**Aula05 – 28/04/25**

**Desenvolvimento Orientado Objeto**

Pode conter vários nomes como **POO** (**OOP** em inglês), **POG** – Programação Orientada Gambiarra.

OOP é a base, para o que chamamos de código limpo, APIs, entre outras...

**Classe x Objeto**

Objeto ele puxa as atribuições da classe, como por exemplo:

**Classe** = Costumer que contém ID, Nome, Sobrenome, Idade. Assim, quando indexado no objeto vai puxar todos os recursos a cima

**Classe:**

* Código definido com a palavra-chave class;
* Formatada por propriedades e funções/métodos que compõem os membros da classe.

**Objeto:**

* Uma instância de uma classe na memória do computador;
* Variável que possibilita o acesso aos métodos e propriedades especificadas pela classe;
* Ao encerrar o programa, o objeto é eliminado da memória, mas a classe permanecerá especificada no código fonte do programa.

Os objetos possuem o formato especificado pela classe, porém, cada um deles podem armazenar valores diferentes em suas propriedades ocasionando comportamento diferentes em seus métodos.

Uma abordagem para projetar e construir aplicações que são:

* Flexíveis;
* Naturais;
* Bem elaboradas;
* Testáveis.

Focada em objetos que interagem uns com os outros.

**Atributos**

**Homogêneos:** em programação são aqueles que possuem **valores ou elementos do mesmo tipo**, como por exemplo, uma lista que contém apenas números ou uma matriz onde todas as células possuem valores do mesmo tipo de dados.

**Heterogêneos:** são aqueles que podem conter **valores ou elementos de diferentes tipos**. Um exemplo seria um objeto ou estrutura que combina tipos variados, como números, strings e outros objetos.

1. Identifique as classes dos requisitos ou especificações para definir as entidades da aplicação;
2. Analise as classes identificadas e separe as responsabilidades. Tenha em mente que a aplicação pode ser decomposta com o mínimo de impacto;
3. Estabeleça os relacionamentos entre as classes para que os objetos possam operar em conjunto na resolução da proposta do software;
4. Reutilize. Construa seu código pensando em reaproveitamento do código para evitar repetições e facilitar manutenções.

**Busines Objects**

* São as classes projetadas para solucionar problemas de um determinado domínio de aplicação de software.
* Certamente teríamos uma definição de classe Comsumidor em um sistema comercial, contendo nome e e-mail como propriedades, como por exemplo:

**Entidade**

**Entity:** Qualquer coisa importante do domínio da aplicação ou do mundo real que precisa ser representado como uma classe.

|  |
| --- |
| **Identifying Classes** |
| **Separating reponsibilities** |
| **Establishing relationships** |
| **Leveraging reuse** |

**Atividade**

Você recebeu uma solicitação por e-mail de uma amiga veterinária, informando que está com problemas para gerenciar os atendimentos dos pets. Na mensagem, ela diz que gostaria de ter registro dos ANIMAIS, que precisa também dos dados dos seus parceiros VETERINÁRIOS que atendem no centro clínico para poder saber quais médicos atenderam quais animais em um determinado PERÍODO DE TEMPO, ou até mesmo, registrar quais PROCEDIMENTOS foram realizados com os animais nos últimos ATENDIMENTOS.

**Aula05 – 19/05/25**

Ler conteúdo dos slides (importante)

**Aula05 – 26/05/25**

Existem praticamente 2 formas de estruturar dados em arquivos, guardar, armazenar de forma estruturada

1. **Estrutura Delimitada:** Escolho um caractere (;) e delimito cada atributo de dados, arquivos **.CSV**.
2. **Tamanho Fixo:** Muito comum e utilizado em códigos de barra, segue um padrão como um boleto.

Pesquisar sobre os a sigla **CQLS(Comands)**

**Atividade**

Fazer o que foi feito no Customer, para o Product!

**Aula05 – 02/06/25**

**Termo:** Injeção de Dependência (Dependency Injection).

**Desafio**

Fazer um método de IMPORTAÇÃO (upload e leitura dos dados)