

 13

O que aprendemos?

Nessa aula você:

- implementou a persistência do AluraCar usando a API *IndexedDB*
- aprendeu que o Ionic possui um serviço chamado *Storage* para lidar com persistência e configurou esse serviço no módulo principal da aplicação
- modificou o modelo de agendamentos para incluir os atributos `confirmado` e `enviado`
- alterou o estado a propriedade `enviado` para `true` sempre que o agendamento for enviado para a API no `agendamentos-service`
- salvou o agendamento na nossa base, através de uma nova propriedade para usar o serviço de Storage do Ionic, convertendo a *promise* retornada por padrão do serviço de Storage para um `Observable`
- concatenou o envio do agendamento com o salvamento, compondo seus observables do RxJs através do `mergeMap`
- tratou os erros de agendamento usando o operador `catch` do RxJs finalizar a persistência
- encapsulou a lógica de persistência para permitir que ela seja utilizada em outras partes da aplicação usando o padrão **DAO** (Data Access Object)
- evitou a sobreescrita de agendamentos repetidos, consultando a base de dados com o método `ehDuplicado`
- concluiu o primeiro módulo do curso! Parabéns!!