Roteiro para Vídeo de Arquitetura I

1. Motivação:
    1. Ensinar *assembly* x86 é o objetivo da cadeira.
    2. Não tenho um computador adequado executando x86, estou na plataforma Mac M1, que é ARM.
    3. Poderia usar um *notebook* antigo, mas seria trabalhoso carregá-lo para a sala, pois o antigo que tenho não possui bateria.
    4. Vamos utilizar a nuvem:
        1. O Campus tem parceria com a AWS, através do programa AWS Academy.
        2. Eu irei utilizar máquinas virtuais para programar em *assembly*, os alunos tem a opção de fazer o mesmo.
2. Ambiente:
    1. O livro *Guide do Assembly Language*:
        1. Voltado para a graduação.
        2. Não se aprofunda em tópicos complexos.
        3. Já tem exercícios, alguns com resolução.
        4. O problema é que é todo baseado em Microsoft Visual Studio no Windows.
    2. Irei criar uma máquina virtual Windows Server na nuvem:
        1. Existe a opção do Windows Server.
        2. Pelo AWS Academy o máximo que podemos criar é uma máquina com 8 GB de RAM e 2 núcleos.
        3. Vou tentar nessa máquina mais limitada, a memória parece ser suficiente, mas os núcleos são poucos.
        4. O teste será executar o primeiro código do livro.
    3. Desvantagens:
        1. O acesso remoto é pelo protocolo RDP, existem clientes para Linux, Windows, Mac, etc.
        2. O problema é que se a conexão não for boa, o acesso pode ficar lento.
        3. Eu já tenho uma ideia de como tratar isso, mas ficará para um segundo vídeo. Primeiro vamos deixar o Windows configurado.
    4. Vantagens:
        1. Qualquer problema no ambiente, é fácil reconfigurá-lo.
        2. A instalação nativa do Microsoft Visual Studio envolve o *download* de vários *gigabytes*. Dentro da rede da Amazon é bem rápido.
        3. Os alunos terão contato com a Nuvem, tecnologia imprescindível para o profissional de TI moderno.
    5. Se o aluno tiver acesso a uma máquina Windows com no mínimo 4 GB de RAM, ele pode instalar o Microsoft Visual Studio (é gratuito) e não usar a nuvem, se assim preferir.

3. Configuração da máquina virtual:
    1. O aluno irá receber um e-mail do AWS Academy e realizar o cadastro. Como já fiz o cadastro, não tenho como demonstrar aqui, mas qualquer dúvida no cadastro é conversar comigo.
    2. Após o cadastro, o aluno pode acessar o AWS Academy pelo endereço https://awsacademy.instructure.com/ e clicar em *Student Login*.
    3. Ao fazer o *login*, deve escolher o curso AWS Academy Learner Lab [41712].
    4. Indo em Módulos, depois em *Learner Lab -> Learner Lab*.
    5. No primeiro acesso, aparecerá um termo de compromisso da Vocareum. Passe o texto até embaixo e aceite (*I Agree*). Irá aparecer uma tela com instruções. Se estiverem em Inglês, procure por EN-US e troque por PT-BR se assim desejar.
    6. Acessando a AWS:
        1. Na barra acima das instruções, existem as opções *Start Lab*, *End Lab*, *AWS Details*, *Readme* e *Reset*.
        2. Clique em *Start Lab*. Isso irá iniciar um ambiente AWS limitado para você usar, com direito a US$ 100 de créditos. Você não será cobrado de forma alguma, mesmo se passar esse limite. Irá demorar alguns minutos na primeira inicialização, o círculo vermelho no canto superior esquerdo irá ficar amarelo enquanto inicia, depois verde.
        3. Depois do círculo verde, clique em *AWS Details*. Procure pela linha *SSH KEY,* com as opções *Show*, *Download PEM* e *Download PPK*. Clique em *Download PEM* e salve o arquivo *labsuser.pem*.
        4. Volte ao canto superior esquerdo, procure pelo nome AWS seguido de um círculo verde. Clique nele. Dependendo do seu navegador, você pode ter que liberar a abertura de novas janelas.
        5. Você verá o console *web* da AWS.
    7. Criando uma instância Windows:
        1. No console *web* da AWS, no canto superior esquerdo, ao lado de *Services*/Serviços, há um campo de busca. Digite EC2 nele. Escolha o serviço EC2, de criação de servidores virtuais (*virtual servers*) na nuvem. Você entrará na página do serviço EC2.
        2. A primeira coisa que precisamos fazer é configurar um *firewall* que permita conexões do protocolo de acesso remoto RDP. Na AWS, *firewalls* são chamados de Grupos de Segurança (*Security Groups*).
            1. Procure nas opções à esquerda a parte de Rede e Segurança (*Network and Security*) e selecione Grupos de Segurança (*Security Groups*).

2. No campo superior direito, selecione Criar grupo de segurança (*Create security group*).
3. Coloque o nome do grupo para ser *arquitetura*. Na descrição, coloque *Arquitetura e Organizacao de Computadores 2023.1*. Não coloque acentos nesses campos, pois não são aceitos.
4. Adicione Regras de Entrada (*Inbound Rule*):
    1. Em Tipo, no lugar de TCP Personalizado (*Custom TCP*), escolha RDP.
        1. Na parte de Origem (*Origin*, quarto campo) selecione Qualquer Local - IPv4 (*Anywhere IPv4*).
        2. Clique novamente em Adicionar Regra e coloque o mesmo protocolo, RDP, com a versão IPv6 (*Anywhere IPv6*).
    2. Refaça os mesmos passos do protocolo RDP, mas agora para o protocolo SSH (usaremos isso no futuro, aguardem).
    3. Não mexa em mais nada. Vá até o final da página, no canto esquerdo, clique em Criar grupo de segurança (*Create security group*). Será exibida uma tela com o resumo das duas regras.
3. Agora vamos criar uma instância que usa o grupo de segurança criado:
    1. Cliente em Instâncias -> Instâncias (*Instances -> Instances*) no lado esquerdo.
    2. Selecione Executar Instâncias (*Run Instances*) no canto superior direito.
    3. Coloque o nome da instância como *WindowsAssemblyX86*.
    4. Escolha a imagem do Windows (Microsoft Windows Server Base 2022).
    5. No Tipo de Instância (*Instance Type*), escolha *t3.large*.
    6. No Par de Chaves (*Key Pair*) escolha *vockey*. Apesar do nome diferente, é equivalente ao arquivo *labsuser.pem* que você já baixou.
    7. Na parte de Configurações de rede, escolha S*elecionar grupo de segurança existente*. Na caixa abaixo haverá as opções *default* e *arquitetura*, escolha *arquitetura*.
    8. Nas Configurações de Armazenamento, informe 50 GB e deixe a opção gp2.
    9. Não altere mais nada e clique em Executar Instância (*Run Instance*) no quadro à direita. Se der algum erro, faça uma captura de tela e envie ao professor.
    10. Clique em *Visualizar todas as instâncias*.
4. Acessando a instância Windows:
    1. Na tela das Instâncias no EC2, você deve esperar até a

instância WindowsAssemblyX86 estar no Estado (*State*) Executando (*Running)*.
2. Clique com o botão direito nela e selecione Conectar.
3. Selecione *Cliente RDP*.
4. Selecione *Fazer download de arquivo de área de trabalho remota*. Salve o arquivo *WindowsAssemblyX86.rdp*. Você deve instalar um cliente RDP na sua máquina. No Windows, a extensão .rdp já leva para o utilitário de Área de Trabalho Remota. No Ubuntu, a versão Desktop já vem com um aplicativo que irá abrir o arquivo .rdp. Existem várias opções no Linux, procure na Internet a que achar melhor.
5. Logo abaixo, vá em *Senha -> Obter Senha*. Clique em *Carregar arquivo da chave privada* e selecione o arquivo *labsuser.pem*. Selecione *Descriptografar Senha*. De volta a tela anterior, recorte a senha gerada (é impossível memorizá-la) e salve em algum lugar.
6. Clique duas vezes no arquivo *WindowsAssemblyX86.rdp* no seu gerenciador de arquivos. Coloque usuário *Administrator* e cole a senha anterior. Se surgir algum pedido de aceite de certificado, selecione continuar ou aceitar.
7. Irá surgir uma tela da sua instância Windows na nuvem.
8. Para sair da instância, não precisa desligá-la pelo SO, apenas feche a janela do aplicativo de conexão remota.

8. Uma vez que a instância tenha sido criada, ao entrar novamente no AWS Academy, você não precisa repetir o processo de criação do grupo de segurança, nem da criação da instância. Mas é preciso baixar um novo arquivo *WindowsAssemblyX86.rdp*, pois o IP público da instância muda a cada utilização.
9. Sempre finalizar a utilização do ambiente, volte à tela do AWS Academy e clique em *End Lab*. Isso irá parar a instância, salvando seus créditos. Quando você retornar e clicar em *Start Lab*, a instância será iniciada automaticamente.

4. Configurando o Ambiente de Programação:
    1. Entre na instância do Windows Server em execução na nuvem usando o RDP.
    2. Clique com o botão direito no relógio, escolher *Adjust date/time* e escolhe *Time zone* para *(UTC -03:00) Cayenne, Fortaleza*.
    3. Criando Usuário:
        1. Vamos criar um usuário fixo, para evitar termos que recuperar a senha.
        2. Na barra de busca do Menu Iniciar, onde está escrito *Type here to search*, digite *Computer Management*. Escolha a primeira opção.
        3. Vá em *Computer Management -> System Tools -> Local Users*

and Groups. Escolha a pasta *Users*.

4. Clique com o botão direito na pasta e escolha *New User*:
    1. Vamos colocar *User name* como *alunoufc*.
    2. Em *Full name* e em *Description* coloque *Aluno UFC Quixadá*.
    3. Coloque uma senha complexa, envolvendo símbolos, letras maiúsculas, letras minúsculas e números, mas que você lembre. Irei colocar *@15UFC66quixada#*.
    4. Desmarque *User must change password at next logon*.
    5. Clique em *Create*.
5. Colocando o usuário *alunoufc* no grupo de administradores:
    1. Volte em *Computer Management -> System Tools -> Local Users and Groups*. Escolha a pasta *Users*..
    2. O usuário recém criado deve estar lá. Clique com o botão direito nele e vá em *Properties*. Clique na aba *Member Of* e no botão *Add*.
    3. Na caixa *Enter the object names to select*, digite *Administrators* e depois em *Check Names*. Irá completar automaticamente, depois vá em no botão *Ok*.
    4. Na próxima tela, confirme que *User must change password at next logon* não está marcada. Também desmarque *Account is disabled* e marque *Password never expires*.
    5. Clique em *Apply* e depois *Ok*
    6. Para confirmar que tudo deu certo, saia do cliente RDP e entre novamente, mas desta vez usando usuário *alunoufo* e senha *@15UFC66quixada#*.
4. Alterando o perfil do teclado:
    1. Vamos deixar o Windows em Inglês mesmo, para ir treinando, mas o teclado é bom poder mudar para aceitar acentos.
    2. Na barra de busca do Menu Iniciar, onde está escrito *Type here to search*, digite *Language Settings*. Escolha a primeira opção.
    3. Na parte debaixo da tela, na seção *Preferred Language* clique uma vez em *English (United States*). Clique no botão *Options*.
    4. Na parte debaixo da tela, na seção *Keyboards*, escolha *Add a keyboard*. Para teclados no formato americano, escolha *United States-International*. Para teclados brasileiros, escolha *Portuguese (Brazil ABNT2)*.
    5. Surgirá na área da bandeja, ao lado do relógio (canto inferior direito), as opções ENG US e a outra opção que você adicionou. Pode escolher a que deseja utilizar.
5. Instalando o Visual Studio:
    1. Abra o Microsoft Edge e aceite as primeiras perguntas. Pode responder negando tudo, não irá afetar nosso uso.
    2. Vá em https://visualstudio.microsoft.com/vs/community/ e

clique e em *Download*. Ele já deve baixar o arquivo *VisualStudioSetup.exe*. Execute esse arquivo para iniciar a instalação. Ao clicar em *Continue* na primeira tela, ele irá baixar o resto do instalador.

3. Na parte de *Workloads*, escolha apenas *Desktop Development with C++*. Irão surgir novas opções no lado direito. Desmarque as seguintes opções:
    1. *Test Adapter for Boost.Test*.
    2. *Test Adaptar for Google Test*.
    3. *LiveShare*.
    4. *IntelliCode*.
    5. *C++ Address Sanitizer*.
    6. *Security Issues Analysis*.
    7. *C++ ATL for latest...*
4. Clique em *Install*. São quase 10 GB, então em uma rede normal, demora um bom tempo.
5. Se você quiser pode usar uma conta da Microsoft, ou até o mesmo o e-mail @alu.ufc.br, para fazer o registro ou criar uma conta. Mas por enquanto, vamos de *Skip this for now*.
6. Escolha o esquema de cor que desejar e clique em *Start Visual Studio*.
7. Testando a instalação:
    1. Escolha *Create a new project*:
    2. Escolha o modelo *Console App*, depois em *Next*.
    3. Coloque o nome *Teste01* em *Project Name*. Clique em *Create*.
    4. Irá surgir uma tela com um olá mundo em C++. Clique *Debug -> Start Without Debug* no menu para executar o programa. Irá surgir uma tela com a mensagem.
8. Primeiro Programa em *Assembly*:
    1. Crie um novo projeto, com o mesmo modelo do exemplo acima. Coloque o nome do projeto como *PrimeiroPrograma*.
    2. No painel *Solution Explorer* à direita, escolha a pasta *Source Files* e remova os arquivos com a extensão .cpp.
    3. Clique com o botão direito na mesma pasta, depois *Add -> New Item*. No lugar de um arquivo .cpp, crie o arquivo *Main.asm*.
    4. No Painel *Solution Explorer,* clique com o botão direito em *PrimeiroPrograma*, o nome do projeto. Escolha *Build Dependencies -> Build Customizations*. Escolha *masm(.targets, .props)*.
    5. Clique com o botão direito no arquivo *Main.asm* e vá em *Properties*. Para o campo *Item Type*, escolha o valor *Microsoft Macro Assembler*, depois *Apply* e *OK*.

6. Coloque o conteúdo de http://joao.marcelo.nom.br/source/PrimeiroPrograma.asm no arquivo *Main.asm*.
7. Na Barra de Ferramentas, à direta do ícone de salvar, onde tem x64, selecione x86. É entre *Debug* e *Local Windows Debugger*.
8. No Painel *Solution Explorer,* clique com o botão direito em *PrimeiroPrograma*, o nome do projeto. Escolha *Properties*. No Painel à esquerda, expanda *Linker* e escolha *Advanced*. Certifique-se que na barra superior em *Platform,* está *Win32*, não *x64*. No atributo *Entry Point*, coloque *main,* depois *Apply* e *OK*.
9. Vá em *Build,* no menu, e depois *Build Solution*. Verifique se a construção ocorre sem erros.
10. Clique *Debug -> Start Without Debug* no menu para executar o programa. Irá surgir uma tela sem nenhuma mensagem, pois ainda veremos entrada e saída.

6. Habilitando o SSH:
    1. Vamos configurar o SSH na máquina Windows para poder acessá-la sem usar a interface gráfica, exigindo menos da rede.
    2. Vamos seguinte estas orientações: https://learn.microsoft.com/pt-br/windows-server/administration/openssh/openssh_install_firstuse
    3. Usando o PowerShell:
        1. Clique com o botão direito no Menu Iniciar. É o símbolo do Windows no canto inferior esquerdo. Escolha a opção *Windows PowerShell (Admin)*. Clicando em *Yes* irá aparecer uma tela de comandos.
        2. Verifique se o SSH está instalado com o comando:
            1. *Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'*
            2. Deve aparecer que o cliente está instalado, mas o servidor não.
        3. Instale o servidor com o comando:
            1. *Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0*
        4. Inicie o servidor com
            1. *Start-Service sshd*
            2. Depois configure par iniciar com o sistema com
                1. *Set-Service -Name sshd -StartupType 'Automatic'*
    4. Para testar, na sua máquina local:
        1. Recupere o IP Público da instância (está no console *web* da AWS ou no canto superior direito no papel de parede da máquina virtual).
        2. Execute

        1. *ssh alunoufc@XXX.XXX.XXX.XX*
        2. XXX.XXX.XXX.XXX é o IP público da instância.
        3. Responda *Yes* e forneça a senha criada para o usuário *alunoufc*.
    3. Lembre que o IP público muda toda vez que a instância é iniciada.
    4. Você irá entrar no *prompt* de comando do Windows Server.
7. O problema do cenário acima é que o *prompt* de comando padrão do Windows Server é muito simples.
8. Vamos configurar um *prompt* mais avançado, o PowerShell (https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_server_configuration):
    1. Entre na instância Windows através da interface gráfica, usando o protocolo RDP.
    2. Clique com o botão direito no Menu Iniciar. É o símbolo do Windows no canto inferior esquerdo. Escolha a opção *Windows PowerShell (Admin)*. Clicando em *Yes* irá aparecer uma tela de comandos.
    3. Execute com o comando:
        1. *New-ItemProperty -Path "HKLM:|SOFTWARE|OpenSSH" -Name DefaultShell -Value “C:|Windows|System32|WindowsPowerShell|v1.0|powershell.exe" -PropertyType String -Force*
    4. Ao entrar de novo via SSH, estará no PowerShell.
9. Agora vamos configurar as variáveis de ambiente dos compiladores no PowerShell:
    1. Clique no Menu Iniciar. Procure pela pasta *Visual Studio 2022*. Clique com o botão direito no ícone azul *Developer PowerShell VS 2022*. Vá em *More -> Open file location*. Irá abrir uma basta com os atalhos do Visual Studio.
    2. Clique novamente com o botão direito no atalho *Developer PowerShell VS 2022*, escolha *Properties*.
    3. Anote o conteúdo do campo *Target* no Bloco de Notas (*Notepad*). Deve ser algo como:
        1. *C:|Windows|SysWOW64|WindowsPowerShell|v1.0|powershell.exe -noe -c "&{Import-Module """C:|Program Files|Microsoft Visual Studio|2022|Community|Common7|Tools|Microsoft.VisualStudio.DevShell.dll"""; Enter-VsDevShell ea935e97}"*
    4. Da linha anterior, recorte e cole no Bloco de Notas para ficar apenas:
        1. *Import-Module "C:|Program Files|Microsoft Visual Studio|2022|Community|Common7|Tools|Microsoft.VisualStudio.DevShell.dll";*

Enter-VsDevShell ea935e97
        2. Basicamente deixar o comando em duas linhas e tirar as aspas extra na primeira linha.
        3. Veja que o valor *ea935e97* pode mudar de instalação para instalação.
    5. Clique com o botão direito no Menu Iniciar. É o símbolo do Windows no canto inferior esquerdo. Escolha a opção *Windows PowerShell*. Não é a versão *Admin*. Irá aparecer uma tela de comandos.
    6. Digite o comando:
        1. *New-Item $profile -Type File -Force*
        2. Navegue até o diretório *C:|Users|alunoufc|Documents|WindowsPowerShell* pelo Windows Explorer. Dentro dele deve ter o arquivo *Microsoft.PowerShell_profile.ps1*. Clique com o botão direito e escolha *Edit*.
        3. No Editor que surge, coloque as linhas do Item 4.1 acima. Salve o arquivo.
    7. Agora retorne ao seu computador local e tente fazer o *login* via SSH. Você será recebido com uma mensagem dizendo que o ambiente de desenvolvimento está configurado. Teste executar os comandos *cl.exe* e *MSBuild.exe*
10. Integrando com o Visual Studio Code:
    1. Poderíamos instalar o Vim no Windows e programar via SSH através da linha de comando.
    2. Entretanto, vamos usar uma versão mais leve do Visual Studio, o Visual Studio Code. Através dele, podemos acessar a instância Windows via SSH.
    3. Instale o Visual Studio Code na sua máquina local.
    4. Clique no Ícone de extensões. Está ao lado esquerdo, são quatro quadrados, com um deles deslocado.
    5. Procure pelas extensões:
        1. *Remote - SSH*
        2. *Remote Explorer*
    6. Instale as duas extensões e reinicie o Visual Studio Code.
    7. No mesmo local do ícone das Extensões, deve ter um novo ícone, no formato de um monitor. É o *Remote Explorer*. Clique nele.
    8. Ao lado de SSH, se você mover o cursor para a direita, irá aparecer uma engrenagem e um símbolo de adição. Clique no +.
    9. Você será levado a um formulário de comando. Coloque:
        1. *ssh alunoufc@XXX.XXX.XXX.XXX*
        2. Lembrando que XXX.XXX.XXX.XXX é o IP público da instância, que muda a cada inicialização.

    3. Depois ele pergunta qual arquivo salvar a configuração. Escolha a primeira opção.
10. Ao apertar Enter, ele avisará que o que servidor foi adicionado.
11. Na tela do *Remote Explorer*, se você recarregar, deve aparecer o novo IP. Clique com o botão esquerdo e escolha *Connect in New Window*.
12. Irá surgir uma nova tela do Visual Studio Code, você deve colocar a senha do usuário *alunoufc*.
13. Essa nova versão acessa o sistema Windows. Tanto o Terminal quando o sistema de arquivos equivale ao da máquina remota.

11. Instalando o Git:
    1. Abra a instância utilizando o cliente RDP.
    2. Acesse https://git-scm.com/download/win usando o Microsoft Edge e baixe o instalador de 64 bits.
    3. Existe várias opções. Deixe todas como padrão e vá clicando em *Next*. No final clique em *Finish*.
12. Instalando a GitHub CLI:
    1. Abra a instância utilizando o cliente RDP.
    2. Acesse https://cli.github.com. Escolha *Download for Windows*.
    3. Abra o executável e siga o mesmo padrão de aceitar todas as opções padrão com *Next*. No final clique em *Finish*.