

Relatório Técnico - Jackut

Autor: Davi Silva de Melo Lins

1. Introdução

O **Jackut** é um sistema de rede social desenvolvido em Java, seguindo princípios de **Orientação a Objetos** e **Design Patterns**. Este relatório detalha:

- Diagrama de classes e justificativas de design.
- Padrões de projeto aplicados.
- Melhorias da Milestone 1 para a Milestone 2.

2. Diagrama de Classes

https://drive.google.com/file/d/1oUzzinr-yD2N7B64VP7dV5fc_WR4bBox/view?usp=sharing

Justificativas do Design

| Decisão | Motivo |
|---------------------------------------|---|
| Facade Pattern | Isolar a complexidade do sistema em uma interface única (Facade). |
| SRP (Single Responsibility Principle) | Cada classe tem uma única responsabilidade (ex: SessionService gerencia usuários, User armazena dados). |
| Composição sobre Herança | User contém Perfil (composição) em vez de herdar dele. |
| Persistência em Arquivos | Armazenamento em .txt para simplicidade, com classes dedicadas (EscritaDeArquivos , LeituraDeArquivos). |

3. Design Patterns Aplicados

a) Facade

- **Onde:** Classe Facade .
- **Por quê:** Simplifica a interação com o sistema, escondendo a complexidade de SessionService , User , e demais componentes.

b) Singleton (implícito)

- **Onde:** `SessionService` (gerenciada pela `Facade`).
- **Por quê:** Garante que apenas uma instância do serviço seja usada durante a execução.

c) Iterator

- **Onde:** Uso de `ArrayList` e `Queue` para percorrer recados, mensagens e relacionamentos.
- **Por quê:** Permite acesso sequencial aos elementos sem expor a estrutura interna.

4. Melhorias (Milestone 1 → Milestone 2)

| Problema na Milestone 1 | Solução na Milestone 2 |
|-------------------------|--|
| Facade não isolada | Refatorada para ser o único ponto de entrada , delegando lógica para <code>SessionService</code> . |
| Exceções genéricas | Implementadas exceções personalizadas (ex: <code>UsuarioNaoRegistradoException</code> , <code>ComunidadeJaExisteException</code>). |
| Baixa coesão | Classes divididas por responsabilidade (ex: <code>User</code> cuida de dados, <code>SessionService</code> de regras). |

5. Conclusão

O Jackut evoluiu significativamente na Milestone 2, com:

- ✓ **Design mais limpo** (Facade, SRP, coesão).
- ✓ **Tratamento de erros robusto** (exceções específicas).
- ✓ **Código legível** e fácil de manter.