



Análise e Desen.
de Sistemas

POO

Programação Orientada a Objetos I

Prof. ^a Letícia Pieper

60 h





O que é OO?

Abreviação de Orientação a Objetos, OO vem mais como uma forma de ver o mundo todo como objetos.

#INTRODUÇÃO





O que é OO?

POO, sendo um dos paradigmas de programação, é a estruturação de um programa de computador em vários objetos de alto nível.

Cada objeto é modelado com algum aspecto de algo que você precisa resolver.

#INTRODUÇÃO





O que é OO?

Todo objeto tem características e comportamentos, e isso leva a estruturação de um objeto.

Trata-se de mapear o mundo real, no mundo computacional, contudo, objetos específicos para cada software podem ser criados.

#INTRODUÇÃO





O que é um Objeto?

Encapsulamento de Estado e Comportamento.

O que sou?

Como me comporto?

#INTRODUÇÃO



ITEM	OPÇÕES	QUANTIDADE	PREÇO	REMOVER
 Relógio de Pulso Invicta Venom	Gênero Masculino Escolha uma cor: Prata	<input type="text" value="1"/>	R\$514.28	x
 Relógio de Pulso Magnum		<input type="text" value="1"/>	R\$512.69	x
 Relógio de Pulso Magnum		<input type="text" value="1"/>	R\$821.55	x

TOTAL

R\$1,848.52

FINALIZAR COMPRA

#INTRODUÇÃO



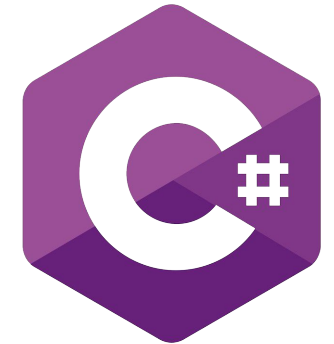
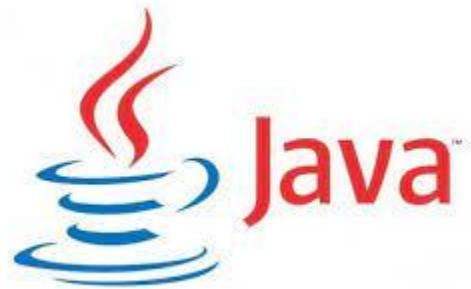
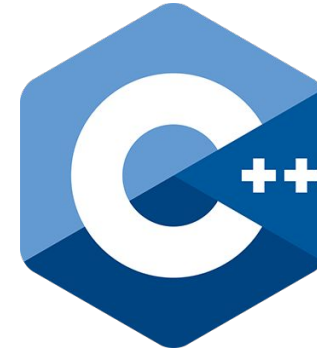


POO

POO pode ser aplicadas em diversas linguagens, sendo a própria linguagem orientada a objetos.

#INTRODUÇÃO





#INTRODUÇÃO





#INTRODUÇÃO





POO

Várias linguagens veem a POO como uma componentização do software, podendo reutilizar o mesmo objeto várias vezes, no mesmo projeto ou até em outros projetos.


#INTRODUÇÃO




SB Admin Pro

Search

Documentation >





CORE

Dashboards

Default

Multipurpose

Affiliate

CUSTOM

Pages

Applications

Flows

UI TOOLKIT

Layout

Components

Utilities

PLUGINS

Charts

Tables


Dashboard

Example dashboard overview and content summary

Feb 05, 2022 - Feb 05, 2022

Welcome to SB Admin Pro!

Browse our fully designed UI toolkit! Browse our prebuilt app pages, components, and utilities, and be sure to look at our full documentation!



Recent Activity

27 min

New order placed! Order #2912 has been successfully placed.

58 min

Your weekly report has been generated and is ready to view.

2 hrs

New user Valerie Luna has registered

1 day

Server activity monitor alert

1 day

New order placed! Order #2911 has been successfully placed.

1 day

Details for Marketing and Planning Meeting have been updated.

2 days

New order placed! Order #2910 has been successfully placed.

Progress Tracker

Server Migration

20%

Sales Tracking

40%

Customer Database

60%

Payout Details

80%

Account Setup

Complete!

Visit Task Center

Earnings (Monthly)

\$40,000

View Report

Earnings (Annual)

\$215,000

View Report

Task Completion

24

View Tasks

Pending Requests

17

View Requests

Earnings Breakdown

Monthly Revenue

Logged in as: Valerie Luna

#INTRODUÇÃO



SB Admin Pro

Search

Documentation

CORE

Dashboards

Default

Multipurpose

Affiliate

CUSTOM

Pages

Applications

Flows

UI TOOLKIT

Layout

Components

Utilities

PLUGINS

Charts

Tables

Logged in as:
Valerie Luna


Dashboard

Example dashboard overview and content summary

Feb 05, 2022 - Feb 05, 2022

Welcome to SB Admin Pro!

Browse our fully designed UI toolkit! Browse our prebuilt app pages, components, and utilities, and be sure to look at our full documentation!



Recent Activity

27 min

New order placed! Order #2912 has been successfully placed.

58 min

Your weekly report has been generated and is ready to view.

2 hrs

New user Valerie Luna has registered

1 day

Server activity monitor alert

1 day

New order placed! Order #2911 has been successfully placed.

1 day

Details for Marketing and Planning Meeting have been updated.

2 days

New order placed! Order #2910 has been successfully placed.

Progress Tracker

Server Migration

20%

Sales Tracking

40%

Customer Database

60%

Payout Details

80%

Account Setup

Complete!

Visit Task Center

Earnings (Monthly)

\$40,000

View Report

Earnings (Annual)

\$215,000

View Report

Task Completion

24

View Tasks

Pending Requests

17

View Requests

Earnings Breakdown

Monthly Revenue

#INTRODUÇÃO

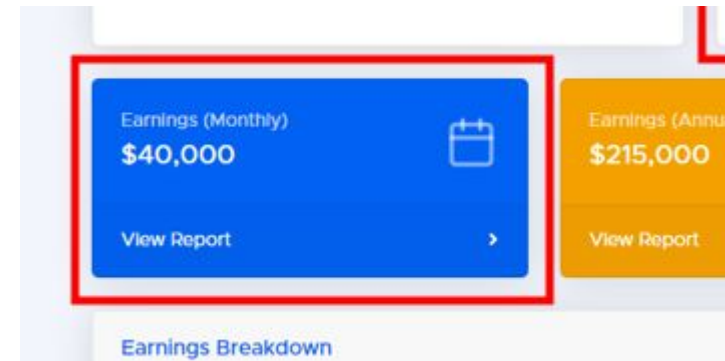




Objetos

Um objeto é a instância de uma classe.

No exemplo anterior, temos um **card**.



#INTRODUÇÃO



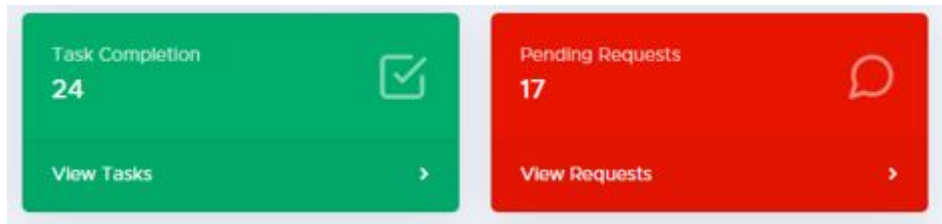
Classe Card

Título

Valor

Logo

Cor



#INTRODUÇÃO





Classe Topo

Classe Menu

Classe Detalhe

Classe Detalhe

Classe Card

Classe Card

Classe Card

Classe Card

#INTRODUÇÃO





Classe Main

Classe Topo

Classe Menu

Classe Detalhe

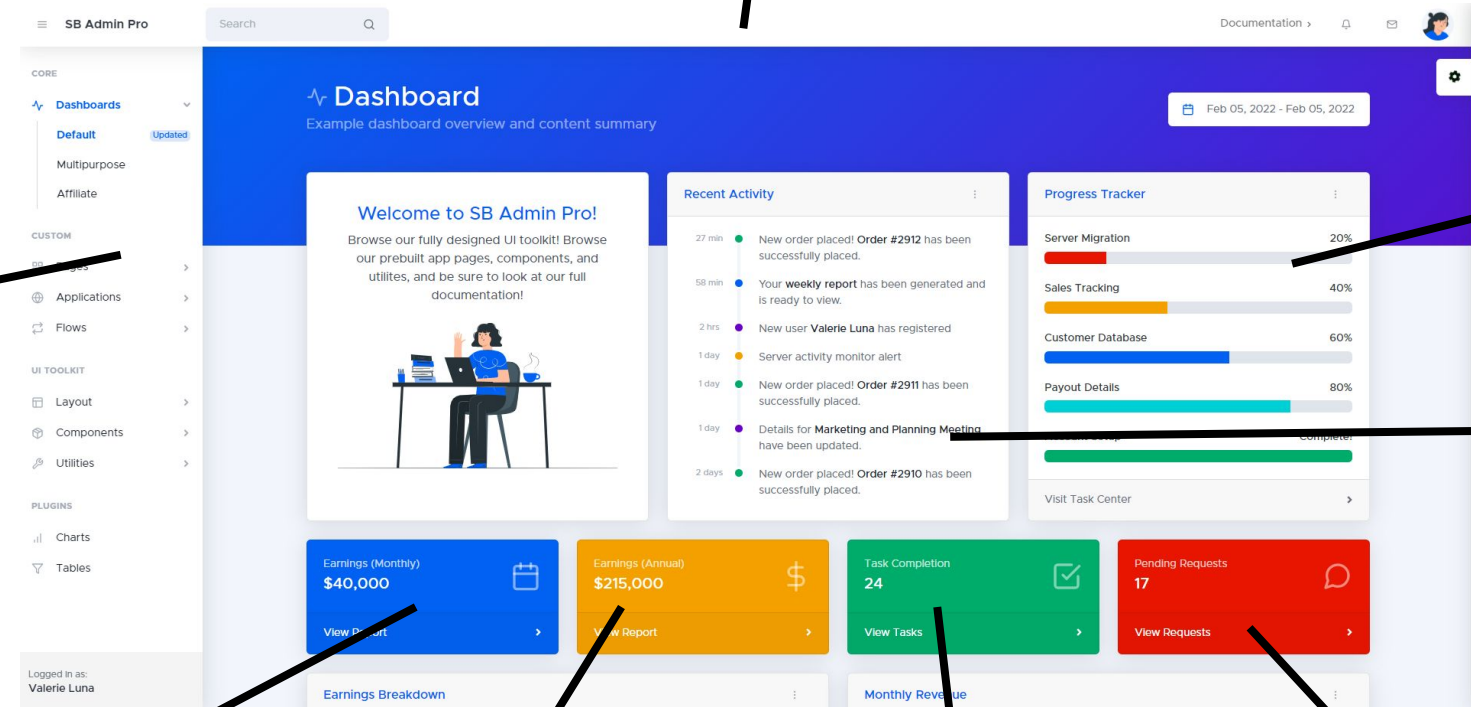
Classe Detalhe

Classe Card

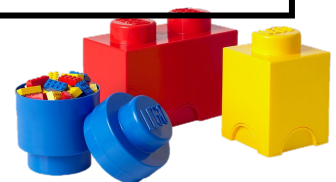
Classe Card

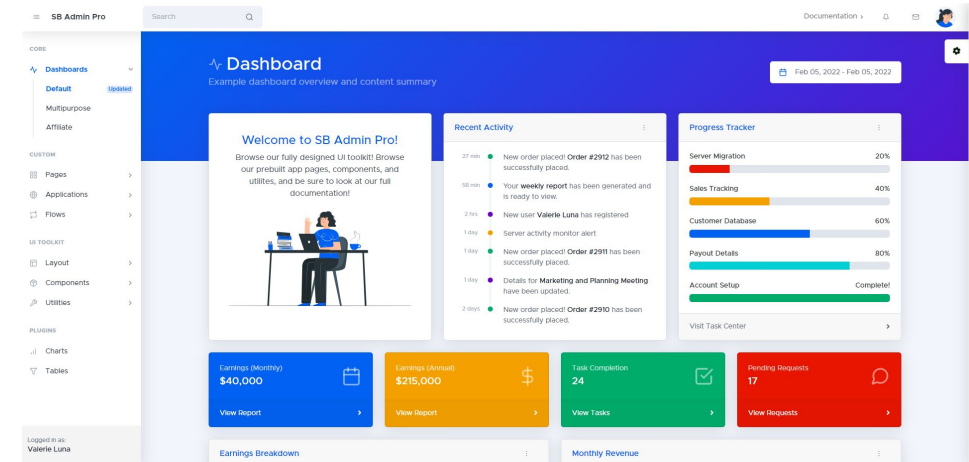
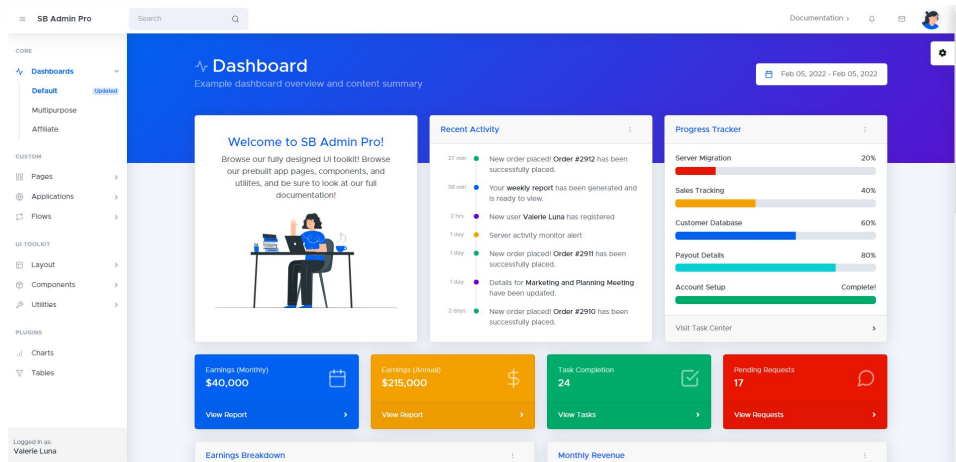
Classe Card

Classe Card

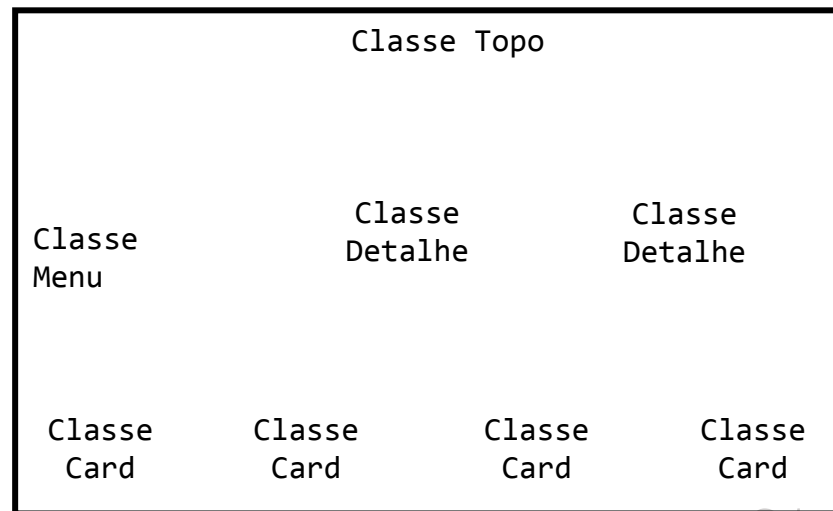


#INTRODUÇÃO

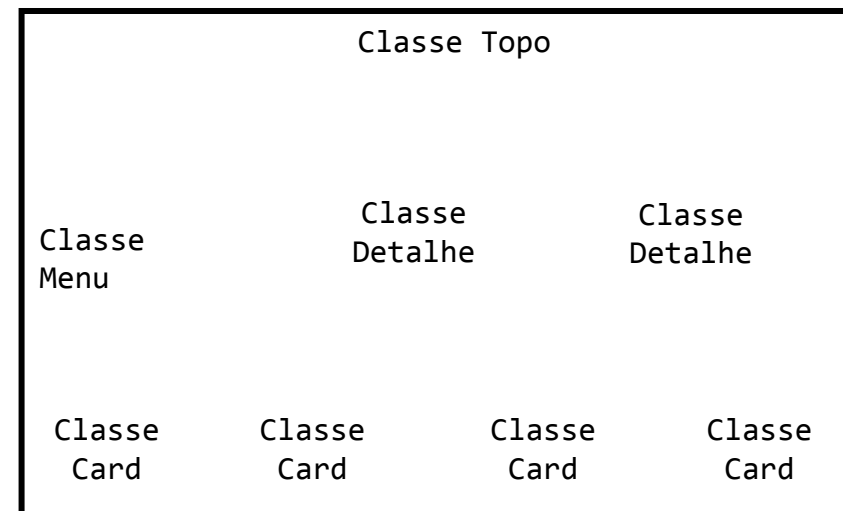




Classe Main



Classe Main



#INTRODUÇÃO





Exercícios de fixação

1. Escreva uma classe "Pessoa" com atributos para o nome, idade e endereço. (pelo menos 5 atributos)
2. Escreva uma classe "Carro" com atributos para a marca, modelo e ano. (pelo menos 8 atributos)

#INTRODUÇÃO





Exercícios de fixação

3. Escreva uma classe "Conta Bancária" com atributos para o nome do titular da conta, número da conta e saldo, e descreva que ações essa classe pode realizar.

4. Escreva uma classe "Animal" com atributos para o nome, idade e tipo.

#INTRODUÇÃO





Métodos

Em programação orientada a objetos, um método é uma funcionalidade ou operação que pode ser realizada por uma classe ou objeto.

É semelhante a uma função ou procedimento em outras linguagens de programação.

#INTRODUÇÃO





Métodos

Os métodos são definidos dentro da classe e permitem que os objetos criados a partir dessa classe realizem ações específicas.

#INTRODUÇÃO





Métodos

Por exemplo, uma classe "**Pessoa**" pode ter um método "**caminhar**" que permita ao objeto "**Pessoa**" realizar a ação de caminhar.

Além disso, os métodos também podem aceitar parâmetros e retornar valores.

#INTRODUÇÃO





Métodos

Em geral, os métodos são usados para encapsular lógicas e comportamentos específicos de uma classe, tornando o código mais organizado e fácil de manter.

#INTRODUÇÃO





Métodos

Em resumo

Os métodos são uma das principais características da programação orientada a objetos e permitem que as classes sejam projetadas de maneira modular, reutilizável e organizada.

#INTRODUÇÃO





Detalhes que fazem a diferença

#INTRODUÇÃO





Armadilha I:

Pensar em Poo apenas como uma linguagem.

Frequentemente, as pessoas equiparam linguagens orientadas a objetos com a POO. O erro surge ao supor que você está programando de maneira orientada a objetos simplesmente porque usa uma linguagem orientada a objetos. Nada poderia estar mais distante da realidade.

#INTRODUÇÃO





A verdadeira POO é um estado da mente que exige que você veja seus problemas como um grupo de objetos e use encapsulamento, herança e polimorfismo corretamente.

#INTRODUÇÃO





Armadilha II:

Medo da reutilização.

Você deve aprender a reutilizar código. Aprender a reutilizar sem culpa é uma das lições mais difíceis de aprender, quando você usa a POO a primeira vez.

#INTRODUÇÃO



2 problemas levam a essa dificuldade:

1º Os programadores gostam de criar.

Se você olhar o reaproveitamento de código de forma errada, você vai achar que ela irá te afastar da alegria de criar, porém, isso vai ajudar a criar algo maior e melhor em menos tempo.

#INTRODUÇÃO





2º Sentimento de não “foi feito por mim”.

Ou seja, não confiar no componente desenvolvido por outra pessoa.

Mesmo sendo uma ferramenta testada e atende sua necessidade, aproveite!

Lembre-se, tempo curto e qualidade...

#INTRODUÇÃO





Exercícios de fixação

1. Crie uma classe "Livro" com atributos como título, autor, número de páginas e editora. Adicione métodos para exibir essas informações e para retornar se o livro é longo ou curto (mais de 300 páginas ou não).
2. Crie uma classe "Pessoa" com atributos como nome, idade, endereço e telefone. Adicione métodos para exibir essas informações e para retornar se a pessoa é maior de idade ou não (mais de 18 anos).

#INTRODUÇÃO





Exercícios de fixação

3. Crie uma classe "Conta Corrente" com atributos como nome do titular da conta, número da conta, saldo e limite de crédito. Adicione métodos para exibir essas informações, para depositar dinheiro, para retirar dinheiro e para retornar se o saldo é positivo ou negativo.
4. Crie uma classe "Veículo" com atributos como modelo, ano, cor e número de rodas. Adicione métodos para exibir essas informações e para retornar se o veículo é antigo ou novo (mais de 10 anos ou não).

#INTRODUÇÃO

