

# PROJETO POO

Nome dos alunos:

---

- Daniel Alencar Penha Carvalho
- Maic de Oliveira Santos
- Pedro Henrique Amaro
- Bruno Matheus
- Italo Tenorio
- Hellian

# Perspectiva estática: pacotes

- Quantidade de pacotes: 12
- Nome dos pacotes  $M = \{\text{cidade, cozinha, estado, forma\_pagamento, grupo, pedido, permissao, produto, restaurante, usuario, util}\}$

# Perspectiva estática: pacotes

- Nome dos pacotes `m={restaurante}`
- Classes presentes em `m`: `Restaurante`
  - Visando montar um ecossistema que funcione de ponta a ponta e focando no ramo alimentício, a classe `Restaurante` é adicionada para ser uma das pontas que dá sentido a existência do código em si, sendo a fornecedora e funcionando em função da Classe usuário.

# Perspectiva estática: pacotes

- Nome dos pacotes  $m = \{\text{usuario, grupo, permissao}\}$
- Classes presentes em  $m$ : Usuario, Grupo, Permissao
  - As classe citadas estão relacionadas com a identificação do usuario, que por sua vez exige a necessidade de se diferenciar os níveis de permissão dados a cada grupo, afinal nem todos os usuários podem ter acesso as mesmas coisas.

# Perspectiva estática: pacotes

- Nome dos pacotes  $m=\{\text{cidade, estado}\}$
- Classes presentes em  $m$ : Cidade, Estado
  - Algumas das classes presentes no projeto necessitam de informações sobre localização para funcionarem com total eficiência.

# Perspectiva estática: pacotes

- Nome dos pacotes  $m=\{\text{pedido}, \text{produto}\}$
- Classes presentes em  $m$ : Pedido, ItemPedido, StatusPedido, Produto, FotoProduto
  - Esses dois pacotes são a ligação imediata entre Usuario e Restaurante e trabalham juntas na identificação e localização dos pedidos dos usuarios, permitindo com que a classe Restaurante funcione sempre de maneira correta, e o Usuario tenha acesso as informações pertinentes.

# Perspectiva estática: pacotes

- Nome dos pacotes  $m=\{\text{cozinha}\}$
- Classes presentes em  $m$ : Cozinha
  - Uma “generalização” de Restaurante, seus métodos e atributos poderiam ser implementados em Restaurante, mais por questão de modularização foi criada uma essa nova classe.

# Perspectiva estática: pacotes

- Nome dos pacotes  $m = \{\text{forma\_pagamento}\}$
- Classes presentes em  $m$ : FormaPagamento
  - O pagamento e as formas aceitas do mesmo são uma parte muito importante do projeto, portanto, se faz necessária uma classe que lide diretamente com os mesmos.



# Perspectiva estática: pacotes

- Nome dos pacotes  $m=\{\text{util}\}$
- Classes presentes em  $m$ : Endereco, Destinatario
  - As classes se encontrava sem pacote e o pacote util é utilizado para guardar coisas uteis ao funcionamento do código.

# Perspectiva estática: classes, métodos e operações

- Quantidade de classes: 16
- P representa um pacote como sendo um conjunto de classes. Para cada  $P_i$  em M nos apresentamos:
  - Nome das classes  $P = \{c_1, c_2, c_3, c_n\}$
  - Motivo da criação de cada classe, TAD (V, O)
  - Os elementos do conjunto de valores
  - Os elementos do conjunto de operações

# Perspectiva estática: classes, métodos e operações

- cidade={Cidade}
- Classes: Cidade
  - Objetivo: Generalizar o conceito de endereço.
- Quais os elementos do conjunto de valores?
  - nome, estado
- Quais os elementos do conjunto de operações?
  - construtor, getters e setters

# Perspectiva estática: classes, métodos e operações

- cozinha={Cozinha}
- Classes: Cozinha
  - Objetivo: Definir o comportamento da cozinha mediante o preparo e tempo de criação de certos produtos comestíveis.
- Quais os elementos do conjunto de valores?
  - nome
- Quais os elementos do conjunto de operações?
  - getters e setters

# Perspectiva estática: classes, métodos e operações

- estado={Estado}
- Classes: Estado
  - Objetivo: Definir detalhes do Estado, para ser possível, por exemplo, calcular a distância da localização do Estado para a localização do Restaurante associado. Assim, é possível calcular diferentes fretes de acordo com o estado.
- Quais os elementos do conjunto de valores?
  - nome
- Quais os elementos do conjunto de operações?
  - getters e setters

# Perspectiva estática: classes, métodos e operações

- `forma_pagamento={FormaPagamento}`
- Classes: `FormaPagamento`
  - Objetivo? Definir diferentes formas de pagamento para os produtos comprados pelo cliente. Especificar quantidade de vezes de divisão no cartão, por exemplo.
- Quais os elementos do conjunto de valores?
  - `descricao`
- Quais os elementos do conjunto de operações?
  - `getters` e `setters`

# Perspectiva estática: classes, métodos e operações

- grupo={Grupo}
- Classes: Grupo
  - Objetivo? Separar os diferentes tipos de usuário do Restaurante. Por exemplo, pode haver clientes comuns e clientes especiais (com desconto por exemplo)
- Quais os elementos do conjunto de valores?
  - Nome, permissao
- Quais os elementos do conjunto de operações?
  - getters e setters

# Perspectiva estática: classes, métodos e operações

- pedido={ItemPedido, Pedido, StatusPedido}
- Classes: ItemPedido
  - Objetivo: Facilitar a implementação de produtos iguais dentro do carrinho de compras (Pedido) do usuário.
- Quais os elementos do conjunto de valores?
  - Quantidade, precoUnitario, precoTotal e observação
- Quais os elementos do conjunto de operações?
  - getters e setters



# Perspectiva estática: classes, métodos e operações

- pedido={ItemPedido, Pedido, StatusPedido}
- Pedido:
  - Objetivo: Implementar o carrinho de compras do usuário. Uma forma de manipular todos os pedidos com diferentes quantidades de produtos feitos pelo usuário
- Quais os elementos do conjunto de valores?
  - código, subtotal, taxaFrete, valorTotal, dataCriacao, dataConfirmacao, dataEntrega, dataCancelamento, statusPedido, restaurante, usuario, item, posItem, formaPagamento e enderecoEntrega
- Quais os elementos do conjunto de operações?
  - Construtor, getters e setters, addItem, delItem

# Perspectiva estática: classes, métodos e operações

- `permissao={Permissao}`
- Classes: `Permissao`
  - Objetivo: É um enum para definir constantes das permissões.
- Quais os elementos do conjunto de valores?
  - `nome`, `descricao`
- Quais os elementos do conjunto de operações?
  - Getters e setters

# Perspectiva estática: classes, métodos e operações

- produto={FotoProduto, Produto}
- Classes: FotoProduto
  - Objetivo: Definir detalhes referentes ao produto
- Quais os elementos do conjunto de valores?
  - nome, descricao, contentType, tamanho
- Quais os elementos do conjunto de operações?
  - Getters e setters

# Perspectiva estática: classes, métodos e operações

- restaurante={Restaurante}
- Classes: Restaurante
  - Objetivo: Toda a lógica de entrada e saída de produtos passa pelo Restaurante
- Quais os elementos do conjunto de valores?
  - nome, taxaFrete, ativo, aberto, dataCadastro, dataAtualizacao, cozinha, Produtos, endereco, QUANT\_PRODUTOS
- Quais os elementos do conjunto de operações?
  - Getters e setters, construtor, cadastrarProduto, encontrarProduto, ativarProduto, desativarProduto, toString, definirResponsavel

# Perspectiva estática: classes, métodos e operações

- usuario={Usuario}
- Classe: Usuario
  - Objetivo: Descrever informações importantes para o acesso do usuário ao restaurante. Exemplo: nome, email e senha.
- Quais os elementos do conjunto de valores?
  - nome, email, senha, dataCadastro, grupo
- Quais os elementos do conjunto de operações?
  - Getters e setters, construtor

# Perspectiva estática: classes, métodos e operações

- util={Endereco}
- Classes: Endereco
  - Objetivo? Reúne todas as informações possíveis de localização para serem utilizadas como um endereço do restaurante e o endereço de entrega
- Quais os elementos do conjunto de valores?
  - cep, logradouro, numero, complemento, bairro, cidade
- Quais os elementos do conjunto de operações?
  - Getters e setters, construtor, calcularFrete, toString

# Perspectiva estática:Herança

- Quantos relacionamentos de herança existem:
  - um relacionamento foi implementado, Usuario é a superclasse e Cliente e Responsavel herdam dela
- Porque cada relacionamento de herança foi criado?
  - A existência da classe Usuario, facilita a programação e evita a escrita do mesmo código várias vezes.

# Perspectiva estática:Classes abstratas

- Classe GerarSenha
  - Esta classe foi criada com o propósito de auxiliar a criação de senhas para o usuário do restaurante.
  - Temos um método abstrato criado na classe Destinatario, e implementado na Endereco
- Temos um relacionamento de classe abstrata.
  - A classe Usuario utiliza esta classe para gerar senha automática para o usuário que foi instanciado pelo construtor.



# Perspectiva estática: Interfaces

- Quantas interfaces existem no projeto
  - Uma interface Destinatario foi implementada
- Explique o porquê da criação de cada interface
  - Foram feitas para facilitar o cálculo do frete
- Quantidade de relacionamentos com interfaces.
  - Endereço implementa Destinatario

# Polimorfismo

- Quantos de subtipo?
  - O método calcular frete foi sobrescrito na classe Endereço;
- Quantos de Overloading?
  - Foi utilizada uma classe abstrata com métodos geradores de senhas. Visto que senhas podem ter quantidade máxima de caracteres especificadas no programa ou não. Assim, caso não for especificada a quantidade de caracteres, a quantidade máxima da senha gerada é de 8 caracteres.

# Conclusão

- Síntese das quantidades
  - Foram implementadas 16 classes. Dentre elas uma interface
- Perspectiva de continuidade.
  - Implementação de interfaces gráficas
  - Conexão com uma API de CEPs (para cálculo dinâmico de frete)
  - Conexão com banco de dados

# Perspectiva dinâmica

- Em tempo de execução, quantos objetos foram criados para cada C em cada P?
  - Foram criados 5 objetos Permissão no pacote permissão, 4 objetos Grupo no pacote grupo, 2 Usuarios no pacote usuario, 1 Pedido no pacote pedido, e 1 Restaurante no pacote restaurante.
- Qual a quantidade total de objetos criados em tempo de execução?
  - Foram criados 13 objetos no total em tempo de execução.