

OAT 1 – Descobrindo Kotlin

Autor: Joalisson Maia
Jadson Sobrinho
Mateus Queiroz
Murilo Marinho

Agenda

Apresentar funcionalidades da
OAT 1 - Descobrindo Kotlin

1. Cadastrar Usuário Comum:

Permite se registrar informando nome, data de nascimento, sexo e e-mail, assim como senha.

3. Reativar Usuário:

Permite informar usuário e senha e caso confirmado, o perfil deve ser reativado.

2. Visualizar Perfil do Usuário:

Permite visualizar nome completo, a data de nascimento (como a idade impressa do lado em anos, meses e dias) e e-mail.

4. Cadastrar Usuário Organizador:

Permite informar dados do usuário comum, quanto outros como, CN PJ, Razão Social, e Nome Fantasia.

Cadastrar Usuário Comum



- Dados para o registro
- Falar é bom, mostre-me o código
- Validação

--- DENDÊ EVENTOS ---

1. Login / Reativar

2. Cadastro Comum

3. Cadastro Org

0. Sair

Opção:

Cadastrar Usuário Comum



- Dados para registro:

- Nome
- Data de Nascimento
- Sexo
- Email

```
--- DENDÊ EVENTOS ---
```

```
1. Login / Reativar
```

```
2. Cadastro Comum
```

```
3. Cadastro Org
```

```
0. Sair
```

```
Opção: 2
```

```
Digite seu Email: joca@gmail.com
```

```
Digite seu Nome: joca
```

```
Digite sua Data de Nascimento no formato (DD/MM/YYYY): 06/10/2003
```

```
Digite seu Sexo: Masculino
```

```
Digite sua Senha: 123
```

```
Cadastrado com sucesso!
```

```
--- DENDÊ EVENTOS ---
```

```
1. Login / Reativar
```

```
2. Cadastro Comum
```

```
3. Cadastro Org
```

```
0. Sair
```

```
Opção: |
```

Cadastrar Usuário Comum



- Código

```
print("Digite seu Email: ")
val emailRespostaMenu = readln()

//Ele verifica em qualquer usuário se tem o email que vem do menu
if (usuariosListaMutavel.any { it.usuarioEmail == emailRespostaMenu }) println("Erro: Email já cadastrado")
else {
    print("Digite seu Nome: ")
    val nomeRespostaMenu = readln()

    print("Digite sua Data de Nascimento no formato (DD/MM/YYYY): ")
    val dataNascimentoRespostaMenu = LocalDate.parse( text = readln(), formatter = formatarData)

    print("Digite seu Sexo: ")
    val sexoRespostaMenu = readln()

    print("Digite sua Senha: ")
    val senhaRespostaMenu = readln()

    val tipoRespostaMenu = if (opcoesMenu == "3") TipoUsuario.ORGANIZADOR else TipoUsuario.COMUM
    val novoUsuario = Usuario(
        usuarioEmail = emailRespostaMenu,
        usuarioNome = nomeRespostaMenu,
        usuarioDataNascimento = dataNascimentoRespostaMenu,
        usuarioSexo = sexoRespostaMenu,
        usuarioSenha = senhaRespostaMenu,
        usuarioTipo = tipoRespostaMenu)

    usuariosListaMutavel.add(novoUsuario); println("Cadastrado com sucesso!")
}
```

Cadastrar Usuário Comum



- Validação

```
--- DENDÊ EVENTOS ---
1. Login / Reativar
2. Cadastro Comum
3. Cadastro Org
0. Sair
Opção: 2
Digite seu Email: joca@gmail.com
Digite seu Nome: joca
Digite sua Data de Nascimento no formato (DD/MM/YYYY): 06/10/2003
Digite seu Sexo: masculino
Digite sua Senha: 123
Cadastrado com sucesso!

--- DENDÊ EVENTOS ---
1. Login / Reativar
2. Cadastro Comum
3. Cadastro Org
0. Sair
Opção: 2
Digite seu Email: joca@gmail.com
Erro: Email já cadastrado.

--- DENDÊ EVENTOS ---
1. Login / Reativar
2. Cadastro Comum
3. Cadastro Org
0. Sair
Opção: |
```

Visualizar Perfil do Usuário



- Retorno dos dados
- Falar é bom, mostre-me o código

Visualizar Perfil do Usuário



- Retorno dos dados:
 - Faz o login
 - Escolhe opção de perfil
 - Visualiza os dados

```
--- DENDÊ EVENTOS ---
```

```
1. Login / Reativar
```

```
2. Cadastro Comum
```

```
3. Cadastro Org
```

```
0. Sair
```

```
Opção: 1
```

```
Email: joca@gmail.com
```

```
Senha: 123
```

```
Olá, joca!
```

```
--- MENU (joca) ---
```

```
1. Perfil
```

```
2. Alterar Perfil
```

```
3. Inativar
```

```
0. Logout
```

```
4. Feed
```

```
5. Comprar
```

```
6. Meus Ingressos
```

```
7. Cancelar Ingresso
```

```
Opção: 1
```

```
Nome: joca | Email: joca@gmail.com
```

```
Idade: 22 anos, 4 meses, 13 dias
```


Visualizar Perfil do Usuário

- Falar é bom, mostre-me o código:

```
"2" -> {  
    //ALTERAR PERFIL  
    print("Deixe em branco para não alterar.")  
    print("Novo nome: ")  
  
    usuarioLogado.usuarioNome =  
        readln().takeIf {  
            it.isNotBlank()  
        }?.let {  
            usuarioLogado.usuarioNome = it  
        }.toString()  
  
    print("Nova senha: ")  
    usuarioLogado.usuarioSenha = readln().takeIf {  
        it.isNotBlank()  
    }?.let {  
        usuarioLogado.usuarioSenha = it  
    }.toString()  
}
```

```
    print("Novo sexo: ")  
    usuarioLogado.usuarioSexo = readln().takeIf {  
        it.isNotBlank()  
    }?.let {  
        usuarioLogado.usuarioSexo = it  
    }.toString()  
  
    print("Nova Data de Nascimento no formato (DD/MM/YYYY): ")  
    usuarioLogado.usuarioDataNascimento = LocalDate.parse( text = readln(), formatter = formatarData)  
  
    if (usuarioLogado.usuarioTipo == TipoUsuario.ORGANIZADOR) {  
        print("Novo Nome Fantasia: "); readln().takeIf {  
            it.isNotBlank()  
        }?.let {  
            usuarioLogado.usuarioNomeFantasia = it  
        }  
    }  
  
    println("Dados atualizados!")  
}
```

Reativar Usuário



- Passo a passo
- Falar é bom, mostre-me o código

```
--- MENU (joca) ---
```

```
1. Perfil
2. Alterar Perfil
3. Inativar
0. Logout
4. Feed
5. Comprar
6. Meus Ingressos
7. Cancelar Ingresso
Opção: 3
Conta inativada.
```

```
--- DENDÊ EVENTOS ---
```

```
1. Login / Reativar
2. Cadastro Comum
3. Cadastro Org
0. Sair
Opção: 1
Email: joc@gmail.com
Senha: 123
Erro: Credenciais inválidas.
```

```
--- DENDÊ EVENTOS ---
```

```
1. Login / Reativar
2. Cadastro Comum
3. Cadastro Org
0. Sair
Opção: 1
Email: joc@gmail.com
Senha: 123
Conta inativa. Reativar? (S/N): S
Reativada!
```

```
--- MENU (joca) ---
```

```
1. Perfil
2. Alterar Perfil
3. Inativar
0. Logout
4. Feed
5. Comprar
6. Meus Ingressos
7. Cancelar Ingresso
Opção: |
```

Reativar Usuário

- Falar é fácil, mostre-me o código:

```
"1" -> {  
    //LOGIN & REATIVAR  
    print("Email: "); val emailRespostaMenu = readln()  
    print("Senha: "); val senhaRespostaMenu = readln()  
  
    val usuarioLista = usuariosListaMutavel.find { it.usuarioEmail == emailRespostaMenu && it.usuarioSenha == senhaRespostaMenu }  
    if (usuarioLista != null) {  
        if (!usuarioLista.usuarioAtivo) {  
            print("Conta inativa. Reativar? (S/N): ")  
            if (readln().uppercase() == "S") {  
                usuarioLista.usuarioAtivo = true  
                usuarioLogado = usuarioLista  
                println("Reativada!")  
            }  
        } else {  
            usuarioLogado = usuarioLista  
            println("Olá, ${usuarioLista.usuarioNome}!")  
        }  
    } else println("Erro: Credenciais inválidas.")  
}
```

Cadastrar Usuário Organizador

- Passo a Passo
- Falar é fácil, mostre-me o código

--- DENDÊ EVENTOS ---

1. Login / Reativar

2. Cadastro Comum

3. Cadastro Org

0. Sair

Opção: 3

Digite seu Email: *joca@gmail.com*

Digite seu Nome: *joca*

Digite sua Data de Nascimento no formato (DD/MM/YYYY): *06/10/2003*

Digite seu Sexo: *masculino*

Digite sua Senha: *123*

Informar empresa? (S/N): *S*

CNPJ: *101010101010101*

Razão Social: *Dende Eventos*

Nome Fantasia: *Dende Eventos LTDA*

Cadastrado com sucesso!

Título da Página



- Falar é fácil, mostre-me o código:

```
val novoUsuario = Usuario(  
    usuarioEmail = emailRespostaMenu,  
    usuarioNome = nomeRespostaMenu,  
    usuarioDataNascimento = dataNascimentoRespostaMenu,  
    usuarioSexo = sexoRespostaMenu,  
    usuarioSenha = senhaRespostaMenu,  
    usuarioTipo = tipoRespostaMenu)  
  
if (tipoRespostaMenu == TipoUsuario.ORGANIZADOR) {  
    print("Informar empresa? (S/N): ")  
    if(readln().uppercase() == "S") {  
        print("CNPJ: ")  
        novoUsuario.usuarioCnpj = readln()  
        print("Razão Social: ")  
        novoUsuario.usuarioRazaoSocial = readln()  
        print("Nome Fantasia: ")  
        novoUsuario.usuarioNomeFantasia = readln()  
    }  
}  
  
usuariosListaMutavel.add(novoUsuario); println("Cadastrado com sucesso!")
```

Considerações finais



- Caso houvesse mais tempo seria implementado mais validações de segurança na entrada do usuário.
- Houve uma dificuldade em implementar certos métodos, como filter, let, it, e any, entretanto após alguns estudos em alguns sites como DevMedia, e alguns fóruns como o da Alura, a própria documentação do Kotlin, e conseguimos implementar.
- Tiramos algumas dúvidas com IA para nos auxiliar durante o processo.

Referências



- <https://kotlinlang.org/docs/collection-filtering.html>
- <https://www.devmedia.com.br/kotlin-faq-conheca-melhor-essa-linguagem/40754>
- <https://cursos.alura.com.br/forum/topico-duvida-nao-entendi-muito-bem-a-questao-do-let-279988>