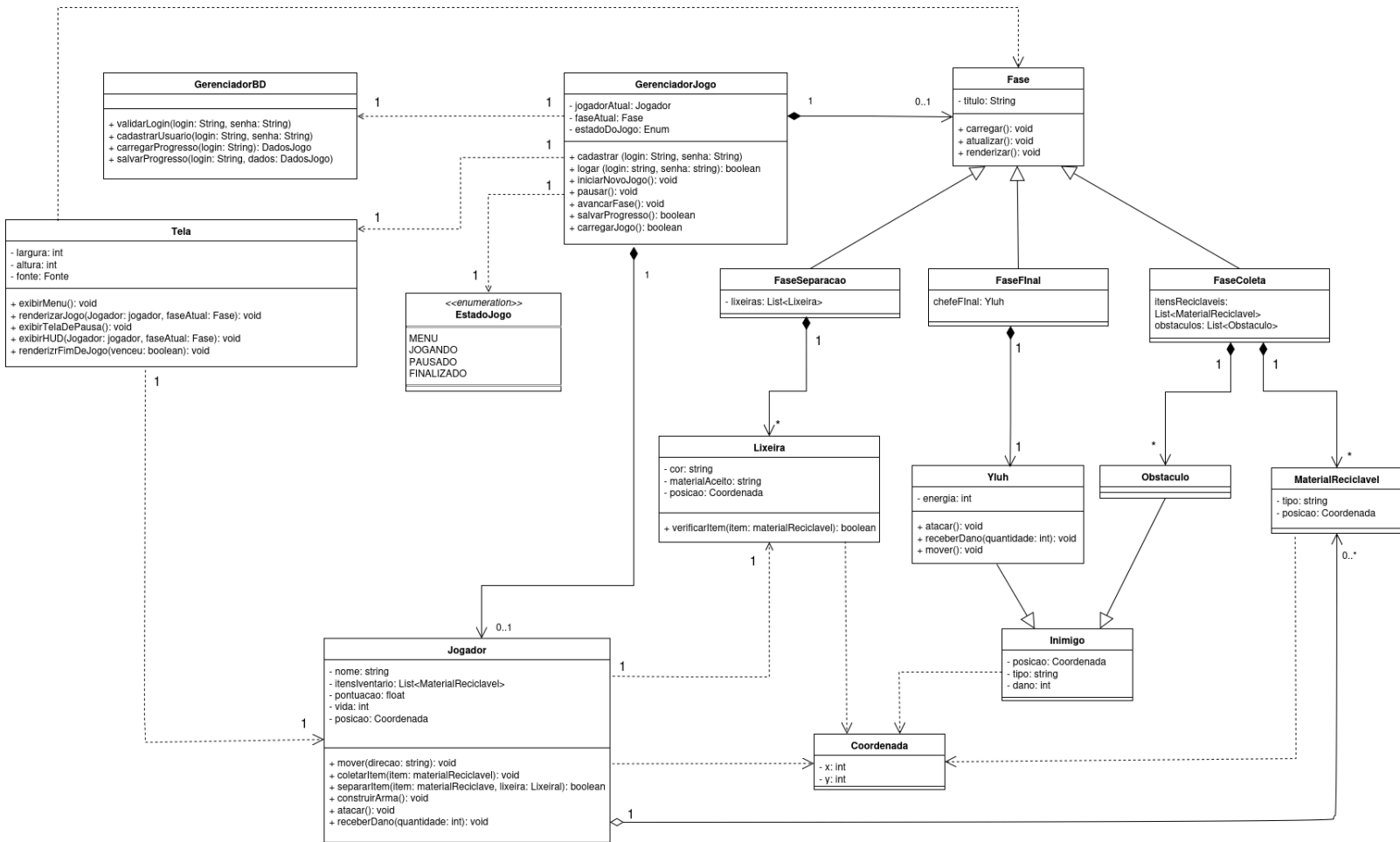


# DIAGRAMA DE CLASSES - ECO RUNNER

## 1. Diagrama de classes



[Visualizar](#)

## 2. Classes do Núcleo do Jogo

- Estas são as classes responsáveis por gerir o estado geral, a renderização e a persistência de dados.

### *GerenciadorJogo*

- **Responsabilidade:** Centraliza toda a lógica de alto nível, controlando o fluxo principal (o game loop).
- **Funções Chave:**
  - Controla o estado atual do jogo (se está no menu, a jogar, pausado ou no fim).
  - Instancia e gere a fase atual e o jogador.
  - Recebe os inputs do utilizador (teclado, mouse) e delega as ações.
  - Coordena as outras classes do núcleo, dizendo à *Tela* o que desenhar e ao *GerenciadorBD* quando salvar.

### *GerenciadorBD*

- **Responsabilidade:** Classe responsável pela persistência e comunicação com o banco de dados
- **Funções Chave:**
  - *cadastrarUsuario()*: Cria um novo registo de utilizador.
  - *logar()*: Verifica as credenciais de um utilizador.
  - *salvarProgresso()*: Guarda o estado atual do jogador (fase, inventário).
  - *carregarProgresso()*: Carrega os dados de um jogador salvo.

### *Tela*

- **Responsabilidade:** Tem como função desenhar coisas no ecrã.
- **Funções Chave:**

- *renderizarJogo(jogador, fase)*: Recebe os objetos do jogo e desenha-os nas suas posições corretas.
- *renderizarHUD()*: Desenha a interface do utilizador (*vida, pontuação*, etc.).
- Classe que cumpre as ordens de desenho do *GerenciadorJogo*.

### *EstadoJogo (Enum)*

- **Responsabilidade:** É o "livro de regras" para os estados do jogo.
- **Funções Chave:**
  - Define um conjunto fixo de estados possíveis (*MENU, JOGANDO, PAUSADO, FINALIZADO*).
  - Previne bugs de digitação e garante que o *GerenciadorJogo* opere sempre num estado válido e conhecido.

### *Coordenada*

- **Responsabilidade:** É o "pilare" fundamental para a posição.
- **Funções Chave:**
  - Armazena um par de valores *x* e *y*.
  - É usada por todas as classes que precisam de ter uma posição no mundo do jogo, centralizando o conceito de localização num único ponto.

## 3. Classes de Lógica e Mundo do Jogo

- Estas classes representam as entidades com as quais o jogador interage e o mundo em que ele habita.

### *Jogador*

- **Responsabilidade:** Representa o avatar do utilizador no jogo.

- **Funções Chave:**

- Mantém o seu estado: vida, posicao, e o inventário de *ItensReciclaveis*.
- Define as ações do jogador: *mover()*, *coletarItem()*, *separarItem()*, *atacar()*.

### *Fase (Superclasse)*

- **Responsabilidade:** É o "molde" para todas as fases do jogo.

- **Funções Chave:**

- Define a estrutura que todas as fases devem seguir, obrigando-as a ter os métodos *carregar()*, *atualizar()* e *renderizar()*.
- Permite que o *GerenciadorJogo* trate todas as fases da mesma forma, sem precisar de saber os detalhes específicos de cada uma.

### *Subclasses de Fase (FaseColeta, FaseSeparacao, FaseFinal)*

- **Responsabilidade:** Implementam a lógica e os desafios únicos de cada nível.

- **Funções Chave:**

- *FaseColeta*: Gere a criação e a lógica dos *ItensReciclaveis* e dos *Obstaculos*.
- *FaseSeparacao*: Gere as Lixeiras e a lógica de verificação da separação correta dos itens.
- *FaseFinal*: Gere a batalha contra o boss, incluindo a criação da *Yluh*

## **4. Classes de Entidades**

- Estas são as classes que representam os objetos e inimigos específicos dentro das fases.

### *Inimigo (Superclasse)*

- **Responsabilidade:** É o "molde" para todas as entidades hostis.
- **Funções Chave:**
  - Define os atributos comuns a todos os inimigos, como *vida* e *posicao*.
  - Permite que o sistema trate diferentes tipos de inimigos de forma genérica.

### *Subclasses de Inimigo (Yluh, Obstaculo)*

- **Responsabilidade:** Implementam o comportamento específico de cada inimigo.
- **Funções Chave:**
  - Yluh: O boss final. Tem a sua própria lógica de movimento e ataque.
  - Obstaculo: Inimigos mais simples da *FaseColeta*.

### *ItemReciclavel*

- **Responsabilidade:** Representa os itens colecionáveis do jogo.
- **Funções Chave:**
  - Armazena o seu tipo (plástico, metal, etc.) e a sua *posicao* no mapa.

### *Lixeira*

- **Responsabilidade:** Representa os pontos de interação na *FaseSeparacao*.
- **Funções Chave:**
  - Armazena o tipo de lixo que aceita e a sua *posicao*.