



Projeto Arquitetural - Sistema de Gerenciamento TCC, Estágio e Monitoria - Sig@ UPE

Alunos:

Danilo Takashi,
Eduardo Dubeux,
Marcos Prudêncio,
Roberto Antônio

1. Objetivo do Projeto da Arquitetura

O objetivo deste projeto arquitetural é organizar a estrutura do sistema Sig@ UPE de forma clara, eficiente e bem definida, para que o desenvolvimento do sistema seja mais fácil de entender, manter e evoluir. A ideia é dividir o sistema em partes que se comunicam entre si, mas que têm responsabilidades bem separadas — como a interface com o usuário, a lógica dos processos (regras de negócio) e o armazenamento dos dados.

Ao projetar essa arquitetura, buscamos garantir que o sistema seja modular, seguro, reutilizável e preparado para crescer no futuro. Também levamos em conta a facilidade de manutenção, a clareza na organização do código e a integração entre os diferentes perfis de usuários (alunos, orientadores, coordenadores e supervisores). Com essa base bem estruturada, o time de desenvolvimento poderá dar continuidade à implementação dos casos de uso com mais confiança e consistência.

2. Mapeamento da Análise para o Projeto

2.2 Classes de Análise → Classes de Projeto

Foram identificadas as seguintes categorias:

- **Entidades:** Usuário, Aluno, Monitor, Orientador, Coordenador, Disciplina, SolicitacaoMonitoria, AlunosInteressadosMonitoria, Monitoria, TCC, Estágio, Relatório, TermoDeCompromisso, Avaliador
- **Fronteira:** TelaLogin, TelaRegistro, TelaPerfilUsuario, TelaPrincipal, TelaTCC, TelaOrientadorTCC, TelaSubmissaoPropostaTCC, TelaRevisaoPropostaTCC, TelaValidacaoPropostaTCC, TelaAtualizacaoStatusTCC, TelaRevisaoParcialTCC, TelaSubmissaoFinalTCC, TelaAgendamentoDefesaTCC, TelaNotasFeedbackTCC, ArquivamentoTCC, TelaMonitoria, TelaSolicitarMonitoria, TelaCadastroOferta, TelaSolicitarSerMonitor, TelaGerenciarSolicitacoes, TelaRegistrarRelatorio, TelaGerenciarMonitoria, TelaAprovarRelatório, TelaAvaliarMonitoria, TelaEmitirRelatorioGeral, TelaEstagio, TelaInscricaoEstagio, TelaAprovacao, TelaValidacaoTCE, TelaSubmissaoRelatorio, TelaSubmissaoTCE, TelaAvaliacaoRelatorio
- **Controladores:** AppFacade, ControladorTCC, CriarTCC, ControladorEstagio, CriarEstagio, ControladorMonitoria, ControladorUsuario, ControladorAluno, ControladorOrientador, ControladorCoordenador

2.3 Refinamento

A arquitetura proposta foi refinada com foco na separação de responsabilidades, adotando o padrão MVC (Model-View-Controller) para garantir modularidade e facilitar a manutenção do sistema. Durante esse processo, as funcionalidades inicialmente descritas foram distribuídas entre três camadas principais:

- **Camada de Entidade (Model):** concentra-se na representação dos dados do domínio da aplicação, incluindo classes como Aluno, Usuário, Orientador, Monitor, TCC, Estágio, Disciplina, entre outras. Essa camada armazena os dados que trafegam entre as outras partes do sistema e representa os objetos reais manipulados no cotidiano acadêmico.
- **Camada de Fronteira (View):** responsável pela interação com o usuário, através das interfaces das funcionalidades. Cada processo relevante do sistema possui uma tela associada, como TelaLogin, TelaSubmissaoPropostaTCC, TelaSolicitarMonitoria, TelaSubmissaoTCE, entre outras. Essas telas foram mapeadas para proporcionar uma navegação fluida e intuitiva, de acordo com o papel de cada usuário (aluno, professor, coordenador).
- **Camada de Controle (Controller):** coordena o fluxo de dados entre a fronteira e as entidades, controlando as regras de negócio e orquestrando os processos do sistema. Classes como ControladorTCC, ControladorEstagio, ControladorMonitoria e AppFacade assumem esse papel, garantindo que cada operação siga os critérios definidos e que os dados sejam tratados corretamente antes de serem persistidos ou exibidos.

3. Estrutura de Camadas da Aplicação

A arquitetura do sistema foi concebida com base no modelo em camadas, promovendo organização, separação de responsabilidades e facilidade de manutenção. As camadas principais são:

- **Apresentação (View):** responsável pela interface com o usuário. Aqui estão localizadas as telas do sistema, como TelaLogin, TelaTCC, TelaEstagio, TelaMonitoria e suas subdivisões.
- **Negócio (Controller):** camada intermediária que contém as regras de negócio. É composta pelos controladores (ControladorTCC, ControladorEstagio, ControladorMonitoria, etc.) e pela AppFacade, que atua como fachada para orquestrar as funcionalidades.
- **Persistência/Dados (Model):** engloba as entidades que representam os dados manipulados no sistema, como Aluno, Usuário, TCC, Estágio, Disciplina, Monitor, Orientador, etc.

4. Componentes do Sistema

O sistema é formado por três módulos principais:

- **Módulo de TCC:** gerencia todo o ciclo do Trabalho de Conclusão de Curso, desde a escolha do orientador até a defesa e arquivamento do trabalho.

- **Módulo de Estágio:** permite a inscrição, avaliação, envio de documentos (como TCE e relatórios) e controle de aprovação do estágio.
- **Módulo de Monitoria:** administra inscrições de alunos, ofertas de disciplinas, avaliações e geração de relatórios da monitoria.

Cada módulo se comunica com o núcleo central do sistema por meio da AppFacade, garantindo uma integração coesa entre os processos.

5. Padrões de Projeto Utilizados

- **MVC (Model-View-Controller):** aplicado para separar as responsabilidades entre interface, controle e dados.
- **Fachada (Facade):** representado pela AppFacade, simplifica o acesso às operações de alto nível do sistema, centralizando os serviços principais.
- **Singleton:** utilizado nas classes de coleção como CollectionAluno, CollectionMonitor, CollectionOrientador, garantindo uma única instância para gerenciar os dados persistentes.

6. Diagrama de Classes

O diagrama de classes do sistema representa detalhadamente a estrutura das classes e suas relações. As classes foram organizadas segundo os papéis:

- **Entidade:** Aluno, Orientador, TCC, Estágio, Monitor, Disciplina, SolicitacaoMonitoria, etc.
- **Fronteira:** classes que representam telas como TelaSubmissaoTCC, TelaAgendamentoDefesaTCC, TelaSubmissaoRelatorio, TelaSolicitarMonitoria, etc.
- **Controle:** ControladorTCC, ControladorEstagio, ControladorMonitoria, ControladorUsuario, entre outros.

Esse diagrama foi essencial para definir a estrutura e responsabilidades de cada parte do sistema.

7. Tecnologias Utilizadas

- **Linguagem de Programação:** Java
- **IDE:** Eclipse ou IntelliJ IDEA
- **Modelagem UML:** Astah
- **Padrões de Arquitetura:** MVC, Singleton, Facade
- **Ferramentas de Apoio:** Lucidchart, Canva ou Draw.io (para diagramas complementares)

8. Reuso de Componentes

O sistema foi projetado com reusabilidade em mente. Exemplos claros:

- As telas de submissão e avaliação de relatórios no módulo de TCC e estágio seguem a mesma estrutura lógica.
- O método de autenticação é compartilhado por todos os perfis de usuário no login.
- Classes de coleção, como `CollectionAluno`, podem ser reutilizadas em diversas partes do sistema para busca e manipulação de dados.

9. Segurança e Controle de Acesso

Cada perfil (Aluno, Orientador, Coordenador) possui permissões específicas de acesso às funcionalidades:

- **Alunos:** podem se inscrever em monitoria, submeter TCC, enviar documentos de estágio e consultar feedbacks.
- **Orientadores:** podem validar propostas, avaliar relatórios e acompanhar o progresso dos orientandos.
- **Coordenadores:** têm poderes administrativos, como aprovação de inscrições e geração de relatórios globais.

A autenticação é realizada via login com validação de dados e controle de sessão para garantir a integridade do acesso.

10. Conclusão

O projeto arquitetural do sistema **Sig@UPE** foi essencial para dar forma e organização a uma ideia que busca facilitar a vida acadêmica de alunos, orientadores e coordenadores. Com a definição clara das camadas, das responsabilidades de cada classe e da estrutura geral do sistema, conseguimos construir uma base sólida que torna o desenvolvimento mais seguro, coeso e eficiente.

A separação entre entidades, controles e interfaces, aliada ao uso de padrões como o MVC e a fachada, garantiu que o sistema fosse modular, fácil de entender e preparado para crescer com o tempo. Pensamos também na experiência de quem vai usar o sistema no dia a dia — buscando tornar os processos mais simples, rápidos e acessíveis.

Mais do que uma etapa técnica, esse projeto de arquitetura representa o compromisso com uma solução real, que conecta a tecnologia às necessidades de uma universidade pública. É a partir dessa base que seguimos para as próximas fases do desenvolvimento, confiantes de que estamos no caminho certo.