

RELATÓRIO FINAL DO PROJETO DE POO

1. Introdução

Este relatório apresenta o desenvolvimento de um sistema de gestão para a escola de idiomas "WorldTalk", criado como parte da disciplina de Programação Orientada a Objetos. O projeto teve como objetivo aplicar, de forma prática, conceitos fundamentais de POO, tais como abstração, encapsulamento, herança, polimorfismo, modularização e boas práticas de engenharia de software.

O sistema foi desenvolvido por uma equipe composta por quatro alunos, cada um responsável por módulos diferentes, garantindo divisão equilibrada das responsabilidades e integração coerente entre os componentes implementados. O trabalho buscou atender aos requisitos funcionais propostos pela atividade, garantindo a construção de uma solução funcional.

2. Atividades Desenvolvidas pelos Integrantes do Grupo

Samuel Rodrigues atuou na implementação do gerenciamento de turmas e horários, incluindo a definição de limites de alunos por turma e a priorização de vagas para alunos VIP. Além disso, liderou o projeto, criou o esqueleto inicial da arquitetura do sistema, realizou correções em diversas classes e gerenciou as versões do código por meio de ferramentas de controle de versão. Sua atuação garantiu a coesão estrutural e a integração entre os módulos desenvolvidos pelo grupo.

Danilo Campos ficou responsável pelo módulo de pagamentos e histórico de alunos. Sua contribuição incluiu o registro de pagamentos, a implementação de planos de fidelidade para alunos que completam múltiplos cursos e a criação de um sistema completo para registro de níveis, notas e progresso individual dos estudantes.

Absalão Neto desenvolveu o módulo de Biblioteca Virtual e materiais de apoio. O trabalho incluiu a criação de um sistema de empréstimos virtuais, com registro de histórico, devoluções e acesso exclusivo para alunos conforme sua categoria. Esse módulo agregou recursos pedagógicos complementares ao sistema.

João Paulo foi responsável pelo módulo de cursos e níveis, implementando o cadastro de cursos e níveis de aprendizado, além do registro de progresso individual. Também construiu o sistema de agendamento de aulas particulares, incluindo regras de cancelamento com taxa de penalidade e designação de professores específicos para alunos VIP.

3. Funcionalidades Implementadas

Entre as principais funcionalidades desenvolvidas estão:

- Cadastro e gerenciamento de cursos e níveis de aprendizagem;
- Registro de progresso individual dos alunos ao longo dos níveis;
- Gerenciamento de turmas, horários e limites de alunos;
- Priorização automática de vagas para estudantes VIP;
- Sistema de agendamento de aulas particulares com regras de cancelamento;
- Biblioteca virtual com controle de empréstimos e materiais de apoio;
- Registro de pagamentos e estruturação de planos de fidelidade;
- Histórico detalhado de cursos concluídos, níveis alcançados e desempenho;
- Menu de interação simples via console;
- Aplicação dos princípios essenciais da Programação Orientada a Objetos.

As funcionalidades implementadas abrangem tanto aspectos acadêmicos quanto administrativos, permitindo uma simulação realista de um ambiente educacional.

4. Funcionalidades Não Implementadas e Justificativas

Embora o sistema esteja em estágio funcional, algumas funcionalidades previstas não foram implementadas devido ao tempo limitado disponível para o desenvolvimento, além da necessidade de priorização dos requisitos essenciais.

Entre as funcionalidades não concluídas estão:

- Geração de relatórios automáticos em PDF;
- Interface gráfica em JavaFX ou Swing;
- Recursos avançados da biblioteca virtual, como recomendações personalizadas (Armazenamento em nuvem, como integração com Google Drive).

Esses itens exigiriam maior aprofundamento técnico, uso de bibliotecas externas ou tempo adicional para testes e validação. Dessa forma, o grupo optou por focar nas funcionalidades principais, garantindo estabilidade e coerência na solução entregue.

5. Experiência de Aprendizado

O desenvolvimento do sistema proporcionou aprendizado significativo aos integrantes do grupo. O projeto permitiu a aplicação prática de conceitos complexos de orientação a objetos, consolidando temas como modularização, tratamento de erros, encapsulamento e organização arquitetural.

Além do aprendizado técnico, o trabalho reforçou a importância da cooperação, do versionamento de código e da comunicação entre integrantes. A experiência destacou como decisões arquiteturais impactam diretamente a qualidade do projeto final e evidenciou a importância de testes contínuos e revisões colaborativas.

O grupo melhorou seu conhecimento no uso de Java, manipulação de arquivos e boas práticas de desenvolvimento, fortalecendo sua capacidade de resolver problemas reais com design orientado a objetos.

6. Bibliotecas Utilizadas

O projeto foi desenvolvido majoritariamente com bibliotecas nativas da linguagem Java, garantindo simplicidade e compatibilidade. Entre os principais recursos utilizados estão:

- `java.util` - coleções e estruturas auxiliares;
- `java.time` - manipulação de datas e horários;
- `java.io` - persistência de dados via arquivos;

A escolha por evitar dependências externas foi motivada pelo foco na compatibilidade entre ambientes e pela necessidade de manter o projeto leve e simples para fins acadêmicos.

7. Referências Consultadas

Ao longo do desenvolvimento foram consultadas diversas fontes de apoio, incluindo:

- Documentação oficial da linguagem Java;
- Vídeos do Youtube;
- Repositórios open source para consulta de padrões de projeto;
- Materiais fornecidos pelo professor.

8. Considerações Finais

O projeto desenvolvido demonstra a capacidade do grupo em aplicar conceitos essenciais de Programação Orientada a Objetos na construção de um sistema realista e funcional. A divisão de responsabilidades, a integração entre módulos e a aplicação de boas práticas resultaram em uma solução coerente e bem estruturada.

Apesar das limitações de tempo, o sistema atende aos principais requisitos funcionais e apresenta potencial para expansão futura. A experiência adquirida durante o processo contribuiu significativamente para a formação técnica dos integrantes, consolidando conhecimentos fundamentais para o desenvolvimento de software orientado a objetos.