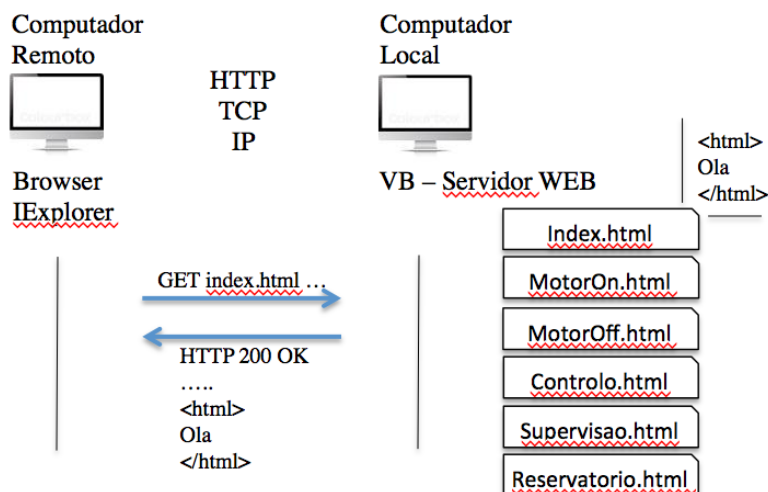




Este trabalho tem por objetivo facilitar a compreensão do protocolo HTTP, lecionado nas aulas teóricas. Para isso, pretende-se desenvolver em VBasic um servidor WEB/HTTP, elementar, que possa responder a pedidos de um browser WEB.

De uma forma simplista podemos afirmar que um servidor WEB não é mais que um programa que aceita ligações TCP/IP, habitualmente na porta 80, e que responde com o conteúdo dos documentos HTML pedidos pelo Browser. Os pedidos do Browser consistem em mensagens de texto, enviadas por TCP/IP para o servidor (ex. GET \index.html). A sintaxe dessas mensagens/pedidos/respostas é definida pelo protocolo HTTP.



Neste trabalho serão utilizados dois computadores (figura):

- Um dos computadores, o **computador local**, tem o **servidorWEB** desenvolvido em VBasic. O ServidorWEB aceita os pedidos de ligação TCP/IP e os pedidos de documentos HTML (GET ...) efetuados pelo browser WEB, de acordo com o protocolo HTTP.
 - Os **documentos HTML** residem no disco do computador local. Quando a aplicação ServidorWEB recebe pedidos de documentos HTML, lê o seu conteúdo e envia-o por TCPIP para o BrowserWEB remoto.
- O segundo computador, o **computador remoto**, possui um **browser WEB** que acede ao computador local através do **protocolo HTTP/TCP/IP**.

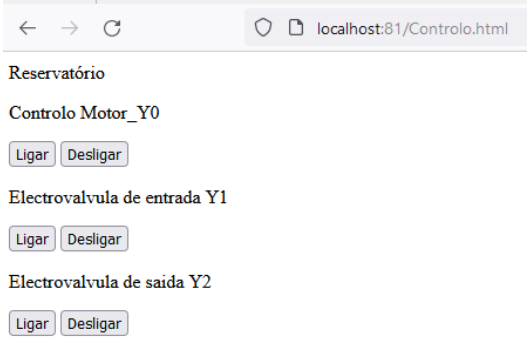
I. Desenvolva em VBasic, um servidor WEB/HTTP:

A **aplicação local**, desenvolvida em VBasic, deverá utilizar objetos do tipo: **TcpListener**, **TcpClient**, e **NetworkStream** para receber ligações TCP/IP e transferir dados; e um objeto do tipo **StreamReader** para ler os documentos disponíveis no disco local.

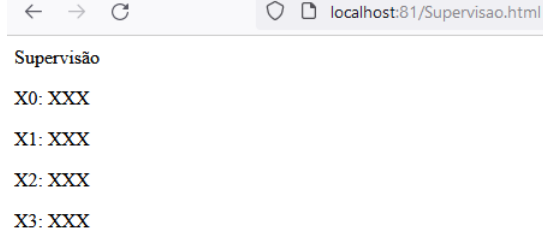
II. Código HTML

Edite os ficheiros HTML e guarde-os num diretório HTML na pasta do projeto VB.

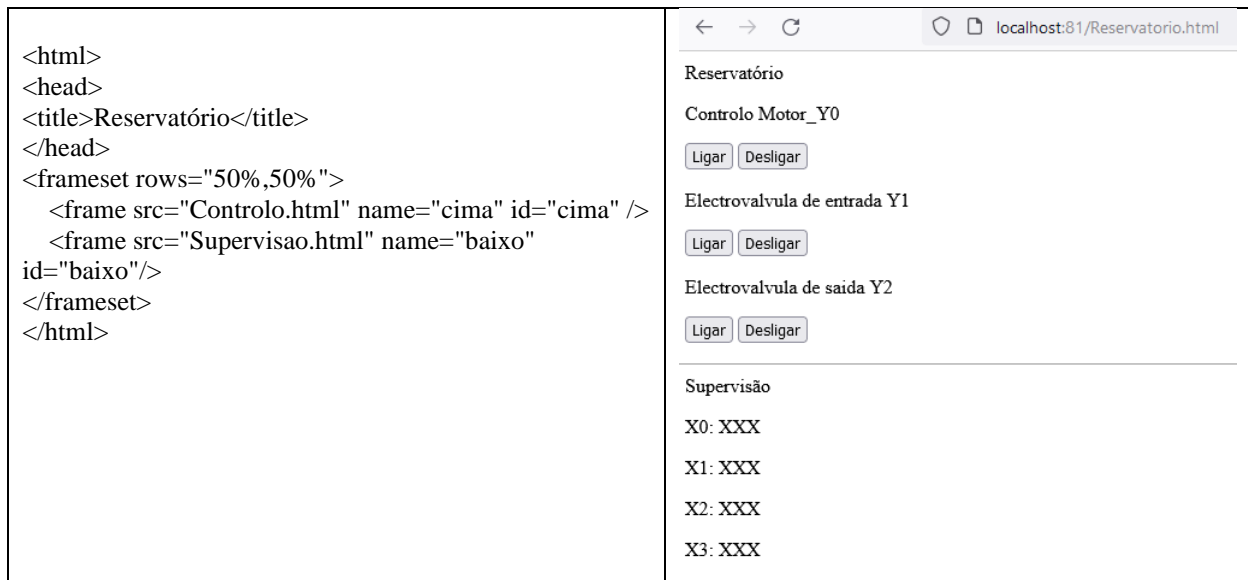
Página *Controlo.html*

<pre><html> <head> </head> <body> <p>Reservatório</p> <form> <p>Controlo Motor_Y0</p> <button type="submit" Name="Y0" Value="1">Ligar</button> <button type="submit" Name="Y0" Value="0">Desligar</button> <p>Electrovalvula de entrada Y1</p> <button type="submit" Name="Y1" Value="1">Ligar</button> <button type="submit" Name="Y1" Value="0">Desligar</button> <p>Electrovalvula de saida Y2</p> <button type="submit" Name="Y2" Value="1">Ligar</button> <button type="submit" Name="Y2" Value="0">Desligar</button> </form> </body> </html></pre>	
--	--

Página *Supervisao.html*

<pre><html> <head> </head> <body> <p>Supervisão</p> <p>X0: XXX</p> <p>X1: XXX</p> <p>X2: XXX</p> <p>X3: XXX</p> </body> </html></pre>	
---	--

Página *Reservatorio.html*



III. Aceda ao servidorWEB através do Browser

Depois de ter criado o programa exemplo, descrito no final deste documento (ServidorWEB), e os documentos HTML, poderá, a partir do computador remoto, usando um **browserWEB**, por exemplo o Firefox, Safari, IExplorer, Opera, ou outro, aceder ao seu servidorWEB e pedir os 3 documentos:

<http://localhost/Controlo.html>
<http://localhost/Supervisao.html>
<http://localhost/Reservatorio.html>

Quando aceder a <http://localhost/Reservatorio.html> o seu Browser deverá apresentar a imagem seguinte:

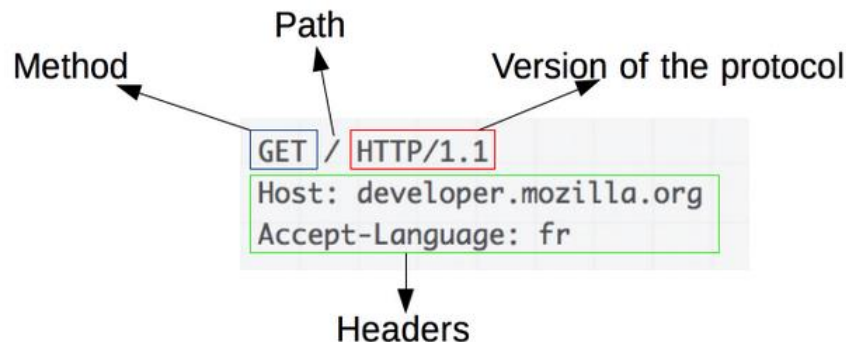


IV. Protocolo HTTP

Existem dois tipos de mensagens, requisições e respostas, cada uma com seu próprio formato.

Requisições

Exemplo de uma requisição HTTP:

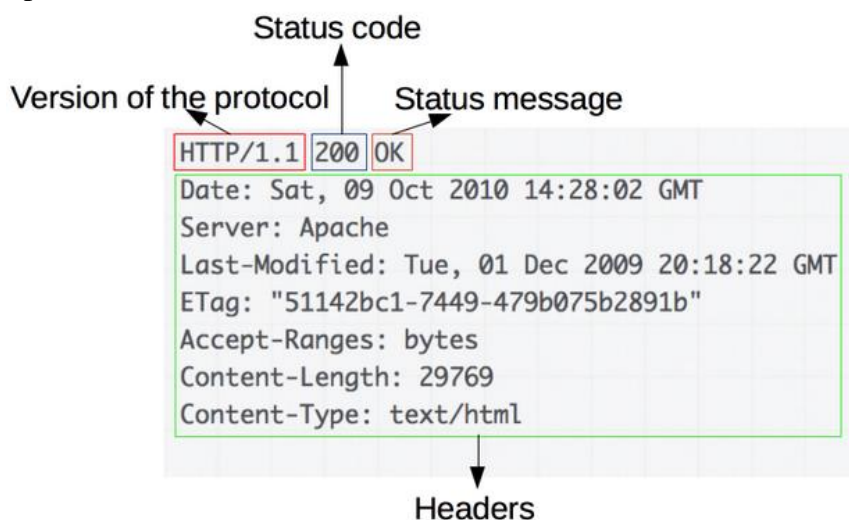


As requisições consistem nos seguintes elementos:

- Um método HTTP, geralmente é um verbo como [GET](#), [POST](#), [DELETE](#), [PUT](#), etc, ou um substantivo como [OPTIONS](#) ou [HEAD](#) que define qual operação o cliente quer fazer. Tipicamente, um cliente quer descarregar um recurso (usando [GET](#)) ou publicar dados de um [formulário HTML](#) (usando [POST](#)), embora mais operações podem ser necessárias em outros casos.
- O caminho do recurso a ser descarregado; a URL do recurso sem os elementos que são de contexto, por exemplo sem o protocolo [protocol](#) ([http://](#)), o domínio [domain](#) (aqui como [developer.mozilla.org](#)), ou a porta [port](#) TCP (aqui indicada pelo 80 que é ocultado por ser o número da porta padrão)
- A versão do protocolo HTTP.
- [Cabeçalhos](#) opcionais que contém informações adicionais para os servidores.
- E um corpo de dados, para alguns métodos como [POST](#), similares aos corpos das respostas, que contém o recurso requisitado.

Resposta

Exemplo de resposta HTTP:



Respostas consistem nos seguintes elementos:

- A versão do protocolo HTTP que elas seguem.
- Um código de status, indicando se a requisição foi bem-sucedida, ou não, e porquê.
- Uma mensagem de status, uma pequena descrição informal sobre o código de status.
- Cabeçalhos HTTP, como aqueles das requisições.

- Opcionalmente, um corpo com dados do recurso requisitado.

V. Servidor WEB VB

O nosso código irá servir ficheiros HTML. Para saber que documento está a ser solicitado a mensagem recebido tem de ser processada. A variável `caminho_pasta_html` irá conter uma string com o caminho da pasta onde estão os documentos HTML.

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    'Ver se há algum pedido de ligação, aceitar se for o caso
    If servidor.Pending = True Then
        ligacao = servidor.AcceptTcpClient()
        stream = ligacao.GetStream()
    End If

    If ligacao.Connected Then
        Dim carateres_disponiveis As Integer
        carateres_disponiveis = ligacao.Available 'ver numero de bytes disponíveis

        Dim carateres(carateres_disponiveis) As Byte
        stream.Read(carateres, 0, carateres_disponiveis) 'ler os bytes

        Dim texto_recebido As String

        'Copiar para variável texto recebido
        Dim i As Integer
        texto_recebido = "" 'Apagar o texto recebido
        For i = 0 To carateres_disponiveis - 1
            texto_recebido += texto_recebido + Chr(carateres(i))
        Next i

        'Copiar para a caixa de texto (só para visualização)
        'TextBox1.Text = TextBox1.Text + vbCrLf + texto_recebido
        TextBox1.Text = texto_recebido

        '-----
        '----- Processamento da mensagem recebida -----
        '-----

        Dim primeira_linha = texto_recebido.Substring(0, texto_recebido.IndexOf(vbCrLf))
        Dim itens_primeira_linha = primeira_linha.Split(" ")
        Dim metodo = itens_primeira_linha(0)
        Dim caminho = itens_primeira_linha(1)

        If InStr(caminho, "Supervisao.html") Then
            EnviarFicheiro(caminho_pasta_html + "Supervisao.html")
        ElseIf InStr(caminho, "Controlo.html") Then
            EnviarFicheiro(caminho_pasta_html + "Controlo.html")
        ElseIf InStr(caminho, "Reservatorio.html") Then
            EnviarFicheiro(caminho_pasta_html + "Reservatorio.html")
        End If

        ligacao.Close() ' o protocolo http prevê o fim da ligação
        ligacao = New TcpClient
    End If
End Sub
```

A resposta a enviar ao browser vai conter o documento requisitado na estrutura da mensagem HTTP. A maior parte dos cabeçalhos de resposta são opcionais, o nosso exemplo apenas vai enviar o “Content-type” a identificar que a resposta contém código HTML.

```
Private Sub EnviarFicheiro(ByVal caminho_do_ficheiro As String)
    If ligacao.Connected Then
        Dim stream_do_ficheiro As StreamReader = New StreamReader(caminho_do_ficheiro)
        Dim string_do_ficheiro As String = stream_do_ficheiro.ReadToEnd()

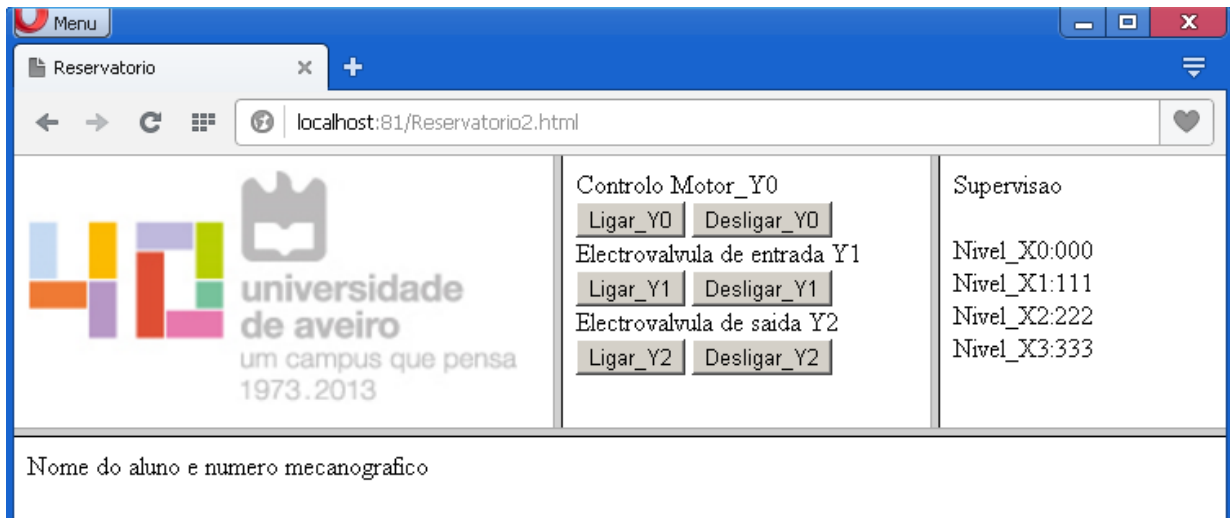
        Dim estado = "HTTP/1.1 200 OK"
        Dim cabecalho = "Content-Type: text/html; charset=utf-8"

        Dim resposta = estado + vbNewLine + cabecalho + vbNewLine + vbNewLine + string_do_ficheiro

        Dim bytes_enviar() As Byte = Encoding.UTF8.GetBytes(resposta)
        stream.Write(bytes_enviar, 0, bytes_enviar.Length)
    End If
End Sub
```

VI. Trabalho de casa

Crie os documentos HTML que forem necessários e altere o ServidorWEB de forma que o Browser apresente as 4 “frames” seguintes:



- A frame da esquerda, apresenta um novo documento, a criar em casa, com o nome “Logo.html”. Esse documento tem a referência à imagem:

- A frame do centro, apresenta o documento “Controlo.html”.
- A frame da direita, apresenta o documento “Supervisao.html”.
- A frame de baixo, apresenta um novo documento, a criar, com nome “Rodape.html”
- O documento que define as frames e que o browser pede é o “Reservatorio2.html”
- O ServidorWEB deve ser alterado de forma a poder enviar para o Browser os documentos: “Logo.html”, “Rodape.html” e “Reservatorio2.html”.

Conhecimentos a adquirir

Introdução ao protocolo HTTP e ao funcionamento dos Servidores WEB.

Familiarização com páginas HTML

- Sequência de interações Cliente WEB – servidor WEB
- Sintaxe dos pedidos HTTP e respetivas respostas.
- Estrutura e sintaxe dos documentos HTML

Importante:

O trabalho será avaliado por questionário individual, nas semanas seguintes à entrega do mesmo.

VII. Bibliografia

[T7_HTTP_Apontamentos.pdf](#)

[T7_HTML_Apontamentos.pdf](#)

[T7_http_html2_pages_PPoint.pdf](#)

- [Rosen 2003] **Web Application Architecture, Principles, Protocols and Practices**, Leon Shklar, Richard Rosen, John Wiley & Sons, 2003
Chapter 3. Birth of the World Wide Web: HTTP
Chapter 4. Web Servers
Chapter 6. HTML and Its Roots

Norma HTTP

<https://www.ietf.org/rfc/rfc2616.txt>

<http://www.w3.org/Protocols/rfc2616/rfc2616.html>

http-protocol.wmv (25 min)

<https://www.youtube.com/watch?v=WGJrLqtX7As>

Introdução ao HTML

<http://www.w3schools.com>

Uma visão geral do HTTP

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>