

GELASIO EBEL JUNIOR

OSMAR BARBOSA GOMES DA CONCEIÇÃO

PADRÕES DE PROJETO

SISTEMA DE SIMULAÇÃO DE CORRIDA F1

O domínio escolhido foi Formula 1, nosso sistema simula corridas de F1 ocorridas durante os anos e que foram guardadas como uma base de dados pela comunidade, no site: <http://ergast.com/mrd/db/> na Ergast Developer API. A título de curiosidade 2024 será o último ano que esta API será atualizada, nos moldes atuais sendo atualizada futuramente no que é chamado Jolpica, que será baseado em JSON, mas que ainda se encontra em versão alfa mas que será retroativo aos dados contidos no Ergast.

USO DA API:

Foi feito o download do site: <http://ergast.com/mrd/db/> das databases do mysql e também em arquivos *.csv e foram convertidos em um banco SQLite para uso no projeto, de modo offline e sem necessidade de configuração por parte do usuário, porque facilidade de uso é algo que nós também consideramos “padrões” de boas práticas . :D

- - Pilotos e suas características (nome, código, nacionalidade, número)
- - Equipes e suas propriedades (nome, nacionalidade)
- - Corridas com informações detalhadas (circuito, data, país)
- - Estados dos pilotos durante a corrida (correndo, abandonou, finalizado)
- - Resultados e estatísticas
- - Visualização em tempo real do progresso da corrida buscando a duração da corrida em milissegundos

Demos um passo a mais nos manifestando em fazer bom uso da tendência atual de usar IA como uma ferramenta que nos auxilia a ir mais longe e a utilizamos para nos ajudar a criar uma interface e apesar de todos os bugs, não puderem ter sido corrigidos a tempo, ela fez nossa aplicação ser mais interessante.

CONCLUSÃO

A aplicação dos padrões de projeto selecionados resultou em uma aplicação prototipal que pode ser reusada como uma base sólida para futuras funcionalidade e manutenções. Com certeza esse deve ser um projeto que tentarei expandir e aprimorar, a ponto de que mesmo com bugs, grosseiros, ela ainda possui alguma funcionalidade ativa, o que torna mais fácil de debugar.

Podemos concluir que:

SINGLETON: Garante gerenciamento eficiente de recursos e consistência de dados.

BUILDER: Proporciona construção segura e flexível os objetos.

FACADE: Simplifica a interface do sistema e o acesso a partes complexas do sistema

STATE: Organiza comportamentos e impõe estados sem necessidade de códigos de validação extensos.

PADRÕES CRIACIONAIS

SINGLETON

O padrão Singleton foi implementado para o:

1. - Gerenciador de Banco de Dados (DatabaseManager)
2. - Simulador de Corrida (SimuladorF1Facade)
3. - Acesso a Dados de Corrida (CorridaDAO)

A escolha do Singleton para estes componentes foi motivada para:

- - Evita sobrecarga do sistema com múltiplas instâncias desnecessárias
- - Garantir a Consistência de Dados e que eles não se repitam,
- - Garantir que todos os componentes acessem os mesmos dados
- - Previne inconsistências que poderiam surgir de múltiplas instâncias, como repetição de linhas
- - Simplificar a sincronização entre os dados recolhidos
-

BUILDER

O padrão Builder foi implementado para a construção dos objetos principais:

1. - Pilotos
2. - Equipes
3. - Corridas

A escolha do Builder foi para:

- - Facilitar a necessidade de construção em etapas
- - Garantir a consistência dos objetos após a construção
- - Prevenir modificações acidentais
- - Código mais legível e manutenível
- - Reduzir bugs de construção

PADRÃO ESTRUTURAL:

FACADE

O Facade foi implementado em SimuladorF1Facade, para:

- -Garantir um ponto central de acesso às funcionalidades do sistema.
- -Redução da complexidade visível do sistema
- - Interface única e coesa para funcionalidades relacionadas
- - Esconder os detalhes da implementação

PADRÃO COMPORTAMENTAL:

STATE

O State foi implementado para:

1. - Estado de corrida
 2. - Estado de abandono
 3. - Estado de finalização – posição
- - gerenciar os diferentes estados dos pilotos durante a corrida, incluindo:
 - - Separar claramente os estado do piloto nas corridas
 - - Facilitar a adição de novos estados, simplificando a manutenção de código
 - - Eliminar condicionais complexas
 - - Organizar a lógica