

Resumo detalhado de cada classe:



Camada de Interface (Views)

Responsável apenas por mostrar dados no console e ler entradas do teclado.

1. TelaBase

- **Função:** Classe mãe que contém métodos utilitários comuns a todas as telas (limpar console, cabeçalhos, ler opções numéricas com tratamento de erro).
- **Conexões:**
 - Nenhuma (É a base).

2. TelaEstudos

- **Função:** Exibe o menu do cronômetro, mensagens de "Estudando..." e lista o histórico de sessões.
- **Conexões:**
 - Herda de: TelaBase.

3. TelaGamificacao

- **Função:** Exibe o status do jogador (Nível, XP, Barra de Progresso).
- **Conexões:**
 - Herda de: TelaBase.

4. TelaLoja

- **Função:** Lista os itens disponíveis para compra e seus preços.
- **Conexões:**
 - Herda de: TelaBase.

5. TelaInventory

- **Função:** Lista os itens que o usuário já comprou e permite selecionar um para equipar/usar.
 - **Conexões:**
 - Herda de: TelaBase.
-

🎮 Camada de Controle (Controllers)

O "Cérebro" do sistema. Gerencia a lógica, conecta a Tela aos Dados e aplica regras de negócio.

6. ControladorPrincipal

- **Função:** Gerencia o loop principal do programa e o menu inicial. Delega a execução para os outros controladores conforme a escolha do usuário.
- **Conexões:**
 - Agrega (possui ponteiros para): ControladorEstudo, ControladorLoja, ControladorInventory, ControladorGamificacao.
 - Associa-se com: Usuario (para exibir nome/boas-vindas).

7. ControladorEstudos

- **Função:** Gerencia o timer. Quando o estudo acaba, cria uma SessaoEstudo, salva no repositório e chama o Usuario para adicionar XP.
- **Conexões:**
 - Compõe (tem uma): TelaEstudos.
 - Associa-se com: RepositorioEstudos (para salvar/ler histórico).
 - Associa-se com: Usuario (para dar XP).

8. ControladorGamificacao

- **Função:** Calcula lógica de nível e exibe dados do usuário.
- **Conexões:**
 - Compõe (tem uma): TelaGamificacao.
 - Associa-se com: RepositorioGamificacao (para persistir dados).
 - Associa-se com: Usuario (para ler dados).

9. ControladorLoja

- **Função:** Permite a compra. Verifica se o usuário tem moedas suficientes, desconta o valor e adiciona o item ao inventário.
- **Conexões:**
 - Compõe (tem uma): TelaLoja.
 - Associa-se com: Loja (para ver o catálogo).
 - Associa-se com: RepositorioInventory (para salvar o item comprado).
 - Associa-se com: Usuario (para gastar moedas).

10. ControladorInventory

- **Função:** Permite visualizar e equipar itens.
- **Conexões:**
 - Compõe (tem uma): TelaInventory.
 - Associa-se com: RepositorioInventory (para buscar os itens do usuário).
 - Associa-se com: Usuario.

Camada de Entidades (Model)

Objetos que representam os dados do sistema. São passivos.

11. Usuario

- **Função:** A entidade central. Guarda o estado atual: Nome, XP, Nível e Moedas. Contém regras simples (ex: subir de nível se XP estourar).
- **Conexões:**
 - Nenhuma saindo (Ele não conhece ninguém).
 - Recebe conexões de todos os Controladores e Repositórios.

12. SessaoEstudo

- **Função:** Representa um bloco de estudo concluído (Data, Duração, Matéria).
- **Conexões:**
 - Nenhuma.

13. Loja

- **Função:** Funciona como um Catálogo. Contém a lista de todos os itens possíveis de serem comprados.
- **Conexões:**
 - Agrega: Item (Lista de produtos).

14. Item (Classe Abstrata/Base)

- **Função:** Define o que é um objeto comprável (Nome, Preço, Tipo).
- **Conexões:**
 - Nenhuma.

15. ItemAudio

- **Função:** Especialização de Item (ex: Música de fundo).
- **Conexões:**
 - Herda de: Item.

16. ItemCosmetico

- **Função:** Especialização de Item (ex: Avatar, Tema).
 - **Conexões:**
 - Herda de: Item.
-

Camada de Infraestrutura (Repositories)

Responsável por Ler e Escrever arquivos .txt.

17. RepositorioBase

- **Função:** Lida com a abertura de arquivos, verificação de existência e leitura/escrita bruta de linhas de texto.
- **Conexões:**
 - Nenhuma.

18. RepositorioEstudos

- **Função:** Converte objetos SessaoEstudo em texto para salvar, e texto em objetos para carregar.
- **Conexões:**
 - Herda de: RepositorioBase.
 - Depende de: SessaoEstudo.
 - Depende de: Usuario (para saber o nome do arquivo).

19. RepositorioGamificacao

- **Função:** Salva e carrega o estado do Usuario (XP, Moedas, Nível).
- **Conexões:**
 - Herda de: RepositorioBase.
 - Depende de: Usuario.

20. RepositorioInventario

- **Função:** Salva quais itens o usuário comprou. Carrega essa lista para a memória (vector<Item>) quando solicitado.
 - **Conexões:**
 - Herda de: RepositorioBase.
 - Agrega: Item (Possui a lista de itens do jogador).
 - Depende de: Usuario (para saber o nome do arquivo).
-