



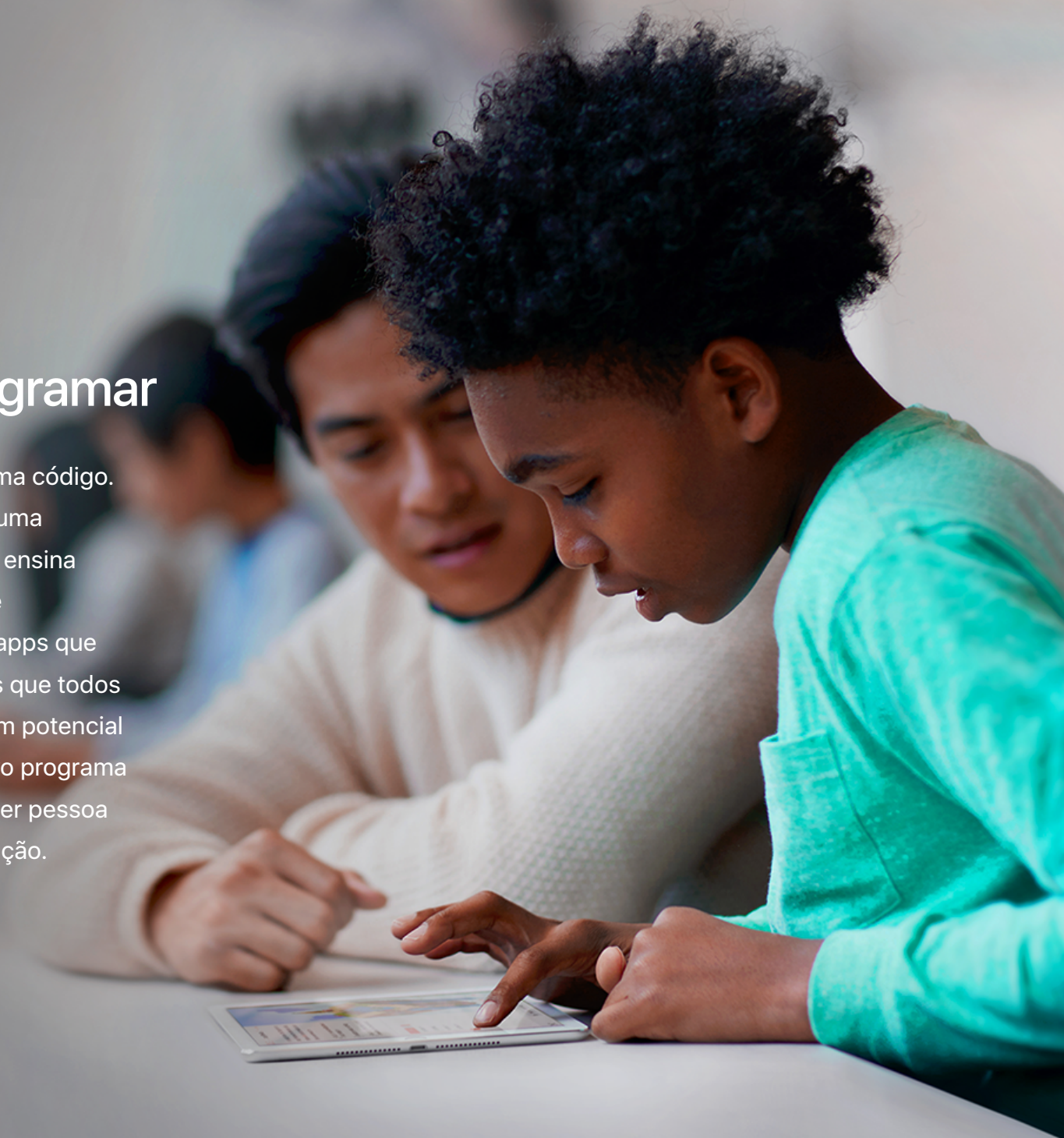
Guia do curso: Introdução ao Desenvolvimento de Apps com Swift

Setembro de 2017



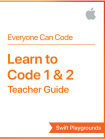
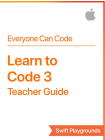
Todo mundo pode programar

A tecnologia tem uma linguagem. Ela se chama código. Acreditamos que dominar essa linguagem é uma habilidade muito importante. A programação ensina a resolver problemas e a trabalhar em equipe de maneiras criativas. Também ajuda a criar apps que colocam suas ideias em prática. Acreditamos que todos deveriam ter a oportunidade de criar algo com potencial de mudar o mundo. Por isso, criamos um novo programa com recursos e ferramentas para que qualquer pessoa possa aprender, formular e ensinar programação.



Currículo do programa Todo mundo pode programar

O programa Todo mundo pode programar conta com diversos recursos que habilitam os alunos a desenvolver seus primeiros apps, mesmo sem nenhuma experiência em programação. A tabela abaixo apresenta uma visão geral de todos os recursos de ensino e aprendizagem gratuitos disponíveis.

Currículo	Dispositivo	Público	Aplicativo	Pré-requisitos	Visão geral	Materiais de aprendizagem	Recursos de apoio	Horas-aula incluídas
 Get Started with Code 1 Teacher Guide		Do jardim de infância ao 2º ano		Nenhum	Usar apps visuais para explorar conceitos de programação de forma prática e começar a pensar como um programador.	- Aulas do app codeSpark Academy - Curso Tynker Space Cadet	Comece a Programar 1: Guia do professor	30 horas, incluindo o Guia do professor e as aulas dos apps
 Get Started with Code 2 Teacher Guide		Do 3º ao 5º ano		Nenhum	Explorar os conceitos básicos da programação e usar apps visuais para praticar o pensamento da programação.	Curso Tynker Dragon Spells	Comece a Programar 2: Guia do professor	36 horas, incluindo o Guia do professor e as aulas dos apps
 Learn to Code 1 & 2 Teacher Guide		A partir do ensino fundamental II		Nenhum	Aprender os conceitos básicos de programação usando código Swift real.	- App Swift Playgrounds - Aulas dos cursos Aprenda a Programar 1 e 2 - Curso do iTunes U	Aprenda a Programar 1 e 2: Guia do professor	Até 85 horas, incluindo o Guia do professor e as aulas dos cursos Aprenda a Programar 1 e 2
 Learn to Code 3 Teacher Guide		A partir do ensino fundamental II		Aprenda a Programar 1 e 2	Expandir as habilidades de programação e começar a pensar como um desenvolvedor de apps.	- App Swift Playgrounds - Aulas do curso Aprenda a Programar 3	Aprenda a Programar 3: Guia do professor	Até 45 horas, incluindo o Guia do professor e as aulas dos cursos Aprenda a Programar 3
 Intro to App Development with Swift Teacher Guide		Ensino médio e superior		Nenhum	Ganhar experiência prática com as ferramentas, técnicas e conceitos necessários para criar um app do iOS do zero.	Livro Introdução ao Desenvolvimento de Apps com Swift e arquivos de projeto	Introdução ao Desenvolvimento de Apps com Swift: Guia do professor	90 horas
 App Development with Swift Teacher Guide		Ensino médio e superior		Nenhum	Aprender conceitos básicos sobre Swift, UIKit e redes com laboratórios práticos e projetos orientados. Até o final do curso, os alunos serão capazes de criar um app idealizado por eles.	Livro Desenvolvimento de Apps com Swift e arquivos de projeto	Desenvolvimento de Apps com Swift: Guia do professor	180 horas

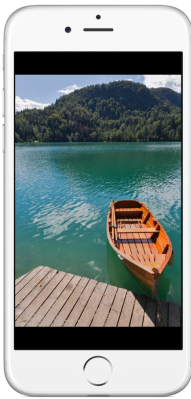
Visão geral

Os cursos Introdução ao Desenvolvimento de Apps com Swift e Desenvolvimento de Apps com Swift foram criados para ensinar alunos do ensino médio e superior com pouca ou nenhuma experiência em programação a desenvolver apps e colocar suas ideias em prática.

O curso Introdução ao Desenvolvimento de Apps com Swift apresenta aos alunos o mundo do desenvolvimento de apps e os conceitos básicos da linguagem Swift e do app Xcode. No projeto final do curso, os alunos podem escolher entre duas opções de apps básicos para iOS para desenvolver.

Já no curso Desenvolvimento de Apps com Swift, os alunos vão mais longe, tanto os que são novos no assunto quanto os que querem expandir suas habilidades. Quem já conhece a linguagem Swift, o app Xcode e sabe programar para iOS pode avançar pelas aulas com mais rapidez ou ir direto para os laboratórios para criar miniprojetos e testar os códigos em playgrounds. Até o final do curso, os alunos serão capazes de criar um app totalmente funcional idealizado por eles.

First App



5.1 New Project

Now that you're getting more comfortable with playgrounds, you might be wondering how to build an app you can use on your iOS device, or even your Apple Watch. A lot of moving parts need to work together to make an app run, and Xcode is the best tool for putting them all together.

In this three-part lesson, you'll build SinglePhoto—a simple iOS app that displays a single photo. In the first exercise, you'll create an app project from scratch. Then you'll use Xcode to explore your project and learn to navigate your coding environment.

You can customize every part of your app—from its icon on the Home screen to the way its buttons behave. There are panels and controls in Xcode that display the many options available to you. You'll practice using the Xcode Interface Builder to continue customizing your first app.

In the final step, you'll add an image to your project and edit the user interface. You'll also get an introduction to Interface Builder—a powerful component of Xcode where you create the user interface of your app. At the end of the second exercise, your app will look like this, but it will display a photo of your own choosing.

Lesson 5.1 | First App: New Project

25

Lesson 1.8

Interface Builder Basics

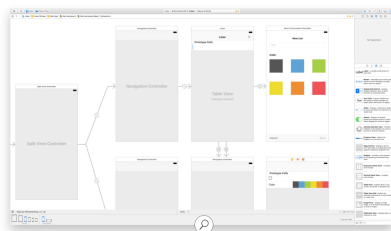
The best way to learn the basics of Interface Builder is to dive into Xcode and explore some of its features. Start by creating a new iOS project using the Single View Application template. Name the project "IBBasics".

STORYBOARDS

Interface Builder opens whenever you select an XIB file (.xib) or a storyboard file (.storyboard) from the project navigator.

An XIB file contains the user interface for a single visual element, such as a full-screen view, a table view cell, or a custom UI control. XIBs were used more heavily before the introduction of storyboards. They're still a useful format in certain situations, but this lesson will focus on storyboards.

In contrast with an XIB, a storyboard file includes many pieces of the interface, defining the layout of one or many screens as well as the progression from one screen to another. As a developer, you'll find that the ability to see multiple screens at once will help you understand the flow within your app.

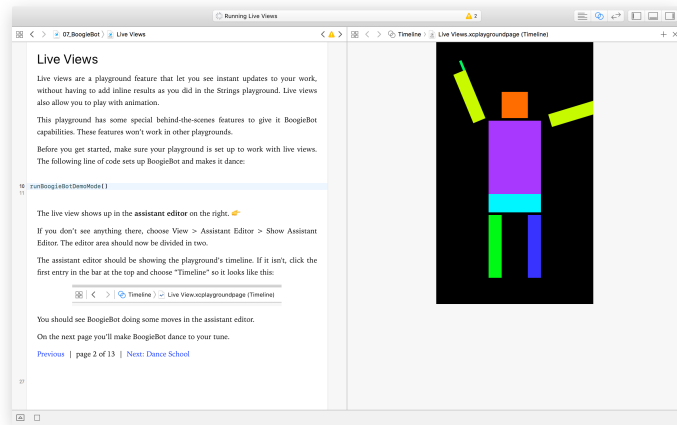


78

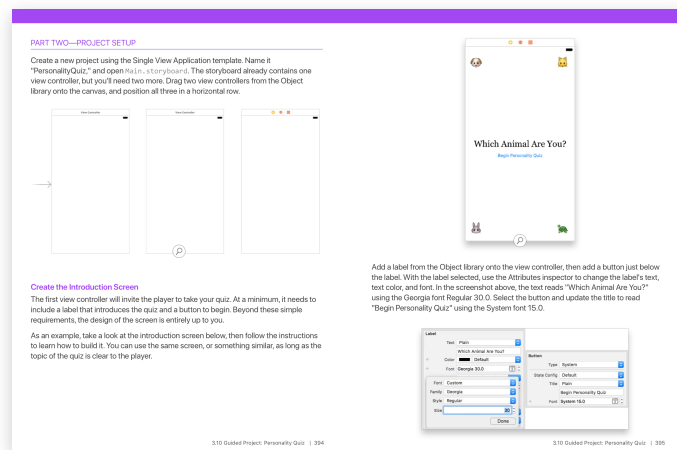
1.8 Interface Builder Basics | 79

Principais características

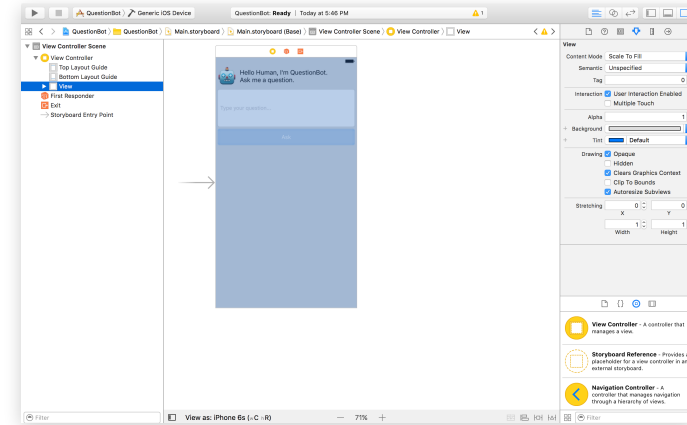
Playgrounds. Os alunos aprendem os conceitos de programação à medida que formulam códigos nos playgrounds, um ambiente de programação interativo que permite fazer experiências com os códigos e ver os resultados imediatamente.



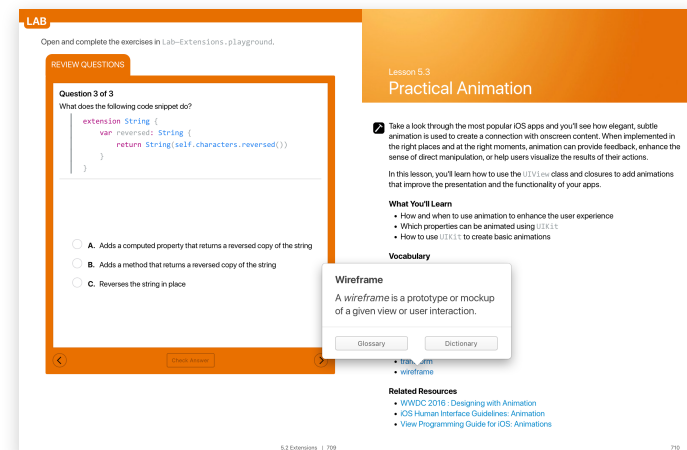
Instruções passo a passo. Os alunos são orientados por todas as etapas de criação de um app no Xcode, com instruções detalhadas, imagens e vídeos.



Projetos de amostra. Usando os arquivos do projeto que acompanham o material, os alunos podem testar partes do código sem precisar criar um app inteiro do zero.



Ferramentas de estudo. Perguntas de revisão, vocabulário principal, links para documentos e outros recursos ajudam os alunos a confirmar se entenderam e a aplicar o que aprenderam.



Estrutura do curso

Introdução ao Desenvolvimento de Apps com Swift

Este curso introdutório de um semestre foi criado para oferecer aos alunos uma base sólida sobre programação com linguagem Swift. Os alunos ganham experiência prática com ferramentas, técnicas e conceitos necessários para criar um app simples para iOS.

As aulas de design de apps orientam os alunos pelo processo de desenvolvimento de um app, incluindo troca de ideias, planejamento, criação de um protótipo e avaliação de um app criado por eles próprios. Mesmo que eles ainda não tenham as habilidades necessárias para criar o app, o trabalho realizado na criação da estrutura servirá como preparação para que continuem se desenvolvendo.

Aula 1 — Noções básicas do playground. Os alunos se familiarizam com o ambiente interativo do playground.

Aula 2 — Nomenclatura e identificadores. Os alunos exploram as noções básicas da solução de problemas usando nomenclatura e identificadores adequados.

Aula 3 — Strings. Os alunos são apresentados ao conceito de strings e interpolação de strings.

Aula 4 — Olá, mundo! Os alunos são apresentados à tradição da programação e aprendem a personalizar o ambiente Xcode e a depurar.

Aula 5 — Primeiro app. Os alunos criam seu primeiro app usando o Xcode e exibem o trabalho em um simulador do iOS.

Aula 6 — Funções. Os alunos aprendem por que as funções são tão poderosas ao combinarem as etapas detalhadas em uma definição que podem usar repetidamente.

Aula 7 — BoogieBot. Os alunos colocam em prática o que aprenderam sobre funções controlando um robô animado que dança no playground.

Aula 8 — Constantes e variáveis. Os alunos expandem a compreensão da nomenclatura ao serem formalmente apresentados aos conceitos de constantes e variáveis.

Aula 9 — Tipos. Os alunos se aprofundam mais nos fundamentos da linguagem Swift ao aprenderem sobre o sistema de tipos, desde a biblioteca padrão em Swift até os tipos personalizados.

Aula 10 — Parâmetros e resultados. Os alunos aprofundam seus conhecimentos em funções ao aprenderem sobre parâmetros e valores de retorno, que tornam as funções mais flexíveis e eficazes.

Aula 11 — Tomada de decisões. Os alunos aprendem a tomar decisões no código usando as instruções condicionais “if”/“else”, valores booleanos “true” e “false” e operadores de comparação.

APPLY AND EXTEND (25 minutes)	REVIEW AND DISCUSS (10 to 20 minutes)
In the lesson, we used the analogy of an artist creating a painting to help students differentiate the four types of functions. Discuss the four different types of functions: ❌ Parameters, ❌ Return value <code>paintPicture()</code> Does work on its own and doesn't return a value. ✅ Parameters, ❌ Return value <code>paintPicture(width: Int, height: Int, dominantColor: UIColor)</code> Does work that changes depending on the arguments but doesn't return a value. ❌ Parameters, ✅ Return value <code>paintPicture()</code> -> Painting Doesn't require any information but does return a value. ✅ Parameters, ✅ Return value <code>paintPicture(width: Int, height: Int, dominantColor: UIColor)</code> -> Painting Accepts information and returns a value.	Review and discuss the reflection questions in the student guide. - You can now build functions that can take information in and use it in their work. But the work that's done is still the same. What if you could do different work that depends on the information that's passed in? - What processes and tasks from real life can you think of that fit into the various ways you can define a function (with or without parameters, with or without a return type)? Here are some examples: <code>func turnOffOven()</code> (Turning off the oven requires no parameters, and nothing is returned from this action.) <code>func preheatOven(temperature: Int)</code> (Preheating the oven requires a temperature as its parameter to begin the preheating process.) <code>func bakeCookies()</code> -> [Cookie] (Baking cookies requires no parameters, and would yield a confection.) <code>func bake(ingredients: [Ingredient])</code> -> [BakedGood] (General baking with the oven requires a list of ingredients, and returns a delicious batch of baked goods.)

Lesson 10 | Parameters and Results 52

O Guia do professor contém atividades de extensão adicionais, perguntas para discussão e atividades de diário sobre apps nas quais os alunos trabalharão ao longo do semestre.

Estrutura do curso (continuação)

Aula 12 — Instâncias, métodos e propriedades. Os alunos ampliam seus conhecimentos acerca dos tipos e aprendem os métodos e as propriedades que formam uma instância desse tipo.

Aula 13 — QuestionBot. Os alunos ganham experiência modificando um projeto do Xcode ao formularem uma nova lógica para um bot de app que responde a diferentes perguntas.

Aula 14 — Matrizes e loops. Os alunos aprendem a criar e a trabalhar com matrizes adicionando e removendo objetos, e aprendem como os loops “for” funcionam com cada objeto em uma matriz.

Aula 15 — Definição de estruturas. Os alunos percebem que, muitas vezes, é útil agrupar informações e funcionalidades relacionadas em um tipo personalizado.

Aula 16 — QuestionBot 2. Os alunos ampliam os conhecimentos sobre o app QuestionBot fazendo o ChatBot, um app que mostra o histórico da conversa. Eles analisam o padrão da fonte de dados e criam um objeto simples de fonte de dados que fornece informações sobre objetos do Mensagens a serem exibidos na visualização da lista de mensagens. Os alunos praticam o acréscimo de dados a uma matriz para armazenar as mensagens no objeto de fonte de dados e manter um histórico de conversa.

Aula 17 — Ações e saídas. Os alunos aprendem a criar interfaces de usuário usando o Interface Builder e a vincular os elementos dessa interface ao código com ações e saídas. Serão praticadas a criação de saídas para acessar as propriedades de uma interface de usuário e as ações para responder à interação do usuário com os botões e outros controles.

Aula 18 — Interfaces de usuário adaptativas. Os alunos aprendem um processo que se repete para criar uma interface de usuário para a menor tela de iPhone que pode ser dimensionada para todas as orientações e tamanhos de iPhone. Eles aprendem sobre layout automático, o sistema que dispõe as restrições que definem a localização e o tamanho de elementos da interface de usuário. Além disso, são usadas visualizações em pilha, um objeto especial criado para definir restrições de layout automáticas com base em ajustes mais simples e um sistema em grade. Nesse processo, os alunos desenvolvem os apps SimpleCenter, ElementQuiz e AnimalSounds.

Aula 19 — Enumerações e switch. Os alunos descobrem as enumerações ou enums (um modo de definir uma determinada lista de opções) e aprendem para que elas são usadas, como defini-las e as maneiras mais comuns de trabalhar com elas. Eles também aprendem a usar a instrução switch para executar condicionalmente um código específico com base em qualquer opção que uma enum defina.

Aula 20 — Projeto de conclusão. Os alunos realizam uma ou as duas opções de projeto de conclusão a partir do zero. A primeira opção é um jogo de Pedra/Papel/Tesoura, e a segunda é um gerador de memes. Eles reveem diversos conceitos abordados no curso e constroem a interface de usuário, os dados de modelo e os objetos do controlador que compõem o app.

Aula 21 — Design de apps. Os alunos passam por um ciclo de design, cujo foco é criar um protótipo, semelhante ao processo que os desenvolvedores profissionais de apps passam.

E agora? Os alunos explorarão uma ampla variedade de recursos para desenvolvimento de apps, da página Apple Developer aos vídeos da Conferência Mundial para Desenvolvedores da Apple (Apple Worldwide Developers Conference, WWDC) que falam sobre as estruturas e as ferramentas mais recentes para a criação de apps para todas as plataformas da Apple.

Estrutura do curso (continuação)

Desenvolvimento de Apps com Swift

Este curso de dois semestres tem 45 aulas, cada uma pensada para ensinar uma habilidade específica relacionada à linguagem Swift ou ao desenvolvimento de apps. Cada tipo de aula tem uma abordagem diferente:

- **Aulas sobre a linguagem Swift.** Enfocam conceitos específicos. Os laboratórios são apresentados em playgrounds. Assim, os alunos podem fazer experiências com os códigos e ver os resultados imediatamente. Nós fornecemos os arquivos de playground.
- **Aulas sobre desenvolvimento de apps.** Enfocam a criação de recursos específicos para apps do iOS. Normalmente, os alunos recebem orientações passo a passo para desenvolver um miniprojeto. Nos laboratórios, eles têm um novo cenário para colocar em prática o que aprenderam.

Ao final de cada uma das primeiras cinco unidades, os alunos realizam projetos orientados, incluindo a descrição dos recursos voltados para o usuário, um plano de projeto e instruções para desenvolver um app totalmente funcional. Nesses projetos, os alunos podem criar recursos de acordo com os interesses deles, realizando o tipo de trabalho que acontece no mercado de desenvolvimento de apps. Na última unidade, eles aprendem a projetar e criar o protótipo e a arquitetura de um app idealizado por eles.

O Guia do professor contém dicas para estender ou adaptar as aulas, a fim de aumentar a colaboração e apoiar os alunos que precisam de mais ajuda. Esse material também inclui apresentações do Keynote para download de cada aula, os gabaritos dos laboratórios e uma ficha para avaliar o trabalho dos alunos.

Unidade 1 — Introdução ao desenvolvimento de apps. Os alunos aprendem conceitos básicos sobre dados, operadores, fluxo de controle na linguagem Swift, documentação, depuração, Xcode, desenvolvimento e execução de apps e Interface Builder. Em seguida, eles colocam esses conhecimentos em prática no projeto orientado, criando um app simples de lanterna chamado Light.

Unidade 2 — Introdução ao UIKit. Os alunos exploram strings, funções, estruturas, coleções e loops na linguagem Swift. Eles também aprendem sobre o UIKit — as visualizações e os controles do sistema que compõem uma interface do usuário — e como exibir dados usando layout automático e visualizações em pilha. Na sequência, eles colocam esses conhecimentos em prática em um projeto orientado, criando um app de jogo de palavras chamado Apple Pie.

Unidade 3 — Navegação e fluxos de trabalho. Os alunos descobrem como criar fluxos de trabalho simples e hierarquias de navegação usando controles de navegação, de barra de abas e transições. Eles também conhecem duas ferramentas poderosas na linguagem Swift: opcionais e enumerações. Em seguida, eles colocam esses conhecimentos em prática no projeto orientado “Teste de Personalidade”, uma pesquisa personalizada que revela uma resposta divertida ao usuário.

Unidade 4 — Tabelas e persistência. Os alunos conhecem as visualizações de rolagem e em tabela e aprendem a criar telas de entrada complexas. Eles também veem como salvar e compartilhar dados com outros apps e como trabalhar com imagens na biblioteca de fotos do usuário. Em seguida, eles utilizam essas novas habilidades no projeto orientado “List”, um app para a organização de tarefas que permite adicionar, editar e apagar itens em uma interface de tabela comum. Os alunos podem personalizar o app para acompanhar qualquer tipo de informação, como uma coleção, tarefas ou playlists.

Unidade 5 — Trabalho com a web. Os alunos aprendem sobre animações, simultaneidade e como trabalhar com a web. Na sequência, eles colocam o que aprenderam em prática no projeto orientado “Restaurant”, um app de cardápio personalizável que mostra os pratos disponíveis em um restaurante e permite que o usuário faça um pedido. Esse app usa um serviço da web para que os alunos possam personalizar o cardápio com os pratos e as fotos que escolherem.

Unidade 6 — Criação de protótipos e planejamento de projetos. Os alunos aprendem a desenvolver e criar o protótipo e a arquitetura de um projeto idealizado por eles. Com tempo suficiente, eles devem ser capazes de realizar esse projeto por conta própria.

Outras informações

Baixe os recursos dos cursos Comece a Programar

- [Tynker](#)
- [codeSpark Academy](#)
- [Comece a Programar 1](#)
- [Comece a Programar 2](#)

Baixe os recursos do Swift Playgrounds

- [Aprenda a Programar 1 e 2: curso do iTunes U](#)
- [Aprenda a Programar 1 e 2: Guia do professor](#)
- [Aprenda a Programar 3: Guia do professor](#)
- [App Swift Playgrounds](#)

Baixe os guias do curso Desenvolvimento de Apps com Swift

- [Introdução ao Desenvolvimento de Apps com Swift](#)
- [Introdução ao Desenvolvimento de Apps com Swift: Guia do professor](#)
- [Desenvolvimento de Apps com Swift](#)
- [Desenvolvimento de Apps com Swift: Guia do professor](#)

Outros recursos

- Saiba mais sobre o programa [Todo mundo pode programar](#).
- Converse com outros educadores nos [Fóruns de desenvolvedores da Apple](#).

Sobre a linguagem Swift

Swift é uma linguagem de programação eficiente e intuitiva desenvolvida pela Apple para a criação de apps. Com ela, programar fica mais fácil, flexível e divertido. Além de facilitar a aprendizagem da programação, a linguagem Swift é muito poderosa. Ela foi projetada para escrever desde o programa mais simples, como “Olá, mundo”, ao software mais avançado do mundo.

Saiba mais sobre a linguagem [Swift](#).

Sobre o Xcode

Xcode é o app do Mac usado para criar vários apps para Mac e iOS. Ele tem todas as ferramentas necessárias para criar apps fantásticos. Ele pode ser obtido por download gratuitamente na Mac App Store.

Saiba mais sobre o [Xcode](#).

Alinhamento do currículo: Introdução ao Desenvolvimento de Apps com Swift

Este é o alinhamento preliminar do curso Introdução ao Desenvolvimento de Apps com Swift com o nível 3A dos Padrões de Serviços Provisórios de Ciência da Computação da Associação de Professores de Ciência da Computação (CSTA) K-12 de 2016. Depois que os novos padrões forem finalizados, o guia passará por uma análise formal de mapeamento da CSTA. O alinhamento aborda os conceitos de algoritmos e programação dentro dos Padrões de Serviços Provisórios de Ciência da Computação da CSTA K-12 de 2016.

Padrões de Ciência da Computação da CSTA K-12 Nível 2 para o 9º ao 10º ano												
Padrão da CSTA	3A-A-2-1 Design de artefatos	3A-A-2-2 Colaboração	3A-A-7-3 Licenciamento	3A-A-5-4 Responder a eventos	3A-A-5-5 Pesquisa	3A-A-5-6 Conceitos de matemática	3A-A-4-7 Hierarquia e abstração	3A-A-4-8 Desconstruir problema	3A-A-4-9 Abstração	3A-A-3-10 Criar algoritmos	3A-A-3-11 Modelagem e simulação	3A-A-6-12 Depuração
Noções básicas do Playground												●
Nomenclatura e identificadores						●		●	●	●		●
Strings						●		●	●	●	●	
Olá, mundo!								●		●		●
Primeiro app				●				●				
Funções						●		●	●	●	●	
Boogie Bot						●		●	●	●	●	
Constantes e variáveis						●		●	●	●	●	●
Tipos						●		●	●	●	●	●
Parâmetros e resultados						●		●	●	●	●	
Tomada de decisões				●		●		●	●	●	●	
Instâncias, métodos e propriedades						●		●	●	●	●	
Bot de perguntas				●		●		●	●	●	●	●
Matrizes e loops						●		●	●	●	●	
Definição de estruturas						●		●	●	●	●	
Bot de perguntas 2				●		●		●	●	●	●	
Ações e saídas				●		●		●		●		●
Interfaces de usuário adaptativas				●		●		●		●		●
Enumerações e switch						●		●	●	●	●	
Projeto de conclusão				●		●		●	●	●	●	
Design de apps	●	●		●	●							

Alinhamento do currículo: Desenvolvimento de Apps com Swift

Este é o alinhamento preliminar do curso Desenvolvimento de Apps com Swift com o nível 3A dos Padrões de Serviços Provisórios de Ciência da Computação da Associação de Professores de Ciência da Computação (CSTA) K-12 de 2016. Depois que os novos padrões forem finalizados, o guia passará por uma análise formal de mapeamento da CSTA. O alinhamento aborda os conceitos de algoritmos e programação dentro dos Padrões de Serviços Provisórios de Ciência da Computação da CSTA K-12 de 2016.

Alinhamento do curso Desenvolvimento de Apps com Swift com o nível 3A dos Padrões de Ciência da Computação da CSTA K-12 para o 9º ao 10º ano												
Padrão da CSTA	3A-A-2-1 Design de artefatos	3A-A-2-2 Colaboração	3A-A-7-3 Licenciamento	3A-A-5-4 Responder a eventos	3A-A-5-5 Pesquisa	3A-A-5-6 Conceitos de matemática	3A-A-4-7 Hierarquia e abstração	3A-A-4-8 Desconstruir problema	3A-A-4-9 Abstração	3A-A-3-10 Criar algoritmos	3A-A-3-11 Modelagem e simulação	3A-A-6-12 Depuração
Unidade 1 — Introdução ao desenvolvimento de apps	●	●	●	●	●	●		●	●	●		●
Unidade 2 — Introdução ao UIKit	●	●		●		●		●	●	●		●
Unidade 3 — Navegação e fluxos de trabalho	●	●		●	●	●		●	●	●		●
Unidade 4 — Tabelas e persistência	●	●		●		●		●	●	●		●
Unidade 5 — Trabalho com a web	●	●		●	●	●		●	●	●		●
Unidade 6 — Criação de protótipos e planejamento de projetos	●	●		●	●	●		●	●	●	●	

Os recursos estão sujeitos a mudanças. Alguns recursos podem não estar disponíveis em todas as regiões ou idiomas.

© 2017 Apple Inc. Todos os direitos reservados. Apple, o logotipo da Apple, iPhone, iTunes U, Keynote, Mac, macOS e Xcode são marcas comerciais da Apple Inc., registradas nos Estados Unidos e em outros países. Swift e Swift Playgrounds são marcas comerciais da Apple Inc. App Store é uma marca de serviço da Apple Inc., registrada nos EUA e em outros países. Os nomes de outros produtos e empresas aqui mencionados podem ser marcas comerciais de suas respectivas empresas. As especificações dos produtos estão sujeitas a alterações sem aviso prévio. Este material é fornecido somente para fins informativos; a Apple não se responsabiliza pelo seu uso. Setembro de 2017