



(<http://dev.rbtech.info>)



## Vantagens de programar com Orientação a Objetos (OOP)

Publicado em: Categoria(s)

29/10/2012 [Planejamento \(http://dev.rbtech.info/assuntos/teoria/planejamento/\)](http://dev.rbtech.info/assuntos/teoria/planejamento/)

Autor:

**Ricardo Bernardi** (<http://dev.rbtech.info/author/ricardo/>)

Interação:

**9 comentários** (<http://dev.rbtech.info/vantagens-de-programar-com-orientacao-a-objetos-oop/#comments>)

Primeiramente gostaria de deixar o comentário que me "inspirou" a escrever este artigo:

Porque será que mesmo tendo vendo todas as aulas, eu ainda não consegui enfiar na cabeça um "por que" usar OO. Eu sei que todos falam que OO é melhor, mas como eu já disse em outro post, eu faço qualquer coisa no php usando programação extruturada(sem OO), ai vi todas as aulas e entendi direitinho. Mas eu ainda não consegui enfiar na cabeça um "porque" usar OO. Eu ainda estou pensando que se for fazer isso tudo ai sem OO é mais fácil. Deus do céu, eu tenho que enfiar esse OO dentro do meu cérebro, mas ele parece não querer entrar. Parece que minha cabeça esta recusando o OO.

Este comentário foi deixado pelo leitor **fbinfo** na aula 15 do curso de [PHP Orientado a Objetos \(http://dev.rbtech.info/curso-de-php-orientado-a-objetos-aula-01/\)](http://dev.rbtech.info/curso-de-php-orientado-a-objetos-aula-01/), e lendo este relato eu percebi que talvez não tenha conseguido expor muito bem as vantagens e possíveis desvantagens de programar com OO.

### Porque as pessoas não percebem vantagem ao estudar OOP?

Isso aconteceu comigo também, quando eu comecei a estudar OO eu pensei "como isso vai me ajudar em algo, se é tudo bem mais complicado do que a programação normal". Foi então que comecei ir atrás de mais conhecimento e também de teorias sobre o assunto, e não demorou muito para eu me empolgar e fazer meu primeiro sistema em OOP.

Foi um sistema simples de login, que eu tinha visto em uma série de aulas em um site, onde o autor levou mais de 2 horas de aula para fazer a codificação. Montei a lógica do sistema em 10 minutos (o mesmo tempo que o autor do curso original) e depois em mais 45 minutos eu tinha o sistema pronto, com as mesmas funcionalidades que o original e em menos da metade do tempo.

Foi neste instante que "caiu a ficha", pois não tinha sido nada complicado, muito pelo contrário, pela primeira vez na minha vida eu tinha achado programação fácil.

### Mas o que eu ganho programando em OOP?

Acho que pelo exemplo acima citado já temos uma noção sobre a agilidade que a OO nos oferece, mas vamos fazer uma lista um pouco mais completa, para que todos tenham mais argumentos para analisar:

- A codificação fica mais próxima do cenário real do problema a ser resolvido: é muito mais simples ter uma classe pessoas que faz o cadastro, atualização, exclusão e busca do que uma função para cada ação necessária.
- A manutenção futura fica mais simples e rápida: você somente altera as classes que são necessárias, sem precisar ficar procurando funções perdidas em meio aos arquivos para alterar algo.
- Maior reutilização de código: imagine que você crie uma classe que faça as operações de INSERT, UPDATE e DELETE do banco de dados, e crie uma classe específica para SELECTs, durante todo o restante do sistema você vai estar usando as mesmas classes, pois se bem planejadas eles se tornam muito mais dinâmicas do que uma simples função.
- Padronização do sistema: criando um sistema em programação procedural, ele fica com o padrão do programador ou da empresa, já em OOP ele fica com o padrão da OOP, ou seja, o sistema será "entendível" para qualquer desenvolvedor futuramente.
- Segurança: é muito mais difícil burlar um sistema orientado a objetos do que um sistema procedural, uma vez que não basta simplesmente inserir alguma função ou algo do gênero, pois é preciso instanciar objetos para que a "coisa funcione". Ainda existe a possibilidade de definir propriedades e métodos públicos, protegidos ou privados, aumentando ainda mais o nível de segurança.
- Herança de código: digamos que você tenha uma classe pessoas com as propriedades nome, email e telefone e precise gerenciar pessoas físicas (com CPF) e jurídicas (com CNPJ). Basta criar uma classe física herdando tudo da classe pessoas e adicionando a propriedade CPF, e fazer a mesma coisa para jurídica.

Isso é o principal, mas se for parar para pensar com calma ainda tem como achar algumas outras vantagens. Mas como diz o ditado, "nem tudo na vida são flores" e programar com OO é um pouco mais complexo para iniciantes, pois a curva de aprendizado é um pouco maior do que uma simples programação procedural.

O fato é que uma vez que você desenvolve seu primeiro sistema em OOP você passa a ter uma grande aversão ao sistema procedural, e dificilmente você sentirá saudades!