

 06

## Faça como eu fiz

### Mensagens personalizadas com o Resource

Modifique a classe `Resource` para receber uma property `erro` do tipo `String?` com o valor padrão `null`.

Então, ao invés do `Boolean`, utilize o `Resource<Boolean>` como valor do `LiveData`. No momento que tiver sucesso, devolva apenas o `true`, na falha, devolva `false` e um erro personalizado.

Modifique os demais locais do código para que lidem com `LiveData<Resource<Boolean>`.

No Fragment de cadastro, adapte a observação do `LiveData` para apresentar a mensagem de sucesso se o dado do `Resource` for `true` e a mensagem personalizada caso for `false`.

Teste o App e confira se apresenta o comportamento esperado.

### Identificando exceptions de cadastro

Identifique as possíveis Exceptions lançadas ao criar um usuário com e-mail e senha.

Para cada Exception, devolva uma mensagem de erro específica para identificar o problema ocorrido, considerando que as exception são:

- `FirebaseAuthWeakPasswordException`
- `FirebaseAuthInvalidCredentialsException`
- `FirebaseAuthUserCollisionException`

Também adicione uma mensagem padrão ao identificar uma exception diferente dessas.

Após implementar o código, teste o App e simule os problemas, então, confira se apresenta as mensagens esperadas.

### Evitando problema de campos vazios

Evite o crash do App ao enviar os campos vazios para o Firebase Authentication. Para isso, faça o seguinte:

- Envolva o método de cadastro com o try catch capturando a `IllegalArgumentException`;
- Ao capturar a exception, atribua um `Resource` com dado `false` para o `LiveData` e indique a mensagem de falha pelos campos vazios.

Em seguida, teste o App e confira se ao tentar cadastrar com os campos vazios apresenta a mensagem indicando o problema sem quebrar o App.

Após o teste, também adicione validações para cada campo. Para isso, faça o seguinte no listener do botão cadastrar:

- Limpe o erro de todos os campos;
- Verifique se o valor de cada campo é vazio, se sim, apresente a mensagem de erro indicando a obrigatoriedade do campo e atribua `false` ao marcador de validação (variável `valido`);

- Chame o cadastro do ViewModel apenas se o marcador for `true`.

Após implementar, teste o App e confira se apresenta os comportamentos esperados ao tentar cadastrar usuários sem informação de e-mail ou senha.