

# Relatório xadrez com MVC Parte 2

Caio Cesar Aguiar Alarcon Nº7241109

30 de Junho de 2019

## Introdução

Nesta parte do trabalho o objetivo foi usar arquivos e implementar serialization, threads, exceptions e sockets. Para isso, o jogo conta agora com timer e é possível jogar contra o computador se houver conexão com a rede. Caso contrário uma mensagem de erro é emitida usando-se o recurso das exceptions. Além disso o jogo salva o estado atual a cada segundo sendo assim possível recuperá-lo assim que o jogo for iniciado.

## Tecnologias Implementadas

### Exceptions:

Foram usadas exceptions para emitir mensagem de erro quando o jogo não conseguir se comunicar com o servidor. Isso foi feito com a classe 'erro' que é herdeira da classe 'Exception' e cujo único método consiste em retornar a mensagem de erro de conexão.

### Threads:

Threads foram usadas para implementar o timer do jogo e também para evitar que o jogo, fique travado até que a resposta chegue da rede. Isso foi feito através da classe TabuleiroController. Agora ela herda as características da classe 'Thread' cujo método 'run' é executado por uma thread própria. Dentro deste método um loop infinito garante que a thread não termine sua execução. Dentro deste loop coisas importantes são feitas, como renderização das partes gráficas e gerenciamento dos timers.

### Socket:

Um socket foi implementado para obter as jogadas de um servidor. Isso foi feito através da classe `player`. Ele se conecta com o servidor por meio de um socket por onde faz requisições através do protocolo HTTPS para um servidor web Apache. Neste servidor há um [script PHP](#) que repassa a solicitação para o executável do [stockfish](#) que foi baixado via git e compilado no servidor previamente. A conexão toda ocorre em thread separada para que a espera pela resposta não cause travamento no sistema inteiro. Além disso uma mensagem de erro é emitida utilizando conceitos de Exceptions caso a conexão com o servidor falhe. Porém isso acontece apenas assim que o programa é iniciado, caso a conexão seja interrompida durante sua execução, este erro não foi tratado adequadamente.

### Arquivos e Serialization:

Ambas tecnologias foram utilizadas para salvar o estado atual do jogo bem como recuperá-lo quando o jogo for iniciado novamente. Isso foi feito utilizando herança nas classes `ModelTabuleiro` e `Peca` para que elas sejam serializáveis. Posteriormente uma função para salvar o estado do jogo foi implementada na classe `TabuleiroController` e o salvamento ocorre a cada segundo, aproveitando o timer. Além disso, ao inicializar o jogo o programa verifica se o arquivo de configuração existe, caso exista ele carrega o estado salvo.

## Conclusão

O uso de orientação a objetos bem como de suas classes prontas que caracterizam a linguagem Java como sendo uma linguagem de alto nível propicia a criação de sistemas completos, robustos e seguros utilizando-se relativamente pouco código se comparado com linguagens de nível mais baixo como a linguagem C, por exemplo, ou mesmo C++, por não terem um ambiente tão completo como Java possui. Desta maneira, é possível usar orientação a objetos com facilidade para criação de interfaces sem perder desempenho pois é possível integrar com facilidade linguagens de desempenho mais alto por meio de rede tornando a aplicação de simples implementação, porém, robusta. Além disso o uso de mvc facilita o remanejamento da parte gráfica com as outras partes, desta forma atualizações de sistema e também o trabalho em equipe são muito facilitados.