

01
SQLite

Transcrição

[00:00] A aplicação atualmente permite que o usuário faça o agendamento do TestDrive, mas o nosso usuário não sabem quais agendamento ele já fez, se por exemplo o nosso usuário marcou um agendamento para segunda-feira, outro para terça-feira e um outro TestDrive para quarta-feira, como ele vai saber quais carros ele agendou para cada dia?

[00:25] Ele não tem como saber, porque a nossa aplicação não está salvando esses dados localmente em lugar nenhum, até poderíamos armazenar uma lista local, em memória dos agendamentos que já foram feitos pelo usuário, por exemplo, se ele escolhe o Azera V6, ele vai lá e escolhe o acessório, clica no próximo, preenche os dados e clica no agendar, a aplicação vai enviar para o servidor da Aluracar esse agendamento e vai salvar remotamente, vai salvar no servidor, mas se ele fechar a aplicação e abrir de novo, o que acontece?

[01:04] Como ele tem uma lista em memória, uma lista volátil, esses dados dessa lista de agendamento vão se perder, porque não salvamos, não persistimos esses dados em lugar nenhum no nosso aparelho, no smartphone do usuário, por esse motivo é preciso o quê?

[01:23] Um mecanismo de persistência de dados, que permita que a nossa lista de agendamentos já feitos seja persistida e que ele possa ser recuperada tantas vezes quanto necessário, porque quando o usuário fechar aplicação e abrir de novo, essa lista vai estar disponível, vai ser recuperada novamente e vai estar disponível para o usuário ver tudo que ele já fez, todo seu histórico de agendamentos de TestDrive.

[01:51] Existem algumas técnicas para poder gravar dados localmente no smartphone, como por exemplo, sistema de arquivos com arquivo texto, arquivo formato XML, formato de JSON, formato CSV, só que uma abordagem mais profissional exige um tipo de armazenamento em banco de dados relacional e qual a diferença de um sistema de gerenciamento de banco de dados relacional e um arquivo texto por exemplo? Ou arquivo CSV, um TXT, ou um JSON, XML?

[02:25] O banco de dados relacional armazena os dados agrupados em tabelas, essas tabelas podem ter relacionamentos, podemos criar um mecanismo sofisticado de armazenamento de dados em que consigamos relacionar dados através de propriedades, através de campos, podemos fazer indexação, podemos trabalhar com transações, o banco de dados relacional é bem mais sofisticado em geral do que um banco de dados, por exemplo, baseado em arquivo.

[02:59] Se for um arquivo texto simples vamos simplesmente ler o arquivo texto e gravar em cima de novo. Só que é um banco de dados relacional podemos definir uma estrutura rica, podemos definir os tipos de Campo, os tipos de índice, podemos definir se um dado é, por exemplo, nulo ou não, se ele pode permitir um nulo, se ele pode, por exemplo, ter uma chave estrangeira, se ele pode ter uma chave primária, se ele exige que existam uma outra entidade, um outro registro, uma outra tabela, podemos forçar uma série de regras, dentro do banco de dados, para ele ficar sempre consistente.

[03:47] Para o desenvolvimento multiplataforma a ferramenta mais utilizada mundialmente é o SQLite, porque ele é um mecanismo de banco de dados relacional muito leve, muito compacto, ele cabe em qualquer aparelho, qualquer dispositivo, isso já é suportado há muitos anos, desde que os smartphones eram muito mais limitados, o SQLite já cumpria essa função de ser um gerenciador de banco de dados relacional.

[04:16] No nosso projeto também vamos utilizar o SQLite, que é uma ferramenta Open Source, é gratuito e vamos baixar, vamos instalar o SQLite no nosso projeto do Xamarin Forms, para o TestDrive. Como o SQLite não vem instalado com o Visual Studio vamos ter que fazer isso manualmente, como instalamos um pacote com Visual Studio?

[04:41] Vamos utilizar a ferramenta, em Tools, ferramentas, vamos abrir ferramentas e vamos no “NuGet Package Manager”, vamos gerenciar os pacotes da nossa solução, vamos primeiro selecionar a solução, “Solution (TestDrive)” e eu vou clicar ferramentas, “Tools”, vamos em “NuGet Package Manager” e vou selecionar o console, “Package Manager Console”.

[05:12] No console eu vou digitar o comando que vai fazer a instalação do pacote da SQLite em todos os projetos da minha solução, como ele é um pacote muito compacto qualquer tipo de projeto pode suportar ele.

[05:31] Vamos rodar o comando de instalação, que é o “install package”, “install-package” e vamos colocar o nome do pacote do “SQLite”, agora, não podemos colocar qualquer SQLite, vamos colocar o SQLite que é a forma mais compacta de todas, o pacote mais compacto de todos que é o que dá o suporte aqui para o nosso projeto Portable.

[05:58] O projeto Portable é muito mais limitado do que os outros tipos de projeto. No Xamarin Forms ele existe para ser um coringa, que pode ser utilizado em qualquer situação, em qualquer outro tipo de projeto.

[06:17] Como ele é um projeto mais limitado, temos que utilizar a versão do SQLite que é mais adequada para o PCL, para o Portable, vamos colocar aqui “sqlite-net-pcl”, o que significa “pcl”? É Portable Class Library, ou seja, biblioteca de classes portáteis, ou portáveis, porque é a biblioteca mais compacta que existe para desenvolvimento com o Xamarin.

[06:54] Com isso vamos instalar o pacote do “sqlite-net-pcl” em todos os projetos da nossa solução, agora está instalando “sqlite-net-pcl”. Terminou de instalar, agora já temos o SQLite disponível para ser utilizado nas nossas soluções.

[07:14] Vamos começar agora a desenvolver uma aplicação baseada em banco de dados relacional, para poder fazer a persistência dos dados no aparelho, no nosso dispositivo, que no caso é o Android.