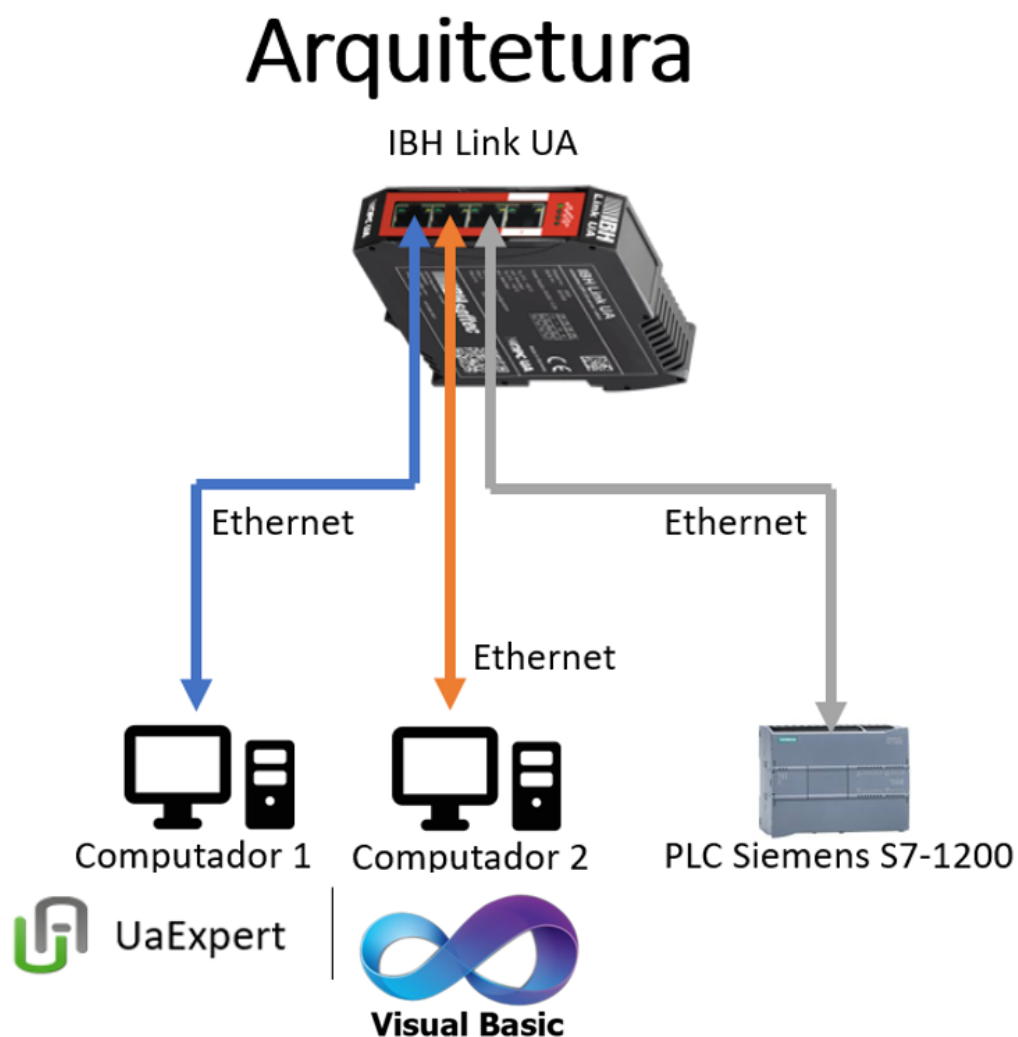


Fig. 1: Exemplo de arquitetura OPC-UA

Para mais informação é só aceder ao repositório [3].

## Trabalho de aula

Nesta aula pretende-se desenvolver um programa em Visual Basic, neste caso um Cliente OPC-UA, para interagir com o Servidor OPC-UA do módulo IBH Link UA que permita realizar a leitura de variáveis e a escrita de nova informação na mesma. Na Fig. 2 é exemplificado o esquema da arquitetura, onde cada aluno deve desenvolver o programa em Visual Basic, enquanto num outro computador será executado o UaExpert para monitorizar o funcionamento do Servidor OPC-UA.

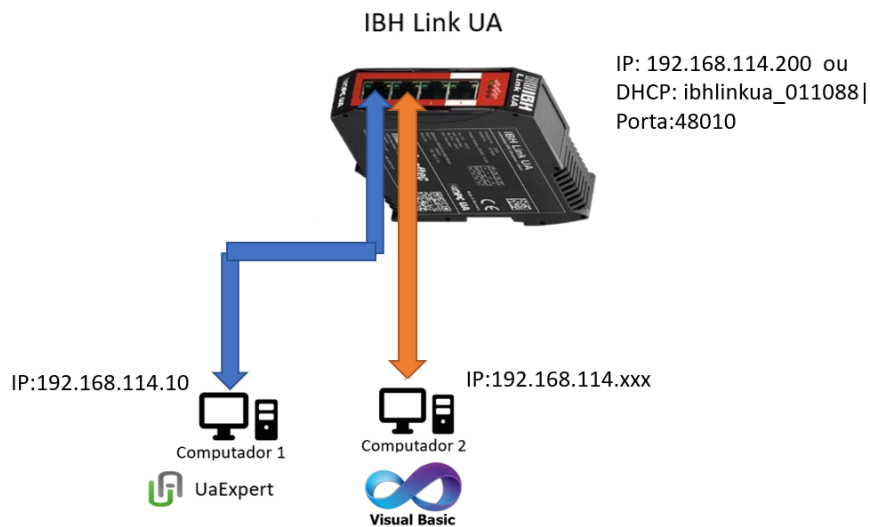


Fig. 2: Arquitetura para trabalho de aula

O programa a desenvolver deverá possuir uma interface semelhante à seguinte.

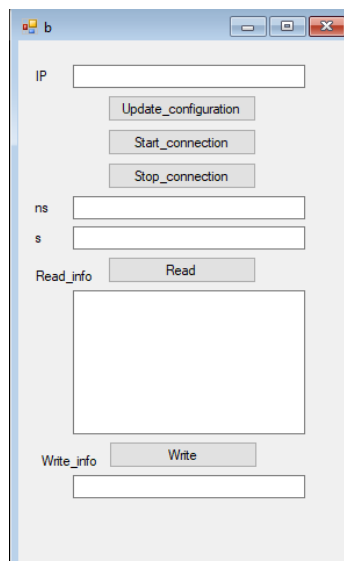


Fig. 3: Exemplo de interface gráfica do programa

Assim, o programa deve permitir as seguintes funcionalidades:

- Configurar o endereço, ou endpoint, do Servidor OPC-UA (botão “Update\_configuration” e caixa de texto “IP”).
- Dar início à ligação OPC-UA com o Servidor OPC-UA (botão “Start\_connection”).
- Interromper a ligação OPC-UA com o Servidor OPC-UA (botão “Stop\_connection”).
- Definir o endereço de uma variável (caixas de texto “ns” e “s”).
- Realizar a leitura de uma variável e visualizar a sua informação (botão “Read” e caixa de texto “Read\_info”).
- Realizar a escrita de informação para uma variável (botão “Write” e caixa de texto “Write\_info”).

## Interações entre equipamentos em OPC-UA

Antes de prosseguir com o início do tutorial, na Fig. 4 e Fig. 5 encontram-se exemplificados os processos para troca de informação relativamente ao pedido de leitura e de escrita de variáveis. Para realizar a leitura de variáveis [Fig. 4], o cliente OPC UA começa por realizar um pedido de leitura, indicando as referências das variáveis a lei (neste caso com um 'n' e um 'ns', mas poderá utilizar outra nomenclatura). Quando este pedido chega ao IBH Link UA (Servido OPC UA) é convertido de TCP/IP para S7, o protocolo de comunicação utilizado pelo PLC da Siemens. Assim que o PLC recebe uma nova mensagem, processa-a e envia a devida resposta (neste caso contendo informação relativamente às variáveis pedidas) para o IBH Link UA, que posteriormente as reencaminha para o Cliente OPC UA. Comparativamente com as mensagens para escrita [Fig. 5], a grande diferença encontra-se na interpretação por parte do PLC, que em vez de realizar o levantamento das variáveis, modifica-as de acordo com os parâmetros da mensagem recebida.

### Leitura de variáveis

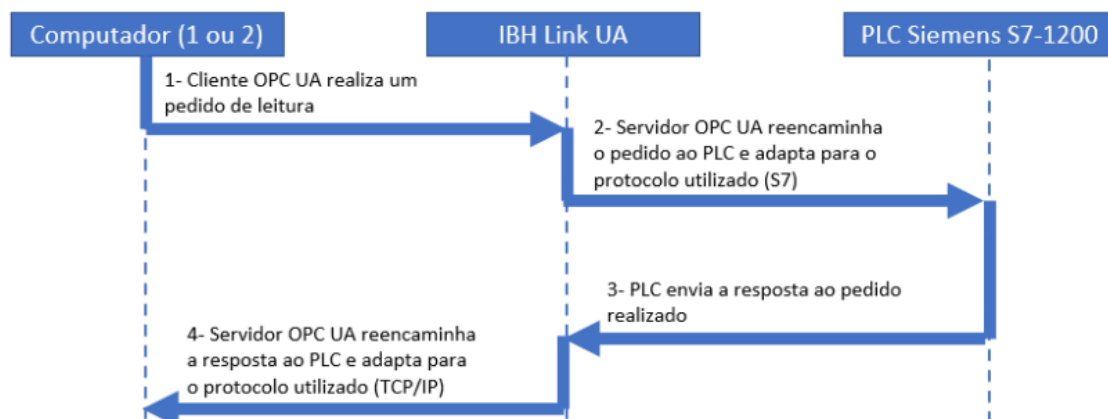


Fig. 4: Gráfico UML da sequência de mensagens no protocolo OPC-UA na leitura de variáveis

### Escrita de variáveis

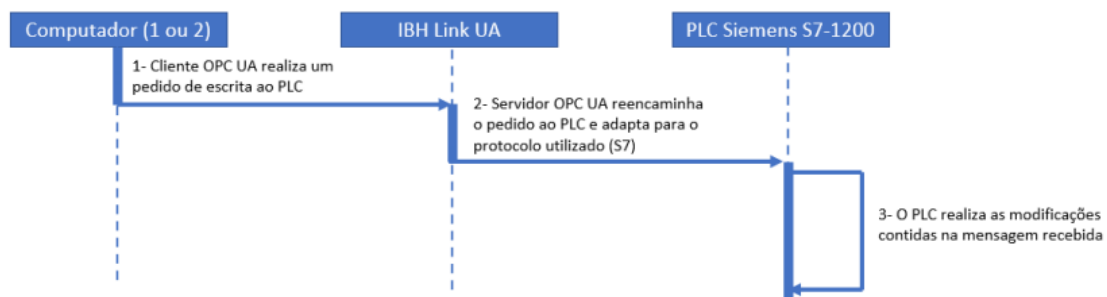


Fig. 5: Gráfico UML da sequência de mensagens no protocolo OPC-UA na escrita de variáveis





# Desenvolvimentos da aplicação

## Criação e configuração do template

A primeira etapa na elaboração do nosso Cliente OPC-UA em visual basic consiste na criação do novo projeto e a instalação da biblioteca (add-on) para o protocolo OPC-UA. As etapas são as seguintes:

1. Na página de abertura do Visual Studio, criar um novo projeto.

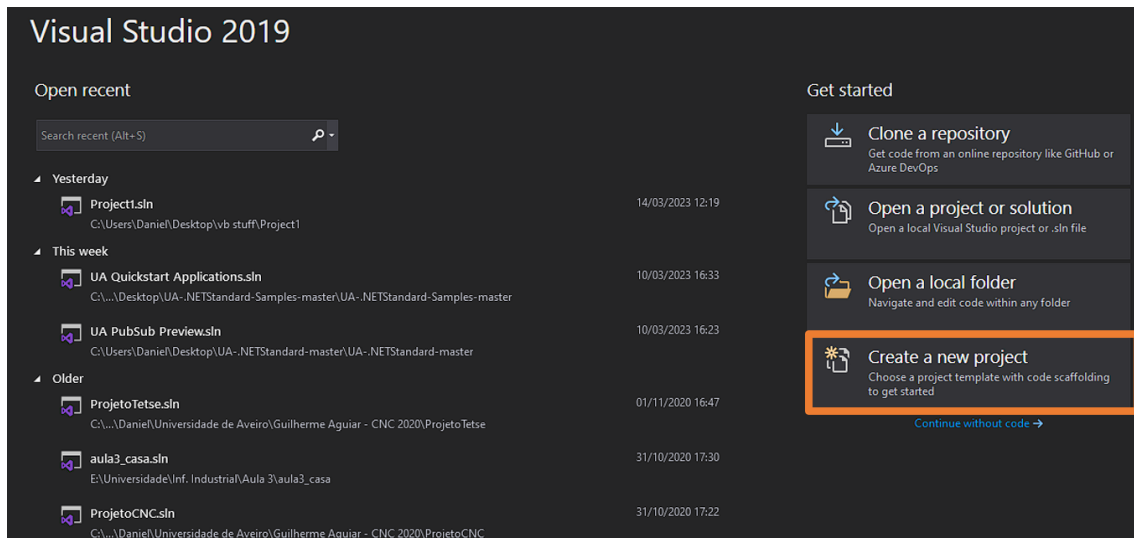


Fig. 6: Página de rosto do Visual Studio

2. Selecionar o template “Empty project” (.NET Framework) em Visual Basic (e não em C#).

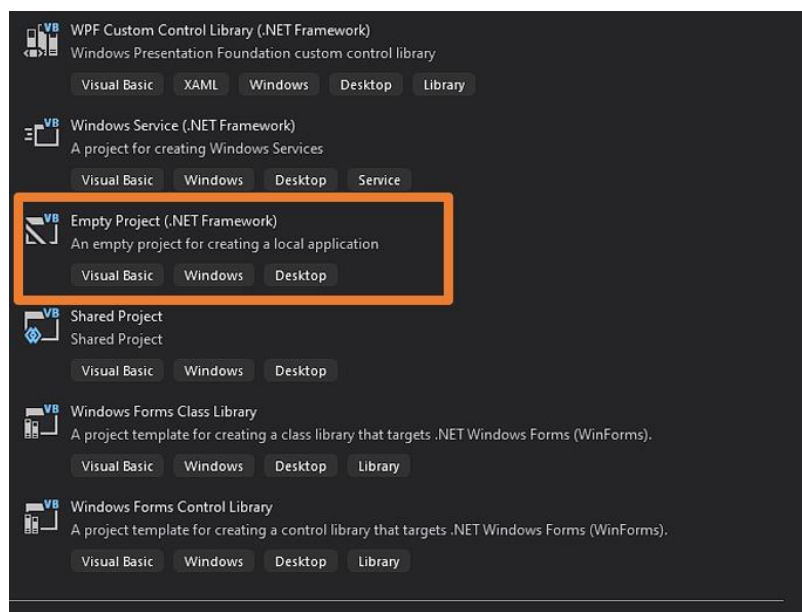


Fig. 7: Seleção do template

3. Atribuir um nome ao projeto e finalizar a sua criação.

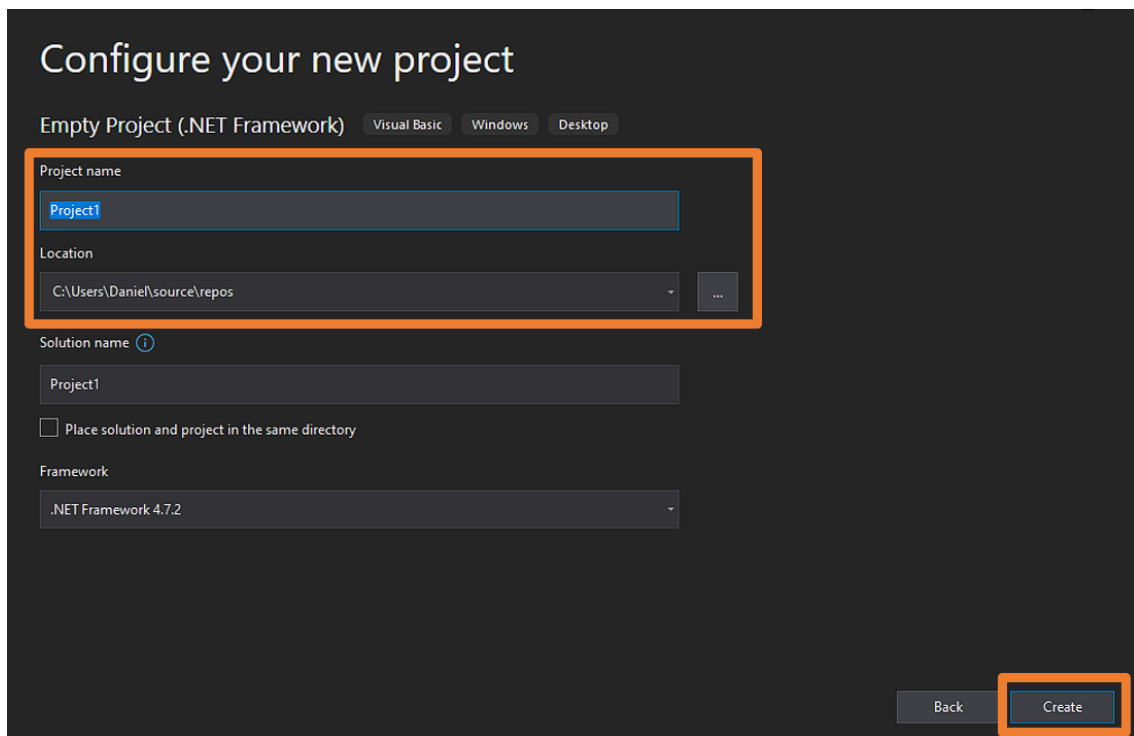


Fig. 8: Finalização da criação do projeto

4. Elaborada a criação do projeto, agora será adicionada a biblioteca OPC-UA. Primeiro selecionar o separador “Project” e depois “Manage NuGet Packages...”

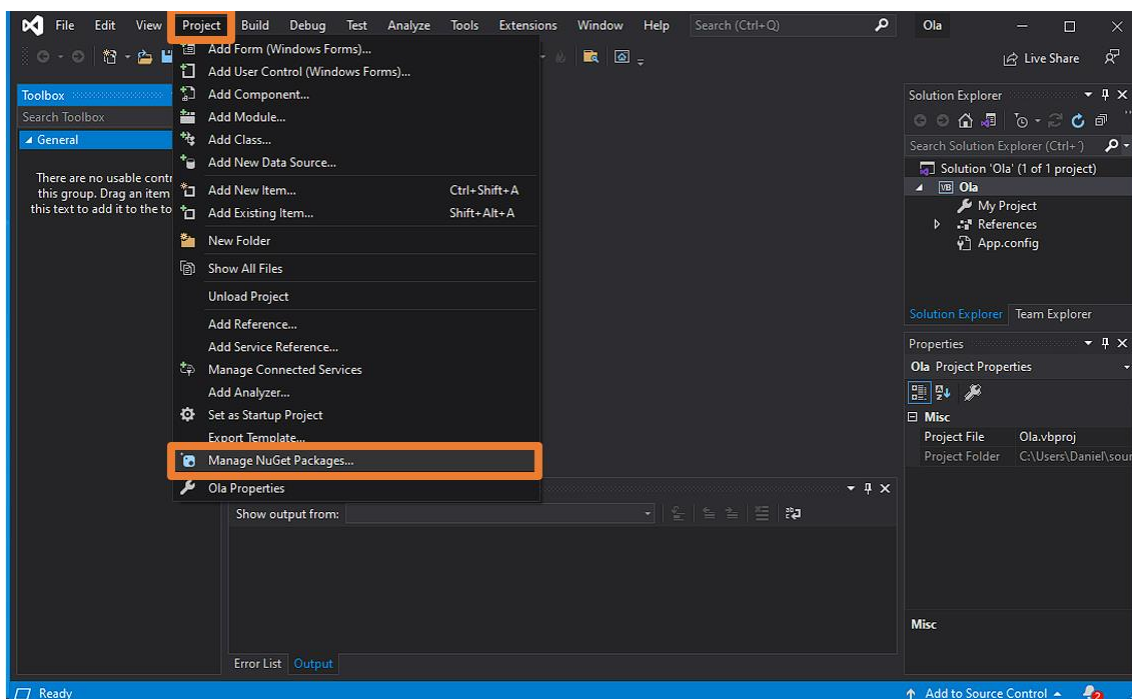


Fig. 9: Gestão das bibliotecas do projeto

5. Em “Browse” procurar por “Opc.UaFx.Client” [4] e instalar.

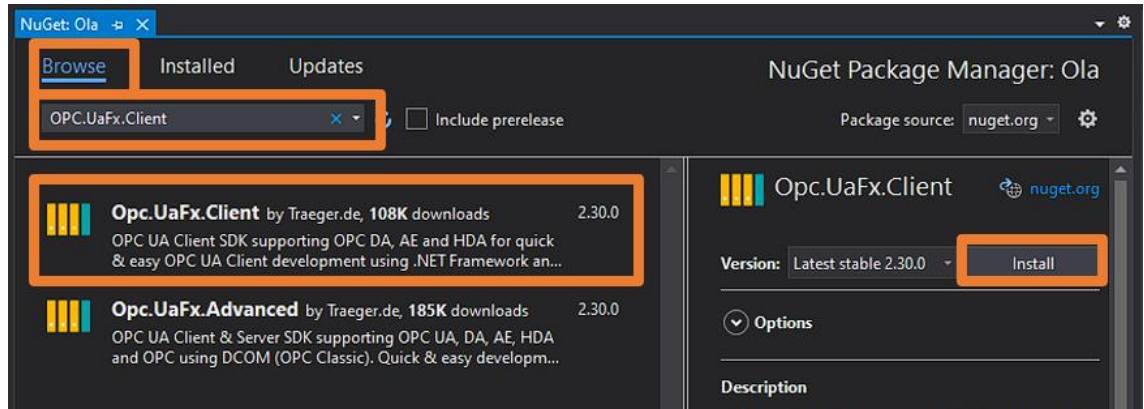


Fig. 10: Instalação da biblioteca OPC-UA

### Criação da interface gráfica (Front end)

Terminada a etapa de configuração do nosso projeto em visual basic, agora segue-se a criação da interface gráfica do nosso programa. Para este exemplo utilizou-se a seguinte interface

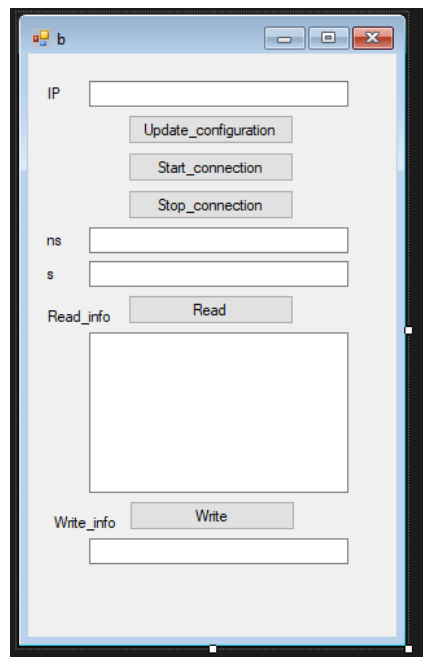


Fig. 11: Exemplo de interface gráfica para o programa

com 5 caixas de texto,

- Uma para configurar o endereço do Servidor OPC-UA (“IP”),
- Uma para indicar o indicador *ns* (“ns”),
- Uma para indicar o indicador *s* (“s”),
- Uma para visualizar a informação de leitura (“Read\_info”),
- Uma para inserir a informação a escrever (“Write\_info”),

e 5 botões,

- Um para submeter o endereço do Servidor OPC-UA ("Update\_configuration"),
- Um para iniciar a conexão ("Start\_connection"),
- Um para encerrar a conexão ("Stop\_connection"),
- Um para realizar a leitura de uma variável ("Read"),
- Um para realizar a escrita de uma variável ("Write").

## Desenvolvimento do código (Back end)

No excerto de código que se segue encontra-se as funcionalidades para a execução do programa em Visual Basic.

```
'Imports the OPC-UA library
Imports Opc.UaFx.Client
Public Class Form1
    'Global variables creation
    Dim IP 'OPC-UA's server endpoint
    Dim client 'Client instance

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'Start up configurations (not needed)
        IP_tb.Text = "ibhlinkua_011088:48010"
        ns_tb.Text = "7"
        s_tb.Text = ".Publish.teste.struct1.u2"
        write_tb.Text = "True"

        'Button Configuration
        Config_btn.Enabled = True
        Start_btn.Enabled = False
        Stop_btn.Enabled = False
        Read_btn.Enabled = False
        Write_btn.Enabled = False
    End Sub

    Private Sub Config_btn_Click(sender As Object, e As EventArgs) Handles Config_btn.Click
        'Updates OPC-UA's server endpoint
        IP = "opc.tcp://" & IP_tb.Text 'OPC-UA's server endpoint

        'Button Configuration
        Config_btn.Enabled = True
        Start_btn.Enabled = True
        Stop_btn.Enabled = False
        Read_btn.Enabled = False
        Write_btn.Enabled = False
    End Sub

    Private Sub Start_btn_Click(sender As Object, e As EventArgs) Handles Start_btn.Click
        'Starts the connection with the OPC-UA Server
        client = New OpcClient(Convert.ToString(IP)) 'Converts the Object from a textbox to a String and
        creates a new OPC-UA Client
        client.Connect() 'Tries to establish a connection with the OPC-UA Server
```

```

'Button Configuration
Config_btn.Enabled = False
Start_btn.Enabled = False
Stop_btn.Enabled = True
Read_btn.Enabled = True
Write_btn.Enabled = True

End Sub

Private Sub Stop_btn_Click(sender As Object, e As EventArgs) Handles Stop_btn.Click
'Ends the connection with the OPC-AU Server
client.Disconnect() 'Finishes to stablish a connection with the OPC-UA Server

'Button Configuration
Config_btn.Enabled = True
Start_btn.Enabled = True
Stop_btn.Enabled = False
Read_btn.Enabled = False
Write_btn.Enabled = False
End Sub

Private Sub Read_btn_Click(sender As Object, e As EventArgs) Handles Read_btn.Click
'Request to read a variable from the OPC-UA Server
Dim id = "ns=" & ns_tb.Text & ";s=" & s_tb.Text 'Variable endpoint identification
Dim OPCValue = client.ReadNode(Convert.ToString(id)) 'Converts the Object from a textbox to a
String and requests the variables info
read_tb.Text = Convert.ToString(OPCValue) 'Converts the Object from the request to a String
and updates the read textbox
End Sub

Private Sub Write_btn_Click(sender As Object, e As EventArgs) Handles Write_btn.Click
'Submits to write a variable from the OPC-UA Server
Dim id = "ns=" & ns_tb.Text & ";s=" & s_tb.Text 'Variable endpoint identification
Dim OPCValue = client.WriteNode(Convert.ToString(id), Convert.ToString(write_tb.Text))
'Converts the Object from a textbox to a String and requests to write a varibale from the OPC-UA
Server
End Sub

End Class

```

Como a premissa deste exemplo é apenas realizar uma introdução ao protocolo OPC-UA, muitas etapas para a consolidação e diagnóstico do código não foram realizadas para a simplificação do mesmo. Assim, muitos erros que possam vir a acontecer forçam o encerramento do programa (p.ex. caso o programa não possa estabelecer uma ligação com o Servidor OPC-UA, este é abortado passado um pequeno intervalo de tempo).

### Configuração do IP do computador (Opcional)

Uma vez que para estabelecer uma ligação TCP/IP com o módulo OPC-UA é necessário utilizar a placa ethernet do computador (caso o módulo não esteja ligado por ethernet ao mesmo router Wi-Fi, por isso a mesma WLAN que a placa wireless do computador) e não se queira utilizar o endpoint por DHCP (utilizado no exemplo anterior), é necessário fixar o IP desta placa para que o computador consiga comunicar com o módulo a partir da mesma subnet. Na

sequência que se segue encontram-se as etapas para atribuir um IP fixo no sistema operativo Windows (10):

1. Selecionar o menu “Start” e abrir as definições.

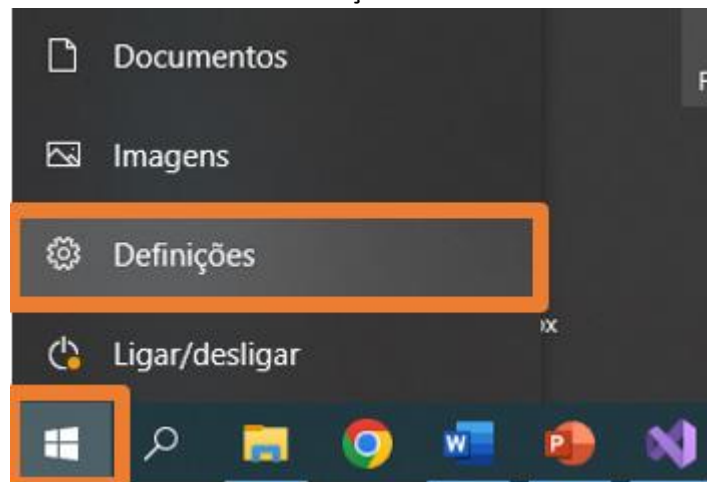


Fig. 12: Definições do Sistema Operativo Windows

2. Selecionar o menu de redes.

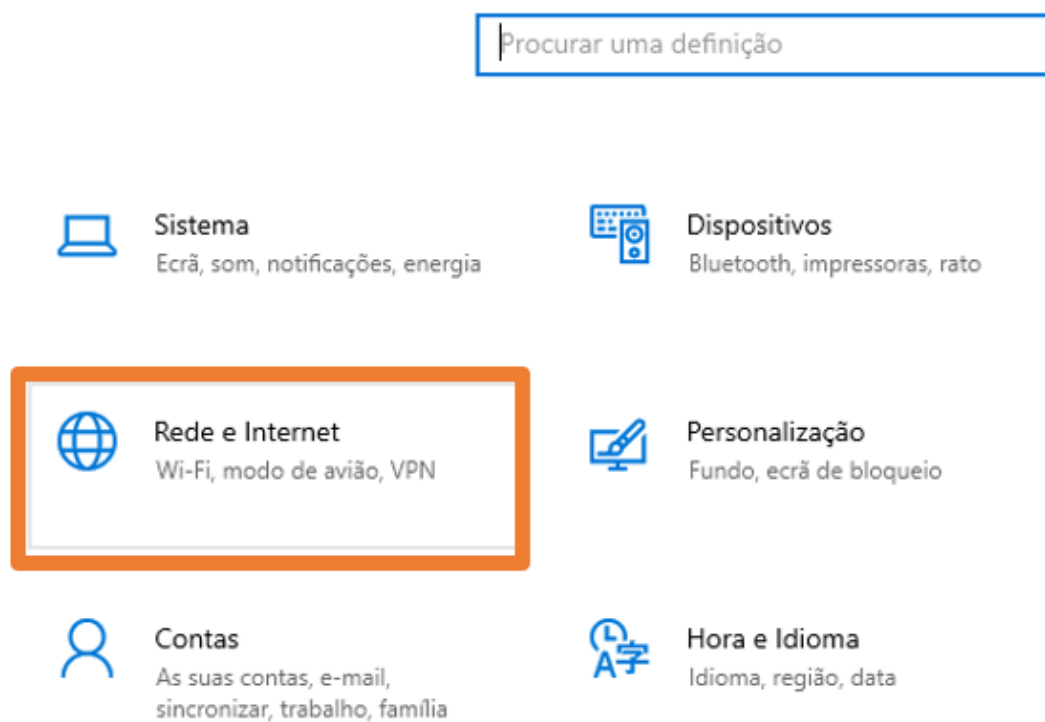


Fig. 13: Definições das placas de rede

3. Selecionar “ethernet” e “Alterar opções do adaptador”.

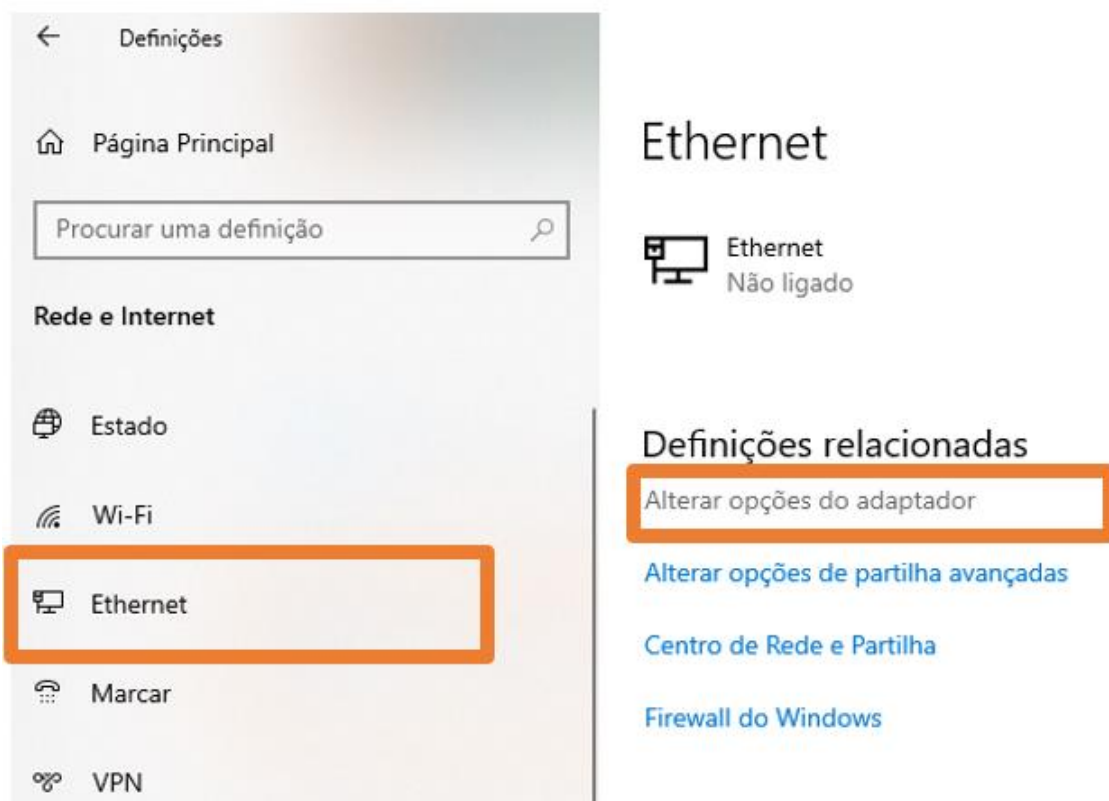


Fig. 14: seleção do adaptador ethernet

4. Com o botão direito do rato sobre a interface “ethernet”, selecionar “Propriedades”.

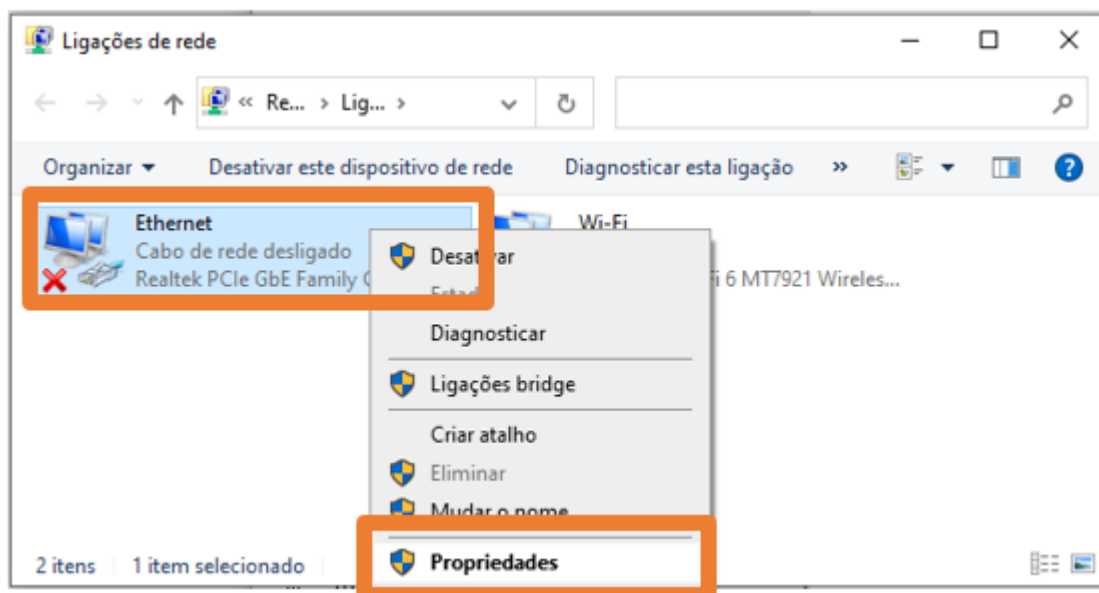


Fig. 15: Menu das propriedades do adaptador ethernet



5. Na checkbox “Protocolo IP Versão 4 (TCP/IPv4)” selecionar “Propriedades”.

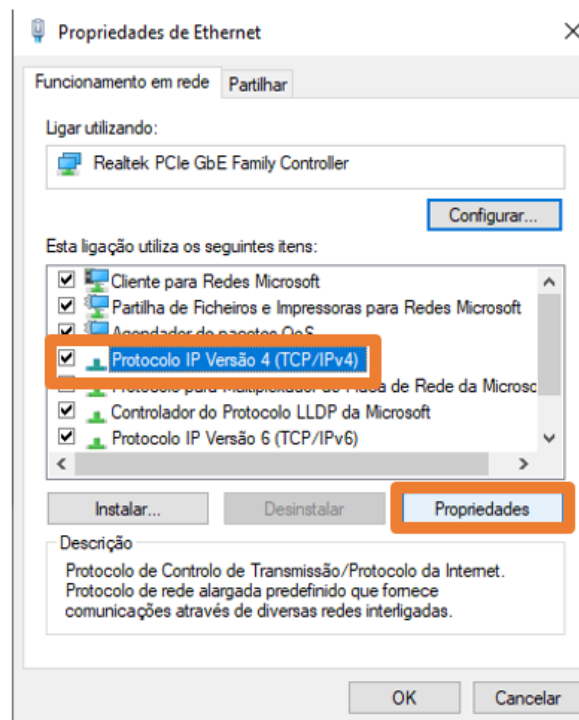


Fig. 16: Modificação das definições IPv4

6. Selecionar “Utilizar o seguinte endereço IP”, atribuir um endereço de IP na mesma subnet que o módulo OPC-UA Server (neste caso o 114) e selecionar a máscara com um clique esquerdo para o seu autopreenchimento. Caso o endereço da subnet do módulo OPC-UA Server seja desconhecido, é só aceder ao menu de configuração (exemplificado no documento original [3]).

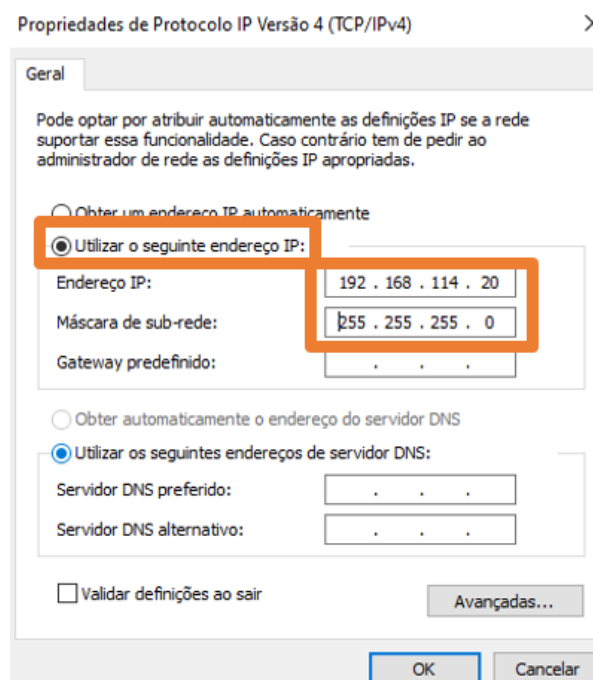


Fig. 17: Designação de um IP fixo

7. Agora é só fechar tudo até à janela da etapa 4. Para confirmar se a configuração foi bem executada, é só abrir uma linha de comandos, escrever “ipconfig” e procurar a interface ethernet. Caso seja necessário saber todos os equipamentos das várias interfaces é só escrever “arp -a” que aparece uma lista dos IPs de todos os equipamentos conectados na mesma subnet.

## Referências

- [1] “IBH Link UA - The compact OPC UA server for S5- and S7-PLC - IBHsofttec GmbH.” [Online]. Available: <https://www.ibhsofttec.com/IBH-Link-UA-Eng>. [Accessed: Apr. 29, 2022]
- [2] “UaExpert ‘UA Reference Client’ - Unified Automation.” [Online]. Available: <https://www.unified-automation.com/products/development-tools/uaexpert.html>. [Accessed: Apr. 29, 2022]
- [3] “danycamarneiro/comunicacao\_OPC-UA: Exemplos de interfaces OPC-UA para Python, Node-Red e Node.js.” [Online]. Available: [https://github.com/danycamarneiro/comunicacao\\_OPC-UA](https://github.com/danycamarneiro/comunicacao_OPC-UA). [Accessed: Mar. 20, 2023]
- [4] “Client Development Guide & Tutorial - TRAEGER Docs.” [Online]. Available: <https://docs.traeger.de/en/software/sdk/opc-ua/net/client.development.guide?lang=vb>. [Accessed: Mar. 20, 2023]