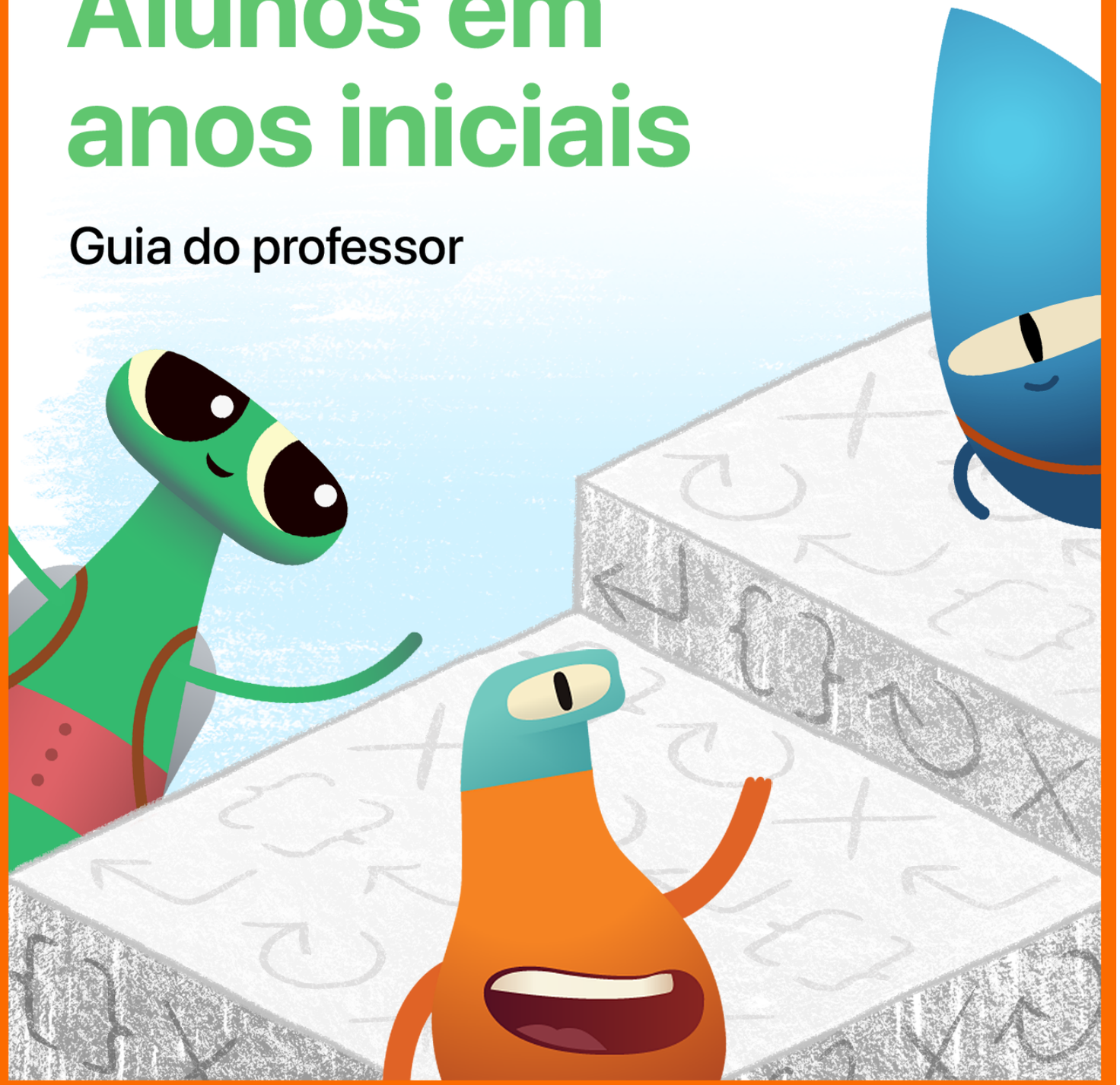
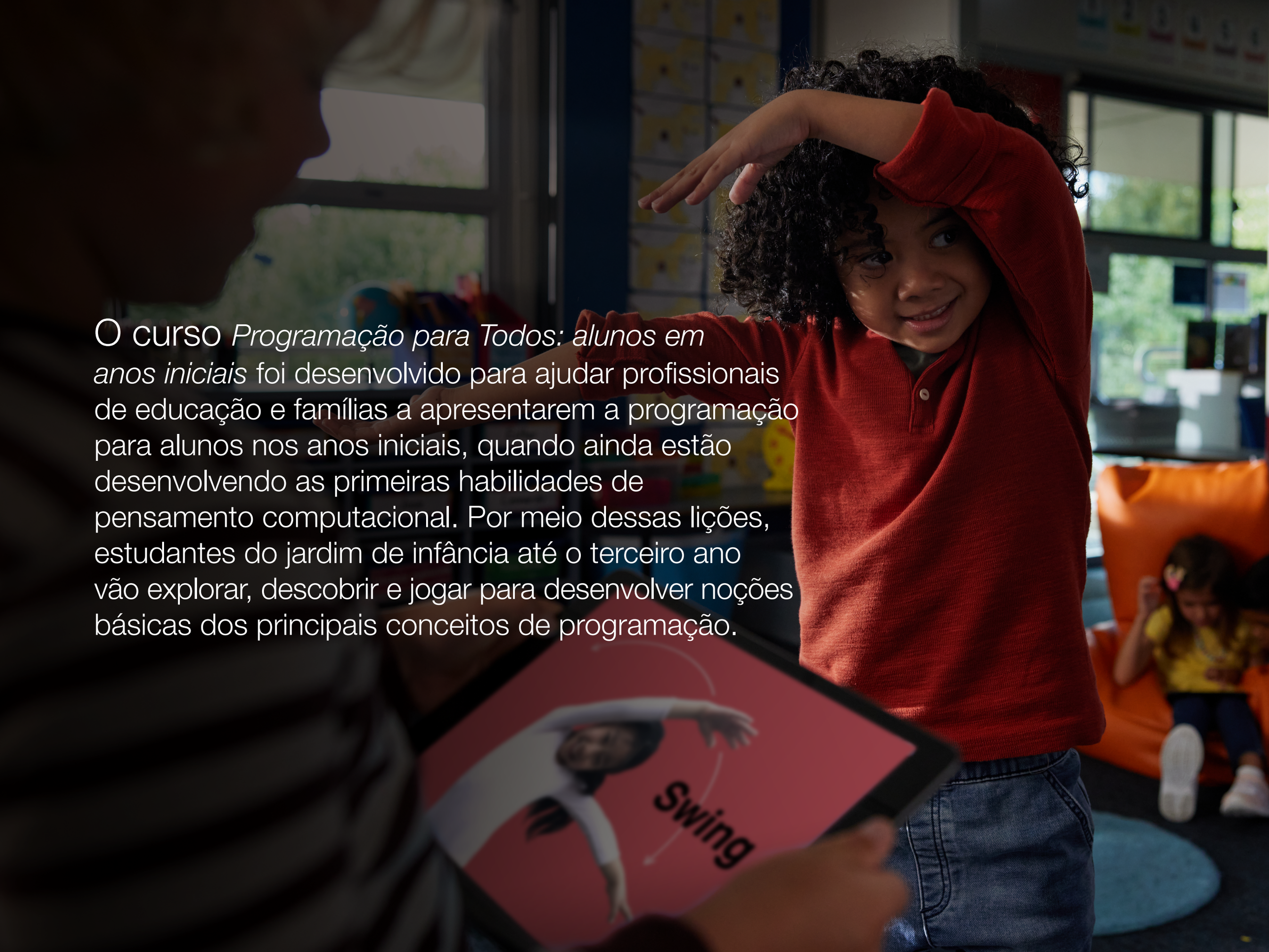
Programação para Todos

Alunos em anos iniciais

Guia do professor



Índice



O curso *Programação para Todos: alunos em anos iniciais* foi desenvolvido para ajudar profissionais de educação e famílias a apresentarem a programação para alunos nos anos iniciais, quando ainda estão desenvolvendo as primeiras habilidades de pensamento computacional. Por meio dessas lições, estudantes do jardim de infância até o terceiro ano vão explorar, descobrir e jogar para desenvolver noções básicas dos principais conceitos de programação.

Design instrucional

Este guia é dividido em quatro módulos e termina com um projeto de design de app. Cada módulo contém três lições, cada uma com foco em um conceito relacionado à programação. Em cada lição, você encontrará três atividades: explorar, descobrir e jogar. As atividades podem ser divididas em várias sessões ou dias.

Dia 1: debate e aprendizagem prática

Explorar

Apresente e debata o conceito de programação

Descobrir

Desenvolva familiaridade com o conceito por meio de atividades criativas

Aprox. 25 minutos

Jogar

- Programe com Byte no app Swift Playgrounds
- Pratique a programação em planilhas complementares e atividades no Keynote
- Leve o mundo de Byte para a realidade com jogos de programação de quebra-cabeça no chão, sem telas

Aprox. 25 minutos

Dia 2: conectando a aprendizagem à programação

Escopo e sequência

Os quatro módulos neste guia foram desenvolvidos para uso do jardim de infância até o terceiro ano. Eles podem ser realizados em qualquer ordem. Recomendamos que você use o módulo Design de Apps a qualquer momento ou até mesmo várias vezes por ano, à medida que estudantes ampliarem a compreensão de programação e apps.

Exemplo:

Ano	Módulo	Projeto final	Tempo total aproximado
Jardim de infância	Comandos	Design de Apps	4 horas
Primeiro	Funções	Design de Apps	4 horas
Segundo	Loops	Design de Apps	4 horas
Terceiro	Variáveis	Design de Apps	4 horas

Aprendizagem contínua

Para ensino do quarto ao oitavo ano, os cursos Programação para Todos: Puzzles, em conjunto com o Diário de Design de Apps e o Guia de Demonstração de Apps, oferecem mais de 45 horas de aprendizagem. Saiba mais sobre o curso Programação para Todos.

Portfólios de estudantes
(opcional)

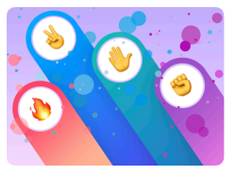
Durante esses módulos, colete itens das atividades para criar portfólios com estudantes.

Módulo	Lição	Itens sugeridos
Comandos	Rotinas diárias	• Planilha Atribuindo Comandos • Planilha Adicionando um Comando
	Ordem da história	• Imagem de ponto na trama de ordem da história • Imagem de grupo de ordem da história
	Passos de dança	• Cartões de passos de dança • Vídeo de passos de dança (opcional)
Funções	Joia de papel	• Forma de joia de papel • Planilha Compondo um Comportamento • Planilha Criando uma Nova Função
	Festival de música	• Vídeo de show do festival de música ou função escrita
	Minha função relaxante	• Desenho ou vídeo de Minha função relaxante • Planilha Coletar, Acionar, Repetir
Loops	Pétalas de Repetição	• Pétalas de Repetição • Planilha Usando Loops • Planilha Loops por Todos os Lados
	Percurso de obstáculos	• Vídeo ou imagens do percurso de obstáculos (opcional)
	Padrões de bateria	• Planilha Ir Até a Ponta e Voltar • Vídeo ou imagens de bateria (opcional)
Variáveis	Afundar ou flutuar	• Afundar ou flutuar • Planilha Mantendo o Controle
	Jogo de palavras	• Jogos de palavras
	Tudo sobre mim	• Tudo sobre mim • Tudo sobre você
Design de Apps		• O que é um app? • Meu Design de Apps • Protótipo de Design de Apps

Introdução ao Swift Playgrounds no iPad ou Mac

Antes de iniciar as lições, é preciso baixar o Swift Playgrounds, o Pages e o Keynote, caso ainda não tenham sido baixados.

Os módulos neste guia usam diferentes combinações de playgrounds. Veja o que será necessário para cada módulo:

Módulo	Playgrounds			Como baixar no Swift Playgrounds
Comandos	 Aprenda a Programar 1	 MeeBot Aprenda a Dançar		Para assinar o feed Playgrounds MeeBot, role até o final da tela Mais Playgrounds e toque em Digitar URL de Assinatura. Em seguida, digite: ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json.
Funções	 Aprenda a Programar 1			
Loops	 Aprenda a Programar 1	 MeeBot Aprenda a Dançar		Para assinar o feed Playgrounds MeeBot, role até o final da tela Mais Playgrounds e toque em Digitar URL de Assinatura. Em seguida, digite: ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json.
Variáveis	 Aprenda a Programar 2	 Pedra, Papel e Tesoura	 Máquina de Código	Pedra, Papel e Tesoura e Máquina de Código estão disponíveis na seção Livros da tela Mais Playgrounds.
Design de apps				

Verifique os requisitos mínimos para Swift Playgrounds na App Store. Acesse o Suporte da Apple para obter ajuda em relação ao Swift Playgrounds.

Dicas do facilitador

Para aproveitar ao máximo as lições com a turma, experimente algumas destas dicas.

Atividades de explorar e descobrir:

- Simplifique qualquer sintaxe ou uso especial de maiúsculas e minúsculas ao formular ou exibir o código — por exemplo:
  - `var names = ["Rose", "Sam", "Joy"]` —> `var names = Rose, Sam, Joy`
  - `var ages = [7, 8, 7, 8, 7]` —> `var ages = 7, 8, 7, 8, 7`
  - `var myFavoriteColor =` ■ —>`var my favorite color =` ■

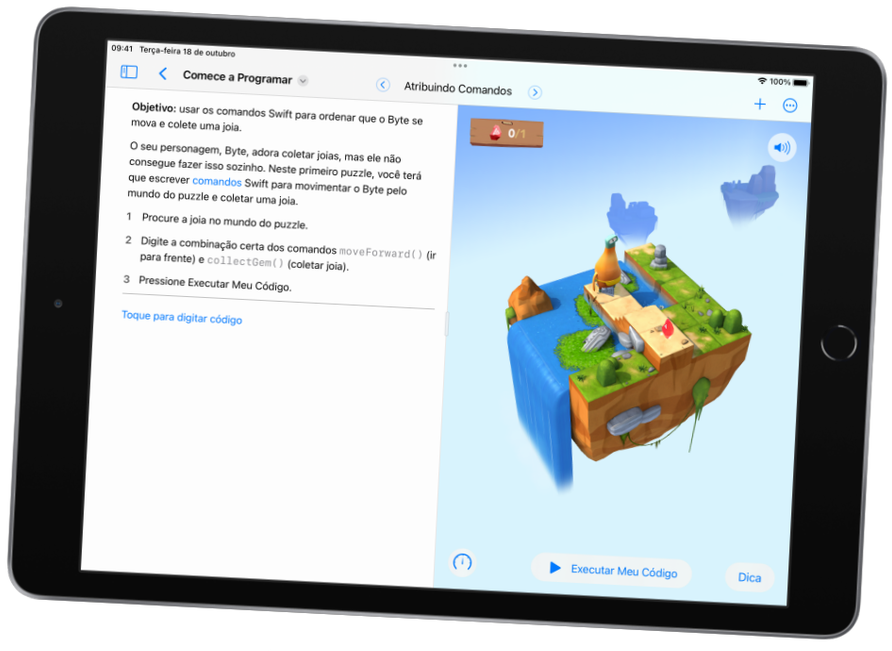
Atividades para Jogar:

- Para tornar o app Swift Playgrounds ainda mais simples para alunos em anos iniciais, siga as instruções nos planos de aula. Por exemplo:
  - Ler as introduções para a turma
  - Dar a estudantes orientações detalhadas sobre as planilhas complementares para que possam criar as suas próprias soluções
  - Usar um iPad ou Mac do facilitador para resolver os puzzles no app
- `let` e `var:` a palavra-chave `let` não é abordada neste guia. Para evitar confusão no Swift Playgrounds, altere quaisquer palavras `let` para `var` antes de mostrar as páginas a estudantes. Nos playgrounds que recomendamos, as duas palavras-chave podem ser trocadas.
  - `let` = variável não muda
  - `var` = variável muda

Extensões:

- Expanda as atividades de jogo no chão para incluir aritmética, alfabetização, palavras ilustradas, ortografia e muito mais. Experimente a atividade de jogo no chão no módulo Funções para se inspirar.
- Personalize as atividades de jogo no chão ao pedir que estudantes criem os seus próprios cartões para comandos, como `twirl()` ou `jump()`.

Página de introdução



Página do playground

Comandos

Visão geral

Lição 1: rotinas diárias

- Explorar: debate que faz a relação entre assar e os comandos
- Descobrir: atividade de rotinas diárias
- Jogar: Atribuindo Comandos e Adicionando um Comando

Lição 2: ordem da história

- Explorar: debate que faz a relação entre a ordem de tramas de histórias e os comandos
- Descobrir: atividade de ordem da história
- Jogar: quebra-cabeça no chão

Lição 3: passos de dança

- Explorar: debate que faz a relação entre os passos de dança e os comandos
- Descobrir: atividade de passos de dança
- Jogar: “Olá, MeeBot!" e Passos básicos

Estudantes serão capazes de:

- Usar exemplos do dia a dia para descrever instruções passo a passo
- Colocar instruções na ordem para que façam sentido
- Testar e depurar instruções e código

Vocabulário

- **Sequência:** a ordem na qual as coisas ocorrem
- **Etapa:** uma ação em um processo mais amplo
- **Modificar:** alterar
- **Comando:** código que diz para um aplicativo realizar uma ação específica
- **Bug:** um erro no código
- **Depurar:** encontrar e corrigir erros em programação

Padrões

1A-AP-08, 1A-AP-10, 1A-AP-12, 1A-AP-14, 1B-AP-16 >

Explorar

Objetivo: apresentar o conceito de comandos ao relacioná-lo ao fato de fazer brownies.

Debate:

- A turma seguiu uma receita de brownie?
- Os passos da receita foram seguidos na ordem?

Dica extra: cada etapa ou instrução em uma receita é como um comando no código. Peça que estudantes criem os seus próprios comandos.

Descobrir

Objetivo: modelar o processo de uma rotina diária ao identificar as instruções passo a passo.

Materiais: cartões sobre Lavar as mãos

Instruções:

1. Embaralhe os cartões sobre Lavar as mãos e os distribua em uma mesa ou no quadro. Os cartões devem estar fora de ordem.
2. Pergunte se a turma acha que há um erro na sequência para lavar as mãos.
3. Peça que estudantes depurem — ou corrijam — as instruções movendo um cartão por vez para o local correto.

Alternativa:

Peça que a turma trabalhe em duplas ou em pequenos grupos. Dê a cada grupo um conjunto de cartões.

Extensão:

Peça que a turma monte o seu próprio conjunto de instruções passo a passo para algo que se faz diariamente e crie imagens das etapas específicas.

Baixar os cartões sobre Lavar as mãos

Jogar

Objetivo: estudantes serão capazes de adicionar os comandos na ordem correta para coletar as primeiras joias no Aprenda a Programar 1 no app Swift Playgrounds.

Instruções:

1. Projete a página de introdução do capítulo "Comandos" no playground Aprenda a Programar 1 em uma tela.
2. Introdução:
   - Leia as páginas com a turma, parando para esclarecer dúvidas, se necessário.
3. Atribuindo Comandos:
   - Revise os dois comandos que a turma deve usar para que Byte colete a joia, `moveForward()` e `collectGem()`.
   - Peça que estudantes experimentem maneiras de direcionar Byte da seta de início até a joia e coletá-la. A turma pode registrar os comandos na planilha ou em uma folha de papel separada.
   - O objetivo é reunir ideias da turma e formular o código no app Swift Playgrounds para concluir o puzzle. Clique ou toque em Executar Meu Código.
   - Tente várias ideias diferentes.
   - Comemore com Byte!

Extensão:

Se a turma estiver pronta, passe para a próxima página, Adicionando um Comando. Aqui, estudantes devem usar um novo comando, `turnLeft()`.

Aprenda a Programar 1

Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground Aprenda a Programar 1
- Projetor ou tela

Materiais de estudantes:

- Planilhas Atribuindo Comandos e Adicionando um Comando
- Lápis
- Papel extra (opcional)

Baixar as planilhas Aprenda a Programar

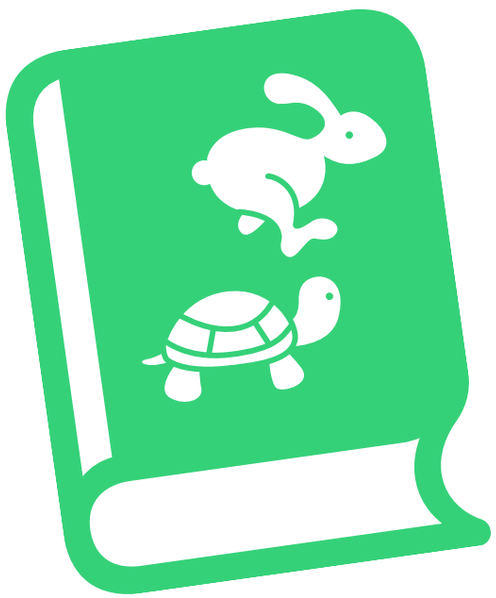
Explorar

Objetivo: explorar quantos livros seguem uma sequência (do início ao meio até o fim) na ordem para que as histórias façam sentido.

Debate:

- Pergunte se os livros seguem uma sequência.
- O que aconteceria se o início, o meio e o fim de um livro estivessem fora de ordem?
- Explore vários exemplos.

Dica extra: faça a conexão com o código, enfatizando como é importante dar comandos de programação na ordem correta, assim como os pontos da trama de uma história.



Descobrir

Objetivo: depois de criar imagens de vários pontos da trama de uma história, estudantes serão capazes de colocar as imagens na ordem para recriar a história com precisão.

Materiais do facilitador:

- Quadro branco
- Marcadores

Materiais de estudantes:

- Papel
- Marcadores ou lápis coloridos
- Alternativa: aparelhos iPad e um app de desenho

Instruções:

1. Leia uma história que estudantes conheçam bem. Com a turma, identifique os principais pontos da trama na história. Idealmente, devem ser criados quatro a seis pontos da trama.
2. Crie pequenos grupos que tenham o mesmo número de estudantes para pontos de trama — por exemplo, quatro pontos de trama equivalem a quatro estudantes por grupo.
3. Peça que cada estudante no grupo desenhe um dos pontos da trama.
4. Os grupos se revezam de pé na frente da sala, com estudantes segurando as imagens da trama fora de ordem.
5. O público reordena as imagens, movendo uma de cada vez.
6. Tire uma foto de cada grupo assim que estudantes estiverem na ordem correta.

Extensão ou alternativa:

Peça que cada grupo de estudantes trabalhe em uma história diferente, determinando os pontos da trama em grupo antes de desenhar as imagens.

Jogar

Objetivo: estudantes poderão orientar Byte por uma grade física até uma joia usando comandos direcionais.

Preparação: estudantes devem trabalhar em grupos de três. Use fita crepe para criar uma grade 4x4 no chão para cada grupo.

Instruções:

1. Distribua os materiais e divida a turma em grupos de três.
2. Leia cada função e atribua a cada pessoa no grupo uma função para o primeiro jogo.
3. Peça que estudantes joguem, começando com a função de designer.
4. A turma deve jogar três vezes, revezando a cada vez os cartões de função.

Funções:

- Designer: posicione a joia e a seta de início na grade.
- Programador: com a ajuda de colegas, posicione os cartões de comando na grade ou ao lado dela para direcionar Byte para a joia e coletá-la.
- Tester: começando com Byte na seta, siga os cartões de comando para mover Byte na grade. Se você coletar a joia, comemore! Caso a joia não seja coletada, vocês devem trabalhar em equipe para depurar ou corrigir o código.

Alternativa:
Se estudantes estiverem trabalhando com você individualmente ou aprendendo remotamente, poderão jogar sem mais ninguém usando a atividade alternativa em Keynote disponível para download.

Materiais do facilitador:

- Fita crepe

Materiais de estudantes:

- Cartões de função
- Cartões de comando: `moveForward()`, `turnLeft()`, `turnRight()` e `collectGem()`
- Joia
- Byte
- Seta

Baixar os materiais

Baixar a atividade alternativa

Explorar

Objetivo: explorar a ideia de que a programação pode ser criativa!

Debate:

- Pergunte a estudantes se já aprenderam alguma coreografia.
- A coreografia tinha uma ordem de passos a ser seguida?
- Como sabiam qual era o próximo passo?
- Os passos de dança têm nomes?
- Estudantes usam os mesmos passos em diferentes momentos durante a dança ou em diferentes coreografias?

Dica extra: ajude estudantes a fazer a conexão de que a programação é criativa e que — como coreografar uma dança — os programadores podem criar novos comandos e agrupá-los de maneiras diferentes e interessantes.

Descobrir

Objetivo: criar uma breve coreografia, junto com cartões para representar os passos de dança. Cada cartão de Passos de dança é como um comando no playground Aprender a Programar.

Materiais de estudantes:

- Aparelhos iPad
- App Keynote
- App Câmera
- Espaço para dançar

Instruções:

1. Peça que duplas ou pequenos grupos de estudantes criem uma breve coreografia.
2. Assim estabelecerem a coreografia, devem criar cartões dos diferentes passos de dança. A turma deve incluir um desenho e um nome do passo em cada cartão, com muita criatividade e diversão.
3. Cada grupo apresentará a sua dança — faça uma festa de dança com toda a turma!

Alternativa:

A turma pode usar os cartões de passos de dança disponíveis para download abaixo para criar a dança. Também pode usar os cartões como exemplos ao criar os seus próprios cartões.

Extensão:

Estudantes devem fazer um vídeo da dança para mostrar para o grupo.

Baixar o cartão Passos de dança

Jogar

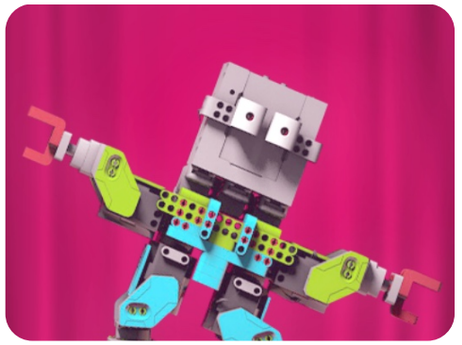
Objetivo: criar uma sequência de etapas para ensinar uma nova dança ao robô MeeBot.

Instruções:

1. Projete o playground MeeBot Aprenda a Dançar em uma tela. Você precisará assinar o playground se ainda não tiver feito isso.
2. Introdução:
   - Leia as páginas com a turma, parando para esclarecer dúvidas, se necessário.
3. Olá, MeeBot:
   - Clique ou toque em Executar Meu Código e assista à dança do robô.
4. Movimentos básicos:
   - Em grupo, em duplas ou individualmente no seu próprio iPad, estudantes devem escolher oito comandos na lista de sugestões e assistir à dança do robô.
   - Peça que a turma compartilhe as suas danças ou crie algumas diferentes em grupo.
   - Dance com o robô!

Extensão:

- Passe para a próxima página, Coreografia, na qual estudantes podem adicionar movimentos na função `myDanceRoutine()` — quantos comandos quiserem.



MeeBot Aprenda a Dançar

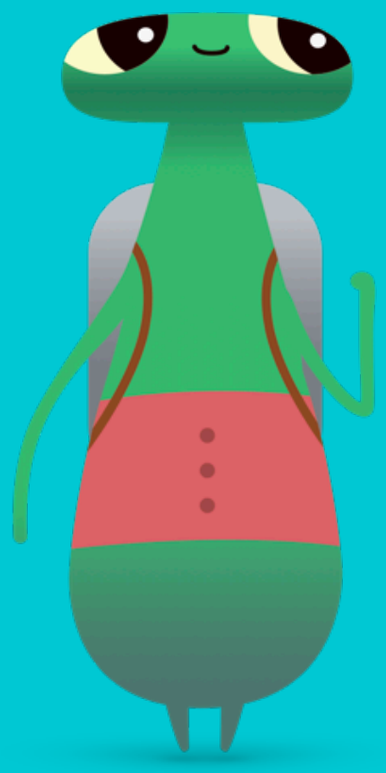
Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground MeeBot Aprenda a Dançar
- Projetor ou tela

Materiais de estudantes:

- Aparelhos iPad (opcional)

Funções



Visão geral

Lição 1: joia de papel

- Explorar: debate sobre instruções passo a passo
- Descobrir: atividade da joia de papel
- Jogar: Compondo um Comportamento e Criando uma Nova Função

Lição 2: festival de música

- Explorar: debate sobre como nomear uma função
- Descobrir: atividade do festival de música
- Jogar: quebra-cabeça no chão

Lição 3: minha função relaxante

- Explorar: debate sobre como resolver problemas de várias maneiras
- Descobrir: atividade Minha função relaxante
- Jogar: Coletar, Acionar, Repetir

Estudantes serão capazes de:

- Desconstruir um grande problema ou tarefa em etapas menores
- Criar uma série de etapas para resolver um problema ou concluir uma tarefa
- Nomear funções
- Testar e depurar código

Vocabulário

- **Função:** conjunto nomeado de comandos que podem ser executados sempre que necessário
- **Acionar:** ativar ou desativar

Padrões

1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14, 1B-AP-16 >

Explorar

Objetivo: explorar a ideia de empacotar uma série de comandos e dar um nome a ela.

Debate: decida sobre uma rotina diária para foco como turma. Peça que estudantes identifiquem o nome da rotina diária e as etapas que a compõem.

Exemplo: rotina da hora de dormir

- Etapa 1: escovar os dentes
- Etapa 2: ir ao banheiro
- Etapa 3: ler
- Etapa 4: dizer boa noite
- Etapa 5: apagar as luzes

Dica extra: criar um conjunto de instruções e dar um nome a ele é o mesmo conceito de criar uma função.

Extensão: pergunte se as instruções para qualquer uma das etapas poderia ser mais específica. Por exemplo, quais são as etapas específicas para escovar os dentes?

Descobrir

Objetivo: estudantes começarão seguindo instruções para criar uma joia de papel. Em seguida, escreverão ou desenharão as instruções para criar outra forma de sua escolha.

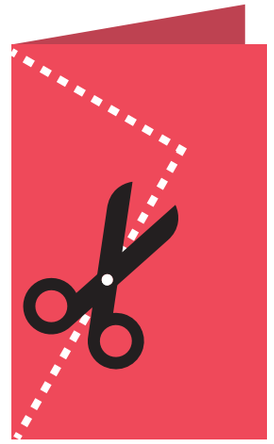
Materiais de estudantes:

- Papel
- Tesoura
- Lápis
- Aparelhos iPad (opcional)

Instruções:

Mostre à turma como criar uma joia de papel:

1. Dobre o pedaço de papel ao meio.
2. Desenhe uma linha do canto superior no lado dobrado até três ou cinco centímetros acima do centro do papel.
3. Desenhe outra linha a partir de onde a primeira linha termina até o canto inferior do lado dobrado.
4. Corte ao longo das linhas que você desenhou.
5. Remova a joia do papel que sobrou e a desdobre.



Peça que estudantes criem as suas próprias formas:

1. Divida a turma em pequenos grupos.
2. Peça que os grupos decidam uma forma para criar.
3. Dê um tempo para que pratiquem a criação da forma uma ou duas vezes.
4. Peça que estudantes escrevam ou desenhem as instruções para criar a forma. Depois, peça que deem um nome às instruções, como "Fazer um círculo" ou “A letra T".

Alternativa:

Grave um vídeo mostrando como criar as formas.

Jogar

Objetivo: trabalhando em grupo, a turma será capaz de detalhar as etapas necessárias para que Byte colete a joia.

Instruções:

1. Projete o playground Aprenda a Programar 1 em uma tela. Acesse o capítulo "Funções" no Aprenda a Programar 1.
2. Introdução:
   - Leia as páginas com a turma, parando para esclarecer dúvidas, se necessário.
3. Compondo um Comportamento:
   - Revise os comandos `moveForward()`, `turnLeft()` e `collectGem()` — lembre-se de que você não tem um comando `turnRight()`.
   - Peça que estudantes experimentem maneiras de direcionar Byte da seta de início até a joia e coletá-la. A turma deve registrar os comandos na planilha ou em uma folha de papel separada.
   - O objetivo é reunir ideias da turma e formular o código no app Swift Playgrounds para concluir o puzzle. Clique ou toque em Executar Meu Código.
   - Tente várias ideias diferentes.
   - Comemore com Byte!
4. Criando uma Nova Função:
   - Com base no que aprenderam na última página do playground, Compondo um Comportamento, peça que estudantes apresentem ideias para criar a função `turnRight()`.
   - Usando a função `turnRight()`, peça que estudantes tentem direcionar Byte de diversas maneiras a partir da seta de início até cada controle fechado e então os acionem.
   - O objetivo é reunir ideias da turma e formular o código no app Swift Playgrounds para concluir o puzzle. Clique ou toque em Executar Meu Código.
   - Tente várias ideias diferentes.
   - Comemore com Byte — esse puzzle foi difícil!

Aprenda a Programar 1

Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground Aprenda a Programar 1
- Projetor ou tela

Materiais de estudantes:

- Planilhas Compondo um Comportamento e Criando uma Nova Função
- Lápis
- Papel extra (opcional)

Baixar as planilhas Aprenda a Programar

Explorar

Objetivo: aplicar conhecimento de comandos e funções a músicas ao dar nomes descritivos a eles.

Debate: peça que estudantes criem várias músicas e deem a cada uma um nome descritivo de função.

Exemplo: para a música "Twinkle, Twinkle, Little Star" (Brilha brilha estrelinha), a função poderia se chamar `singTwinkle()`, mas `singSong1()` não seria um bom nome, pois a primeira música poderia ser alterada.

Dica extra: usar nomes descritivos em funções é importante, pois facilita a compreensão da programação para você e para outras pessoas.

Descobrir

Objetivo: estudantes criarão um show musical invocando diferentes comandos de som em uma função de show musical.

Materiais do facilitador:

- iPad ou Mac
- Projetor ou tela
- Quadro branco
- Marcadores

Instruções:

1. Ajude estudantes a criar nomes de função para várias músicas — por exemplo, `singHappyBirthday()`.
2. Em grupo, escolha a ordem na qual cantar as músicas.
3. Escreva uma definição de função para um show musical e preencha a função com os comandos de música.

Exemplo:

```
func createConcert() {
    singHappyBirthday()
    singTwinkleTwinkle()
    singMaryHadALittleLamb()
}
createConcert()
```

Alternativa:

Estudantes devem cantar em pequenos grupos, cada um com a sua própria lista de músicas, nomes de função de música e ordem para cantar as músicas. Cada grupo deve então apresentar as músicas e fazer um vídeo do seu show musical.

Jogar

Objetivo: estudantes resolverão uma equação simples, colocarão uma joia na resposta e orientarão Byte pela grade usando comandos direcionais.

Preparação: estudantes devem trabalhar em grupos de três. Use fita crepe para criar uma grade 4x4 no chão para cada grupo. Posicione a seta de início dentro de um quadrado e um número dentro de cada quadrado restante.

Instruções:

1. Distribua os materiais e divida a turma em grupos de três.
2. Leia cada função e atribua a cada pessoa no grupo uma função para o primeiro jogo.
3. Peça que estudantes joguem, começando com a função de designer.
4. A turma deve jogar três vezes, revezando a cada vez os cartões de função.

Funções:

- Designer: jogue dois dados. Com a ajuda de colegas, some os números e coloque a joia em um quadrado da grade que contenha a soma.
- Programador: com a ajuda de colegas, posicione os cartões de comando na grade ou ao lado dela para direcionar Byte para a joia e coletá-la.
- Tester: começando com Byte na seta, siga os cartões de comando para mover Byte na grade. Se você coletar a joia, comemore! Caso a joia não seja coletada, vocês devem trabalhar em equipe para corrigir o código.

Alternativa:
Se estudantes estiverem trabalhando com você individualmente ou aprendendo remotamente, poderão jogar sem mais ninguém usando a atividade alternativa em Keynote disponível para download.

Materiais do facilitador:

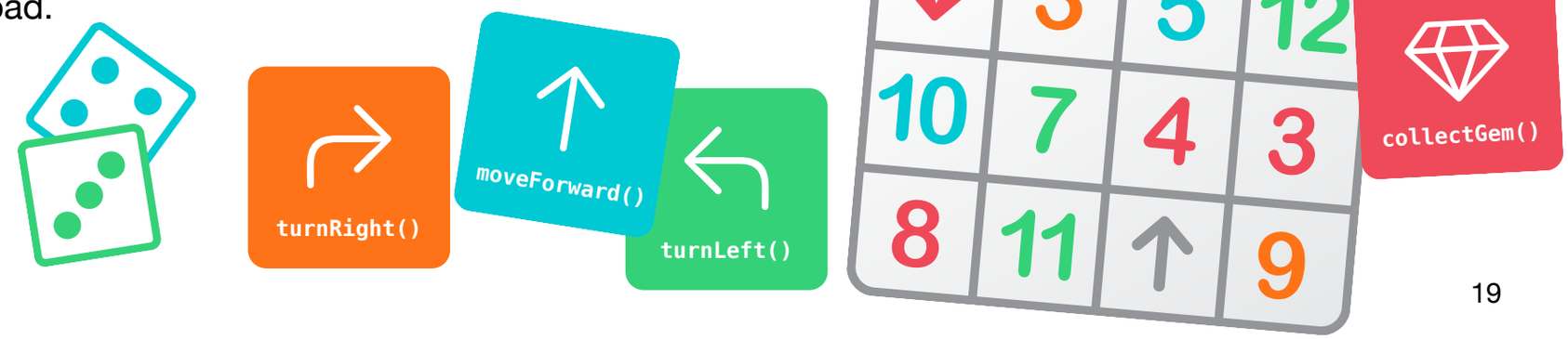
- Fita crepe
- Um conjunto de números impressos para cada grade

Materiais de estudantes:

- Cartões de função
- Cartões de comando: `moveForward()`, `turnLeft()`, `turnRight()` e `collectGem()`
- Joia
- Byte
- Seta
- Dois dados

Baixar os materiais

Baixar a atividade alternativa

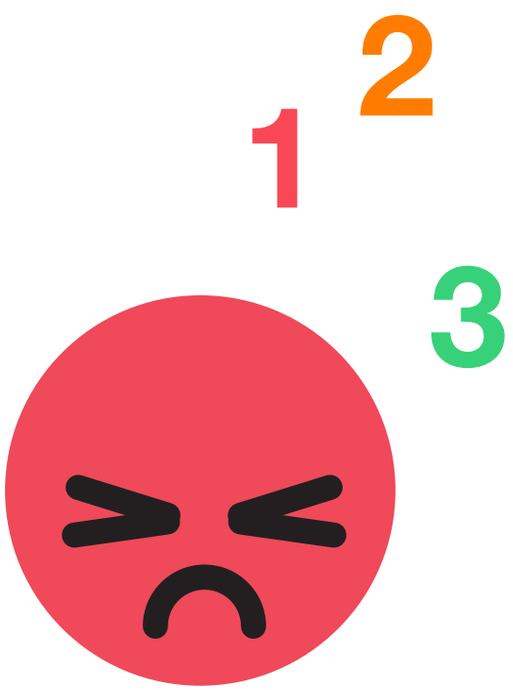


Explorar

Objetivo: estudantes compreenderão que geralmente há mais de uma maneira de resolver um problema.

Debate: peça que estudantes pensem sobre um problema que já tiveram. Depois, peça que digam como o resolveram. Pergunte ao grupo se alguém resolveria esse problema de outra maneira. Explore vários problemas e soluções diferentes.

Dica extra: ajude estudantes a fazer a conexão com código e aprender que geralmente há mais de uma maneira de resolver um problema de programação.



Descobrir

Objetivo: estudantes escreverão uma função para a sua técnica de relaxamento e darão um nome a ela.

Materiais de estudantes:

- Planilha Minha função relaxante
- Lápis
- Canetas ou lápis de cor

Instruções:

Dica: é melhor que estudantes trabalhem individualmente nesta atividade, se possível.

1. Peça que cada pessoa troque ideias sobre o que a acalma em casa ou na escola quando está nervosa. Peça que detalhem as técnicas de relaxamento em etapas.
2. Distribua a planilha Minha função relaxante e peça que estudantes desenhem as etapas da técnica de relaxamento.
3. Peça que estudantes deem um nome à técnica de relaxamento. Podem usar “camel case" — por exemplo, `countToTen()` — ou apenas uma frase curta, como “Count to ten” (Contar até 10).

Extensões:

Desconectado: peça que estudantes mostrem a técnica de relaxamento em pequenos grupos ou na frente da turma.

Com o iPad: estudantes devem criar um vídeo da técnica de relaxamento para compartilhar com a turma.

Baixar a planilha Minha função relaxante

Jogar

Objetivo: estudantes serão capazes de criar uma função composta por vários tipos de comandos. Depois, usarão essa função para concluir um puzzle.

Instruções:

1. Projete em uma tela a página Coletar, Acionar, Repetir no playground Aprenda a Programar 1, mostrando a função em branco que estudantes ajudarão a criar.
2. Coletar, Acionar, Repetir:
   - Revise os comandos `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` e `toggleSwitch()`.
   - Estudantes devem tentar identificar as peças do puzzle que se repetem. Em seguida, devem usar as ideias para completar a função no app e dar um nome a ela.
   - Peça que inventem um símbolo para a função. Em seguida, peça que registrem o símbolo e o nome da função na legenda de comandos na planilha.
   - Com o comando adicional, estudantes devem descobrir maneiras de orientar Byte a coletar todas as joias e acionar todos os controles. A turma deve registrar os comandos na planilha ou em uma folha de papel separada.
   - O objetivo é reunir ideias da turma e formular o código no app Swift Playgrounds para concluir o puzzle. Clique ou toque em Executar Meu Código.
   - Tente várias soluções diferentes.
   - Comemore com a turma — esse foi um puzzle difícil!

Aprenda a Programar 1

Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground Aprenda a Programar 1
- Projetor ou tela

Materiais de estudantes:

- Planilha Coletar, Acionar, Repetir
- Lápis
- Papel extra (opcional)

Baixar a planilha Aprenda a Programar

Loops

Visão geral

Lição 1: pétalas de repetição

- Explorar: debate sobre como relacionar etapas repetidas em programação na vida real
- Descobrir: atividade de Pétalas de repetição
- Jogar: Usando Loops e Loops por Todos os Lados

Lição 2: percurso de obstáculos

- Explorar: debate sobre pontos de parada em um loop
- Descobrir: atividade de percurso de obstáculos
- Jogar: quebra-cabeça no chão

Lição 3: padrões de bateria

- Explorar: debate sobre loops em música
- Descobrir: atividade Padrões de bateria
- Jogar: Ir Até a Ponta e Voltar e Passos de dança

Estudantes serão capazes de:

- Identificar um loop no código
- Desconstruir um grande problema ou tarefa em etapas menores
- Criar uma sequência de comandos e repetir essa sequência usando um loop
- Testar e depurar instruções e código

Vocabulário

- **Loop:** bloco de código que é repetido um determinado número de vezes

Padrões

1A-CS-01, 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14 >

Explorar

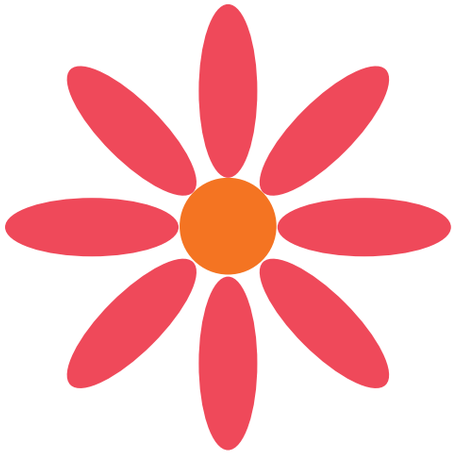
Objetivo: conectar a ideia de loops à vida real.

Debate: explorar quando estudantes podem repetir uma tarefa ou etapa na vida real.

Exemplos:

- Caminhada
- Bicicleta
- Costurar, tricotar ou fazer crochê

Dica extra: os loops repetem um comando ou um conjunto de comandos quantas vezes você especificar.



Descobrir

Objetivo: estudantes começarão a explorar o conceito de loops ao criar uma flor única.

Materiais de estudantes:

- Planilha Pétalas de Repetição
- Papel colorido
- Lápis
- Tesoura
- Hastes colantes
- Dados

Instruções:

1. Estudantes devem desenhar uma única pétala — aproximadamente do comprimento da palma da mão — em um pedaço de papel colorido e recortá-la. Esse será o modelo de pétala da flor.
2. Cada estudante jogará os dados duas vezes, somará os números e preencherá o número faltando no loop na planilha Pétalas de Repetição. Esse é o número de pétalas que a flor deles terá.
3. Usando o modelo de pétala, estudantes devem desenhar a pétala no papel colorido e recortar o número correto de pétalas para a flor.
4. Usando a planilha Pétalas de Repetição, estudantes devem montar a flor e colar as peças no lugar.

Baixar a planilha Pétalas de Repetição

Jogar

Objetivo: estudantes serão capazes de formular código em um loop para coletar todas as joias.

Instruções:

1. Projete a página de introdução do capítulo "Loops For" no playground Aprenda a Programar 1 em uma tela.
2. Introdução:
   - Leia as páginas com a turma, parando para esclarecer dúvidas, se necessário.
3. Usando Loops:
   - Mostre à turma como portais funcionam e revise os comandos `moveForward()`, `turnLeft()`, `turnRight()` e `collectGem()`.
   - Peça que estudantes experimentem maneiras de orientar Byte da seta de início até as joias e coletá-las, observando como os comandos se repetem. A turma deve registrar os comandos na planilha ou em uma folha de papel separada.
   - Reúna ideias da turma e formule código no app Swift Playgrounds para orientar Byte a coletar a primeira joia e caminhar até o portal.
   - Pergunte quantas joias são contadas e adicione esse número ao loop. Clique ou toque em Executar Meu Código.
   - Tente várias soluções diferentes.
   - Comemore com Byte!
4. Loops por Todos os Lados:
   - Peça que estudantes experimentem maneiras de coletar todas as joias, observando quais comandos se repetem.
   - Para adicionar um loop `for`, use as sugestões de código na parte inferior do editor ou toque em + na parte superior da tela.
   - Reúna ideias da turma e formule o código no app Swift Playgrounds para concluir o puzzle. Clique ou toque em Executar Meu Código.
   - Tente várias ideias diferentes.
   - Comemore com Byte!

Aprenda a Programar 1

Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground Aprenda a Programar 1
- Projetor ou tela

Materiais de estudantes:

- Planilhas Usando Loops e Loops por Todos os Lados
- Lápis
- Papel extra (opcional)

Baixar as planilhas Aprenda a Programar

Explorar

Objetivo: explorar o motivo pelo qual loops sempre precisam de um ponto final específico.

Debate: peça que estudantes imaginem uma roda-gigante ou outro brinquedo que conheçam. O que aconteceria se o operador não pressionasse o botão para parar a roda depois de cinco voltas? Peça outros exemplos do que aconteceria se um loop não fosse parado.

Dica extra: ajude estudantes a compreender que, se não colocarem uma parada em um loop, ele se repetirá infinitamente.

Descobrir

Objetivo: estudantes descobrirão como loops funcionam ao aplicar um loop em um percurso de obstáculos que criaram.

Materiais:

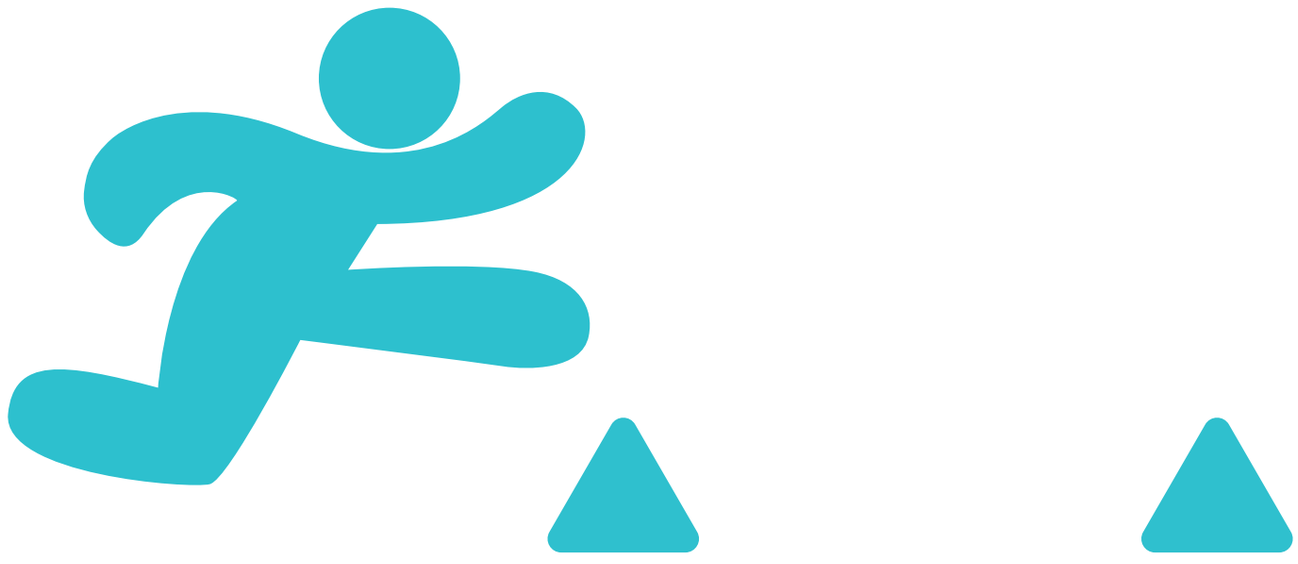
- Espaço para fazer uma atividade física
- Acessórios para percurso de obstáculos
- Dado

Instruções:

1. Crie um breve percurso de obstáculos, na sala de aula ou fora dela.
2. Jogue um dado e peça que estudantes repitam o percurso quantas vezes o dado mostrar.

Alternativa:

Estudantes devem criar uma série de movimentos — por exemplo, tocar os dedos do pé, pular, pular com uma perna. Jogue um dado e peça que estudantes repitam os movimentos quantas vezes o dado mostrar.



Jogar

Objetivo: estudantes serão capazes de criar um puzzle com um padrão de repetição. Depois, resolverão o puzzle em grupo.

Preparação: estudantes devem trabalhar em grupos de três. Use fita crepe para criar uma grade 4x4 no chão para cada grupo.

Instruções:

1. Distribua os materiais e divida a turma em grupos de três.
2. Leia cada função e atribua a cada pessoa no grupo uma função para o primeiro jogo.
3. Peça que estudantes joguem, começando com a função de designer.
4. A turma deve jogar três vezes, revezando a cada vez os cartões de função.

Funções:

- Designer: com a ajuda de colegas, posicione três joias em um padrão de repetição na grade. Posicione a seta de início na grade.
- Programador: com a ajuda de colegas, posicione os cartões de comando na grade ou ao lado dela para direcionar Byte para as joias e coletá-las. Use os cartões de Loop para dizer ao tester quantas vezes você repetirá os comandos.
- Tester: começando com Byte na seta, siga os cartões de comando para mover Byte na grade. Se coletar todas as joias, comemore! Caso a joia não seja coletada, vocês devem trabalhar em equipe para corrigir o código.

Alternativa:
Se estudantes estiverem trabalhando com você individualmente ou aprendendo remotamente, poderão jogar sem mais ninguém usando a atividade alternativa em Keynote disponível para download.

Materiais do facilitador:

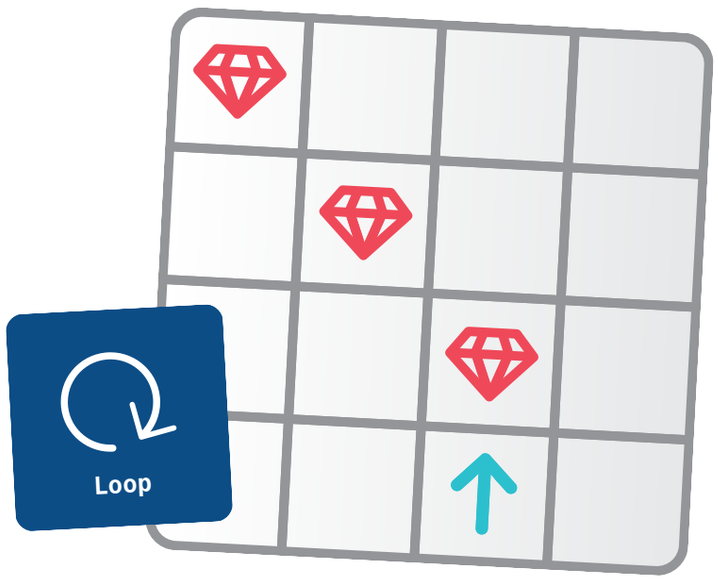
- Fita crepe

Materiais de estudantes:

- Cartões de função
- Cartões de comando: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` e `Loop`
- Joias
- Byte
- Seta

Baixar os materiais

Baixar a atividade alternativa



Explorar

Objetivo: explorar padrões que se repetem em música.

Debate: peça que estudantes conversem sobre instrumentos que tocam ou músicas que gostam. Pergunte se já repetiram uma batida ou refrão ao tocar ou cantar. Veja se conseguem lembrar de outras partes de uma música que se repetem.

Dica extra: reforce a ideia de que loops consistem em duas partes:

- Os comandos
- O número de vezes a ser repetido

Descobrir

Objetivo: estudantes serão capazes de repetir um padrão de batida, fazendo uma conexão entre o código em loop e um exemplo físico da vida real.

Materiais:

- Algo para bater, como chão, pernas ou livros
- Espaço para sentar em círculo

Instruções:

1. Peça que estudantes sentem em círculo.
2. Diga à turma para repetir a batida que você criou conforme o número de dedos que você mostrar na mão. Por exemplo, se você mostrar quadro dedos, a turma deve repetir a batida quatro vezes e então parar.
3. Percorra o círculo ou divida a turma em pequenos grupos para que cada estudante tenha a chance de ser o percussionista principal.

Extensão:

Peça que estudantes criem batidas.

Jogar

Objetivo: estudantes devem acionar vários comandos diferentes dentro de um loop e identificar quantas vezes o loop deve ser invocado.

Instruções:

1. Projete o playground Aprenda a Programar 1 em uma tela. Acesse a terceira página, Ir Até a Ponta e Voltar, do capítulo "Loops".
2. Ir Até a Ponta e Voltar:
   - Revise os comandos `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` e `toggleSwitch()`.
   - Peça que estudantes tentem direcionar Byte de diversas maneiras a partir da seta de início até cada controle fechado e então acioná-lo.
   - Para adicionar um loop `for`, use as sugestões de código na parte inferior do editor ou toque em + na parte superior da tela.
   - O objetivo é reunir ideias da turma e formular o código no app Swift Playgrounds para concluir o puzzle. Clique ou toque em Executar Meu Código.
   - Tente várias ideias diferentes.
   - Comemore com Byte!
3. Saia do Aprenda a Programar 1 e abra o playground MeeBot Aprenda a Dançar, acessando a página Passos de dança. (Não há uma planilha de estudante para a página deste playground.)
4. Passos de dança:
   - Peça que a turma trabalhe em grupo, em duplas ou individualmente em seu próprio iPad para concluir o passo e assistir à dança do robô.
   - Peça que estudantes compartilhem as suas danças ou criem algumas diferentes em grupo.
   - Dance com o robô!

Aprenda a Programar 1

MeeBot Aprenda a Dançar

Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground Aprenda a Programar 1
- Playground MeeBot Aprenda a Dançar
- Projetor ou tela

Materiais de estudantes:

- Planilha Ir Até a Ponta e Voltar
- Lápis
- Aparelhos iPad (opcional)
- Papel extra (opcional)

Baixar a planilha Aprenda a Programar

Variáveis

Visão geral

Lição 1: afundar ou flutuar

- Explorar: debate sobre como atualizar uma variável
- Descobrir: atividade Afundar ou flutuar
- Jogar: Mantendo o Controle e Exemplo de Jogo

Lição 2: jogo de palavras

- Explorar: debate sobre tipos de respostas a perguntas
- Descobrir: atividade de jogo de palavras
- Jogar: quebra-cabeça no chão

Lição 3: tudo sobre mim

- Explorar: debate sobre como responder a perguntas com listas
- Descobrir: atividade Tudo sobre mim
- Jogar: Usando um Loop

Estudantes serão capazes de:

- Associar um nome de variável a um determinado valor
- Alterar o valor atribuído a uma variável
- Compreender os diferentes tipos de Swift que você pode atribuir a uma variável, incluindo verdadeiro/falso (booleanos), números (Ints), palavras (Strings), cores (literais de cores) e imagens (literais de imagens)
- Testar e depurar instruções e código

Vocabulário

- **Variável:** contêiner nomeado que armazena um valor e pode ser alterado
- **Dados:** informações
- **Booleano:** tipo que tem um valor de `true (verdadeiro)` ou `false (falso)`

Padrões

1A-AP-09, 1B-AP-09, 1B-AP-10, 1B-AP-16 >

Explorar

Objetivo: explorar o conceito de variáveis contando objetos e atualizando o número de variável.

Materiais do facilitador:

- Quadro branco
- Marcador
- Borracha
- Contêiner
- Cinco lápis (ou cinco unidades de outro objeto)

Instruções:

1. Comece escrevendo uma instrução de variável no quadro branco para manter o controle dos objetos.
   - Exemplo: `var numberOfPencils = 0`
2. Segure um contêiner vazio e diga que ele representa a sua variável, `numberOfPencils`.
3. Adicione um lápis ao contêiner e pergunte qual é a contagem da variável agora. Quando responderem corretamente, apague o *0* e escreva *1*.
4. Continue até ter adicionado todos os lápis e o código for: `var numberOfPencils = 5`.
5. Em seguida, comece tirando os lápis do contêiner, atualizando a variável à medida que retirá-los.

Dica extra: ajude estudantes a entender que variáveis armazenam uma parte de informação. Nesse caso, as informações são um número, e o número diz quantos lápis há no contêiner.

Descobrir

Objetivo: usando os objetos encontrados, estudantes devem fazer experimentos para verificar se os itens afundam ou flutuam. Depois, devem registrar os dados usando imagens (literais de imagens) e valores verdadeiro/falso (booleanos).

Materiais de estudantes:

- Aparelhos iPad
- App Keynote
- Planilha Afundar ou flutuar
- Balde de água
- Vários objetos para testar

Instruções:

1. Divida a turma em pequenos grupos.
2. Peça que coletem vários itens para testar.
3. Para cada item, peça que estudantes:
   - Tirem uma foto do item e incluam a imagem na planilha.
   - Testem o item na água.
   - Registrem os resultados na planilha circulando `true (verdadeiro)` ou `false (falso)`.

Baixar a planilha Afundar ou flutuar

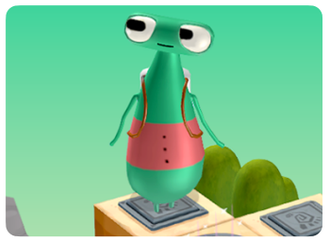
Jogar

Objetivo: estudantes serão capazes de criar e atualizar variáveis em dois contextos diferentes de programação.

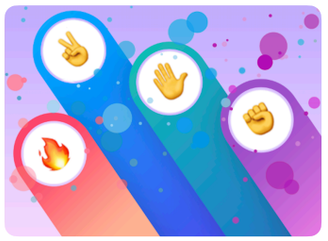
Instruções:

1. Projete o playground Aprenda a Programar 2 em uma tela. Acesse o capítulo "Variáveis".
2. Introdução:
   - Leia as páginas com a turma, parando para esclarecer dúvidas, se necessário.
3. Mantendo o Controle:
   - Peça que a turma experimente maneiras de direcionar Hopper da seta de início até a joia e coletá-la. A turma deve registrar os comandos na planilha ou em uma folha de papel separada.
   - O objetivo é reunir ideias da turma e formular o código no app Swift Playgrounds para concluir o puzzle. Clique ou toque em Executar Meu Código.
   - Tente várias ideias diferentes.
   - Comemore com Hopper!
4. Saia do Aprenda a Programar 2 e passe para a última página do playground Pedra, Papel e Tesoura, chamado de Exemplo de Jogo. (Não há uma planilha de estudante para a página deste playground.)
5. Exemplo de Jogo:
   - Clique ou toque em Executar Meu Código para jogar antes de mudar qualquer coisa.
   - Descreva em grupo quais peças do jogo você quer personalizar. Algumas coisas divertidas que você pode fazer são `game.roundsToWin`, `game.challenger.emoji`, `game.addOpponent` e `game.roundPrize`.
   - Jogue várias vezes, mudando algo diferente a cada vez.

Extensão: muitas variáveis são estabelecidas no arquivo Game.swift. Se estudantes tiverem curiosidade sobre o motivo pelo qual algumas variáveis não têm `var` na frente delas, abra o arquivo Game.swift para mostrar onde as propriedades do jogo foram criadas.



Aprenda a Programar 2



Pedra, Papel e Tesoura

Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground Aprenda a Programar 2
- Playground Pedra, Papel e Tesoura
- Projetor ou tela

Materiais de estudantes:

- Planilha Mantendo o Controle
- Lápis
- Papel extra (opcional)

Baixar a planilha Aprenda a Programar

Explorar

Objetivo: explorar vários tipos de resposta no mundo real e relacioná-las aos vários tipos da linguagem Swift, incluindo sim/não ou verdadeiro/falso (booleanos), números (Ints), palavras (Strings), cores (literais de cores) e imagens (literais de imagens).

Materiais do facilitador:

- Quadro branco
- Marcadores

Debate: com a turma, pense em perguntas que exijam diferentes tipos de respostas e as escreva no quadro.

Exemplos:

- Qual é a cor dos seus olhos? —> color
- Você tem um animal de estimação? —> sim/não
- Você tem irmãos ou irmãs? —> sim/não
- Quantos anos você tem? —> número
- Qual é seu nome? —> palavra

Dica extra: explique que variáveis também têm diferentes tipos, incluindo números, palavras, cores, imagens e respostas sim ou não. Dependendo de como cria uma variável, você precisará manter o mesmo tipo, mesmo se atualizar a variável para algo novo. Por exemplo, `var myAge = 8` pode mudar para 9, mas não pode mudar para `“nine"`.

Descobrir

Objetivo: estudantes serão capazes de concluir um jogo de palavras preenchendo o tipo de resposta correta.

Materiais de estudantes:

- Planilhas do Jogo de palavras
- Lápis
- Lápis de cor

Instruções:

Peça que estudantes concluam um ou mais jogos de palavras em pequenos grupos. O ideal é que cada grupo tenha pelo menos uma pessoa para ler ou uma pessoa auxiliar. Se ninguém souber ler, faça jogos em grupo.

Extensão: se souberem ler, peça que estudantes criem um jogo de palavras para outra pessoa preencher. Incentive a turma a usar números, palavras, cores, imagens e respostas do tipo sim ou não para os espaços em branco.

Baixar as planilhas do Jogo de palavras

Jogar

Objetivo: estudantes serão capazes de orientar Byte a coletar várias joias, adicionar cada joia a um contêiner e atualizar uma variável.

Preparação: estudantes devem trabalhar em grupos de três. Use fita crepe para criar uma grade 4x4 no chão para cada grupo.

Instruções:

1. Distribua os materiais e divida a turma em grupos de três.
2. Leia cada função e atribua a cada pessoa no grupo uma função para o primeiro jogo.
3. Peça que estudantes joguem, começando com a função de designer.
4. A turma deve jogar três vezes, revezando a cada vez os cartões de função.

Funções:

- Designer: posicione várias joias e a seta de início na grade.
- Programador: com a ajuda de colegas, posicione os cartões de comando na grade ou ao lado dela para direcionar Byte para as joias e coletá-las.
- Tester: começando com Byte na seta, siga os comandos para movimentar Byte na grade, adicionando as joias ao contêiner à medida que forem coletadas. Se você coletar todas as joias, atualize a variável `numberOfGems` no contêiner e comemore! Caso não sejam coletadas todas as joias, vocês devem trabalhar em equipe para corrigir o código.

Alternativa:

Se estudantes estiverem trabalhando com você individualmente ou aprendendo remotamente, poderão jogar sem mais ninguém usando a atividade alternativa em Keynote disponível para download.

Materiais do facilitador:

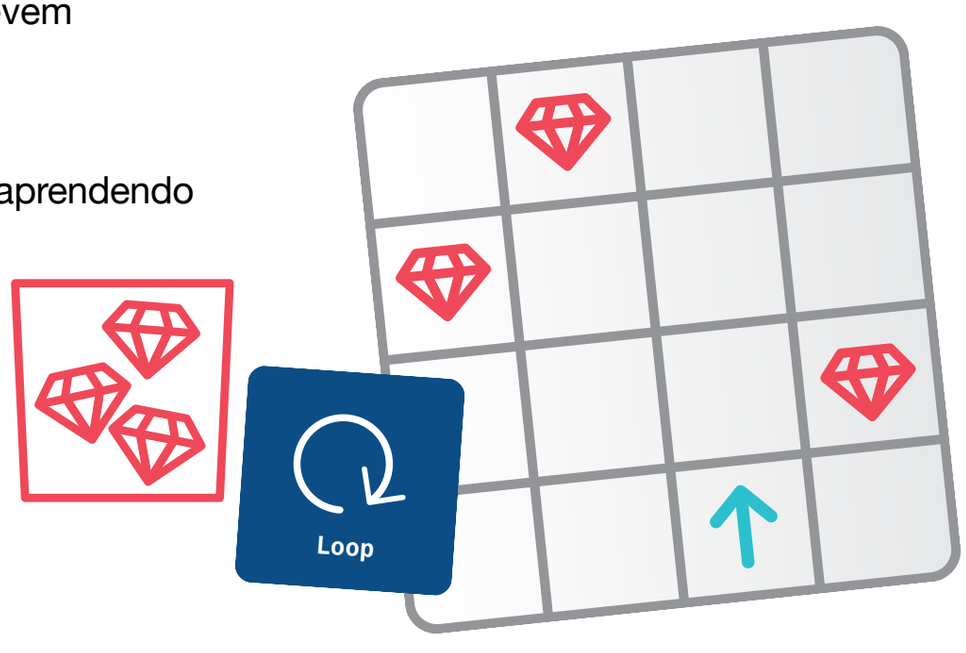
- Fita crepe

Materiais de estudantes:

- Cartões de função
- Cartões de comando: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` e `Loop`
- Joias
- Byte
- Seta
- Contêiner intitulado: `var numberOfGems = _____`
- Caneta

Baixar os materiais

Baixar a atividade alternativa



Explorar

Objetivo: explorar como usar listas — ou *vetores* — ao criar variáveis.

Debate: o que aconteceria se uma planilha perguntasse a cada estudante qual o nome do irmão ou irmã e essa pessoa tivesse mais de um? Colete ideias da aula. Se sugerirem fazer uma lista, diga que é isso que programadores fazem! Quando uma variável tem mais de uma resposta, estudantes devem criar uma lista.

Peça que estudantes criem perguntas que poderiam ter várias respostas.

Exemplos:

- Nomes de amigos —> Rose, Sam, Joy
- Idades de estudantes —> 7, 8, 7, 8, 7, 8, 9, 7, 8, 9, 8
- Cores favoritas —> , , , ,
- Animais favoritos —> , , ,

Dica extra: as listas criadas por estudantes em programação são semelhantes a listas em uma frase.

Descobrir

Objetivo: estudantes serão capazes de preencher variáveis para descrever algo sobre si e outra pessoa da turma. A turma pode ter a oportunidade de usar um vetor como um tipo de variável.

Materiais de estudantes:

- Planilhas Tudo sobre mim e Tudo sobre você
- Lápis
- Lápis de cor

Instruções:

1. Peça que estudantes preencham a planilha Tudo sobre mim.
   - Quem tiver mais de um irmão/irmã ou animal de estimação deve criar uma lista de itens separados por vírgulas.
2. Estudantes devem formar duplas para preencher a planilha Tudo sobre você.

Alternativa: estudantes podem usar o iPad e o Keynote para preencher a planilha, tirando fotos das respostas das imagens e colorindo os literais usando as opções de formatação.

Baixar as planilhas Tudo sobre

Jogar

Objetivo: estudantes serão capazes de identificar uma variável no código e explorar maneiras de usar vetores com loops.

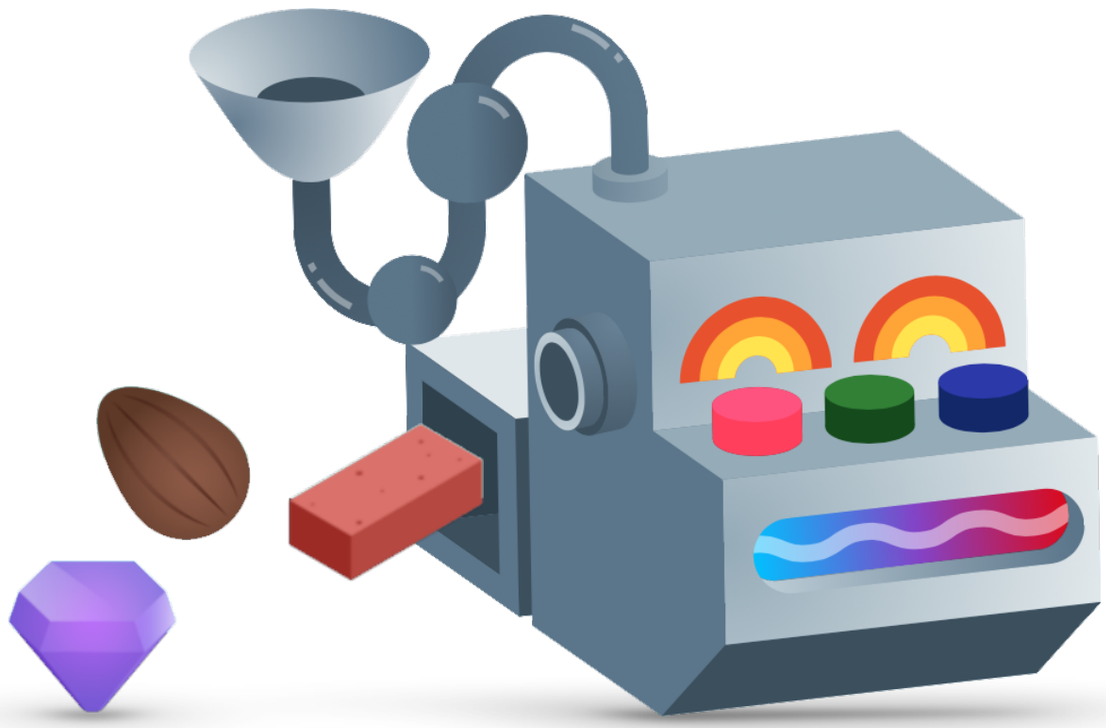
Instruções:

1. Projete o playground Máquina de Código em uma tela.
2. Introdução:
   - Leia as páginas com a turma, parando para esclarecer dúvidas, se necessário.
   - Opcional: jogue as duas primeiras páginas, Explorando a Máquina e Combinando com Cores.
3. Usando um Loop:
   - Nesta página, estudantes combinarão seu conhecimento de loops com variáveis.
   - Veja se a turma consegue identificar a variável no código que usa um vetor.
   - Clique ou toque em Executar Meu Código para ver o que a máquina cria.
   - Passe para a segunda etapa nas instruções e atualize o código para incluir uma segunda variável, itens e um loop aninhado. Clique ou toque novamente em Executar Meu Código para ver o que a máquina criará.
   - Nota: tente usar esta página antes de fazer a atividade com estudantes.

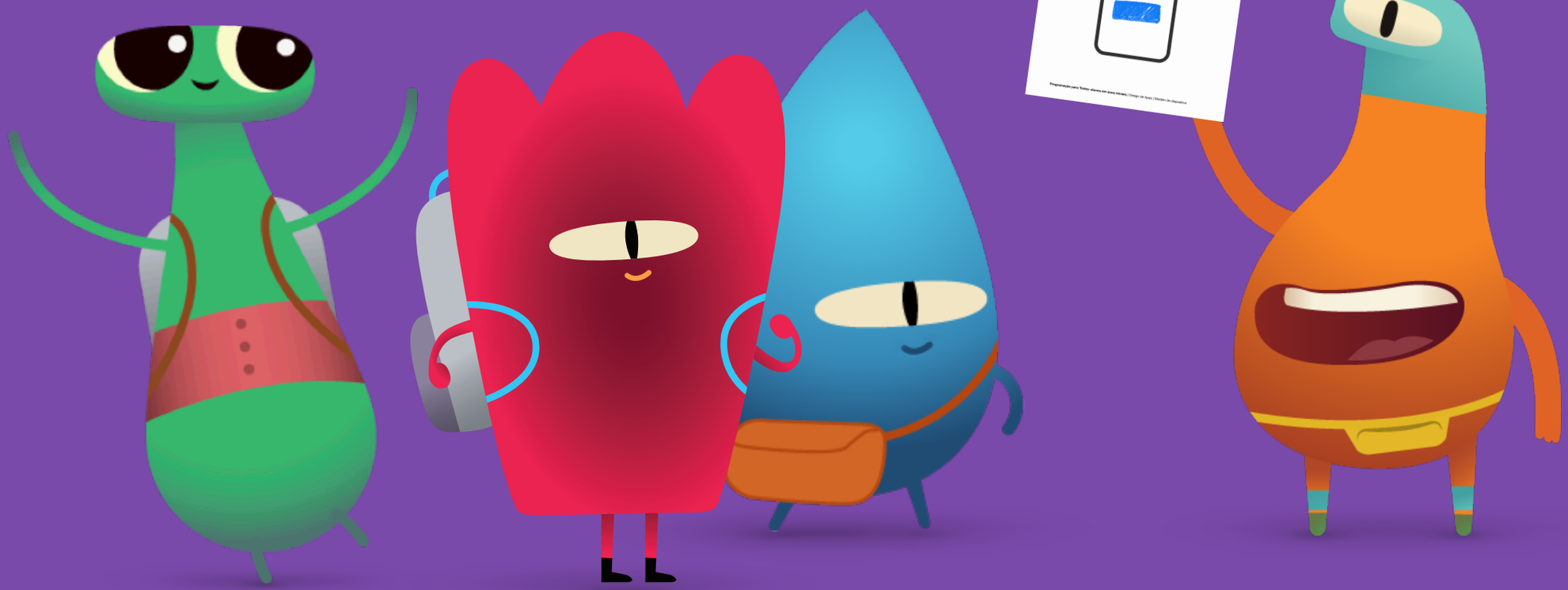
Máquina de Código

Materiais do facilitador:

- iPad ou Mac
- App Swift Playgrounds
- Playground Máquina de Código
- Projetor ou tela



Design de apps



Explorar

Objetivo: explorar apps conhecidos em vários aparelhos.

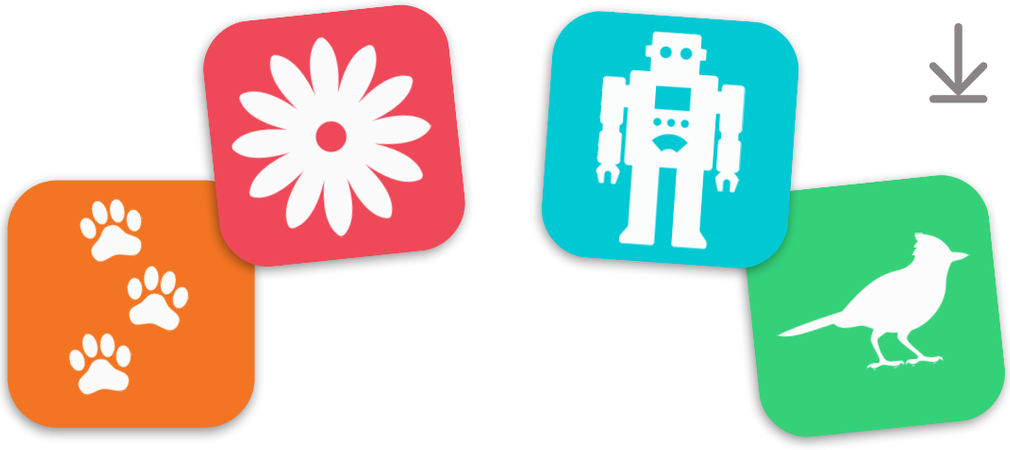
Instruções: inicie um debate sobre apps que estudantes usam no iPad em casa ou na escola. Em seguida, converse sobre apps que estudantes ou seus pais, mães ou responsáveis usam em aparelhos em casa.

Dica extra: reforce a ideia de que apps não estão apenas em telefones celulares, mas também em relógios inteligentes, tablets, computadores e até mesmo na TV.

Extensão: aprofunde-se em alguns exemplos de apps, perguntando a estudantes para quem o app foi desenvolvido, o que ele faz e por que acham que o app foi criado.

Exemplo:

- App: Swift Playgrounds
- Para quem é: pessoas que querem aprender sobre Swift
- O que ele faz: ajuda pessoas a aprender como programar por meio de puzzles e lições
- Por que ele foi criado: para ensinar programação a pessoas com pouco ou nenhum conhecimento do assunto



Descobrir

Objetivo: preparar estudantes para que criem os seus próprios apps ao analisar um app conhecido.

Materiais de estudantes:

- Aparelhos iPad
- Planilha "O que é um app?"
- Lápis
- Canetas ou lápis de cor

Instruções:

1. Divida a turma em pequenos grupos ou peça que trabalhem individualmente.
2. Peça que escolham um app para iPad.
3. Peça que usem a planilha "O que é um app?" para orientar a exploração do app.
4. Peça que compartilhem suas conclusões sobre o app, seja para a turma toda ou em duplas.

Dica do facilitador: quanto mais jovem for a turma, mais ajuda estudantes precisarão para preencher essa planilha. Para um grupo de anos iniciais, considere fazer dois ou três apps com toda a turma.

Baixar a planilha "O que é um app?"

Jogar

Objetivo: estudantes devem criar os seus próprios apps!

Materiais de estudantes:

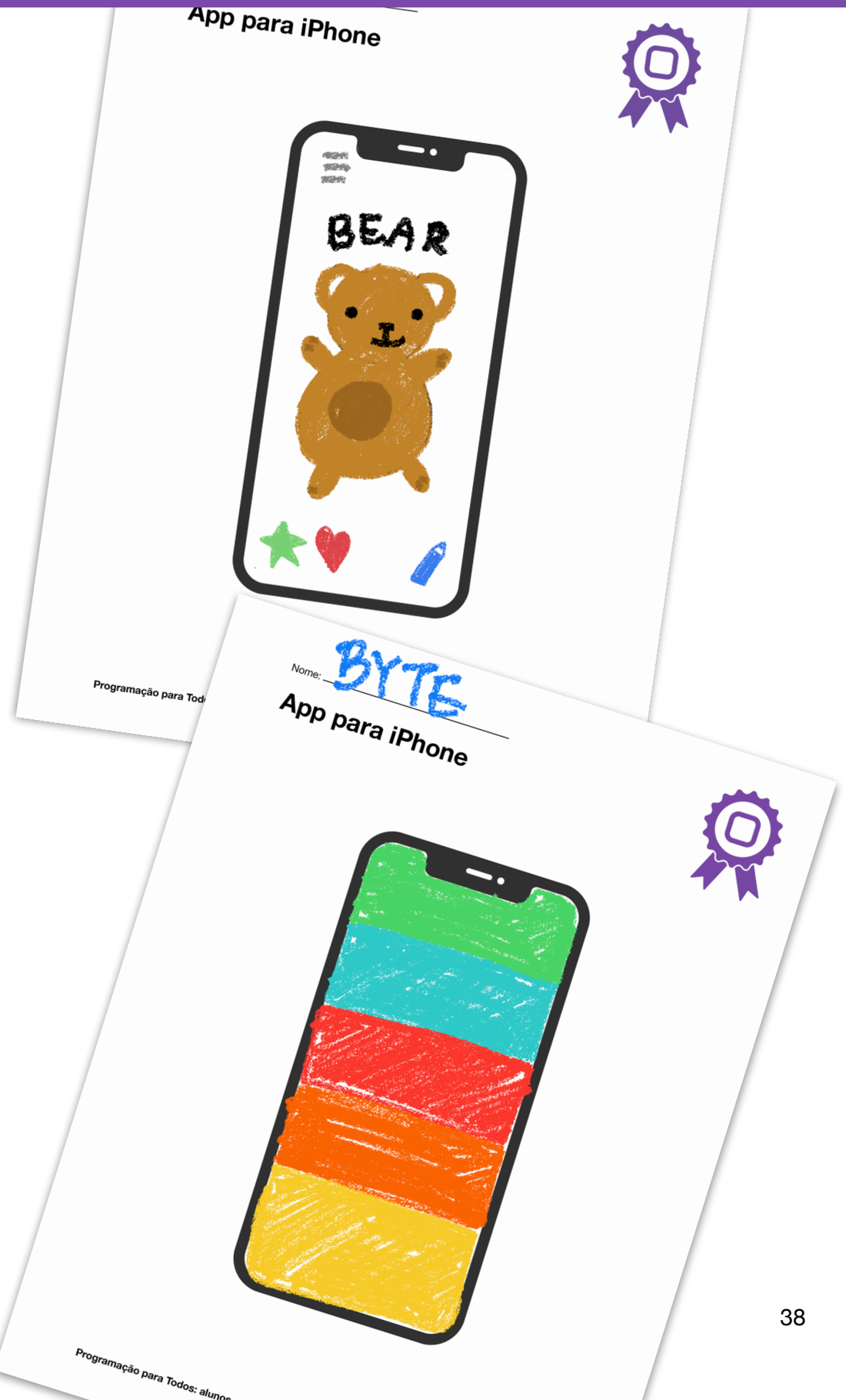
- Planilha Meu Design de Apps
- Modelos de aparelho
- Papel extra
- Lápis
- Canetas ou lápis de cor

Instruções:

1. Divida a turma em pequenos grupos ou peça que trabalhem individualmente.
2. Auxilie a turma em relação à planilha Meu Design de Apps para fornecer orientações sobre o processo inicial de criação de um app.
3. Peça que estudantes criem as páginas de seus apps usando papel extra ou os modelos de aparelho.
4. Instrua a turma a criar uma versão final dos protótipos de app usando os modelos de aparelho.
5. Convide cada estudante ou grupo de estudantes a apresentar suas ideias para todo o grupo.

Baixar a planilha Meu Design de Apps

Baixar os modelos de aparelho



Recursos do facilitador

Glossário

- **Acionar:** ativar ou desativar
- **Booleano:** tipo que tem um valor de `true` (verdadeiro) ou `false` (falso)
- **Bug:** um erro no código
- **Comando:** código que diz para um aplicativo realizar uma ação específica
- **Dados:** informações
- **Depurar:** encontrar e corrigir erros em programação
- **Etapa:** uma ação em um processo mais amplo
- **Função:** conjunto nomeado de comandos que podem ser executados sempre que necessário
- **Loop:** bloco de código que é repetido um determinado número de vezes
- **Modificar:** alterar
- **Sequência:** a ordem na qual as coisas ocorrem
- **Variável:** contêiner nomeado que armazena um valor e pode ser alterado

Padrões da CSTA >

1A-AP

- 1A-AP-08: representar processos diários criando e seguindo algoritmos (conjuntos de instruções passo a passo) para concluir as tarefas.
- 1A-AP-09: representar a maneira como programas armazenam e manipulam dados usando números ou outros símbolos para representar informações.
- 1A-AP-10: desenvolver programas com sequências e loops simples para expressar ideias ou resolver um problema.
- 1A-AP-11: decompor (detalhar) as etapas necessárias para resolver um problema em uma sequência precisa de instruções.
- 1A-AP-12: desenvolver planos que descrevam a sequência do programa de eventos, metas e resultados esperados.
- 1A-AP-14: depurar (identificar e corrigir) erros em um algoritmo ou programa que inclua sequências e loops simples.

1A-CS

- 1A-CS-01: selecionar e operar software apropriado para realizar várias tarefas, bem como reconhecer que usuários têm diferentes necessidades e preferências em relação à tecnologia que usam.

1B-AP

- 1B-AP-09: criar programas que usam variáveis para armazenar e modificar dados.
- 1B-AP-10: criar programas que incluam sequências, eventos, loops e condicionais.
- 1B-AP-16: realizar várias funções, com orientações do facilitador, em colaboração com colegas durante as fases de criação, implementação e revisão de desenvolvimento do programa.

Respostas de exemplo

Estas próximas páginas apresentam uma possível solução para cada puzzle do Swift Playgrounds — mas os puzzles podem ser resolvidos de mais de uma maneira. Peça para estudantes tentarem diferentes maneiras de direcionar Byte ou outros personagens.

Comemore todos os tipos de programação e metas que estudantes possam ter. Algumas pessoas podem querer explorar todo o espaço do puzzle além de coletar as joias, enquanto outras podem querer tentar o maior número possível de maneiras de coletar joias. Não se esqueça — a programação deve ser divertida!

Aprenda a Programar 1

Capítulo Comandos
Atribuindo Comandos

```
moveForward()
moveForward()
moveForward()
collectGem()
```

Capítulo Comandos
Adicionando um Comando

```
moveForward()
moveForward()
turnLeft()
moveForward()
moveForward()
collectGem()
```

Capítulo Funções
Compondo um Comportamento

```
moveForward()
moveForward()
moveForward()
turnLeft()
turnLeft()
turnLeft()
moveForward()
moveForward()
moveForward()
collectGem()
```

Capítulo Funções
Criando uma Nova Função

```
func turnRight() {
    turnLeft()
    turnLeft()
    turnLeft()
}

moveForward()
turnLeft()
moveForward()
turnRight()
moveForward()
turnRight()
moveForward()
turnRight()
moveForward()
turnLeft()
moveForward()
toggleSwitch()
```

Aprenda a Programar 1

Aprenda a Programar 2

Capítulo Funções
Coletar, Acionar, Repetir

```
func collectToggle() {
  moveForward()
  collectGem()
  moveForward()
  toggleSwitch()
  moveForward()
}

collectToggle()
turnLeft()
collectToggle()
moveForward()
turnLeft()
collectToggle()
turnLeft()
collectToggle()
```

Capítulo Loops
Usando Loops

```
for i in 1 … 5 {
  moveForward()
  moveForward()
  collectGem()
  moveForward()
}
```

Capítulo Loops
Loops por Todos os Lados

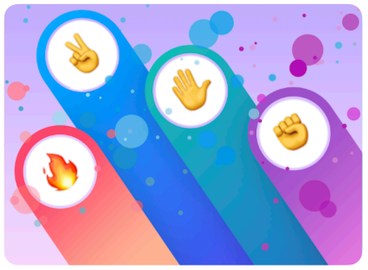
```
for i in 1 ... 4 {
  moveForward()
  collectGem()
  moveForward()
  moveForward()
  moveForward()
  turnRight()
}
```

Capítulo Loops
Ir Até a Ponta e Voltar

```
for i in 1 ... 4 {
  moveForward()
  moveForward()
  toggleSwitch()
  turnLeft()
  turnLeft()
  moveForward()
  moveForward()
  turnLeft()
}
```

Capítulo Variáveis
Mantendo o Controle

```
var gemCounter = 0
moveForward()
moveForward()
collectGem()
gemCounter += 1
```



Pedra, Papel e Tesoura

Exemplo de Jogo

Não há um exemplo de solução para essa página, pois o jogo é inteiramente personalizável. Você pode jogá-lo como quiser!

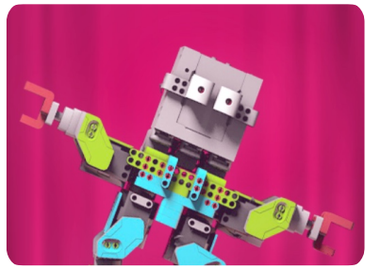
Máquina de Código

Usando um Loop

```
var colors = [Light.red, Light.green,
Light.blue]

var items = [Item.metal, Item.stone,
Item.cloth, Item.dirt, Item.DNA,
Item.spring, Item.wire, Item.egg,
Item.tree, Item.gear, Item.seed,
Item.crystal, Item.mushroom,
Item.unidentifiedLifeForm]

for item in items {
  setItemA(item)
  setItemB(.dirt)
  switchLightOn(.green)
  forgeItems()
}
```



MeeBot Aprenda a Dançar

Passos básicos

```
bendAndTwist()
happy()
moveBackward()
shake()
skip()
split()
swagger()
twist()
```

Passos de dança

```
for i in 1 ... 5 {
  bend()
  bend(beats: 2)
  bendAndTwist()
  moveBackward(beats: 9)
}
```