# Livro do Educador

## Desenvolvimento de Aplicativos Móveis

### iOS

# Unidade 6

## Aula 1 – Firebase

Há uma grande diferença no trabalho do desenvolvedor de apps front-end para o desenvolvedor de serviços web (back-end) mas é essencial que estes dois recursos interajam entre si. Para o desenvolvedor de apps, o mundo do back-end parece muito distante, mas o Firebase veio para acabar com isso.

### CARGA HORÁRIA

Conforme o plano de aula, esta aula terá duração de 1h30 e deverá ser conduzida de acordo com as orientações pedagógicas.

### OBJETIVO DA AULA

Ao final da aula, você deverá garantir que o aluno tenha subsídios para

* Saber as funcionalidades do Firebase;
* Saber as funcionalidades do Firebase Database e Firebase;
* Criar um projeto Firebase.

### ORIENTAÇÕES PEDAGÓGICAS

Para atender os objetivos de aprendizagem, você deverá conduzir o processo de ensino considerando a organização didática apresentada a seguir:

* 40 minutos de aula expositiva;
* 10 minutos para tirar as dúvidas dos alunos;
* 40 minutos para desenvolver as atividades propostas para a turma e apresentação da TDP.

### TÓPICOS DE ESTUDO

Todos os tópicos a seguir, conforme livro do aluno, devem ser trabalhados de forma dinâmica, criativa, com embasamento teórico e prático voltado ao mercado de trabalho.

* Recursos Firebase;
* Firebase Database
* Firebase Authentication;
* Criando um Projeto Firebase;

### PONTOS IMPORTANTES

É importante que o aluno se atente a essa aula pois será apresentados bastante conteúdo teórico, que são de crucial entendimento para o decorrer da aula.

### Recursos do Firebase

Educador, inicie esta aula relembrando os conceitos de front-end e back-end, vistos em Android. Explique também a necessidade de utilizarmos um back-end dinâmico para que se faça um bom app.

Explique também da sobre a grande carência de serviços back-end para aumentarmos muito mais as possibilidades para criação de nossos apps. Porém carecemos de conhecimentos (que são bastante exigentes e específicos) e principalmente de tempo para desenvolvermos nós mesmo qualquer que seja o serviço web que necessitamos, para isso temos o Firebase.

Educador, de uma introdução de Firebase dizendo que ele é uma plataforma web para a centralização de dados e informações de uma aplicação. No próximo item falaremos um pouco mais. O que é o Firebase

Educador, explique um pouco melhor o que é o Firebase, como ele ajuda na melhora, abrangência e lucro do app e também fale um pouco sobre a história do Firebase e o que a atual detentora, Google tem feito por essa plataforma.

No Firebase existem diversos recursos que funcionam de forma independente. Alguns dos recursos são: a rapidez de desenvolvimento, graças a recursos complementares que podemos combinar com nossas necessidades; é possível criar sem se preocupar com infraestruturas complexas; temos o Firebase Analytics que analisa dados coletados e toma decisões baseadas neles; é multiplataforma e gratuito.

### 1.2. Firebase Database

Uma das duas ferramentas que iremos utilizar neste curso é o Firebase Database, que é um banco de dados hospedado na nuvem, onde é possível sincronizar dados com todos clientes conectados em tempo real. Educador, termine a explicação definindo de forma básica o fluxo de dados e seu aspecto multiplataforma.

#### 1.2.1. Principais funcionalidades

Educador, aqui temos três principais funcionalidades a serem explicadas, onde a principal é a sincronização de dados em tempo real. A segunda é a funcionalidade dos aplicativos serem responsivos mesmo off-line e a última é a possibilidade de acesso ao Firebase Realtime Database sem a necessidade de possuir um servidor, de maneira segura.

#### 1.2.2. Como funciona?

Resumindo, o Firebase Databe funciona da seguinte forma:

Através do código no front-end, obtemos, online, acesso ao banco de dados de maneira segura. Estes dados são mantidos localmente, podendo ser acionados em tempo real, mesmo off-line, assim como eventos. Quando o dispositivo torna a ficar online ele se sincroniza co o banco de dados remoto e atualiza o conteúdo local.

Também explique sobre o conjunto de regras que tornam segura a estruturaçãodos dados e quando podem ser lidos ou gravados. Por ultimo, fale sobre a ótima capacidade de resposta que o Firebase Database pode oferecer e de sua integração com o Firebase Authentication.

### 1.3. Firebase Authentication

Educador, explique ao aluno que o Firebase Authentication fornece serviços de backend, SDKs fáceis de usar e bibliotecas de UI para autenticar os usuários em seu aplicativo, prontas para o uso. Ele oferece suporte a autenticação usando senhas, provedores populares de identidades de identidades federadas, como o Google, Facebook e Twitter, e muito mais. Além de tudo ele é facilmente integrado com backends personalizados.

#### 1.3.1. Principais funcionalidades

Educador, apresente as principais funcionalidades do Firebase Authentication como sendo: autenticação baseada em e-mail e senha; integração de provedores de identidades federadas; integração de sistemas de autenticação personalizados; autenticação anônima, entre outros. De uma breve explicação de cada uma destas funcionalidades.

#### 1.3.2. Como funciona?

Educador, tente explicar que o usuário fornece suas credenciais, que são passadas para o Firebase Authentication SDK. Seus serviços verificarão essas credenciais e retornarão com uma resposta ao cliente, permitindo a ele que acesse informações básicas de perfil. Os usuários autenticados podem ver e gravar dados no Firebase Realtime Database e Firebase Storage.

### 1.4. Criando um projeto no Firebase

Educador, apresente o Firebase Console, que é onde podemos encontrar documentações e tutoriais e gerenciar suas aplicações online, de onde estiver. Explique um pouco mais sobre o Firebase Console e, já que adicionamos o Firebase na aula passada via CocoaPods, vamos criar um novo projeto nele.

#### 1.4.1. Criando um novo projeto no Firebase Console

Para criar um novo projeto no Firebase:

* Entrar na página: <https://firebase.google.com/?hl=pt-br> , fazer login na conta Google (se necessário, utilizar a conta fornecida pela escola) ou criar uma e em seguida, clicar em **CRIAR NOVO PROJETO**. Utilizar o nome MusicProject e selecionar o Brasil em País/Região. Clicar em **CRIAR PROJETO**
* Observar a tela que será aberta e partir para as configurações.

#### 1.4.2. Configurando o projeto no Xcode

Para configurar o Firebase para se comunicar com nosso aplicativo:

* Clicar na opção **Adicionar Firebase ao seu aplicativo iOS**  e preencher o campo **Código do pacote do iOS**  na tela seguinte. Para obter este código, abrir o projeto **MusicProject**  do Xcode e localizar o arquivo de configuração de projeto, como mostrado no passo 3, e copiar o código em **Bundle ID**  e copiar para o campo de código no Firebase. Clicar em **ADICIONAR APLICATIVO**.
* Adicionar o arquivo **GoogleService-Info.plist** no seu projeto como está descrito na imagem do site. Na tela de configurações que aparecerá no Xcode, escolher a opção **Copy items if needed**  e clicar em  **FINISH**. Observar que o arquivo foi instalado, como mostrado no passo 8.
* Ao solicitar a instalação do Pod, pule, pois já o fizemos na unidade passada.
* Para inicializar o código no nosso projeto, copiar o código fornecido pelo Firebase abaixo da classe principal App Delegate, como no passo 10.

Assim podemos utilizar o serviço Firebase Auth e o Firebase Database, para fazer login, obter JSONs com informações de músicas respectivamente e Firebase Storage.

### 1.5. Resumo

Nesta aula, o aluno aprendeu o que é o Firebase, uma ferramenta poderosa e útil para criarmos apps inteligentes, responsivos e dinâmicos, centralizar as informações da nossa aplicação, assim o app em cada plataforma só se preocupa em exibi-la ou manda-la para o back-end do Firebase. Aprendeu um pouco sobre as ferramentas Firebase Database e Firebase Authentycation. Por fim, o aluno aprendeu a instalar e configurar um projeto no Firebase.

## Aula 2

### 2. Login com Firebase

Chegou a hora de aprendermos cada método responsável para fazer login de usuários

### CARGA HORÁRIA

Conforme o plano de aula, esta aula terá duração de 1h30 e deverá ser conduzida de acordo com as orientações pedagógicas.

### OBJETIVO DA AULA

Ao final da aula, você deverá garantir que o aluno tenha subsídios para

* Compreender conceitos de Firebase Authentication e autentificação de usuário;
* Aprender conceitos de Usuários e Firebase User;
* Utilizar diversos métodos para realização de login.

### ORIENTAÇÕES PEDAGÓGICAS

Para atender os objetivos de aprendizagem, você deverá conduzir o processo de ensino considerando a organização didática apresentada a seguir:

* 40 minutos de aula expositiva;
* 10 minutos para tirar as dúvidas dos alunos;
* 40 minutos para desenvolver as atividades propostas para a turma e apresentação da TDP.

### TÓPICOS DE ESTUDO

Todos os tópicos a seguir, conforme livro do aluno, devem ser trabalhados de forma dinâmica, criativa, com embasamento teórico e prático voltado ao mercado de trabalho.

* Firebase Authentication;
* Gerenciar usuários no Firebase;
* Implementando login por email e senha.

### PONTOS IMPORTANTES

É importante que o aluno se atente a essa aula pois será apresentado bastante conteúdo teórico, que são de crucial entendimento para o decorrer da aula.

### 2.1. Firebase Authentication

Educador, explique que o Firebase Authentication fornece serviços de backend SDKs fáceis de usar e bibliotecas de IU prontas para autenticar os usuários em seu aplicativo. Também tem a opção de autenticar usuários entrando com seus perfis de redes sociais.

#### 2.1.1. Usuários em projetos do Firebase

Para que o aluno tenha uma base sobre o assunto, é preciso que você defina o que é um objeto Firebase User e que ele é independente do Firebase Auth. Apresente algumas propriedades do usuário como um ID exclusivo, um endereço de e-mail primário, um nome e um URL de foto e a possibilidade de adicionar propriedades adicionais, armazenando-as em seu Firebase Realtime Databe.

Também é importante que o aluno saiba quais dados são preenchidos quando o usuário se inscreve em seu app, mais a frente serão solicitados nas implementações.

#### 2.1.2. Provedores de login

Sobre provedores de login, o aluno deverá estar ciente de que é possível que o usuário se conecte em seu app de formas variadas, como através de e-mail e senha, Google Sign-In, sistemas de autentificação personalizados, etc. Com tudo, as informações vazias decorrentes da variação de formas de autenticação, podem ser fornecidas por um provedor, já que um objeto Firebase User acompanha todos os provedores vinculados ao usuário.

#### 2.1.3. Usuário atual

O aluno também precisa saber que, quando um usuário faz login, ele não perderá suas informações de usuário ao atualizar a página em um navegador ou reiniciar o aplicativo, isto porque a instância de Firebase Auth persiste o estado do usuário. Já quando realiza log-out, a instancia de Auth deixa de ser referenciada e o usuário perde suas informações de perfil.

#### 2.1.4. Ciclo de vida do usuário

Educador, discorra sobre o ciclo de vida do usuário. Ele é descrito através do uso de ouvintes que são notificados sempre que algo relevante acontece com o objeto Auth. De alguns exemplos de notificações, oferecendo explicações objetivas.

#### 

### 2.2. Gerenciar usuários no Firebase

#### 2.2.1. Criar um usuário

A partir de agora vamos mostrar a maneira de se implementar diversos elementos de um gerenciador de usuário. Começando em chamar o método createUser(withEmail:password:completion:) ou conecte um usuário pela primeira vez usando um provedor de identidades federadas, como o Google Sign-In ou o login do Facebook. Para criar um usuário.

#### 2.2.2. Obter o usuário atualmente conectado

A maneira recomendada de obter o usuário atual é definir um ouvinte no objeto Auth:

|  |  |
| --- | --- |
| Código | Descrição |
| FIRAuth.auth()?.addStateDidChangeListener { auth, user in  if let user = user {  // O usuário está logado.  } else {  // O usuário não está logado.  }  } | Definição de um ouvinte para obter um estado do usuário. |
| if let user = FIRAuth.auth()?.currentUser {  // O usuário está logado.  } else {  // O usuário não está logado.  } | Definição de um ouvinte para obter um estado do usuário. Se o usuário não estiver conectado, currentUser será nulo. |

#### 2.2.3. Obter o perfil de um usuário

Para obter as informações de perfil de um usuário:

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| if let user = FIRAuth.auth()?.currentUser {  let name = user.displayName  let email = user.email  let photoUrl = user.photoURL  let uid = user.uid;  } else {  // O usuário não está logado.  } | Obtém informações do perfil do usuário se ele estiver logado. |

#### 2.2.4. Obter as informações de perfil específicas de provedor de um usuário

Para obter informações espécíficas de provedores, fazemos:

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| if let user = FIRAuth.auth()?.currentUser {  for profile in user.providerData {  let providerID = profile.providerID  let uid = profile.uid; // Provider-specific UID  let name = profile.displayName  let email = profile.email  let photoURL = profile.photoURL  }  } else {  // O usuário não está logado.  } | Obtem informações de perfil recuperadas de provedores de login, com o providerData, caso o usuário esteja logado. |

#### 2.2.5. Atualizar o perfil de um usuário

Com a classe FIRUserProfileChangeRequest, é possível atualizar as informações de perfil de usuários.

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| let user = FIRAuth.auth()?.currentUser  if let user = user {  let changeRequest = user.profileChangeRequest()    changeRequest.displayName = "Jane Q. User"  changeRequest.photoURL =  NSURL(string: "https://example.com/jane-q-user/profile.jpg") as URL?  changeRequest.commitChanges { error in  if let error = error {  // Um erro aconteceu.  } else {  // Perfil atualizado.  }  }  } | O método changeRequest.commitChanges permite que possamos atualizar informações básicas de perfil. |

#### 2.2.6. Definir o endereço de e-mail de um usuário

Pode-se definir o endereço de e-mail de um usuário com o método updateEmail(completion:). Por exemplo:

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| let user = FIRAuth.auth()?.currentUser  user?.updateEmail("user@example.com") { error in  if let error = error {  // Um erro aconteceu.  } else {  // Email atualizado.  }  } | O método updateEmail() permite que possamos definir nosso emaill. |

#### 2.2.7. Definir a senha de um usuário

Semelhante, podemos definir a senha de um usuário com o método updatePassword(completion:). Por exemplo:

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| let user = FIRAuth.auth()?.currentUser  let newPassword = getRandomSecurePassword()  user?.updatePassword(newPassword) { error in  if let error = error {  // Um erro aconteceu.  } else {  // Password atualizado.  }  } | O método updatePassword() permite que possamos definir nossa senha. |

#### 2.2.8. Enviar um e-mail de redefinição de senha

Você pode enviar um e-mail de redefinição de senha para um usuário com o método sendPasswordResetWithEmail:completion:. Por exemplo:

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| let email = "user@example.com"  FIRAuth.auth()?.sendPasswordReset(withEmail: email) { error in  if let error = error {  // Um erro aconteceu.  } else {  // E-mail de redefinição de senha enviado.  }  } | O método sendPasswordReset () é usado para enviar email de redefinição de senha. |

#### 2.2.9. Excluir um usuário

Podemos excluir uma conta de usuário com o método deleteWithCompletion. Por exemplo:

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| let user = FIRAuth.auth()?.currentUser  user?.delete { error in  if let error = error {  // Um erro aconteceu.  } else {  // Conta deletada  }  } | O método delete() permite que possamos excluir um usuário. |

#### 2.2.10. Reautenticar um usuário

Para reautenticar usuários usamos o seguinte método:

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| let user = FIRAuth.auth()?.currentUser  var credential: FIRAuthCredential  user?.reauthenticate(with: credential) { error in  if let error = error {  // Um erro aconteceu  } else {  // Usuário re-autenticado.  }  } | O método reauthenticate() permite que possamos reautenticar um usuário. |

### 2.3. Implementando login por e-mail e senha

Nesta aula iremos implementar o básico para obtermos um login via e-email e senha utilizando o Firebase.

* Primeiramente, abrir o material de partida em **Unidade\_6\_-\_Aula\_2\_-\_Exemplo\_1.zip**

#### 2.3.1. Ativando o provedor de login

Para habilitar a opção de login via email/senha no Firebase console. Siga os passos:

* Entrar no Firebase Console e em seguida no projeto MusicProject. Clicar na opção Athentication no menu lateral e em CONFIGURAR MÉTODO DE LOGIN.
* Selecionar o provedor de login **E-mail/senha**, ativá-lo e Finalizar.

#### 2.3.2. Instalando dependências

Para instalar o serviço de autenticação via CocoaPods:

* Abrir o Podfile e adicionar mais um pod com o código mostrado no passo 1. Observar a mudança de seu profile. Em seguida, abrir o terminal e instalar o comando digitando o código como mostra o passo 3.

#### 2.3.3. Implementando os métodos do SDK Firebase Authentication

Para implementar os métodos do Firebase Auth,

* Observar a tela de login que foi criada
* Abrir o arquivo **LoginViewController.swift**, e importar a biblioteca Firebase, em seguida inserir a linha conforme o passo 1.
* Localizar o método de cadastro registerButtonTouchUpInside() e inserir o método mostrado no passo 3. Explicar o código (que mostra certo alerta) e utilizar o método de validação descrito no passo 4.
* Acrescentar mais o código dos passos 5 e 6 para chamar o serviço do Firebase e tratar as mensagens de erro do cadastro.
* Executar o app e realizar o processo de cadastro. Em seguida, observar no Firebase o novo usuário cadastrado. O processo de login é similar.

Para implementar o recurso de login automático seguir os passos:

* Abrir o arquivo **AppDelegate.swift**, localizar o método **application(didFinishLaunchingWith Options:)**  e inserir as instruções mostradas no passo 2.
* No Storyboard Main, configurar a MainViewController como Initial View Controller. Explicar o que esse trecho faz.