**Trabalho sobre Clean Code**

**Explicação ao professor:** Então professor, primeiramente eu estou fazendo esse trabalho sozinho porque o meu colega simplesmente entra nas aulas e não fala nada, entra no WhatsApp e não responde as mensagens. Eu decidi comentar sobre os capítulos 10, 11 e 12, pois estão interligados indiretamente. Vai ser um trabalho bem porco mesmo, porque eu não tenho muito tempo na semana, eu moro longe do centro que é onde fica a sala da empresa.

**Breve resumo:** Os capítulos 10, 11 e 12 cobrem conceitos de desenho de alto nível: classes, sistemas e desenho emergente. O Capitulo 10 está fortemente focado no Princípio de Única Responsabilidade, um conceito que vem de um trabalho anterior do Martin, Agile Software Development: Principles, Patterns, and Practices. Aqui, o tratamento não foi tão profundo, mas é um bom ponto de partida. O Capitulo 11 fala de código limpo no contexto de técnicas a nível de sistema como Injeção de Dependências e Programação Orientada a Aspectos. O capítulo 12 nos lembra para dependermos dos testes e da refatoração para guiar tanto o desenho de alto nível, como as decisões de baixo nível. Nesta parte específica do livro, temos 4 regras. **1ª Efetue todos os testes**, ou seja, certifique-se de que o código está funcionando 100%. **2ª Refatoração, basicamente**, para cada linha de código nós nos perguntamos. Acabei de prejudica-lo? Se sim, nós devemos refatorar ele. **3ª Sem código repetido**, como o próprio nome diz, nada de linhas repetidas, sempre procure poupar suas linhas. **4ª Expressividade**, o código deve expressar claramente o propósito de seu autor. O capítulo também comenta sobre não repetir código, podemos ter implementações separadas para cada método. Para evitar essas repetições, gradualmente refatoramos o código.