

# **Dendê Eventos**

## **EQUIPE PORTO**

**Autor:** Carlos Henrique de Souza Santana Santiago  
Gustavo Bezerra Nonato  
Hudnei Sued Passos Santana  
João Guilherme Gonçalves Pinheiro

# Agenda

Demonstrar a implementação e o funcionamento do sistema CLI Dendê Eventos, destacando as estruturas de dados, as validações rigorosas de regras de negócio e a experiência do usuário no terminal, desenvolvidas inteiramente em Kotlin.

## **1. Item 1: Estruturas de Dados:**

Modelagem das 4 entidades principais e persistência em memória (Listas).

## **2. Item: Gestão de Usuários:**

Cadastro (PF/PJ), login seguro, inativação e validações de idade/senha.

## **3. Item: Gerenciamento de Eventos:**

Criação, alteração e regras de bloqueio (datas passadas e lotação).

## **4. Item: Experiência do Cliente:**

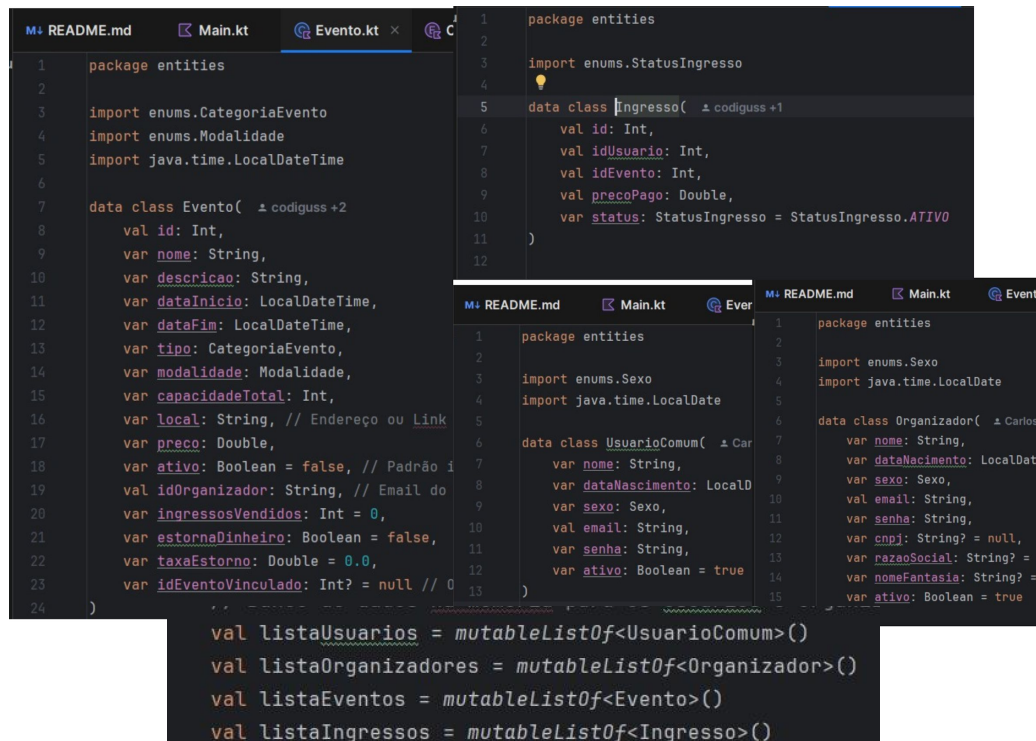
Algoritmo do feed interativo, filtros ativos e compra de ingressos.

## **5. Item: Considerações Finais:**

Desafios da lógica linear, limpeza de memória e trabalho em equipe.

# Estrutura de Dados do Sistema

- Uso de Data Classes para modelar 4 entidades principais: UsuarioComum, Organizador, Evento, Ingresso.
- Uso de Enums para Sexo, CategoriaEvento e StatusIngresso.
- Uso de Coleções: Persistência em memória utilizando `mutableListOf<>` para todas as listas principais.



```
1 package entities
2
3 import enums.CategoriaEvento
4 import enums.Modalidade
5 import java.time.LocalDateTime
6
7 data class Evento( ± codiguss+2
8     val id: Int,
9     var nome: String,
10    var descricao: String,
11    var dataInicio: LocalDateTime,
12    var dataFim: LocalDateTime,
13    var tipo: CategoriaEvento,
14    var modalidade: Modalidade,
15    var capacidadeTotal: Int,
16    var local: String, // Endereço ou Link
17    var preco: Double,
18    var ativo: Boolean = false, // Padrão
19    val idOrganizador: String, // Email do
20    var ingressosVendidos: Int = 0,
21    var estornaDinheiro: Boolean = false,
22    var taxaEstorno: Double = 0.0,
23    var idEventoVinculado: Int? = null // 0
24 )
25
26 val listaUsuarios = mutableListOf<UsuarioComum>()
27 val listaOrganizadores = mutableListOf<Organizador>()
28 val listaEventos = mutableListOf<Evento>()
29 val listaIngressos = mutableListOf<Ingresso>()
```

```
1 package entities
2
3 import enums.StatusIngresso
4
5 data class Ingresso( ± codiguss+1
6     val id: Int,
7     val idUsuario: Int,
8     val idEvento: Int,
9     val precoPago: Double,
10    var status: StatusIngresso = StatusIngresso.ATIVO
11 )
12
```

```
1 package entities
2
3 import enums.Sexo
4 import java.time.LocalDate
5
6 data class UsuarioComum( ± Carlos
7     var nome: String,
8     var dataNascimento: LocalDate,
9     var sexo: Sexo,
10    val email: String,
11    var senha: String,
12    var cnpj: String? = null,
13    var razaoSocial: String? = null,
14    var nomeFantasia: String? = null,
15    var ativo: Boolean = true
16 )
17
```

```
1 package entities
2
3 import enums.Sexo
4 import java.time.LocalDate
5
6 data class Organizador( ± Carlos
7     var nome: String,
8     var dataNascimento: LocalDate,
9     var sexo: Sexo,
10    val email: String,
11    var senha: String,
12    var cnpj: String? = null,
13    var razaoSocial: String? = null,
14    var nomeFantasia: String? = null,
15    var ativo: Boolean = true
16 )
17
```

# Gestão de Usuários



- Separação de Pessoa Física e Jurídica no momento do cadastro.
- Regras de Negócio e Validações: Bloqueio de senhas com menos de 8 caracteres, validação de e-mail único em ambas as listas, controle de idade mínima (12 e 18 anos) usando Period.between.
- Segurança: Inativação e reativação da conta bloqueando o acesso ao menu logado (if (ativo == true)).

```
---- LOGIN ----
Digite seu Email:
roberto@dendeproducoes.com
Digite sua Senha:
SenhaForte123
-----
Login realizado com sucesso!
-----

---- MENU LOGADO ----
-----
Olá Organizador Roberto Almeida!
1) Meu Perfil
2) Alterar dados do Perfil
3) Inativar Minha Conta
4) Gerenciar Eventos
-----
0) Sair (logout)
-----
Escolha:
|
```

```
---- MENU LOGADO ----
-----
Olá Usuario Mariana Costa!
1) Meu Perfil
2) Alterar dados do Perfil
3) Inativar Minha Conta
4) Ver feed de Eventos
-----
0) Sair (logout)
-----
Escolha:
|
```

# Gerenciamento de Eventos



- Lógica para atrelar o Evento ao Organizador usando índice/e-mail como chave estrangeira.
- Regras de Negócio e Validações: Impede a criação de eventos no passado (`LocalDateTime.now()`) e bloqueia a alteração de dados sensíveis (preço/local) de eventos que já possuem ingressos vendidos (`ingressosVendidos > 0`).

```
-----
--- GERENCIAMENTO DE EVENTOS ---
1) Cadastrar Novo Evento
2) Listar Meus Eventos
3) Alterar Evento
4) Ativar/Desativar Evento
0) Voltar ao Menu Principal
-----

Escolha: 2

--- MEUS EVENTOS ---
ID      NOME      DATA      PREÇO      CAPAC.      STATUS
-----
1      Hackathon Dendê S.  25/10 08:00  R$0,00      50      INATIVO
2      Festival Sons da .  14/11 20:00  R$45,50     800      INATIVO

Pre
|
--- ALTERAR EVENTO ---
Seus eventos disponíveis para alteração:
ID: [1] - Hackathon Dendê SoftHouse

Digite o ID do evento que deseja alterar (ou 0 para cancelar): 1
Deixe em branco para manter o valor atual.
Novo Nome (Hackathon Dendê SoftHouse): Hackathon SI UNEX
Novo Local (Laboratorios de Informatica - UNEX): Laboratorios de Informatica - UNEX
Novo Preço (0.0): 0
Evento atualizado com sucesso!
Pressione ENTER para voltar...
|
```

# Feed de Eventos e Ingressos



- Motor de filtragem para exibir apenas o que interessa ao usuário.
- Regras de Negócio e Validações: O feed filtra eventos inativos, eventos que já encerraram a data e eventos lotados (`ingressosVendidos >= capacidadeTotal`).
- Implementação: Algoritmo de ordenação (ex: Bubble Sort) para organizar os eventos por data mais próxima.

## FEED DE EVENTOS

Eventos disponíveis: 1

[1] Hackathon Dendê SoftHouse

📅 Início: 25/10/2026 08:00

🕒 Fim: 26/10/2026 08:00

📍 Local: Laboratorios de Informatica - UNEX

🏷️ Tipo: OUTRO

👤 Modalidade: PRESENCIAL

🎟️ GRATUITO

👥 Vagas: 50/50

👤 Organizador: Roberto Almeida

Pressione ENTER para voltar...

Escolha:

4

## ===== FEED DE EVENTOS =====

1) Feed de Eventos (Ver Todos)

2) Comprar Ingresso

3) Cancelar Ingresso

4) Meus Ingressos

0) Voltar

Escolha uma opção: 1

# Considerações finais



- Desafios de manter a lógica de forma estritamente linear no main.
- A importância da limpeza de variáveis no modo Console (resetar atributos para evitar lixo de memória ao refazer formulários).
- A evolução no trabalho em equipe em um único arquivo de execução.

# Referências



- [Documentação Oficial do Kotlin.](#)
- Materiais das aulas de Lógica de Programação da disciplina.