ETEC PROF. HORÁCIO AUGUSTO DA SILVEIRA

PROGRAMAÇÃO DE APLICATIVOS MOBILE – II – 2024/1

Luiz Fernando Souza

Display Action Sheets - Atualizando dados das disputas e dos personagens

Nesta aula, faremos a configuração para restaurar os pontos de vida dos personagens e os dados das disputas.

1. Adicione os métodos abaixo na classe **PersonagemService** que consumirão o serviço da API.

```
public async Task<int> PutRestaurarPontosAsync(Personagem p)
{
    string urlComplementar = "/RestaurarPontosVida";
    var result = await _request.PutAsync(apiUrlBase + urlComplementar, p, _token);
    return result;
}
1 reference
public async Task<int> PutZerarRankingAsync(Personagem p)
{
    string urlComplementar = "/ZerarRanking";
    var result = await _request.PutAsync(apiUrlBase + urlComplementar, p, _token);
    return result;
}
0 references
public async Task<int> PutZerarRankingRestaurarVidasGeralAsync()
{
    string urlComplementar = "/ZerarRankingRestaurarVidas";
    var result = await _request.PutAsync(apiUrlBase + urlComplementar, new Personagem(), _token);
    return result;
}
```

2. Abra a **ListagemPersonagemViewModel** e crie os métodos que acionarão a classe de serviço

```
public async Task ExecutarRestaurarPontosPersonagem(Personagem p)
{
    await pService.PutRestaurarPontosAsync(p);
}
1 reference
public async Task ExecutarZerarRankingPersonagem(Personagem p)
{
    await pService.PutZerarRankingAsync(p);
}
0 references
public async Task ExecutarZerarRankingRestaurarVidasGeral()
{
    await pService.PutZerarRankingRestaurarVidasGeralAsync();
}
```

3. Adicione o método que processará todas as operações possíveis para o personagem

```
public async void ProcessarOpcaoRespondidaAsync(Personagem personagem, string result)
        {
            if (result.Equals("Editar Personagem"))
            {
                await Shell.Current
                        .GoToAsync($"cadPersonagemView?pId={personagem.Id}");
            }
            else if (result.Equals("Remover Personagem"))
            {
              if (await Application.Current.MainPage.DisplayAlert("Confirmação",
               $"Deseja realmente remover o personagem {personagem.Nome.ToUpper()}?",
"Yes", "No"))
                {
                    await RemoverPersonagem(personagem);
                    await Application.Current.MainPage.DisplayAlert("Informação",
                        "Personagem removido com sucesso!", "Ok");

                    await ObterPersonagens();
                }
            }
            else if (result.Equals("Restaurar Pontos de Vida"))
            {
                if (await Application.Current.MainPage.DisplayAlert("Confirmação",
                    $"Restaurar os pontos de vida de {personagem.Nome.ToUpper()}?", "Yes", "No"))
                {
                    await ExecutarRestaurarPontosPersonagem(personagem);
                    await Application.Current.MainPage.DisplayAlert("Informação",
                        "Os pontos foram restaurados com sucesso.", "Ok");

                    await ObterPersonagens();
                }
            }
            else if (result.Equals("Zerar Ranking do Personagem"))
            {
                if (await Application.Current.MainPage.DisplayAlert("Confirmação",
                    $"Zerar o ranking de {personagem.Nome.ToUpper()}?", "Yes", "No"))
                {
                    await ExecutarZerarRankingPersonagem(personagem);
                    await Application.Current.MainPage.DisplayAlert("Informação",
                        "O ranking foi zerado com sucesso.", "Ok");

                    await ObterPersonagens();
                }
            }
      }
```

4. Adicione o método que vai exibir as opções ao usuário e passará ao método de processamento, a palavra selecionada no click da caixa de opções.

```
public async Task ExibirOpcoesAsync(Personagem personagem)
{
    try
    {
        personagemSelecionado = null;
        string result = string.Empty;

        result = await Application.Current.MainPage
            .DisplayActionSheet("Opções para o personagem " + personagem.Nome,
                "Cancelar",
                "Editar Personagem",
                "Restaurar Pontos de Vida",
                "Zerar Ranking do Personagem",
                "Remover Personagem");

        if (result != null)
            ProcessarOpcaoRespondidaAsync(personagem, result);
    }
    catch (Exception ex)
    {
        await Application.Current
            .MainPage.DisplayAlert("Ops...", ex.Message, "Ok");
    }
}
```

5. Altere a propriedade **PersonagemSelecionado** inserindo o trecho sinalizado em verde e removendo o trecho em vermelho. Neste caso estamos invocando a exibição da caixa de opções.

```
private Personagem personagemSelecionado;
0 references
public Personagem PersonagemSelecionado
{
    get { return personagemSelecionado; }
    set
    {
        if (value != null)
        {
            personagemSelecionado = value;

            Shell.Current
                .GoToAsync($"cadPersonagemView?pId={personagemSelecionado.Id}");

            _ = ExibirOpcoesAsync(personagemSelecionado);
        }
    }
}
```

6. Programe o método que será responsável por zerar o ranking geral

```
public async Task ZerarRankingResturarVidasGeral()
{
    try
    {
        if (await Application.Current.MainPage.DisplayAlert("Confirmação",
           $"Deseja realmente zerar todo o ranking?", "Yes", "No"))
        {
            await ExecutarZerarRankingRestaurarVidasGeral();

            await Application.Current.MainPage
                .DisplayAlert("Informação", "Ranking zerado com sucesso.", "Ok");

            await ObterPersonagens();
        }
    }
    catch (Exception ex)
    {

        await Application.Current.MainPage
            .DisplayAlert("Ops...", ex.Message + " Detalhes: " + ex.InnerException, "Ok");
    }
}
```

7. Declare o ICommand abaixo do fechamento do construtor no grupo onde existe outros ICommands

```
public ICommand ZerarRankingRestaurarVidasGeralCommand { get; set; }
```

8. Atribua o método ao ICommand dentro do construtor

```
ZerarRankingRestaurarVidasGeralCommand =
    new Command(async () => { await ZerarRankingResturarVidasGeral(); });
```

9. Posicione o botão para zerar o Ranking geral no Design da View, antes do ListView.

```
<Button Text="Zerar Ranking Geral" Command="{Binding ZerarRankingRestaurarVidasGeralCommand}"
          FontAttributes="Bold" VerticalOptions="FillAndExpand"/>
```

- Execute o aplicativo e faça os testes das funcionalidades programadas.

10. Adicione uma regra no método ExibirOpcoesAsync de View/Personagens/ListagemView.xaml.cs para caso os pontos de vida do Personagem estejam menores ou iguais a zero, o menu do personagem em questão seja exibido com menos opções.

```
public async Task ExibirOpcoesAsync(Personagem personagem)
{
    try
    {
        personagemSelecionado = null;
        string result = string.Empty;

        if (personagem.PontosVida > 0)
        {
            result = await Application.Current.MainPage
                .DisplayActionSheet("Opções para o personagem " + personagem.Nome,
                    "Cancelar",
                    "Editar Personagem",
                    "Restaurar Pontos de Vida",
                    "Zerar Ranking do Personagem",
                    "Remover Personagem");
        }
        else
        {
            result = await Application.Current.MainPage
                .DisplayActionSheet("Opções para o personagem " + personagem.Nome,
                    "Cancelar",
                    "Restaurar Pontos de Vida");
        }

        if (result != null)
            ProcessarOpcaoRespondidaAsync(personagem, result);
    }
    catch (Exception ex)
    {
        await Application.Current
            .MainPage.DisplayAlert("Ops...", ex.Message, "Ok");
    }
}
```

Luiz Fernando Souza

- Resultados esperados