**APS 2**

Identifique uma oportunidade de *abstração* em seu projeto final da disciplina *Design de Software*. **Com certeza alguma existe.** Sugira uma modificação que aproveite essa oportunidade. Se você está fazendo a APS em dupla, escolham o projeto de um dos membros. Não precisa ser o mesmo projeto da APS 1.

Você não precisa necessariamente escrever código, mas lembre que eu devo entender sua resposta mesmo sem nunca ter visto seu projeto antes. Você pode adicionar figuras e/ou links para códigos no *GitHub* se achar que deixam seu texto mais compreensível.

Apesar do espaço enorme que estou disponibilizando aqui, sua resposta não precisa **nem deve** ser muito longa. Capacidade de síntese também faz parte do que estou avaliando.

Meu projeto final de Desoft foi o jogo Archery Physics, basicamente um jogo de arco e flecha em que você possui 5 tentativas para acertar uma maçã na cabeça de uma pessoa. Caso você mate a pessoa ou suas tentativas se esgotem, você perde. Para ganhar basta acertar a maçã:

Uma imagem contendo céu, ao ar livre

Descrição gerada com muito alta confiança

Duas componentes do jogo são a barra de vida e a barra de velocidade, a implementação de ambas foi feita exatamente da mesma maneira como é possível visualizar na figura abaixo:

**APS 2 2**

Uma imagem contendo monitor, tela, parede

Descrição gerada com muito alta confiança

Podemos observar que foi escrito exatamente a mesma coisa em *def* ***barra\_devida()*** e *def* ***barra\_speed()***, mudando apenas alguns parâmetros, como os de dimensões e cor.

Essas duplicações de código poderiam ser solucionadas com abstrações, criando uma classe que implementa um tipo genérico de barra na tela e posteriormente instanciando objetos dessa classe para implementar a barra de vida e a barra de velocidade.

Com essa abstração não há necessidade de tanta repetição de código e a manutenção fica mais fácil, caso o código precise ser modificado. Sem contar que desta forma o código fica mais coeso.