

[< VOLTAR](#)

Criação das funcionalidades avançadas

Nessa aula vamos aprender a recuperar dados do Firebase, ordená-los e exibi-los de forma amigável aos usuários, realizar muitos testes em nosso aplicativo para finalizá-lo. Será disponibilizado, também, os arquivos criados no projeto ao final.

NESTE TÓPICO

- › Widget de listagem das mensagens
- › Recuperando os dados no Firebase
- › Execução da aplicação e testes
- › Download do projeto completo
- › Referências



Widget de listagem das mensagens

Agora que já temos toda lógica de autenticação e de envio de mensagens pronta, podemos começar a listar as mensagens, ou seja, recuperá-las do Firebase. E lembre-se que essa recuperação não é apenas das mensagens, mas das imagens, também.

Para realizar a recuperação dos dados no Firebase basta uma linha de código, porém a lista de mensagens (textuais e de imagens) precisa ser devidamente montada e trabalhada para ser exibida.

Por isso, vamos começar desenvolvendo o Widget que será responsável por exibir cada uma das mensagens, já nos preocupando com a formatação dele, especialmente se a mensagem for do próprio usuário autenticado ou de algum outro usuário, para diferenciá-las e deixar nossa aplicação ainda mais bonita.

O vídeo abaixo mostra como desenvolver o Widget para as mensagens.

Widget de listagem das mensagens



Com este Widget pronto, podemos partir para a lógica de recuperação das mensagens, portanto.

Recuperando os dados no Firebase

Agora que já temos o Widget que mostrará uma mensagem, precisamos montar a lista de mensagens e exibi-las o utilizando.

Para isso, vamos recuperar todas as mensagens do Firebase, mas o próprio sistema cuidará para que não venham todas as mensagens simultaneamente, para não sobrecarregar o aplicativo. Serão carregadas apenas as necessárias para exibição.

Para isso precisamos utilizar um recurso novo que é o "builder" que retorna um Widget para cada iteração, automaticamente para nós, evitando assim um código grande e sujo.

Para entendermos melhor este processo, veja o vídeo abaixo que mostra como as mensagens são recuperadas para uma lista e configuradas para o Widget da sessão anterior.



Listagem Das Mensagens Firebase



Execução da aplicação e testes

Agora que praticamente já finalizamos a aplicação, podemos acertar alguns detalhes finais, executar em um Android em um iOS (se aplicável) e realizar todos os testes necessários.

Para isso o vídeo abaixo mostra tais pequenos ajustes e a execução em dois dispositivos simultaneamente.



Execução do App, pequenas correções e testes



Download do projeto completo

Como você pode observar este projeto é um pouco mais completo e complexo que os anteriores, por ter vários mecanismos diferentes de interação e, por isso, fora dividido em mais de uma aula.

Caso você tenha qualquer tipo de dúvida, você pode baixar o projeto completo abaixo para consultas, mas não deixe de o desenvolver acompanhando cuidadosamente cada um dos vídeos disponibilizados para esse projeto.

É muito importante mencionar que o projeto abaixo está sem os arquivos do Firebase (iOS e Android) pois isso é individual de cada conta e não adiantaria o professor disponibilizar o arquivo do Firebase utilizado. As chaves de segurança e acesso são totalmente diferentes, mas o projeto pode ser facilmente configurado para ser executado, bastando a inclusão destes arquivos, baseados em sua conta. Dentro deste arquivo comprimido você encontrará todos os arquivos dart (código-fonte), a pasta de assets com o logo do Google e o arquivo pubspec.yaml. O projeto não pode ser aberto no Android Studio, mas os arquivos podem ser utilizados para referência.

Lembre-se, também, que os arquivos de conexão com o Firebase são arquivos de depuração e desenvolvimento. Quando você tiver um projeto pronto para ser publicado na Apple Store na Play Store, você precisará de contas de desenvolvedor em cada um dessas plataformas (essas contas são pagas) e com a conta de desenvolvedor você poderá gerar novas chaves de release (publicação), que são comerciais e de produção, com o SHA-1 gerado automaticamente pelo próprio container (servidor) do Google.

MATERIAL COMPLEMENTAR

(<https://img.uninove.br/static/0/0/0/0/0/9/3/0/4/6/9304652/Uni-Zap-2.7z>)



Quiz

Exercício Final

Criação das funcionalidades avançadas

INICIAR ➤

Referências

DART. **Dart documentation**. *Site*. Disponível em: <https://dart.dev/>. Acesso em: 08 dez. 2020.

MATERIAL DESIGN. **Material Design documentation**. *Site*. Disponível em: <https://material.io/>. Acesso em: 08 dez. 2020.

FLUTTER. **Flutter docs**. *Site*. Disponível em: <https://flutter.dev>. Acesso em: 09 dez.2020.

WINDMILL, Eric. **Flutter in action**. Nova Iorque: Manning publications, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-in-action/9781617296147/>. Acesso em: 08 dez. 2020.

SINHA, Sanjib. **Quick start guide to Dart programming**: create high performance applications for the web and mobile. Lompoc, CA, EUA: Apress, 2019. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/quick-start-guide/9781484255629/>. Acesso em: 09 dez. 2020.

ALESSANDRIA, Simone. **Flutter projects**: a practical, project-based guide to building real-words cross-platform mobile applications and games. Birmingham, Reino Unido: Packt Publishing, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-projects/9781838647773/>. Acesso em: 09 dez. 2020.

ZACCAGNINO, Carmine. **Programming Flutter**. [s.l.]: The Pragmatic Bookshelf, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/programming-flutter/9781680507621/>. Acesso em: 09 dez. 2020.



Avalie este tópico



ANTERIOR

<

Criação da funcionalidades principais

Biblioteca

Criação da funcionalidades principais

(<https://www.uninove.br/conhec>

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Índice

≡

Ajuda?

(<https://ava.un>

Proximo

Curso=>

>

© Todos os direitos reservados

>