Procedimento de Entrega - Verificações - Projeto - O projeto está sem erros de compilação?

Procedimento de Entrega - Envio - Enviar o trabalho pelo moodle (link do projeto + relatório) até 08/06/14.

Procedimento de Entrega - Verificações - Diagrama - O modelo está de acordo com o projeto?

Procedimento de Entrega - Verificações - Projeto - O projeto está sem erros de lógica? Procedimento de Entrega - Verificações - Relatório - O Relatório está explicando os padrões utilizados?

Procedimento de Entrega - Verificações - Diagrama - O Modelo está organizado e dentro dos padrões de qualidade?

Procedimento de Entrega - Verificações - Relatório - O relatório deve conter conclusão sobre o uso dos padrões no auxilio dos padrões de projeto.

Procedimento de Entrega - Verificações - Relatório - O relatório deve conter explicação da utilização da arquitetura mvc.

Procedimento de Entrega - Verificações - Relatório - O relatório deve conter conclusão numerica apartir de dados do sonar.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o movimento do bispo.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o movimento do torre.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o movimento do cavalo.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o movimento do rei.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o movimento da rainha.

Implementação - Requisito Funcional - Partida - Pontuação - O sistema deve registrar a vitória (vale 3 pontos).

Implementação - Requisito Funcional - Partida - Pontuação - O sistema deve registrar o empate (vale 1 ponto).

Procedimento de Entrega - Verificações - Relatório - O Relatório está mostrando as tarefas que foram implementada e as que não foram?

Implementação - Requisito Funcional - Peças - Movimentos - Especiais - O sistema deve permitir o movimento roque grande.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o movimento do peão.

Implementação - Requisito Funcional - Peças - Movimentos - Especiais - O sistema deve permitir o movimento roque menor.

Implementação - Requisito Funcional - Peças - Movimentos - Especiais - O sistema deve permitir o movimento "en passant".

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o xeque.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o xeque-mate.

Implementação - Requisito Funcional - Peças - Movimentos - Básicos - O sistema deve permitir o movimento do peão (inicial).

Implementação - Requisito não Funcional - O sistema deve utilizar interface gráfica swing.

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve permitir que o usuario pause o jogo para continuar depois.

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve permitir que o usuario desista.

Implementação - Requisito de Implementação - Recursos - O projeto deve utilizar generics e interfaces.

Implementação - Requisito não Funcional - O sistema deve utilizar um chat para as conversas.

Implementação - Requisito não Funcional - O sistema deve utilizar danco de dados para salvar as entidades.

Implementação - Requisito não Funcional - O sistema deve utilizar interface amigável.

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve registrar o numero de vitórias, derrotas e empates.

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve permitir o jogador alterar seus dados (apenas os seus).

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve cadastrar cada jogador (nome,email,apelido).

Implementação - Requisito Funcional - Partida - Pontuação - O sistema deve registrar a derrota (vale 0 pontos).

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve permitir que o usuario peça empate.

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve permitir que o usuario jogue em dupla.

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve permitir que usuario jogue sozinho.

Implementação - Requisito Funcional - Partida - Jogador - O sistema deve ter um ranking de jogadores.

Implementação - Requisito de Implementação - Padrões - Criátivo - O projeto deve utilizar singular (singleton).

Implementação - Requisito de Implementação - Padrões - Criátivo - O projeto deve utilizar protótipo.

Implementação - Requisito de Implementação - Padrões - Criátivo - O projeto deve utilizar método fabrica.

Implementação - Requisito de Implementação - Padrões - Criátivo - O projeto deve utilizar fábrica abstrata.

Implementação - Requisito de Implementação - Recursos - O projeto deve utilizar hypersonic com dao para manipulação de dados.

Procedimento de Entrega - Verificações - Relatório - O relatório deve conter explicação de cada padrão utilizado.

Implementação - Requisito de Implementação - Padrões - Estrutural - O projeto deve utilizar compositor ou outros (se necessário).

Implementação - Requisito de Implementação - Conceitos - O projeto deve utilizar nomenclatura padrão java.

Implementação - Requisito de Implementação - Conceitos - O projeto não deve utilizar variáveis globais.

Implementação - Requisito de Implementação - Recursos - O projeto deve utilizar o maven

Implementação - Requisito de Implementação - Recursos - O projeto deve utilizar reflection.

Implementação - Requisito de Implementação - Conceitos - O projeto deve utilizar o menor numero de classes possíveis.

Implementação - Requisito de Implementação - Conceitos - O projeto deve utilizar orientação a objetos.

Implementação - Requisito de Implementação - Padrões - Arquitetural - O projeto deve utilizar modelo-visão-controle.

Implementação - Requisito de Implementação - Recursos - O projeto deve utilizar testes unitários nas classes de negócios.

Implementação - Requisito de Implementação - Conceitos - O projeto deve utilizar clean code.

Padrão Arquitetural MVC

Na fase de projeto, o sistema foi dividido seguindo dois estilos arquiteturais sendo eles o estilo Partição e Camadas. Por opção e circunstancias (de tempo e até da estrutura herdada) foi decidido manter apenas uma partição e organizar a estrutura interna em camadas.

Para estrutura das camadas foi utilizado o padrão Modelo-Visão-Controle e suas subdivisões, sendo elas: Camada de Interação Humana, Camada de Controle de Interface, Camada de Domínio do Problema, Camada de Gestão de Tarefas e Camada de Gestão de Dados. Essa estrutura permite que o projeto seja organizado de acordo com os tipos de tarefas, que basicamente são:

|  |  |  |
| --- | --- | --- |
| Camada | Subcamada | Descrição |
| Modelo | Gestão de Tarefas | Camada responsável pelo comportamento das entidades do projeto (ex. casos de usos do sistema). |
| Domínio do Problema | Camada responsável por representar todas as entidades envolvidas no projeto. |
| Gestão de Dados | Camada responsável por gerenciar a manipulação da base de dados. |
| Visão | Interação Humana | Camada responsável por reunir as classes que manipulam a interface. |
| Controle | Controle de Interação | Camada Responsável por ser uma ponte entre a comunicação dos modelos e as visões. |

Conclusão

A utilização dos conceitos de orientação a objetos, desenvolvimento voltado à interface, utilização de composição ao invés de heranças, os padrões de projeto e arquitetural, controle de versão, qualidade de código e modelagem permitiram que o projeto pudesse ser desenvolvido com um foco em melhorar organização e manutenabilidade do código alem do próprio desempenho. A utilização destes conceitos impacta muito no que diz respeito à complexidade da tarefa de desenvolver, mas impacta para melhor visto que não se trata apenas de desenvolver e sim desenvolver com qualidade. O resultado é um produto de alto padrão de qualidade extremamente diferenciado no mercado.