

Documentação dos Padrões de Projeto — Camada de Persistência

Objetivo

Explicar que a camada de persistência foi implementada com boas práticas, usando padrões de projeto para garantir desacoplamento, organização e testabilidade.

Patterns utilizados (cada um em 1 parágrafo curto)

SingletonArquivo: [src/config/database.ts](#)

Explicação: Centraliza a conexão com o banco em uma única instância ([AppDataSource](#)), evitando múltiplas pools e facilitando a configuração.

Trecho de código:

```
export const AppDataSource = new DataSource({});
```

Benefício: Eficiência, consistência e fácil manutenção da conexão.

Repository Pattern

Arquivo: [src/db/repositories/UserRepository.ts](#) e [CavaloRepository.ts](#)

Explicação: Encapsula operações CRUD e queries específicas, isolando a lógica de persistência.

Trecho de código:

Usuário:

```
async findById(id: string): Promise<User | null> {  
  
    // Try to find in Comprador first  
  
    let user: User | null = await this.compradorRepository.findOne({  
where: { id } });
```

Cavalo:

```
async findById(id: string): Promise<Cavalo | null> {  
  
    return await this.cavaloRepository.findOne({  
  
        where: { id },  
  
        relations: ['dono']  
  
    });  
  
}
```

Benefício: Desacopla serviços da lógica de banco, facilitando testes e troca de SGBD.

DAO (Entities)

Arquivo: [src/db/entities/User.ts](#), [Cavalo.ts](#), [Anuncio.ts](#), etc.

Explicação: Cada classe representa e mapeia uma tabela do banco de dados, usando decoradores do TypeORM.

Trecho de código:

Usuário:

```
@Entity('users')

@TableInheritance({ column: { type: 'varchar', name: 'type' } })

export abstract class User {

    @PrimaryGeneratedColumn('uuid')

    id!: string;

    @Column({ type: 'varchar', length: 100 })

    nome!: string;}
```

Cavalo:

```
@Entity('cavalos')

export class Cavalo {

    @PrimaryGeneratedColumn('uuid')

    id!: string;

    @Column({ type: 'varchar', length: 100 })

    nome!: string;

}
```

Benefício: Facilita o mapeamento objeto-relacional, mantendo organização e clareza.