

Tratando erro de comunicação

Transcrição

Até o momento vimos duas situações: na primeira, o usuário irá se conectar ao servidor e preenche corretamente os campos de "Usuário" e "Senha", conectando-se com sucesso. Na segunda situação, algum campo é preenchido incorretamente, e o servidor nega sua autenticação.

Há ainda outra situação que não testamos, que é quando existe um problema de comunicação entre o *device* do usuário e o servidor.

Vamos rodar a aplicação simulando um cenário em que ocorre algum problema de comunicação, a partir da desabilitação da conexão dos dados do usuário. No emulador do Android, iremos às configurações ("*Settings*"), selecionando o "*Data usage*", que se refere ao uso de dados.

Desabilitaremos o "*Cellular data*" e, confirmando isto, voltaremos à aplicação do *TestDrive*, preenchendo corretamente os campos de login, com "joao@alura.com.br" e "alura123" sendo "Usuário" e "Senha", respectivamente.

Ao clicarmos em "Entrar", vemos um erro de "*Unhandled Exception*" (exceção não tratada), que diz "*System.Net.WebException: Error: NameResolutionFailure*". O que está acontecendo aqui? Vamos utilizar um *breakpoint* para verificarmos com mais detalhes no momento em que se faz a conexão e tenta-se fazer a autenticação com o servidor, na linha `var resultado = await cliente.PostAsync("/login", camposFormulario);` do arquivo `LoginService.cs`.

Feito isto, rodaremos novamente a aplicação, com as pausas nestes pontos. Repetiremos o procedimento de preenchimento dos dados com as informações corretas.

Após a pausa, precisamos do resultado para saber se o código de `IsSuccessStatusCode` foi bem-sucedido ou não. Rodando a partir deste ponto, temos exatamente o erro surgido anteriormente. Como uma exceção está sendo lançada, implementaremos um bloco "*Try/Catch*" para capturarmos a exceção e, no bloco `catch`, teremos que de alguma forma tratá-la para que o usuário não receba uma mensagem de erro na tela, fechando a aplicação abruptamente, por exemplo.

Para isto, vamos envolver o bloco relativo a `using`, de `LoginService.cs`, envolvendo-o em um bloco "*Try/Catch*". Clicando na seleção com o lado direito do mouse, selecionaremos "*Surround With*". Escolhemos "*try*" na lista, e o programa já cria o código automaticamente. Este código captura uma exceção, porém, mas faremos isso posteriormente, mantendo o código simples.

Caso ocorra algum erro de comunicação de dados, queremos capturá-lo lançando uma mensagem nova e utilizando o sistema de mensagens (`MessagingCenter`). Vamos copiar o código da mensagem que aparece quando a autenticação falha e substituir a mensagem em si.

Onde estava "Usuário ou senha incorreto", deixamos "Ocorreu um erro de comunicação com o servidor. Por favor verifique a sua conexão e tente novamente mais tarde.", com um "@" na frente, que permite que pulemos de linha. Assim, tratamos a exceção.

```
public class LoginService
{
    public async Task FazerLogin(Login login)
```

```
{
    try
    {
        using (var cliente = new HttpClient())
        {
            var camposFormulario = new FormUrlEncodedContent(new[]
            {
                ...
            });
            ...
        }
    }
    catch
    {
        MessagingCenter.Send<LoginException>(new LoginException(@"Ocorreu um erro de comunicação com o servidor. Por favor verifique a sua conexão e tente novamente mais tarde."),
            "FalhaLogin");
    }
}
```

Vamos rodar a aplicação de novo e verificar o que acontece quando tentamos nos comunicar com o servidor sem uma linha de dados. Digitemos "joao@alura.com.br" e "alura123", e apertaremos "Entrar". A aplicação pausa na linha do `PostAsync`, continuamos rodando, e aparece uma janela com o erro:

"Login - Ocorreu um erro de comunicação com o servidor. Por favor, verifique a sua conexão e tente novamente mais tarde. OK"

Clicaremos no botão de "OK" e religaremos a linha de dados do emulador, voltando ao "Setting", às configurações, habilitando os dados do celular. Nisto, vamos voltar à aplicação do *TestDrive*, clicaremos em "Entrar", ela para de rodar no `PostAsync`, continuaremos rodando, e o mesmo erro aparece.

Se repetirmos os procedimentos, o erro persiste.

Isto porque a aplicação está com uma informação defasada; Não sabemos se o problema vem do emulador, se é algo proveniente do sistema Android. Já ligamos a linha de dados, porém, somos informados que estamos sem conexão.

Pausaremos a aplicação, rodando-a em seguida. Vamos ver se assim o status da linha é atualizado. A comunicação com o servidor foi reestabelecida e o usuário foi autenticado com sucesso. Com isto, conseguiremos cercar as três situações que podem ocorrer: os dados do usuário estão corretos e o login é realizado satisfatoriamente, ou incorretos (quando a autenticação falha), ou não conseguimos nos conectar com o servidor porque o celular está com problema de conexão.