# Paradigmas da Orientação a Objetos

Vivemos em um mundo que para desenvolvermos um software existem centenas de linguagens de programação disponíveis. Dentro dessas linguagens algumas possuem regras rigidas que ajudam a manter o padrão de desenvolvimento e assim melhorar a manutenção, extensão e o proprio desenvolvimento em si.

Eu considero a programação orientada a objetos uma evolução da programação estruturadas, que é uma forma de executar comandos, quase linearmente permitindo uma organização no código a nível estrutural.

Na orientação a objetos essa organização passa a ser um objeto com propriedades, ações, eventos, para que possamos entender melhor temos que conhecer os pilares dessa ‘estrutura’ de objetos:

## Os 4 pilares do desenvolvimento orientado a objetos

### Abstração

Abstração de um objeto seria praticamente copia-lo com sua estrutura e propriedades para o código, o programador abstrai um objeto ou entidade da vida real, para o codigo criando suas caracteristicas desejadas, que serão utilizadas no código. Esse pilar permite um entendimento muito mais elevado do desenvolvimento de código do que na programação estruturada. Por exemplo: Classe Pessoa tem os atributos Nome, endereço, idade...

### Encapsulamento

Quando criamos a classe Pessoa ela tem propriedades e métodos que somente a propria classe poderá saber e acessar, ou seja quando ela for instanciada essas propriedades e métodos não poderão ser utilizados. Por exemplo se criássemos o metodo Comer(alimento), o proprio método chamaria outro método escondido chamado FazerDigestao(alimento) e isso não dependeria da gente e sim uma responsabilidade interna da classe.

### Herança

Mais um pilar fascinante, como na vida real os objetos podem herdar de um outro objeto podendo usar suas caracteristicas por exemplo a classe Caucasiano herda a classe pessoa, e implementa outro pilar que é o

### Polimorfismo

É a capacidade de “mutar” a classe herdada, sobre-escrevendo suas propriedades e métodos por exemplo o objeto atleta herdado da classe pessoa quando chamado o método Pular() pode alcancar alturas maiores que 2 metros.

Enfim os paradigmas da programação orientada a objetos, nos fornece uma maneira prática, intuitiva e podemos dizer até “gostosa” de se desenvolver software, embora pareça complicado no começo o grau de aprendizado aumenta muito com pouco tempo de prática.

Giuliano Lemes Pereira (obs.: O texto está bem básico porque foi escrito de cabeça sem consultar)