



Currículo Technovation

Semana 4

[Unidade de Negócios 3: Modelo de negócios](#)

[Unidade de Programação 4](#)

Continuar Unidade de Marketing 1

Unidade de Negócios: Módulo 3. Modelo de negócios

Objetivos de aprendizagem:

Neste módulo, você aprenderá a...

- Desenvolver um modelo de receita
- Avaliar custos operacionais
- Determinar projeções de lucro

Aprender: Como desenvolver um modelo de negócios

Negócios têm sucesso quando conseguem fazer dinheiro bastante para se sustentar e gerar lucro. Para isso, é necessário planejar e desenvolver um **modelo de negócios**, que explica como você quer gerar dinheiro (**receita**). Isso é diferente do seu plano de negócios, que inclui mais informações sobre como dar início ao negócio. É necessário pensar em todos os custos das diferentes atividades, recursos, equipamentos e materiais necessários para manter o negócio funcionando e como ele vai gerar **lucro** depois de pagar as contas. Há diferentes maneiras de montar um negócio, como visto no [Módulo 1](#); existem negócios sem fins lucrativos, com fins lucrativos e empreendimentos sociais, além de muitos outros. Pensem em que tipo de classificação vocês querem para sua empresa enquanto começam a desenvolver o modelo de negócios. Independentemente do tipo de negócio, receita é necessária para dar à sua empresa sustentabilidade a longo prazo.

Há muitas partes a se pensar em um novo negócio, mas vamos nos ater às 4 principais:

1. **Capital inicial (capital semente)** - dinheiro inicial investido na empresa para dar início ao negócio
2. **Modelo de receita** - como você planeja receber dinheiro pelos bens e/ou serviços
3. **Custos operacionais** - incluem custos fixos e variáveis para operar o negócio. Coisas que precisam ser pagas constantemente se chamam custos fixos, e coisas que variam dependendo de outros fatores se chamam custos variáveis.
4. **Projeção de lucro** - o dinheiro que sobra depois de pagar contas: quanto você gerou no total. Esse "dinheiro extra" que sobrou também pode ser reinvestido (p.ex. para contratar pessoas, aumentar o local de trabalho, etc.) para que a empresa possa crescer.

Capital inicial



Vocês tiveram uma ótima ideia de aplicativo para desenvolver e agora têm uma ótima marca. Agora está na hora de pensar em como gerar dinheiro para começar o seu negócio!

Vocês têm que buscar investidores para ajudar a investir dinheiro (ou **capital**) no seu negócio. Isso também é chamado de "captação de recursos". Investidores fazem um **investimento** em vocês e no seu produto e, em troca, vocês ganham dinheiro para dar início ao negócio. Depois de começar o negócio, os investidores esperam dinheiro de volta mais um valor adicional (ou participação na empresa) por ter dado a você o capital inicial. Isso é chamado de **retorno** sobre o investimento, mais conhecido como ROI (**Return On Investment**).

Há [muitas maneiras](#) de arrecadar capital para sua start-up. Alguns exemplos:

- **Crowdfunding** - um grupo grande de pessoas dá fundos que juntos resultam em uma quantia grande. Isso oferece oportunidades únicas para pessoas comuns investirem diretamente em empresas, mas isso não é recomendado para captação a longo prazo. Também é útil ter uma rede social grande em que você possa divulgar sua empresa.
 - Exemplos: [Indiegogo](#), [Kickstarter](#), [Kiva](#), [Go Fund Me](#)
- **Investidor anjo** - pessoas muito ricas que investem seu próprio dinheiro em empresas emergentes, mas que esperam em troca participação no negócio
 - Exemplos: [Peter Cowley](#), [Jeff Bezos](#), [Paul Buchheit](#)
- **Capital de risco** - grupo de investidores profissionais que entendem oportunidades de investimento e riscos em start-ups. Eles fazem pesquisas extensas e investem em start-ups

que imaginam que terão sucesso no mercado. O dinheiro investido pode ter uma grande variedade de fontes.

- Exemplos: [3i](#), [Khosla Ventures](#), [SV Angel](#)

Todos os tipos de investidores aceitam uma certa quantidade de risco, com a esperança de retorno após um período de tempo. É por isso que eles investem: eles aplicam dinheiro para vê-lo crescer, mas há uma chance de não o receber de volta.

Exemplo do carrinho de comida

Vejam um exemplo de um carrinho de comida. Se quiserem começar seu próprio carrinho de comida, vocês precisam de um investimento. Vocês podem pedir para seus pais investirem no negócio. Quando pedir a seus pais, eles vão querer saber:

- De quanto dinheiro vocês precisam?
- O que vão comprar com o dinheiro que estão recebendo?
- Quanto eles vão receber de volta e quando (retorno sobre o investimento, ROI)?

Ao falar com investidores, vocês precisam estar preparadas para responder perguntas importantes. No mínimo, eles precisam saber de quanto dinheiro vocês precisam, como vão gastar esse dinheiro e o que eles receberão em retorno. É aqui que o **modelo de negócios** será importante. Seu modelo de negócios deve responder as perguntas deles e mais.

Vamos começar seu modelo de receita para responder a algumas das perguntas!

Modelo de receita

Um **modelo de receita** é uma descrição ou planejamento de como seu negócio vai gerar dinheiro. **Receita** é o dinheiro que um negócio gera de suas atividades normais, como a venda de bens e a prestação de serviços a clientes. Ao subtrair os custos operacionais da receita, vocês têm o **lucro**, que é o dinheiro gerado que vocês podem guardar.

$$[\text{Receita}] - [\text{Custos operacionais}] = [\text{Lucro}]$$

Há muitas maneiras de criar modelos de receita, mas, para aplicativos mobile, existem apenas as seguintes:

Taxa única para download do aplicativo	Compras dentro do aplicativo (inclusive assinaturas)	Propagandas dentro do aplicativo
A cobrança ocorre uma única vez para baixar o aplicativo, fornecendo uma experiência	Venda de bens, serviços ou recursos especiais. Conhecido também como "freemium", que	Pago através de propagandas, que podem ser banners, pop-ups

única.	significa que o aplicativo pode ser baixado de graça, mas o cliente paga uma assinatura ou taxas adicionais para acessar recursos especiais.	(propagandas que aparecem no meio de atividades).
Exemplos		
Swype Keyboard  Minecraft  HeadsUp! 	Airbnb  Pokemon GO  Spotify 	Facebook  Yelp  2048 

Veja este [vídeo](#) se quiser saber mais!

Para dar início e fazer cálculos de receita, vamos voltar para a pesquisa de consumo feita na unidade de [pesquisa de mercado](#). Pensem nos usuários que vocês têm e se eles pagariam mais dinheiro para recursos desenvolvidos no aplicativo. Vocês querem gerar receita vendendo o aplicativo, oferecendo espaço para propagandas, compras dentro do aplicativo ou uma combinação dessas coisas?

Ao desenvolver o modelo de receita, vocês também devem incluir o preço das coisas. Quanto vale seu aplicativo para o cliente (demanda ou atração)? Ou o espaço para propagandas? Ou cada uma das compras dentro do aplicativo? Quanto seus concorrentes cobram pelo aplicativo? Trabalhem em equipe para decidir o que é melhor para o aplicativo. Voltem também para a análise de concorrência na unidade de [pesquisa de mercado](#).

Estudem os modelos de receita de aplicativos similares para entender melhor como deve ser o de vocês. Outra coisa para se ter em mente é que um [estudo de 2013](#) mostrou que a maioria dos aplicativos é gratuita. Eles dependem principalmente da venda de bens e serviços através de

compras dentro do aplicativo. Por exemplo, para baixar o aplicativo do Minecraft, você precisa pagar uma taxa, mas também é possível comprar mais coisas para o jogo. Seu modelo de vendas pode ser simplesmente só vender o aplicativo, ou mais complicado, com várias maneiras de fazer dinheiro (esse modelo também é conhecido como fluxo de receitas).

Pensem em todas as situações complexas que podem acontecer quando clientes compram ou pagam por coisas no seu aplicativo. Isso gerará dinheiro bastante para sustentar seu negócio? Se seus custos operacionais forem maiores que a receita em um período de tempo, seu negócio não lucrará e vocês não conseguirão se sustentar. Peçam ajuda para o mentor e trabalhem em equipe para tomar essas decisões. Para ver mais ideias sobre diferentes fluxos de receita para aplicativos mobile, clique [aqui](#) ou [aqui](#).

Se ficou um pouco confuso, vamos revisar o que aprendemos usando nosso exemplo do carrinho de comida.

Exemplo do carrinho de comida

Vocês precisam decidir entre uma das seguintes opções para gerar receita para o carrinho:

- Modelo de aplicativo pago - cada comida é vendida por um preço determinado por você
- Compras dentro do aplicativo - você dá a comida de graça, mas vende os temperos e os pratos maiores por um preço definido
- Propagandas dentro do aplicativo - você dá a comida de graça, mas vende o espaço de propaganda nos pratos

Alguns recursos para ajudar a definir o preço do produto:

- Consulte este [guia detalhado da Sequoia Capital sobre como definir o preço de um produto](#). A Sequoia Capital é uma empresa reconhecida de capital de risco que já investiu em empresas como Apple, Google, PayPal, Stripe, YouTube e Instagram
- [Mashable: COMO: Determinar o preço correto para seu aplicativo](#)
- [iBrainBursts: Modelo de receita Google Android](#)
- [Quora: Quais são alguns dos modelos de receita mais bem sucedidos para aplicativos mobile?](#)

Custos operacionais

Depois de pensar em quanto dinheiro vão fazer, através do modelo de receita, vamos pensar em quanto custará chegar lá. Neste momento, determinaremos o custo daquilo que é necessário para manter ou operar o negócio, o que é chamado de **custos operacionais**. Há muitos tipos de custos envolvidos. Vocês precisarão pensar sobre quais custos terão quando desenvolverem o aplicativo, o que pode incluir várias coisas, desde [taxas de lojas de aplicativos](#) até um produto que vocês possam vender pelo aplicativo.

Vocês podem baixar nosso [modelo](#) para começar. Talvez nem todos os itens listados sejam necessários e vocês também podem tentar cortar custos. Por exemplo, o espaço de escritório pode ser algum lugar que possam usar de graça.

Projeção de lucro

A última parte do modelo de negócios é a **projeção de lucro**. Para começar, vamos dividir essa expressão. **Lucro** é o dinheiro que sobra depois de pagar todas as contas. **Projeção** é uma estimativa fundamentada sobre quanto vocês terão lucrado após um certo período de tempo. Geralmente, a projeção de lucro é a soma de todos os cálculos feitos no modelo de receita, menos os custos de operação. Você podem calculá-la somando a receita e subtraindo os custos operacionais. Lembre-se de que empresas sem fins lucrativos não geram lucro, mas geram receita para cobrir seus custos operacionais.

Para desenvolver uma projeção, é necessário decidir como o negócio crescerá. Vocês esperam que mais usuários comprem o aplicativo? Ou esperam que mais usuários façam compras dentro do aplicativo? Vocês devem determinar quais fatores aumentarão com o tempo ou se precisarão de mais receita de outras fontes. Quanto esperam ganhar? Há muitas incertezas: toda empresa carrega um risco e uma incerteza, mesmo depois de fazer pesquisas e avaliações.

Alguns guias para ajudar:

- [Chron: Como estimar o lucro](#)
- [BPlans: 5 formas de tornar sua projeção lucro realista](#)
- [Entrepreneur: Como prever receita e crescimento](#)

Criar: Modelo de negócios

Agora está na hora de juntar seu capital inicial, modelo de receita e custos operacionais para desenvolver um modelo de negócios. Para lembrar, utilize a equação a seguir para calcular o lucro:

$$[\text{Receita}] - [\text{Custos operacionais}] = [\text{Lucro}]$$

Vocês podem baixar nosso [modelo](#) ou criar o seu próprio! [incluir link para baixar o modelo]

Vocês podem sempre voltar para outras seções e mudar algumas coisas. Lembrem-se, o seu modelo de negócios é apenas uma estimativa fundamentada sobre como vocês acham que o negócio funcionará financeiramente.

Aplicar: Finalize seu modelo de negócios

Com a ajuda do mentor e da equipe, vocês desenvolveram um modelo de negócios para apoiar tanto a equipe quanto o desenvolvimento do aplicativo. Isso envolveu muito trabalho e

raciocínio, mas terá um ótimo uso! Vocês devem incluí-lo no planejamento de negócio para a entrega da equipe. Caso precisem de ajuda, consultem os seguintes guias:

Recursos adicionais

- [App Lab: Que modelo de receita devo escolher para meu aplicativo mobile?](#)
- [Analytics Academy: Modelos de receita de aplicativos com o Google](#)

Reflexão

Seu modelo de negócios está pronto. Parabéns!

Aprendemos vários termos novos nesta unidade. Vamos revisar os principais que vocês precisarão para desenvolver seu modelo de negócios:

- **Capital inicial** - valor inicial que ajuda vocês a começar, normalmente cedido por diferentes tipos de investidores
- **Receita** - valor total recebido com as atividades da empresa (como venda)
- **Modelo de receita** - um plano sobre como gerar dinheiro, ou receita
- **Custos operacionais** - o dinheiro necessário para que a empresa funcione.
- **Lucro** - dinheiro que sobrou depois de pagar as contas

Ao falar de empreendedorismo, há muitas coisas complicadas para se pensar e, às vezes, nem sabemos a resposta. Podemos precisar de estimativas ou palpites para pensarmos com os pés no chão. Algumas questões a se considerar depois de terminar o modelo de negócios:

- O que precisa acontecer para sua empresa ter sucesso?
- Quanto dinheiro é necessário para começar sua empresa e gerar receita?
- Vocês pensaram em outras formas de melhorar seu aplicativo para aumentar a projeção de lucro da sua empresa?
- Por que vocês escolheram esse modelo de receita?
- Como esperam que o seu negócio cresça?

Agora, vocês podem começar a usar o modelo de negócios! Na próxima unidade vocês desenvolverão um plano de negócios, no qual o modelo de negócios será essencial.

Atividade para o mentor

Sua equipe criou um modelo de negócio que será incluído no plano de negócios. Como mentor, você deve supervisionar esse trabalho. Certifique-se de que ele faz sentido, é factível e prático para os objetivos de negócio da equipe. Quando der seu feedback, seja específico e ajude

a guiar as estudantes. Não apenas dê a resposta. Elas precisam descobrir isso sozinhas. Faça perguntas e ouça o que elas têm a dizer.

Unidade de Programação 4

Recursos mobile e conexão com a web

Objetivos de aprendizagem:

Nesta unidade, você aprenderá:

- Componentes de interface de usuário
- Componentes de mídia
- Sensores
- Componentes sociais
- Bases de dados online

Componentes mobile

Parabéns por chegarem à quarta unidade de programação! Até agora, exploramos maneiras de conversar com o celular e desenvolver aplicativos. Nesta unidade, vocês aprenderão sobre recursos mobile do seu celular e como aproveitá-las, além de se conectar com outros lugares na web.

Esta unidade pode ser usada como referência para as capacidades que seu aplicativo pode oferecer. Vocês não precisam experimentar todos os componentes listados, mas podem testar aqueles que fazem sentido para o aplicativo.

Para referência, aqui estão os [parâmetros de avaliação Technovation 2017](#) e a [lista técnica](#).

Componentes de interface de usuário

São recursos que podem ser incluídos no aplicativo para interação com o usuário. Acessem o [site do MIT](#) para saber mais.

Notifier (notificador)

Pode enviar uma mensagem ou alerta para o usuário. O usuário pode eliminar a caixa de diálogo com um botão, ou ela pode desaparecer sozinha após alguns segundos. O notificador pode enviar informações para o usuário, além de ser útil para depurar bugs, o que vocês aprenderão na unidade de programação 6. Aqui está um [tutorial](#).

Canvas

Esse componente permite ao usuário usar sprites, desenhar e mover imagens. Pode ser muito útil para o desenvolvimento de jogos! Alguns tutoriais:

- [Paintpot 1](#)
- [Paintpot 2](#)
- [Paintpot 3](#)
- [Paintpot 4](#)
- [Ball MIT Tutorial](#)
- [MIT digital doodle](#)

WebView

Esse componente permite incorporar um navegador da web no aplicativo. Se quiserem que o usuário possa acessar um site específico, podem usar esse componente. A URL exibida pode ser especificada, mas este não é um navegador completo. Por exemplo, apertar o botão para voltar do celular faz sair do aplicativo, ao invés de voltar a página no navegador. Aqui está um [tutorial](#).

Date and Time Picker (Selecionador de data e hora)

Esse componente permite aos usuários escolher uma data e/ou hora. Isso é especialmente útil para aplicativos em que os usuários devem planejar eventos. Aqui está um [tutorial](#).

Componentes de mídia

Esses são recursos de mídia, como fotos, áudio e vídeo. Acessem o [site do MIT](#) para saber mais.

Camcorder (filmadora)

Esse componente permite ao usuário gravar vídeos. Pode ser usado para aplicativos sociais, aplicativos de compartilhamento de vídeo ou quando quiserem que o usuário grave vídeos. Aprendam como usá-lo [aqui](#).

Camera (Câmera)

Esse componente permite ao usuário tirar fotos. Pode ser útil para aplicativos que tenham perfis com foto ou que tirem foto de alo. Aqui está um [tutorial](#).

ImagePicker (Selecionador de imagem)

Permite ao usuário escolher uma foto da galeria. É uma boa ideia incluir esse recurso em conjunto com os componentes Camcorder ou Camera. O usuário pode também selecionar fotos que tenham sido tiradas fora do aplicativo. Aqui está um [tutorial](#).

VideoPlayer (Leitor de vídeo)

Permite incorporar um vídeo ao aplicativo para o usuário clicar e assistir. O vídeo precisa estar no formato .wmv, .3gp ou .mp4 e não pode ser maior que 1MB. Aqui está um [tutorial](#).

YandexTranslate (tradutor)

Permite traduzir textos para outro idioma. É necessário que seu aplicativo tenha acesso à internet, porque ele depende do serviço de tradução [Yandex](#). Aqui está um [tutorial](#).

SoundRecorder (gravador de som)

Permite que o usuário grave um som ou ruído. Aqui está um [exemplo](#) de uso da Pura Vida Apps.

Player (leitor de áudio)

Este componente de áudio toca sons. Funciona melhor para sons mais "longos", como músicas, discursos ou poemas. Aqui está um [tutorial](#) e o [resumo do tutorial](#). O Video Player é parecido, mas funciona com vídeos ao invés de áudio!

Sound (Som)

Este componente é parecido com o Player, mas é feito para sons curtos, como campainhas de notificação. Aqui está um [tutorial](#) para aprender como usá-lo.

SpeechRecognizer (reconhecimento de voz)

Traduz fala em texto. Isso é especialmente útil para aplicativos que necessitem de recursos de mãos livres. Aqui está um [exemplo](#).

TexttoSpeech (texto para fala)

Isso faz o contrário do que faz o SpeechRecognizer! Com esse recurso, usuários podem digitar textos e o aplicativo "falará" o que estiver escrito. Aqui está um [tutorial](#).

Sensores

Como vocês estão desenvolvendo um aplicativo para um dispositivo móvel, podem aproveitar todos os recursos e sensores que celulares têm. Esses recursos e sensores são exclusivos de dispositivos móveis, portanto eles podem não fazer parte de um computador comum. Esses recursos serão o que fazem o seu aplicativo um app mobile em vez de um aplicativo web ou até um site. Veja este vídeo sobre os tipos de sensores encontrados em smartphones!

<video>

O App Inventor ajuda a programar esses sensores! Pode ser que os seus celulares não tenham todas as coisas vistas no vídeo, mas é bastante provável que tenham os que serão discutidos aqui. Esta é uma lista de componentes que vocês podem programar com o App Inventor. Acessem o [site do MIT](#) para saber mais.

Acelerômetro

Esse componente determina se o celular está em movimento ou de cabeça para baixo. Esse recurso é útil para quando quiserem que a tela se oriente da maneira como o celular está sendo segurado, ou se quiserem que o aplicativo reaja a movimentos. Aqui está um [tutorial](#).

Pedômetro

Esse sensor usa o acelerômetro para medir quantos passos o usuário deu com o celular, além de estimar a distância percorrida. Aqui está um [tutorial](#).

Giroscópio

Esse componente capta a inclinação do celular. Ele é mais preciso que o acelerômetro e também pode medir alterações no grau de orientação do celular. Aqui está um [tutorial](#).

Relógio

Permite que seu aplicativo saiba as horas ou use um cronômetro. Esse componente é útil para configurar alarmes, cronometrar algo ou saber se é dia ou noite.

Alguns tutoriais:

- [Tutorial de cronômetro](#)
- [Documentação de relógio MIT](#)

Geolocalização

Esse sensor coleta dados de latitude e longitude da localização do celular. Pode ser útil para quando quiser buscar negócios ou pontos de interesse próximos ao usuário. Para saber mais, clique [aqui](#).

Sensor de proximidade

Diz se o celular está próximo do rosto de alguém ou não. No entanto, nem todos os celulares têm esse recurso. Aqui está um [tutorial](#).

Componentes sociais

Esses são recursos que permitem aos usuários fazer ligações, enviar e-mails, mensagens de texto e compartilhar coisas através do aplicativo. Acessem o [site do MIT](#) para saber mais.

ContactPicker (seletor de contatos)

Mostra os contatos do usuário e permite selecionar pessoas dessa lista. Aqui está um [tutorial](#).

EmailPicker (seletor de e-mail)

Permite ao usuário digitar um endereço de e-mail. Aqui está um [tutorial](#).

PhoneNumberPicker (seletor de número de telefone)

Permite ao usuário selecionar um número de uma lista de contatos, como amigos ou família. Aqui está um [tutorial](#).

PhoneCall (ligação telefônica)

Permite ao usuário fazer uma ligação através do aplicativo. Aqui está um [tutorial](#).

Texting (mensagens de texto)

Permite ao usuário enviar mensagens de texto a outros celulares através do aplicativo. Aqui está um [tutorial](#).

Sharing (compartilhamento)

Permite aos usuários compartilhar mensagens, imagens ou outras coisas do seu aplicativo com outros aplicativos em seu celular. Aqui está um [tutorial](#).

Twitter

Permite a comunicação entre seu aplicativo e o Twitter. Usuários podem pesquisar tweets, enviar e receber mensagens, obter listas de seguidores e mais. Aqui está um [exemplo](#).

Componentes de conectividade

Permite ao aplicativo interagir com outros programas, como a web ou outros aplicativos. Acessem o [site do MIT](#) para saber mais.

ActivityStarter (iniciar atividade)

Permite ao aplicativo iniciar outros aplicativos instalados no celular do usuário. Eles podem ser aplicativos criados no App Inventor ou não. Podem ser aplicativos como a Câmera e Mapas que já estejam instalados no dispositivo. Alguns tutoriais:

- [Tutorial 1](#)
- [Tutorial 2](#)
- [Tutorial 3](#)

Web

Permite ao aplicativo obter informações de sites. Com isso, vocês podem obter e utilizar informações sem precisar de um navegador. Aqui está um [tutorial](#).

BluetoothClient e BluetoothServer (cliente e servidor Bluetooth)

Esses componentes permitem ao aplicativo se conectar através de Bluetooth. Aqui está um [exemplo](#).

"Um dos segredos para permanecer jovem é fazer coisas que você não sabe fazer para continuar aprendendo".

– Ruth Reichl

Bases de dados online

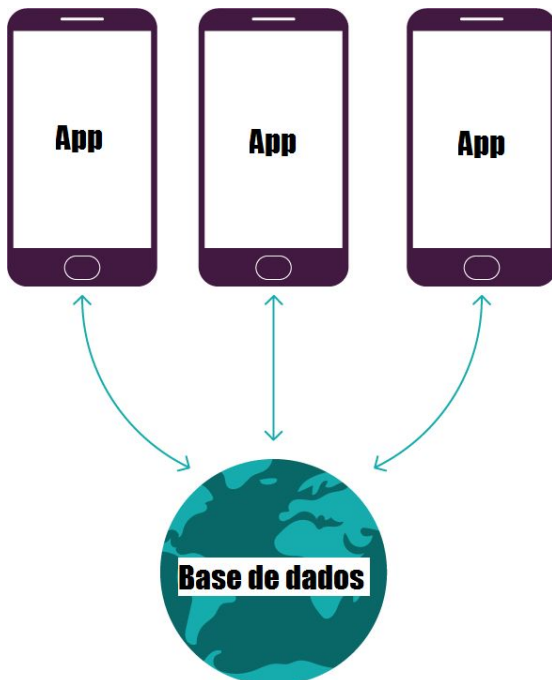
Uma das coisas que vocês definitivamente deveriam usar no aplicativo é a possibilidade do celular de se conectar com a internet. Pode ser mais que só acessar a internet ou o e-mail pelo aplicativo. É possível até armazenar dados na internet e acessá-los de mais de um celular com o aplicativo! Lembram-se da TinyDB, da unidade de programação 2? A TinyDB é uma base local de dados em

que seu aplicativo pode armazenar e obter dados. Quando um celular usa a TinyDB, ele armazena dados somente naquele celular, de modo que um usuário não consegue acessar a TinyDB de outro. Uma base de dados online funciona de forma um pouco diferente. Digamos que você tem dois celular com um aplicativo, o Celular A e o Celular B. Quando o Celular A armazena dados na TinyDB, o Celular B não consegue acessar, porque os dados estão no disco rígido do A. Com uma base de dados online, o A pode armazenar dados em uma base de dados e o B pode acessá-la e obter dados dela. Ao invés de a base de dados estar disponível em apenas um celular, os dois podem compartilhá-la com o aplicativo. Outra vantagem de utilizar bases de dados online com o App Inventor é que o programador (no caso, vocês) pode acessar e editar seu conteúdo pela internet.

Armazenamento local:



Armazenamento em base de dados online:



Bases de dados online podem ser atualizadas a qualquer hora e as mudanças se refletirão nos celulares. Digamos que queiram adicionar uma lista de restaurantes no TinyDB e colocar o aplicativo na loja de aplicativos. Qualquer pessoa que baixá-lo terá a mesma lista de restaurantes. Depois de um tempo, vocês aumentam o número de restaurantes na lista. Vocês atualizam o código e o aplicativo na loja. No entanto, os celulares que tiverem baixado o aplicativo antes disso não terão a lista atualizada de restaurantes. Esses usuários deverão deletar o aplicativo e baixá-lo novamente da loja! Se usar uma base de dados online para os restaurantes, o aplicativo atualizará automaticamente a lista. Cada vez que o celular se conectar à internet e acessar a base de dados, ele receberá a versão atual da lista.

Vocês já compartilharam fotos no Facebook ou Instagram? Esses aplicativos utilizam bases de dados online. Quando um amigo compartilha uma foto, todos os seus seguidores conseguem vê-la. Bases de dados online permitem compartilhar e enviar dados a todos através do aplicativo.

Então, quando devemos usar uma base de dados online? Alguns usos comuns:

- Compartilhar dados de jogos, como lista de recordes
- Permitir que usuários compartilhem imagens através de um feed
- Exibir um feed que atualiza o que os usuários veem
- Exigir login e senha para usuários
- Recordar tudo sobre um usuário, como suas transações, histórico financeiro ou médico

Por sorte, existem algumas maneiras de configurar uma base de dados online no App Inventor! As próximas seções discutirão algumas das bases de dados online que o App Inventor permite incorporar ao aplicativo com facilidade. Ao ler as seções, pensem sobre quais podem funcionar com o seu aplicativo.

Exercício para pensar

Vocês conseguem pensar em cinco aplicativos usados por vocês que dependam de bases de dados online?

TinyWebDB

A TinyWebDB é como a TinyWeb, porque usa pares de valores de tag para armazenar e obter dados. A TinyWebDB é boa para dados que normalmente aparecem em pares, como nomes de usuário e senhas. Ao configurar a TinyWebDB, vocês podem fazer calls para armazenar e obter dados a partir dela. Diferentemente da TinyDB, qualquer celular com o aplicativo pode armazenar e obter dados. A TinyWebDB também tem o bônus de poder ser acessada pela internet para ver as tags existentes e deletar as que não são mais necessárias. Seu provedor de internet deve se parecer com isso:

App Inventor (TinyWebDB) Web Database Service



This web service stores and retrieves values for an [App Inventor for Android](#) app. App Inventor apps can access this service using the TinyWebDB component and setting the ServiceURL to the URL of this site.

Search database for a tag

Tag:

Returned as value to TinyWebDB component: **Password1**

Store a tag-value pair in the database

Tag:

Value:

Key	Value	Created (GMT)	
User5	Password5	Nov. 18, 2016, 10:35 p.m.	Delete
User4	Password4	Nov. 18, 2016, 10:35 p.m.	Delete
User3	Password3	Nov. 18, 2016, 10:34 p.m.	Delete
User2	Password2	Nov. 18, 2016, 10:34 p.m.	Delete
User1	Password1	Nov. 18, 2016, 10:34 p.m.	Delete

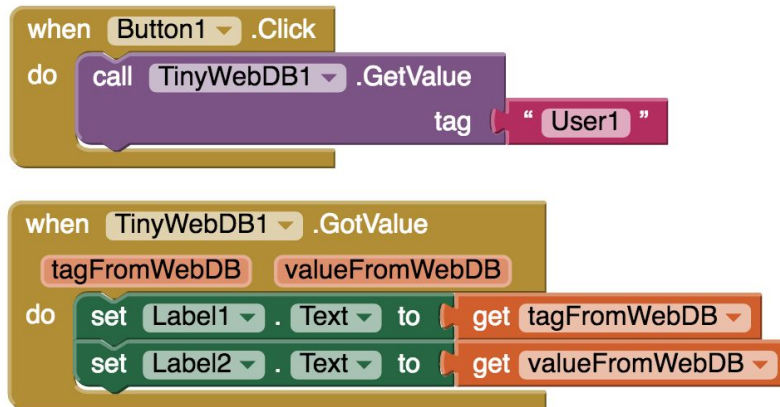
Percebam como temos a opção de deletar as tags que estão ali. É muito mais fácil gerir dados dessa forma, então, mesmo que não seja necessário que dados sejam compartilhados pelos celulares, a TinyWebDB pode ser melhor!

Para usar a TinyWebDB, é necessário configurar um provedor de internet próprio para armazenar e fornecer os dados. Seguem instruções do MIT para fazer isso: [TinyWebDB personalizada](#). Ao configurar sua TinyWebDB, pode ser que não consigam passar de um ponto, então aqui está um vídeo com as [instruções](#).

A TinyWebDB é um componente invisível, que não aparece na tela e é mais ou menos assim:



A programação da TinyWebDB é um pouco diferente da programação da TinyDB. Ao fazer uma call para obter dados, é necessário que o aplicativo solicite esses dados do provedor de internet. Assim que o aplicativo obtiver os dados, vocês podem fazer coisas com eles. Observem o código abaixo:



Ao apertar o botão Button1, o aplicativo solicita ao provedor de internet o valor da tag chamada "User1". Assim que o aplicativo obtiver o valor da TinyWebDB, ele definirá a Label1 e a Label2 para a tag e o valor. A função "when TinyWebDB1.GotValue" usa o último valor obtido da base de dados.

Solução de problemas para configurar a TinyWebDB:

- Se tiverem problemas para baixar o Google App Engine para Python, [cliquem aqui](#).
- Não é necessário colocar informações de cartão de crédito no Google, pois todos os serviços de nuvem que usaremos são gratuitos.
- Assim como a TinyDB, se usar o mesmo nome de tag duas vezes na TinyWebDB, os dados serão sobrescritos.

Fusion tables

Fusion tables são outro tipo de base de dados online que pode ser utilizado no App Inventor com facilidade. Fusion tables usam tabelas (tables), ao invés de pares de valores de tag, para armazenar e recuperar dados. Isso permite armazenar conjuntos mais complexos de dados. Assim como na TinyWebDB, vários usuários do aplicativo podem armazenar e obter valores de Fusion Tables, e vocês poderão gerir esses dados usando um provedor de internet.

Fusion tables permitem armazenar dados em uma tabela. Tabelas são ótimas quando há muitos dados para controlar e organizar em linhas e colunas. Vamos ver um exemplo para entender melhor.

Digamos que vocês estão encarregadas de pedir camisas novas para seu time de futebol. Todas recebem uma camisa com seu nome e número. Vocês precisam encontrar uma maneira de manter o controle do nome, número e tamanho da camisa de cada pessoa. A maneira mais fácil de se fazer isso é a seguinte:

Nome	Número	Tamanho de camisa
Marissa	2	G
Ruchi	6	M
Hadar	3	P
Stephanie	7	M
Iyonce	4	P

Ao invés de utilizar papel e caneta, vocês decidem usar o fusion tables para manter o controle dessas informações. Vocês criam um aplicativo e pedem às integrantes da equipe para baixá-lo. Elas podem informar o nome, número e tamanho de camisa de seus próprios celulares. As informações de seus celulares serão armazenadas na web por meio de uma fusion table. Vocês podem entrar, ver e editar as informações de todas!

Vejam a seguir alguns exemplos de outras equipes Technovation que usaram fusion tables, para que vocês entendam melhor como elas são usadas. Vejam se esses exemplos são similares ao que vocês querem fazer com o aplicativo!

- Criar um diretório de centros que estejam aceitando doações. As informações da tabela incluem latitude, longitude, números de telefone e itens que aceitam.
- Armazenar informações para criar perfis personalizados. Dados como local, nome de usuário e favoritos
- Fazer mapas e pontos em mapas, vejam o seguinte tutorial
- Visualizar dados em gráficos
- Armazenar informações de usuários
- Armazenar postagens e comentários em um fórum

Vejam estes dois aplicativos que usaram fusion tables

- "CramJam" e "BCNeeds".
 - [Vídeo demo CramJam](#)
 - [Vídeo do pitch da CramJam](#)
 - [Vídeo demo BCNeeds](#)
 - [Vídeo do pitch da BCNeeds](#)
- A "CramJam" usou para integrar o carregamento de Google Docs e imagens para visualizar anotações online. A "BCNeeds" usou para registrar pontos de coleta de serviços e bens de doações para caridade. Ambos usaram fusion tables para armazenar dados persistentes e integrá-los com seletores de lista

Aqui está um tutorial do MIT para começar a usar fusion tables: [Fusion table AI Pizza Party](#). Esse tutorial não nos ajudou muito, portanto, usamos [este vídeo](#) para ajudar.

Solução de problemas:

- Não é necessário colocar informações de cartão de crédito no Google, pois todos os serviços de nuvem que usaremos são gratuitos.
- Baixem o código-fonte do Pizza Party e brinquem com ele!

APIs

Digamos que vocês estão desenvolvendo um aplicativo que diz ao usuário como ele deve se vestir com base no tempo. Vocês podem passar horas carregando dados em uma fusion table, mas sabem que há muitos sites que informam às pessoas sobre o tempo lá fora. Uma solução mais fácil é pegar os dados desses sites e exibi-los aos usuários através do aplicativo!

Programadores fazem esse tipo de coisa o tempo todo. Isso é usar uma API, ou interface de programação de aplicativos (application programming interface, em inglês) de um site. Uma API é a maneira com que seu aplicativo pode conversar com a internet e obter as informações necessárias. É provável que vocês entrem o tempo todo em sites que utilizam APIs.

<video>

Há muitas informações na internet e também existem muitas maneiras de acessá-las. Estas são APIs que programadores usam com frequência:

Dados	Onde obter	Tutoriais para o App Inventor
Mapas	Google Maps	http://appinventor.mit.edu/explore/displaying-maps.html
Mercado de ações	Yahoo! Finance	http://explore.appinventor.mit.edu/ai2/stockquotes
Tempo	National Weather Service	https://appinventorapi.com/weather/

Vocês devem usar APIs sempre que quiserem obter informações que souberem que existem em outro site ou base de dados.

Atividade: Explorar APIs

Pensem sobre como uma API pode ser útil para o aplicativo. Perguntas a se fazer:

- De que tipos de dados eu precisarei para meu aplicativo funcionar?
- De onde posso obter esses dados?

Desafio de programação 4:

Escolham: 1- um componente mobile, e 2- uma base de dados online ou uma API que sejam úteis para o aplicativo. Configurem e pratiquem o uso de ambas as escolhas. Isso ajudará na hora de

desenvolver o aplicativo final!

Reflexão

Após revisar essa aula, vocês terão uma boa ideia de quais componentes precisarão usar no desenvolvimento do aplicativo para que ele faça o que querem fazer. Parabéns!

Provavelmente, vocês passaram muito tempo nas semanas seguintes configurando esses recursos no aplicativo. Se houver componentes que precisem, mas não tenham visto na unidade, vejam [aqui](#).

Recursos adicionais

- [Tutorial Thunkable Google Maps](#)
- [Vídeo de demonstração de componentes Spinner](#)