Avaliação PAM (parte escrita)

1.para quê serve o MVVM

O padrão MVVM serve para ajudar na separação da parte lógica e da interface do usuário de um projeto

2. O que é o MVVM community toolkit?

É uma ferramenta pública, usada principalmente por aqueles que utilizam do modelo MVVM de arquitetura de softwares

3. Para quê serve o appshell? Como adicionar uma view para ser exibida no lugar da main Page?

O appshell é utilizado por vários motivos, podendo melhorar o desempenho, possibilitar uma experiência básica da aplicação em caso da máquina estar fora da rede e seguir alguns padrões de arquitetura se software.

Para exibir a view no lugar da main Page basta dicionar a view criada como Content Template no AppShell

4. Para que serve o <TabBar> appshell

A TabBar funciona como uma forma de organizar as páginas de uma aplicação, sendo exibidas na maior parte das vezes no inferior da interface da aplicação

5. Em aula, como instalamos pacotes usando Nuget no C#?

Para a instalação usando o Nuget, é necessário que clicar na solução do nosso projeto com o botão direito, clicar em “Manage NuGet Packages for Solution…”, é procurar o que deseja instalar

6.O que é o Binding? Em que arquivo criamos os bindings?

Binding é a conexão entre a view e o ViewModel, que faz a atualização automática da interface. São criados normalmente em arquivos .XAML

7.O que é um Command?

Em .NET MAUI e no padrão MVVM, um `Command` é basicamente uma ação que você pode atrelar a um botão ou outra interação do usuário. Ele diz o que fazer quando o usuário clica no botão (ou faz outra interação), e pode dizer também se o botão deve estar habilitado ou não em certos momentos.

8.O que é o Binding context? Escreva como associamos o binding context a classe ViewModel

O BindingContext é uma propriedade de um elemento visual (por exemplo, uma página, um controle, etc.) que define a fonte de dados para os bindings declarados no XAML. Quando você define bindings em XAML, como {Binding Message), o mecanismo de binding busca essa propriedade no objeto definido pelo BindingContext.

Primeiro, você define a classe ViewModel com as propriedades e comandos que você quer bindar. No arquivo XAML, você define a interface de usuário e especifica os bindings. No arquivo de code-behind (por exemplo, MainPage.xaml.cs), você associa o BindingContext da página a uma instância do ViewModel.

9.Para que serve a ViewModel?

A ViewModel atua como uma forma intermediária entre View (parte da interface do user) e Model (Parte lógica do projeto)

10.Qual a diferença entre construir uma view ne MVVM e fora do MVVM?

Uma view dentro do padrão MVVM é mais organizada e consequentemente possibilita uma maior facilidade em atuar no código, já uma view fora do padrão MVVM pode proporcionar má organização e em projetos mais complexos oferecer maior dificuldade em atuar no código

11.O que quer dizer que um atributo é uma propriedade observável? [Observable Propriety]?

Uma propriedade observável pode ser classificada como uma propriedade que notifica sobre a sua mudança de valores

12.Quando criamos uma observable Property, que cuidado devemos ter com os atributos?

Quando uma propriedade observável é criada é necessário fazer corretamente o encapsamento