

Identificador	Prompt	Referência
1	Crie uma classe python BancoDeDados aplicando o padrão Singleton, essa classe retorna uma instância da conexão com o banco.	<a href="https://medium.com/@christianmellolima/padrão-singleton-com-o-funcionam-onde-vivem-do-que-se-alimentam-6291fb72b22d">https://medium.com/@christianmellolima/padrão-singleton-com-o-funcionam-onde-vivem-do-que-se-alimentam-6291fb72b22d</a>
2	Crie uma classe chamada Singleton1 que cria um objeto com um nome, e uma classe Singleton2, que faz o mesmo, caso um objeto ainda não exista. Construa o algoritmo em Python e aplique o padrão Singleton.	<a href="https://github.com/kelvins/design-patterns-python/blob/main/criacao/singleton/main.py">https://github.com/kelvins/design-patterns-python/blob/main/criacao/singleton/main.py</a>
3	Crie uma classe python BancoDeDados aplicando o padrão Singleton, essa classe retorna uma instância da conexão com o banco.	<a href="https://github.com/kamranahmedse/design-patterns-for-humans?tab=readme-ov-file#-singleton">https://github.com/kamranahmedse/design-patterns-for-humans?tab=readme-ov-file#-singleton</a>
4	Crie uma classe ExampleSingleton em Python e aplique a ela o padrão Singleton.	<a href="https://www.coursera.org/learn/design-patterns/lecture/hWt11/2-1-3-singleton-pattern">https://www.coursera.org/learn/design-patterns/lecture/hWt11/2-1-3-singleton-pattern</a>
5	Crie uma classe Python FileLogger, com um método log e aplique a ela o padrão Singleton.	<a href="https://www.coursera.org/learn/desenvolvimento-agil-com-padroes-de-projeto/lecture/bYi2f/padrão-lazy-singleton">https://www.coursera.org/learn/desenvolvimento-agil-com-padroes-de-projeto/lecture/bYi2f/padrão-lazy-singleton</a>
6	Crie uma classe Python FileLogger, aplique a ela o padrão Singleton, utilizando o pacote Enum.	<a href="https://www.coursera.org/learn/desenvolvimento-agil-com-padroes-de-projeto/supplement/sPvz6/enum-singleton-o-melhor-singleton-em-java">https://www.coursera.org/learn/desenvolvimento-agil-com-padroes-de-projeto/supplement/sPvz6/enum-singleton-o-melhor-singleton-em-java</a>
7	Crie uma classe Python Tabuleiro e aplique a ela o padrão Singleton alterando o funcionamento do comando <code>__new__</code> .	<a href="https://www.coursera.org/learn/lab-poo-parte-1/supplement/MQFPu/exemplos-do-padrão-singleton-em-python">https://www.coursera.org/learn/lab-poo-parte-1/supplement/MQFPu/exemplos-do-padrão-singleton-em-python</a>
8	Crie uma classe python chamada Database e garanta que jamais haverá mais de uma instância dela.	<a href="https://refactoring.guru/pt-br/design-patterns/singleton">https://refactoring.guru/pt-br/design-patterns/singleton</a>
9	Crie uma classe Python que aplique o padrão Singleton.	<a href="https://www.geeksforgeeks.org/singleton-design-pattern/?ref=lbp">https://www.geeksforgeeks.org/singleton-design-pattern/?ref=lbp</a>
10	Crie uma classe python Database para fazer a conexão com um banco de dados sqlite3, aplique a esse algoritmo o padrão Singleton.	GIRIDHAR, Chetan. Learning python design patterns. Packt Publishing Ltd, 2016.
11	Crie uma classe "Sheep" com atributos name e category. Aplique o padrão Prototype sobre essa classe em python.	<a href="https://github.com/kamranahmedse/design-patterns-for-humans">https://github.com/kamranahmedse/design-patterns-for-humans</a>
12	Crie uma classe python Product que tenha como filhas as classes Book e DVD contendo os atributo SKU, description e length, no caso de DVD substitua length por duration, aplique o padrão Prototype.	<a href="https://edeleastar-portfolio.github.io/design-patterns-with-enterprise-arch/topic00-catalogues/talk-4/GoF_DesignPatterns_Java_UML.pdf">https://edeleastar-portfolio.github.io/design-patterns-with-enterprise-arch/topic00-catalogues/talk-4/GoF_DesignPatterns_Java_UML.pdf</a>
13	Crie uma classe Address, com os atributos street_address, city e country, e um método que retorne os atributos em uma linha. Crie também uma classe Person com os atributos name e address, além de um método igual ao de Address. Construa isso em python e aplique o padrão Prototype.	<a href="https://github.com/tuvo1106/python_design_patterns/blob/master/prototype/prototype.py">https://github.com/tuvo1106/python_design_patterns/blob/master/prototype/prototype.py</a>
14	Crie uma classe Python Circle que contenha um atributo	<a href="https://www.geeksforgeeks.org/p">https://www.geeksforgeeks.org/p</a>

	color e um método draw que escreve essa cor na tela, aplique a esse código o padrão Prototype.	<a href="https://medium.com/@jonesroberto/design-patterns-parte-7-prototype-98962514728f">rototype-design-pattern/?ref=lbp</a>
15	Crie uma classe python CombatRobot com os atributos name e weapon. Depois crie um objeto dessa classe e outro a partir dele, usando o padrão Prototype.	<a href="https://dev.to/bshadmehr/unleashing-the-prototype-design-pattern-in-python-42dn">https://dev.to/bshadmehr/unleashing-the-prototype-design-pattern-in-python-42dn</a>
16	Crie a classe python IEmployee com um método get_details, essa classe é pai das classes Manager e Developer, ambas contendo os atributos words_per_minute, name e role, porém, a classe Developer possui mais um atributo, o preferred_language. Aplique a esse algoritmo o padrão Prototype.	<a href="https://medium.com/@jonesroberto/design-patterns-parte-7-prototype-98962514728f">https://medium.com/@jonesroberto/design-patterns-parte-7-prototype-98962514728f</a>
17	Crie uma classe python Shape com um método draw, além dos atributos id e type, que deve servir de base para as classes Square, Rectangle e Circle, crie também uma classe ShapeCache que retorna cópias sob demanda, para isso aplique o padrão Prototype.	<a href="https://www.tutorialspoint.com/design_pattern/prototype_pattern.htm">https://www.tutorialspoint.com/design_pattern/prototype_pattern.htm</a>
18	Crie uma classe python EmployeeRecord com os atributos id, name, designation, salary e address, além do método showRecord, um objeto dessa classe deve ser criado a partir de valores informados pelo usuário, e outro deve ser formado clonando o primeiro, para isso aplique o padrão Prototype, não utilize bibliotecas para facilitar a aplicação do padrão.	<a href="https://www.javatpoint.com/prototype-design-pattern">https://www.javatpoint.com/prototype-design-pattern</a>
19	Crie uma classe python Vehicle com os atributos seats, tyres, color e fuel além de um método de exibição desses valores. A classe deve servir de base para outras duas: Bike e Car, aplique a esse algoritmo o padrão Prototype, não utilize bibliotecas para facilitar a aplicação do padrão.	<a href="https://www.scaler.com/topics/design-patterns/prototype-design-pattern/">https://www.scaler.com/topics/design-patterns/prototype-design-pattern/</a>
20	Crie uma classe python Color com os atributos red, green e blue, após isso crie três objetos dessa classe: red green e blue, informando os valores rgb das três cores, depois crie três cores personalizadas, por fim crie três cópias dessas cores, utilizando o padrão Prototype.	<a href="https://www.dofactory.com/net/prototype-design-pattern">https://www.dofactory.com/net/prototype-design-pattern</a>
21	Crie uma classe python Burger com os atributos size, cheese, pepperoni, tomato e lettuce. Aplique a esse código o padrão Builder.	<a href="https://github.com/kamranahmmed/design-patterns-for-humans?tab=readme-ov-file#-builder">https://github.com/kamranahmmed/design-patterns-for-humans?tab=readme-ov-file#-builder</a>
22	Crie uma classe ProdutoPizza em Python, contendo os atributos tipo_pizza, tamanho_pizza, massa, molho e cobertura. Crie também classes para as pizzas italiana, marguerita e portuguesa. Aplique ao código o padrão Builder.	<a href="https://www.coursera.org/learn/desenvolvimento-agil-com-padroes-de-projeto/supplement/1bypr/padrao-de-projeto-builder-completo">https://www.coursera.org/learn/desenvolvimento-agil-com-padroes-de-projeto/supplement/1bypr/padrao-de-projeto-builder-completo</a>
23	Crie uma classe python Car, com os atributos seats, engine, trip_computer e gps. Faça com que uma instância dessa classe possa ser construída de forma padrão ou personalizada, utilize para isso o padrão Builder.	<a href="https://refactoring.guru/pt-br/design-patterns/builder">https://refactoring.guru/pt-br/design-patterns/builder</a>
24	Crie uma classe python Computer com os atributos ram, cpu e storage, crie também uma classe ComputerDirector para construir o computador, aplique a esse algoritmo o padrão Builder.	<a href="https://www.geeksforgeeks.org/builder-design-pattern/">https://www.geeksforgeeks.org/builder-design-pattern/</a>
25	Crie uma classe python Computer com os atributos hdd, ram, graphics_card_enabled e bluetooth_enabled. O construtor dessa classe deve criar um computador com o hdd e ram passados por parâmetro, os outros atributos devem ser falsos por padrão, podendo ser ativados	<a href="https://www.digitalocean.com/community/tutorials/builder-design-pattern-in-java">https://www.digitalocean.com/community/tutorials/builder-design-pattern-in-java</a>

	depois. Aplique a esse algoritmo o padrão Builder.	
26	Crie um algoritmo python que simule a preparação de refeições, as refeições podem ser de dois tipos: vegana e não vegana, com o prato, bebida e preço variando a depender da refeição, um ponto importante também é a embalagem que envolve tanto a bebida quanto o prato, mas é de tipos diferentes para cada um. Aplique a esse algoritmo o padrão Builder.	<a href="https://www.tutorialspoint.com/design_pattern/builder_pattern.htm">https://www.tutorialspoint.com/design_pattern/builder_pattern.htm</a>
27	Crie uma classe python Endereco com os atributos logradouro, numero, complemento, bairro, cidade, estado e pais. Crie também uma classe Telefone com os atributos ddd, numero e tipoTelefone, sendo o último uma subclasse, que pode ser residencial ou celular. Por fim, crie uma classe Pessoa que contenha os atributos nome, sobrenome, data de nascimento, endereço e telefones. Aplique a esse algoritmo o padrão Builder.	<a href="https://www.mballem.com/post/simplificando-com-builder-pattern/">https://www.mballem.com/post/simplificando-com-builder-pattern/</a>
28	Crie uma classe python BankAccount com os atributos account_number, owner, branch, balance e interest_rate aplique a esse algoritmo o padrão Builder.	<a href="https://dzone.com/articles/design-patterns-the-builder-pattern">https://dzone.com/articles/design-patterns-the-builder-pattern</a>
29	Crie uma classe python Burger com os atributos bun, meat, salad, cheese e sauce, que por padrão são vazios, essa classe tem uma ramificação chamada CheeseBurger, que deve ser montada usando o padrão Builder.	<a href="https://integu.net/builder-pattern/">https://integu.net/builder-pattern/</a>
30	Crie uma classe python Profession com a enumeração Mage, WARRIOR e THIEF, crie também a classe HairType com a enumeração LONG_STRAIGHT, LONG_CURLY, SHORT e BALD, além da classe HairColor de enumeração BLOND e BLACK. Em seguida crie as classes Armor e Weapon, por fim crie a classe Hero contendo todos os atributos anteriores. Aplique a esse algoritmo o padrão Builder.	<a href="https://java-design-patterns.com/patterns/builder/#programmatic-example-of-builder-pattern-in-java">https://java-design-patterns.com/patterns/builder/#programmatic-example-of-builder-pattern-in-java</a>
31	Crie uma interface python Interviewer com uma função askQuestions implementada por duas classes, Developer e CommunityExecutive, aplique a esse código o padrão Factory Method.	<a href="https://github.com/kamranahmmed/design-patterns-for-humans?tab=readme-ov-file#-singleton">https://github.com/kamranahmmed/design-patterns-for-humans?tab=readme-ov-file#-singleton</a>
32	Crie uma classe python Product com um método write_name que imprime o nome do objeto, essa classe tem duas filhas: ProductA e ProductB, no caso de "ProductB", imprime o inverso, aplique ao código o padrão Factory Method.	<a href="https://edeleastar-portfolio.github.io/design-patterns-with-enterprise-arch/topic00-catalogues/talk-4/GoF_DesignPatterns_Java_UML.pdf">https://edeleastar-portfolio.github.io/design-patterns-with-enterprise-arch/topic00-catalogues/talk-4/GoF_DesignPatterns_Java_UML.pdf</a>
33	Crie uma classe python Point com os atributos x e y, além de um método que os retorne em uma linha, crie também dentro dela os métodos new_cartesian_point e new_polar_point, que retornam respectivamente as coordenadas cartesianas e polares. Crie também uma classe CoordinateSystem CARTESIAN = 1 e POLAR = 2 com as enumerações. Aplique ao algoritmo o padrão Factory Method.	<a href="https://github.com/tuvo1106/python_design_patterns/blob/master/factories/factories.py">https://github.com/tuvo1106/python_design_patterns/blob/master/factories/factories.py</a>
34	Crie uma classe python Section com um método describe, e suas filhas PersonalSection, AlbumSection, PatentSection, PublicationSection. Crie também as classes LinkedIn, Facebook e Profile, gerencie a criação de sessões com o padrão Factory Method.	GIRIDHAR, Chetan. Learning python design patterns. Packt Publishing Ltd, 2016.
35	Crie uma classe Dialog com os métodos create_button,	<a href="https://refactoring.guru/pt-br/desi">https://refactoring.guru/pt-br/desi</a>

	render e close_dialog, o comportamento dessa classe deve variar de acordo o método de acesso (desktop ou web) que será informado pelo cliente, construa o algoritmo em Python e aplique o padrão Factory Method.	gn-patterns/factory-method
36	Crie uma classe python Vehicle e suas filhas TwoWheeler e FourWheeler, garanta que ambos os tipos de veículo possam ser criados de forma eficiente aplicando o padrão Factory Method.	<a href="https://www.geeksforgeeks.org/factory-method-for-designing-pattern/?ref=lbp">https://www.geeksforgeeks.org/factory-method-for-designing-pattern/?ref=lbp</a>
37	Crie uma classe python Song com os atributos song_id, title e artist, crie também uma classe SongSerializer, as informações da música podem ser organizadas em formato json ou xml, utilize o padrão Factory Method para organizar essa formatação.	<a href="https://realpython.com/factory-method-python/">https://realpython.com/factory-method-python/</a>
38	Crie uma classe python Localizer, três localizações podem ser informadas pelo usuário: França, Inglaterra e Espanha, baseado no local informado a mensagem, que sempre será constituída dos meios de transporte: carro, moto e bicicleta, deverá ser traduzida para a língua local, utilize para isso o padrão Factory Method.	<a href="https://medium.com/@amirm.lavasani/design-patterns-in-python-factory-method-1882d9a06cb4">https://medium.com/@amirm.lavasani/design-patterns-in-python-factory-method-1882d9a06cb4</a>
39	Crie as classes python Dog e Cat, contendo um método speak. Aplique a esse código o padrão Factory Method.	<a href="https://dev.to/sk_rajibul_9ce58a68c43bb5/building-better-harnessing-the-factory-method-pattern-in-python-56hp">https://dev.to/sk_rajibul_9ce58a68c43bb5/building-better-harnessing-the-factory-method-pattern-in-python-56hp</a>
40	Crie as classes python Racecar e Van, contendo um método drive, aplique a esse código o padrão Factory Method.	<a href="https://pythonspot.com/factory-method/">https://pythonspot.com/factory-method/</a>
41	Crie uma interface "Door" com um método "getDescription" implementada por duas classes, "WoodenDoor" e "IronDoor", gere também uma interface chamada "DoorFittingExpert" com um método "getDescription" implementada pelas classes "Welder" e "Carpenter" aplicando o padrão Abstract Factory em python.	<a href="https://github.com/kamranahmedse/design-patterns-for-humans?tab=readme-ov-file#-abstract-factory">https://github.com/kamranahmedse/design-patterns-for-humans?tab=readme-ov-file#-abstract-factory</a>
42	Crie uma classe python HotDrink com um método consume, com as filhas Coffee e Tea. Crie também a classe HotDrinkMachine com um método make_drink com instruções para o usuário escolher uma bebida e a quantidade. Aplique o padrão Abstract Factory.	<a href="https://github.com/tuvo1106/python_design_patterns/blob/master/factories/abstract_factories.py">https://github.com/tuvo1106/python_design_patterns/blob/master/factories/abstract_factories.py</a>
43	Crie uma classe python com os métodos create_non_veg_pizza e create_veg_pizza. Crie também as classes USPizzaFactory e IndianPizzaFactory, herdando a primeira classe. Por fim crie uma classe PizzaStore. Construa o algoritmo em python e aplique o padrão Abstract Factory.	<a href="https://github.com/kelvins/design-patterns-python/blob/main/criacao/abstract_factory/main.py">https://github.com/kelvins/design-patterns-python/blob/main/criacao/abstract_factory/main.py</a>
44	Crie as classes python Windows, Linux e Mac, crie também as classes botão e janela para cada um dos SOs, construa esse algoritmo e aplique o padrão Abstract Factory.	<a href="https://www.coursera.org/learn/ab-poo-parte-1/supplement/OOXmR/exemplos-dos-padroes-metodo-fabrica-e-fabrica-abstrata-em-python">https://www.coursera.org/learn/ab-poo-parte-1/supplement/OOXmR/exemplos-dos-padroes-metodo-fabrica-e-fabrica-abstrata-em-python</a>
45	Crie um algoritmo simula um jogo com 2 tipos de temas diferentes: Faroeste e Espacial. São definidas as famílias de produtos para cada tema e criadas as fábricas concretas para cada um dos dois temas. Ao executar o script é possível escolher em tempo de execução entre os temas implementados.	<a href="https://www.coursera.org/learn/ab-poo-parte-1/supplement/OOXmR/exemplos-dos-padroes-metodo-fabrica-e-fabrica-abstrata-em-python">https://www.coursera.org/learn/ab-poo-parte-1/supplement/OOXmR/exemplos-dos-padroes-metodo-fabrica-e-fabrica-abstrata-em-python</a>

46	Gere um algoritmo python que crie botões e caixas de seleção, o comportamento desses itens varia conforme o SO informado (Windows ou Mac), Aplique sobre esse código o padrão Abstract Factory.	<a href="https://refactoring.guru/pt-br/design-patterns/abstract-factory">https://refactoring.guru/pt-br/design-patterns/abstract-factory</a>
47	Crie uma classe python Car com o método assemble e uma CarSpecification com o método display, crie também as classes Sedan e Hatchback como filhas de Car e as classes NorthAmerica e Europe, como filhas de CarSpecification aplique a esse código o padrão Abstract Factory.	<a href="https://www.geeksforgeeks.org/abstract-factory-pattern/?ref=lbp">https://www.geeksforgeeks.org/abstract-factory-pattern/?ref=lbp</a>
48	Crie as classes python Square e Rectangle, bem como suas versões arredondadas, para gerenciar a criação dessas formas aplique o padrão Abstract Factory.	<a href="https://www.tutorialspoint.com/design_pattern/abstract_factory_pattern.htm">https://www.tutorialspoint.com/design_pattern/abstract_factory_pattern.htm</a>
49	Crie as classes python Laptop e Smartphone, ambas contendo o método display, a partir disso crie um algoritmo que permita ao usuário criar eletrônicos das marcas Samsung e Dell, utilize para isso o padrão Abstract Factory.	<a href="https://dev.to/bshadmehr/mastering-the-abstract-factory-design-pattern-in-python-25gc">https://dev.to/bshadmehr/mastering-the-abstract-factory-design-pattern-in-python-25gc</a>
50	Gere um algoritmo python que crie dois tipos de veículos, carros e motos, cada um deles pode ser da marca BMW ou Mercedes, aplique a esse código o padrão Abstract Factory.	<a href="https://medium.com/@fhuseyno/v803/the-abstract-factory-design-pattern-in-python-b32acdb70857">https://medium.com/@fhuseyno/v803/the-abstract-factory-design-pattern-in-python-b32acdb70857</a>

- Código Ideal (representado pela letra A): o código gerado atendeu ao padrão requisitado e não se desviou do contexto passado via prompt.
- Falha Parcial (representado pela letra B): códigos que apresentaram qualquer variação de Falha no Prompt, mas não apresentaram Falha na Geração ou Falha no Padrão, conforme apresentado na sessão de Análise Subjetiva do Material e Métodos, logo, com certas alterações, poderiam ser aproveitados.
- Falha Crítica (representado pela letra C): códigos que apresentaram Falha na Geração ou Falha no Padrão, e, portanto, não poderiam ser utilizados.

#### Singleton

ID	Chat-GPT	Gemini	Copilot
1	C	A	A
2	A	A	A
3	A	A	A
4	A	A	A
5	A	B	B
6	C	B	C
7	B	C	A
8	A	C	A

9	A	A	A
10	A	A	A

#### Prototype

ID	Chat-GPT	Gemini	Copilot
1	A	A	A
2	C	A	B
3	A	A	C
4	A	A	C
5	A	A	A
6	A	A	C
7	B	B	C
8	B	A	A
9	B	B	B
10	A	A	C

#### Builder

ID	Chat-GPT	Gemini	Copilot
1	C	A	A
2	A	A	B
3	B	A	C
4	A	C	C
5	C	A	A
6	A	C	B
7	A	A	A
8	C	A	C
9	A	A	A
10	A	A	A

#### Factory Method

ID	Chat-GPT	Gemini	Copilot
1	A	A	A
2	A	C	C
3	A	C	C
4	A	A	A
5	B	C	A

6	B	C	C
7	A	A	A
8	A	B	B
9	A	A	A
10	A	C	A

#### Abstract Factory

ID	Chat-GPT	Gemini	Copilot
1	A	A	A
2	A	A	C
3	B	A	A
4	A	C	A
5	A	A	C
6	A	A	C
7	C	A	A
8	A	A	C
9	A	A	C
10	A	A	A