

Desenvolvendo para iPhone e iPad

4ª Edição

Ricardo R. Lecheta

Copyright © 2012, 2013, 2014, 2016 Novatec Editora Ltda.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998.

É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Revisão gramatical: Priscila A. Yoshimatsu

Editoração eletrônica: Carolina Kuwabata

Capa: Luciane Mori/Carolina Kuwabata

ISBN: 978-85-7522-480-9

Histórico de impressões:

Março/2016 Quarta edição

Novembro/2014 Terceira edição (ISBN: 978-85-7522-401-4)

Dezembro/2013 Segunda edição (ISBN: 978-85-7522-384-0)

Março/2013 Primeira reimpressão

Novembro/2012 Primeira edição (ISBN: 978-85-7522-303-1)

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

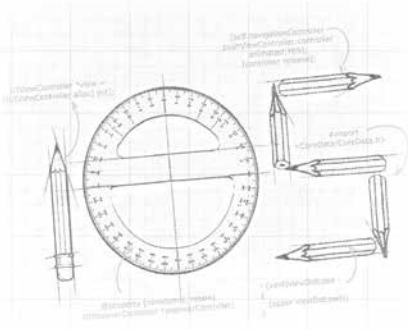
E-mail: novatec@novatec.com.br

Site: novatec.com.br

Twitter: twitter.com/novateceditora

Facebook: facebook.com/novatec

LinkedIn: linkedin.com/in/novatec



CAPÍTULO 1

Introdução

1.1 Introdução

O sistema operacional iOS da Apple se destaca como uma das principais plataformas de desenvolvimento mobile do mercado, pois é uma das preferidas pelos usuários e desenvolvedores. Uma característica do iOS é sua única e exclusiva integração com o hardware da Apple, famosa por desenvolver bons conjuntos de hardware, o que garante um ótimo desempenho geral do sistema.

A primeira versão do iPhone foi lançada em 2007, durante o WWDC (Apple Worldwide Developers Conference), que é o evento para desenvolvedores organizado pela Apple. Mas o SDK para desenvolvimento de aplicativos foi anunciado somente no ano seguinte, em 2008, junto com a abertura da App Store, a loja que facilita o processo de distribuição de aplicativos pelo mundo.

Já no primeiro ano, o iPhone impressionou com sua tela multitouch e com ótima sensibilidade, e também pela forma como todos os seus recursos eram integrados, como acelerômetro, câmera, GPS, internet, serviços e muito mais, e desde o seu lançamento até os dias atuais ele tem sido alvo das atenções da mídia, de desenvolvedores e empresas do mundo inteiro.

Contudo o que realmente impulsionou o desenvolvimento de aplicativos foi a App Store, mais uma grande ideia de Steve Jobs, permitindo que aplicativos fossem distribuídos e instalados facilmente em qualquer aparelho, e ainda com uma proposta de remuneração aos desenvolvedores, o que causou uma grande revolução no mercado.

Steven Paul Jobs nasceu em São Francisco, na Califórnia, em 24 de fevereiro de 1955, e faleceu em 5 de outubro de 2011. Ele foi um inventor e um verdadeiro artista da tecnologia, sempre revolucionando e nunca desistindo de seus sonhos e objetivos. Suas criações influenciaram a vida de pessoas no mundo inteiro, e a Apple Inc, criada por ele, se tornou a empresa mais valiosa do mercado. Se você está lendo este livro, provavelmente deve conhecer sua história.

Se você está lendo este livro deve conhecer o iPhone, o iPad e a Apple TV, assim como as criações de Steve Jobs. O que podemos fazer agora é estudar a tecnologia que ele criou e desenvolver aplicativos que serão utilizados por pessoas no mundo inteiro, desde utilitários, jogos, aplicativos financeiros, de entretenimento, esportes, notícias, redes sociais, saúde, educação e muitos outros. Seja qual for o objetivo de seu aplicativo, defina como meta um ótimo design, pois, segundo Jobs, esse é um dos aspectos mais importante de um software.

1.2 A linguagem Objective-C

A linguagem de programação utilizada para o desenvolvimento de aplicativos para o iOS é o Objective-C, que foi criada com base na sintaxe da linguagem C ANSI e nos conceitos do Smalltalk, uma das primeiras linguagens orientadas a objetos, e, portanto, possui todos os recursos para usufruir deste paradigma.

Além disso, o Objective-C traz recursos encontrados em linguagens que apresentam um paradigma funcional, como, por exemplo, passar funções como parâmetro para funções; tudo isso vamos estudar no livro.

A linguagem Objective-C é popularmente chamada de Objective C ou apenas ObjC / Obj-C, e sua sintaxe é baseada na linguagem C ANSI.

No capítulo 2, vamos estudar a história dessa linguagem e aprender sua sintaxe utilizando muitos exemplos de código, desde como declarar variáveis, imprimir mensagens de logs, criar classes e objetos, atributos, métodos, interfaces (protocolos), herança, até descobriremos como funciona o gerenciamento de memória, conhecido como contador de referências.

1.3 A linguagem Swift

No evento WWDC 2014 (Apple Worldwide Developers Conference) foi lançada a linguagem Swift, com uma sintaxe simples e moderna.

O objetivo do Swift é deixar o desenvolvimento de aplicativos mais produtivo, e por ser mais simples também facilita o aprendizado de quem está começando.

Depois de estudarmos o Objective-C e o básico do desenvolvimento para iOS, vamos estudar o Swift e continuar a leitura do livro com ele. Você vai desenvolver um projeto passo a passo que evoluirá a cada capítulo em Swift.

A sintaxe da linguagem ficou muito mais simples, e conseqüentemente o desenvolvimento ficou mais produtivo.

1.4 Frameworks Foundation e Cocoa Touch

Embora a linguagem de programação utilizada para desenvolver para iOS seja o Objective-C, foram criados alguns frameworks para deixar o desenvolvimento mais produtivo. O primeiro deles é o *Foundation*, que contém um conjunto de classes escritas em Objective-C para tornar o desenvolvimento de aplicações mais simples e rápido. Algumas dessas classes são `NSObject`, `NSString`, `NSDate`, `NSNumber`, `NSArray`, `NSDictionary`, `NSURL`, e muitas outras, e basicamente elas facilitam o trabalho do dia a dia.

Em cima desse conjunto de classes foi criado o framework *Cocoa Touch*, que é bem integrado ao ambiente de desenvolvimento do Xcode e facilita a utilização de recursos de multimídia, networking, persistência, animações e muitos outros, escondendo a complexidade desses assuntos do desenvolvedor.

O *Cocoa Touch* é baseado no padrão MVC (Model View Controller), e nele a camada de View é criada utilizando-se o Interface Builder, que permite desenhar a tela no editor visual. A camada de controller são as classes filhas de `UIViewController`, e basicamente são elas que vão definir o ciclo de vida das telas, tratar os eventos do usuário, controlar a navegação do aplicativo e, é claro, interagir com a camada Model, que contém as classes e os objetos responsáveis pela lógica de negócios da aplicação. A figura 1.1 exibe um diagrama do modelo MVC utilizado pelo *Cocoa Touch*.

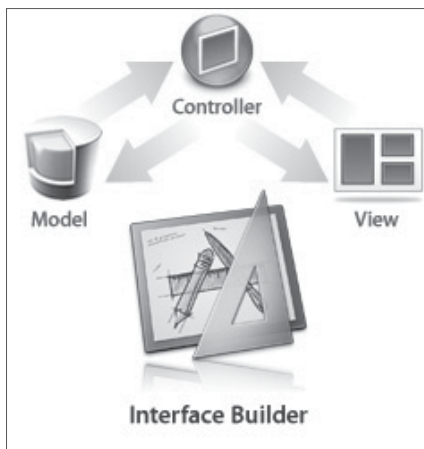


Figura 1.1 – Framework Cocoa Touch é MVC.

1.5 Xcode

O Xcode é o IDE (ambiente de desenvolvimento) oficial da Apple, e vamos utilizá-lo durante todo o livro para desenvolver os projetos. Para instalar o Xcode, acesse a sua página oficial em <https://developer.apple.com/xcode/>, que podemos visualizar na figura 1.2.

Figura 1.2 – Site do Xcode.

Para prosseguir com a instalação, clique no botão View in Mac App Store, que vai abrir o Xcode na loja da Apple diretamente de seu Mac, o que vai permitir que você faça o download dessa incrível ferramenta de desenvolvimento.

Uma das vantagens de se baixar os softwares pela Mac App Store é a facilidade de atualização, pois você receberá notificações sempre que existir uma nova versão disponível.

A figura 1.3 exibe a página da Mac App Store, com o Xcode disponível para download. Como no meu caso a ferramenta já está instalada, apareceu no canto esquerdo um botão atualizar, mas se for a primeira vez que você está acessando a loja, será exibida a opção para baixar e instalar o software.



Figura 1.3 – Xcode na Apple App Store.

Depois de instalar, você pode executar o Xcode normalmente.

Lembre-se de que para instalar o Xcode é necessário ter um computador com Mac OS.

A figura 1.4 exibe a tela de abertura do Xcode, que permite criar novos projetos ou abrir os mais recentes que estão em desenvolvimento.

O botão **Get started with a playground** cria um Playground, que é um recurso novo do Xcode para escrever um código Swift e obter o resultado rapidamente. E o botão **Create a new Xcode project** vai criar um novo projeto.



Figura 14 – Tela inicial do Xcode.

1.6 Criando um projeto no Xcode

Para validar a instalação do Xcode é interessante criar um projeto simples e executá-lo no simulador do iPhone ou iPad. Como existem templates para a criação de projetos, podemos simplesmente escolher um para validar a instalação do ambiente de desenvolvimento.

Então, mãos à obra! Na tela inicial do Xcode, clique no link **Create a new Xcode project** ou acesse o menu **File > New > Project > iOS > Application** para abrir o wizard. Depois selecione o template **Tabbed Application** conforme a figura 1.5, que vai criar um projeto com o template de navegação por tabs, que é um padrão de interface muito utilizado no iPhone.

Na próxima tela, digite o nome do projeto, como, por exemplo, *HelloTabs*, e selecione no combo **Devices** a opção **Universal**, que indica que você está criando um projeto compatível com iPhone e iPad ao mesmo tempo, conforme a figura 1.6.

Feito isso, prossiga com o wizard e escolha um diretório para salvar o projeto. Quando o wizard finalizar, o projeto será automaticamente aberto no Xcode.

Nos próximos capítulos, vamos verificar em detalhes os arquivos gerados pelo wizard e toda a estrutura do projeto, mas neste momento nosso objetivo é simplesmente executá-lo no simulador do iPhone para validar a instalação do ambiente de desenvolvimento.

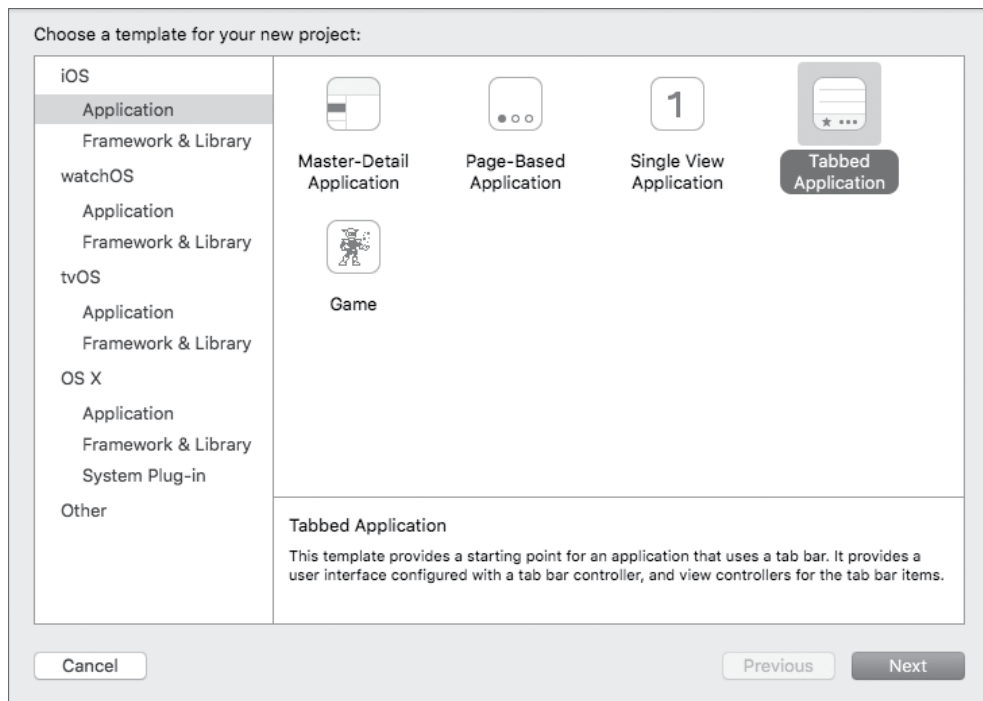


Figura 1.5 – Wizard para criar projeto.

Ao criar um projeto no Xcode, você pode optar pelas linguagens Objective-C ou Swift. Neste livro você vai aprender as duas. O Objective-C é importante pelo legado e para entender o código de projetos ou bibliotecas já existentes, e o Swift é a linguagem recomendada para criar um projeto novo.

Para começar a brincadeira, vamos executar o projeto no simulador do iPhone e conferir o resultado, conforme demonstrado na figura 1.7.

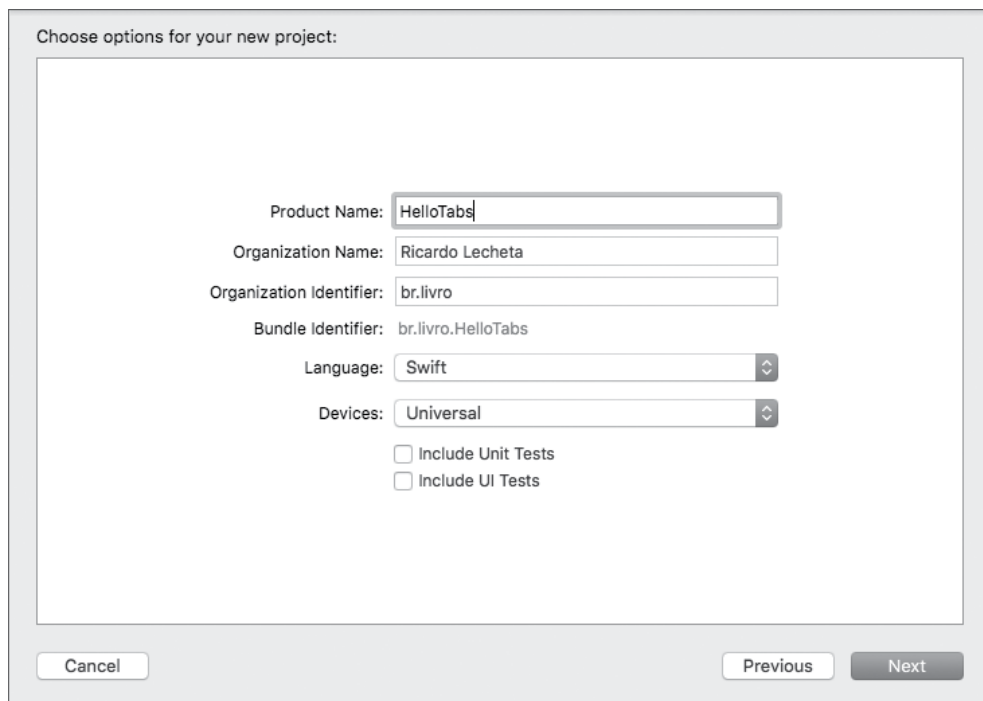


Figura 1.6 – Criando o projeto.

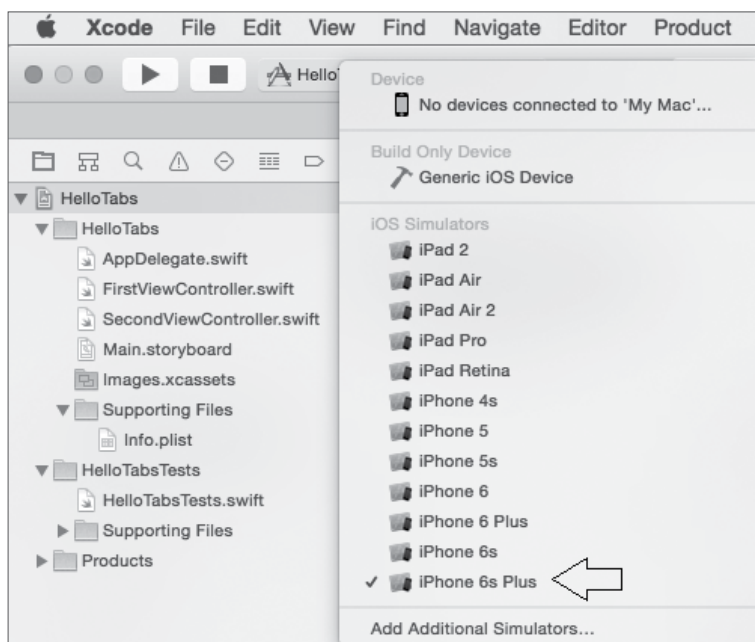


Figura 1.7 – Selecionando o simulador do iPhone.

Logo em seguida, clique no botão **Run** para executar o projeto. Se preferir, pode utilizar a tecla de atalho **command+R** para executar o simulador selecionado automaticamente.

A figura 1.8 exibe o projeto executando no simulador do iPhone.

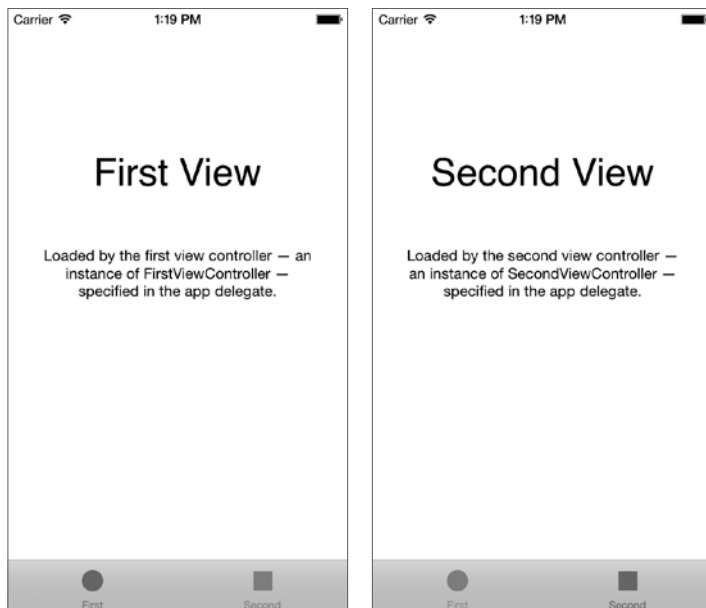


Figura 1.8 – Simulador do iPhone com as tabs.

O wizard **Tabbed Application** criou um projeto com navegação por Tabs, e o interessante é que este projeto é compatível com iPhone e iPad, pois marcamos no wizard o campo **Target** como Universal.

Portanto execute o projeto também no simulador do iPad, para validarmos que a instalação do Xcode está funcionando perfeitamente. Selecione o iPad na lista de simuladores, conforme a figura 1.9.

Feito isso, execute o projeto com a tecla de atalho **command+R**. A figura 1.10 exibe o resultado no simulador do iPad.

Uma informação bem importante que lhe pode ser útil são algumas teclas de atalho que você pode utilizar para ajustar o simulador do iPad, para que ele não fique tão grande na tela de seu computador. Utilize as teclas **command+1**, **command+2** e **command+3** para ajustar o tamanho do simulador, teste e veja a diferença.

Outra dica interessante é usar a tecla **command+seta para esquerda** e **command+seta para direita** que utilizamos para rotacionar o simulador, para visualizar a tela na vertical e horizontal.

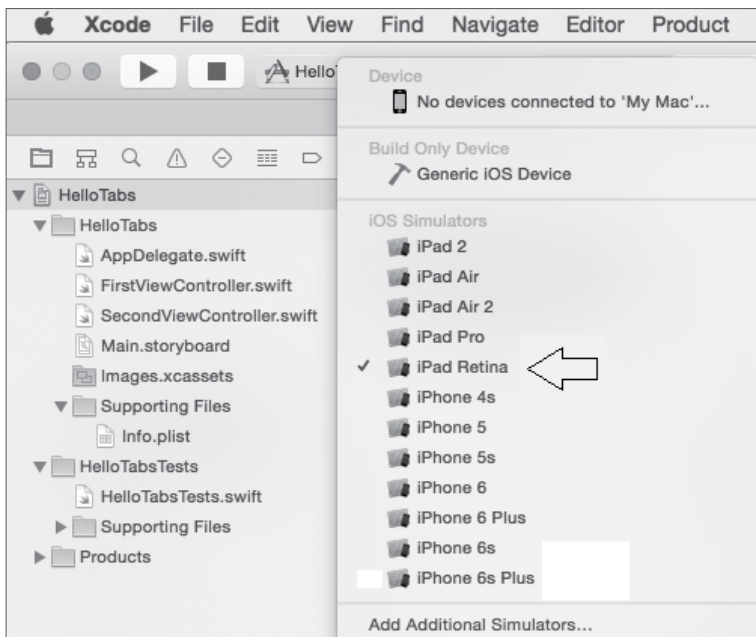


Figura 1.9 – Selecionando o simulador do iPad.

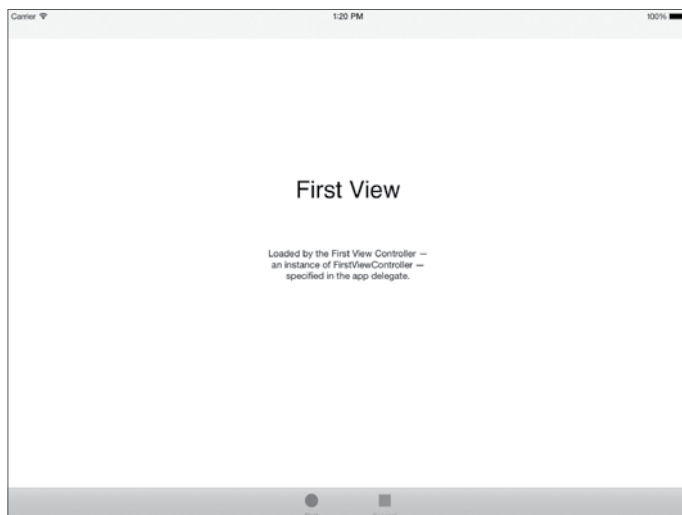


Figura 1.10 – Simulador do iPad.

Nos próximos capítulos, vamos estudar as linguagens Objective-C e Swift, assim como os detalhes sobre o desenvolvimento de aplicativos para a plataforma iOS.

1.7 Projeto dos carros

Os primeiros capítulos do livro vão lhe ensinar as linguagens Objective-C e Swift e os conceitos principais do desenvolvimento para iOS, sempre com exemplos simples, mas com o objetivo de fornecer uma boa base para prosseguir com a leitura do livro.

Antes de iniciar o projeto dos carros, você já terá aprendido a linguagem de programação e os principais conceitos sobre o desenvolvimento de aplicativos para iOS, como telas e formulários, alguns componentes visuais e várias maneiras de navegar de uma tela para outra, dependendo da sua necessidade.

Depois de aprender os conceitos fundamentais, você estará apto a iniciar o projeto, que vamos desenvolver juntos, passo a passo, durante todo o livro. Este projeto possui muitas características encontradas em aplicativos comerciais, como diversos componentes visuais, controle de telas na vertical e horizontal, integração com web services, XML e JSON, banco de dados e arquivos, recursos de multimídia, mapas e GPS e muitos outros. Eu adotei esse projeto como forma de ensino porque acredito que a melhor maneira de aprender é aprender fazendo.

É muito mais divertido do que fazer um projeto hello world em cada capítulo, pois aqui a coisa vai ficar bem mais séria e diversos problemas surgirão durante o desenvolvimento do projeto, e nossa finalidade é discutir sempre boas práticas e técnicas de design para solucionar problemas comuns durante a construção de um aplicativo.

A figura 1.11 exibe algumas telas do aplicativo executando no simulador do iPhone.



Figura 1.11 – Aplicativo dos carros para iPhone.

Muitos temas interessantes foram incorporados no projeto, como a parte de mapas e multimídia. Por exemplo, na tela de detalhes do carro, podemos escolher para visualizar no mapa o local da fábrica do carro, conforme a figura 1.12.



Figura 1.12 – Mapa no modo satélite.

Às vezes, alguns conceitos ficam fora do escopo do projeto dos carros, mas sempre que isso acontece um projeto separado é criado para explicar toda a teoria necessária. Por exemplo, neste capítulo dos mapas também é explicado detalhadamente como monitorar o GPS do aparelho.

Na tela de detalhes do carro, também podemos assistir ao vídeo de cada modelo, conforme está na figura 1.13. Este capítulo de multimídia é outro que aguarda muitas surpresas além do projeto, como reprodução e gravação de áudio, acesso à galeria de fotos do iPhone e como utilizar a câmera pelo aplicativo para tirar fotos.

Depois de concluir a versão para iPhone, vamos estudar os principais componentes utilizados para iPad e deixar universal o aplicativo dos carros para funcionar no iPhone e no iPad, conforme mostra a figura 1.14.

Vale ressaltar que todos os dados necessários para o aplicativo funcionar, como as informações dos carros, as fotos e os vídeos, serão retornados do servidor por meio de um web service.

```
// Carros clássicos (XML e JSON)
http://www.livroiphone.com.br/carros/carros_classicos.xml
http://www.livroiphone.com.br/carros/carros_classicos.json
```

```
// Carros esportivos (XML e JSON)
http://www.livroiphone.com.br/carros/carros_esportivos.xml
http://www.livroiphone.com.br/carros/carros_esportivos.json

// Carros de luxo (XML e JSON)
http://www.livroiphone.com.br/carros/carros_luxo.xml
http://www.livroiphone.com.br/carros/carros_luxo.json
```

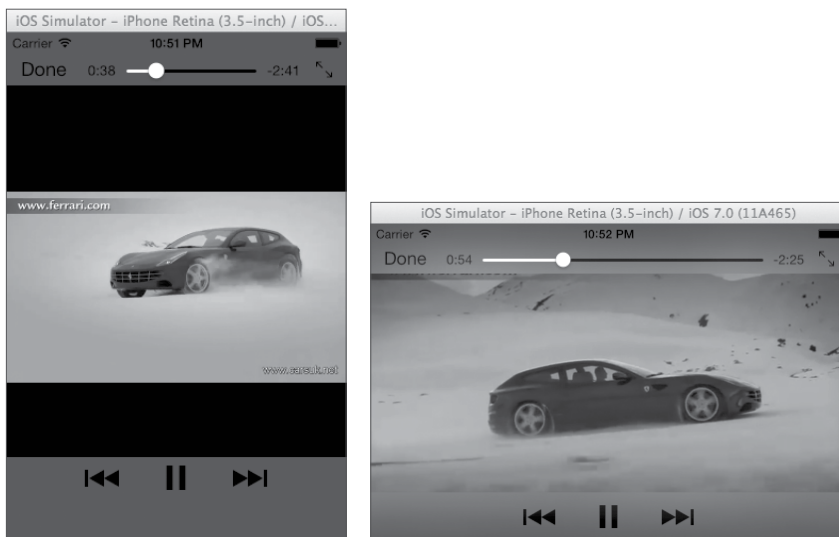


Figura 1.13 – Vídeo de cada carro.



Figura 1.14 – Aplicativo dos carros para iPad.

Lembre-se de que durante o livro também vamos abordar vários temas com exemplos simples e práticos, os quais não estão incluídos no projeto, como por exemplo sensores de acelerômetro e bússola, GPS, câmera e fotos, áudio, reconhecimento de gestos, frameworks de desenho, views avançadas, games com SpriteKit etc.

Mas, antes de iniciar este projeto, temos muito que aprender; então nós daremos um passo de cada vez. No próximo capítulo, vamos estudar os fundamentos da linguagem Objective-C, e logo depois vamos aprender a linguagem Swift.

1.8 Links úteis

No próximo capítulo, vamos estudar a linguagem Objective-C, e a partir de agora, durante sua vida como desenvolvedor de aplicativos para iOS, você precisa ter alguns links em seus favoritos, que vão lhe auxiliar durante a sua jornada.

Seguem algumas páginas que você precisa ter sempre em mãos:

- **iOS Dev Center** – Neste portal, você vai encontrar diversas informações sobre o desenvolvimento de aplicativos.

<https://developer.apple.com/devcenter/ios/>

- **Documentação das classes do iOS** – Nesta página, podemos consultar a documentação das classes do iOS e também diversos exemplos sobre os mais variados temas.

<https://developer.apple.com/library/ios/navigation/>

- **Página do WWDC** – O WWDC (Apple Worldwide Developers Conference) é o evento oficial para desenvolvedores, organizado anualmente pela Apple, e sempre traz as novas tendências, tecnologias e as novidades e os lançamentos sobre Mac OS e iOS.

<https://developer.apple.com/wwdc/>

- **Página oficial da linguagem Swift**

<https://developer.apple.com/swift/>

- **Página do livro** – Visite a página do livro para obter o código-fonte dos exemplos e mais detalhes sobre esta e outras obras.

<http://www.livroiphone.com.br>