

UNIVERSIDADE POSITIVO

SIMULADOR DE CORRIDA
PROJETO JAVA

CURITIBA/2025

UNIVERSIDADE POSITIVO

SIMULADOR DE CORRIDA
PROJETO JAVA

ALUNOS:

AMANDA EDLING

LIGIA MARIA

LUIZ STORRER

WELLINGTON COQUEIRO

CURITIBA/2025

ÍNDICE

INTRODUÇÃO	01
RESUMO	02
OBJETIVOS	03
MATERIAIS E MÉTODOS	04
RESULTADOS	05
CONCLUSÃO	06
REFERÊNCIAS	07

INTRODUÇÃO

Este trabalho tem como objetivo o desenvolvimento de um simulador de corrida em linguagem Java, para o projeto de faculdade da matéria de desenvolvimento de software, com aplicação de conceitos fundamentais de programação orientada a objetos, arquitetura em camadas, persistência de dados e uso de estruturas de controle e fluxo.

O projeto de sistema simula um campeonato entre veículos, pilotos e equipes, contendo variações climáticas, escolha de pneus, desempenho parcial e total dos participantes. O projeto será desenvolvido e testado pelo terminal, com a utilização de menus interativos. A integridade dos dados é garantida por arquivos JSON e a serialização por ObjectOutputStream.

A aplicação também registra logs em arquivos .log. O projeto foi desenvolvido de acordo com os requisitos técnicos exigidos pelo professor André Luiz, promovendo boas práticas de codificação e arquitetura de software e implantação de MVC.

O presente trabalho apresenta a construção de um simulador de corrida utilizando Java, com foco em aplicar conceitos teóricos na prática. O projeto simula uma competição entre veículos de diferentes equipes e pilotos, abordando temas como encapsulamento, herança, polimorfismo, abstração, interfaces, persistência e controle de fluxo.

Este trabalho também pode ser acessado através do repositório do GitHub, segue link:

https://github.com/Ligiamariadasilva/JAVA_A1

OBJETIVOS

Este trabalho tem como objetivo principal aplicar na prática os conceitos da disciplina de desenvolvimento de software, que usa a linguagem JAVA, dentre eles estão:

- a) Herança.
- b) Abstração.
- c) Polimorfismo.
- d) Encapsulamento.
- e) Criação e aplicação no terminal.
- f) Organização utilizando o padrão MVC (Model View Controller).
- g) Armazenamento de dados por meio de arquivos JSON.
- h) Serialização e desserialização de objetos com ObjectOutputStream.
- i) Geração e leitura de arquivos de log com FileWrite.

MATERIAIS E MÉTODOS

1. FERRAMENTAS UTILIZADAS:

- a) Linguagem: Java 17.
- b) IDEs: IntelliJ IDEA / Visual Studio Code.
- c) Biblioteca: Gson (para manipulação de JSON).
- d) Terminal de linha de comando para interface com o usuário.
- e) GitHub.

2. METODOLOGIA:

O projeto foi desenvolvido com base nos princípios de modularização e responsabilidade única. A estrutura do código foi dividida em camadas, de acordo com o modelo MVC:

- a) Models: contém as principais entidades, Veículo, Piloto, Equipe, Pneu e suas subclasses.
- b) Controllers: responsáveis pela lógica a criação, edição e exclusão de objetos.
- c) Repositories: responsáveis pela persistência dos dados em arquivos .JSON.
- d) Views: camada de apresentação, com menus interativos no terminal.
- e) Services: serviços auxiliares como log do sistema.
- f) Exceptions: tratamento de erros customizados.
- g) Interfaces: definição de comportamentos genéricos, como o contrato da Oficina.

Dados, essa parte foi implementada por meio de leitura e gravação de arquivos .json utilizando a biblioteca GSON, que tem foco em dados e separação clara de responsabilidades.

3. FUNCIONALIDADES DO SISTEMA

O projeto de simulador de corridas tem as seguintes funcionalidades:

- a) Cadastro, listagem, edição e remoção de pilotos, equipes e veículos.
- b) Simulação de corridas com escolha de mapa e variações climáticas.
- c) Cálculo de desempenho com base na combinação de clima e tipo de pneu.
- d) Registro da atividade do sistema em um arquivo de log.
- e) Armazenamento persistente de dados em formato JSON.
- f) Organização em listas de pontuação ao final de cada corrida.

RESULTADOS

O desenvolvimento do projeto que compreende criar um simulador de corrida auxiliou na aprendizagem, possibilitou o exercício prático de todos os temas abordador na aula. A estrutura modular permitiu fácil adição de novas funcionalidades, como a troca de pneus durante a corrida ou a simulação da mudança climática entre voltas.

O uso de uma interface textual via terminal contribuiu para a agilidade da implementação e o foco nos conceitos fundamentais, sem a complexidade adicional de interfaces gráficas. A manipulação de dados por arquivos JSON demonstrou-se eficaz para cenários educacionais e de testes simples, além de facilitar a leitura e depuração dos dados.

A aplicação também registrou corretamente logs de ações do sistema, incluindo erros e eventos importantes. Os dados foram organizados de forma clara e os menus foram validados por meio de testes com múltiplos cenários e combinações de entrada.

CONCLUSÃO

O projeto cumpriu os objetivos propostos pelo professor André, permitindo a interação da equipe e troca de ideias para desenvolver uma aplicação simples e objetiva. O uso de conceitos como herança, polimorfismo, encapsulamento, abstração e interfaces foi analisado, entendido e plenamente demonstrado na prática.

Para além, o uso de recursos como logs, serialização e manipulação de dados com o JSON contribuiu para o aprofundamento no desenvolvimento de aplicações reais. A estrutura MVC adotada e aplicada para garantir uma estrutura hierárquica com arquitetura limpa e que possamos reutilizar em projeto futuros.

A IA ChatGPT e GEMINI foi utilizada como apoio técnico para revisar e refinar o código JAVA, garantindo excelência na entrega do projeto e busca para a nota máxima.

REFERÊNCIAS

DEITEL, H. M.; DEITEL, P. J. Java: Como Programar. 10. ed. São Paulo: Pearson, 2016.

GSON - Google. Disponível em: <https://github.com/google/gson>

ORACLE. Documentação oficial do Java. Disponível em: <https://docs.oracle.com/en/java>

OPENAI. ChatGPT. <https://chat.openai.com>