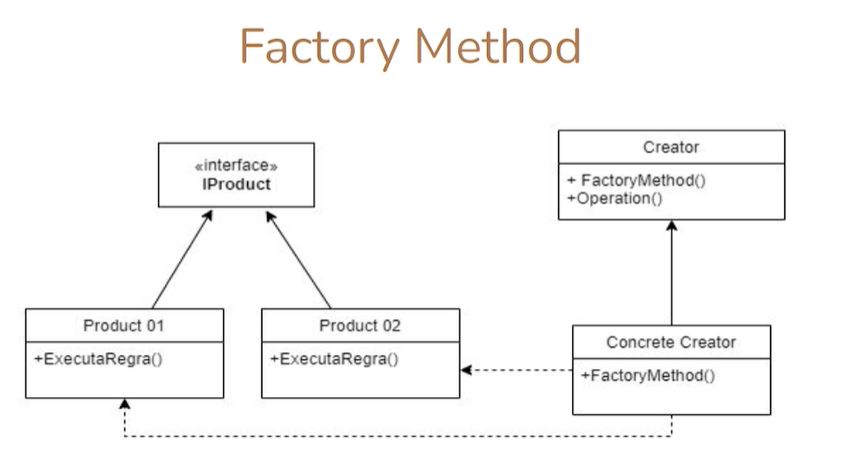
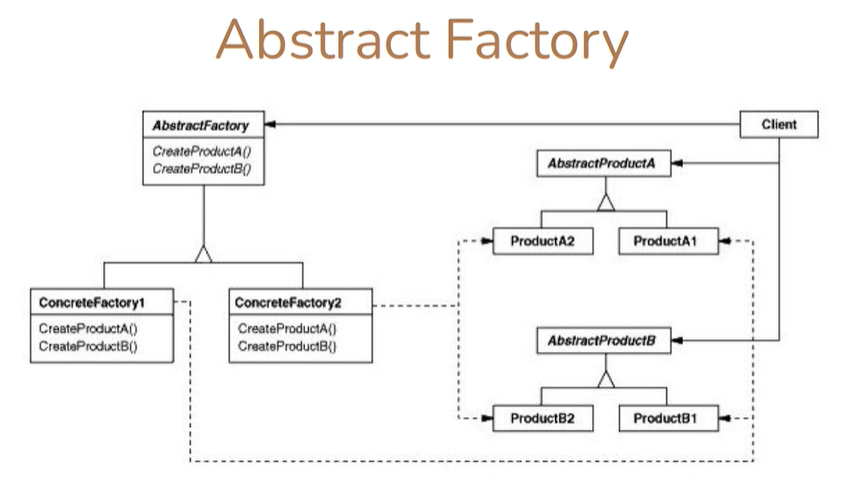
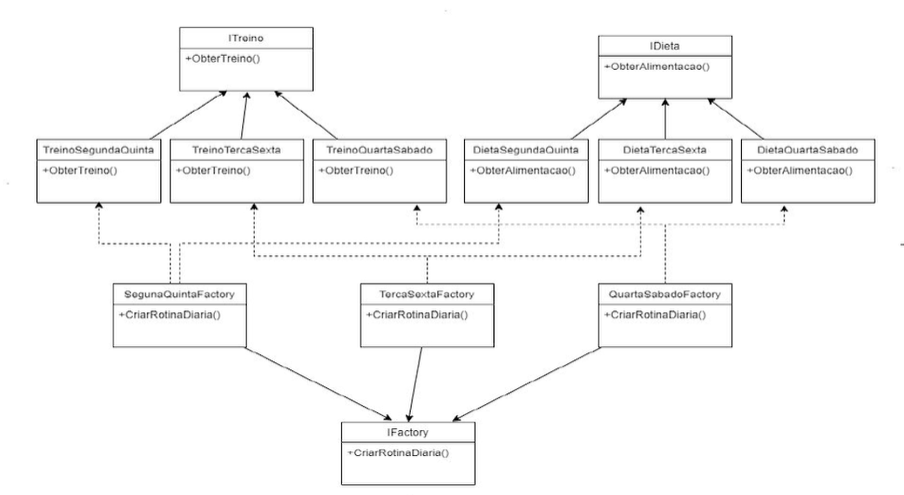
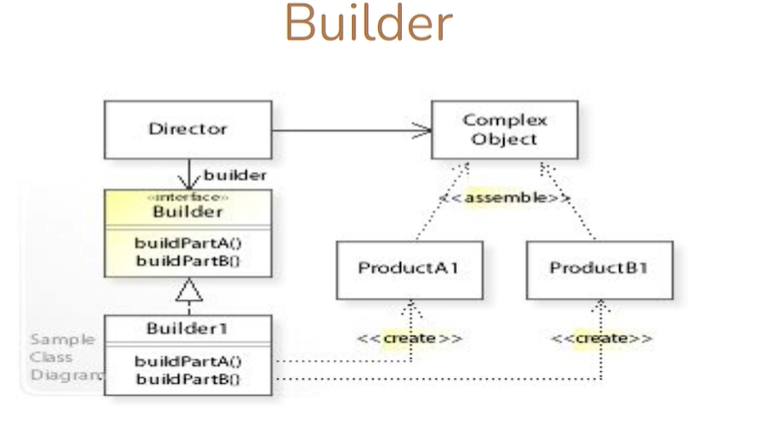
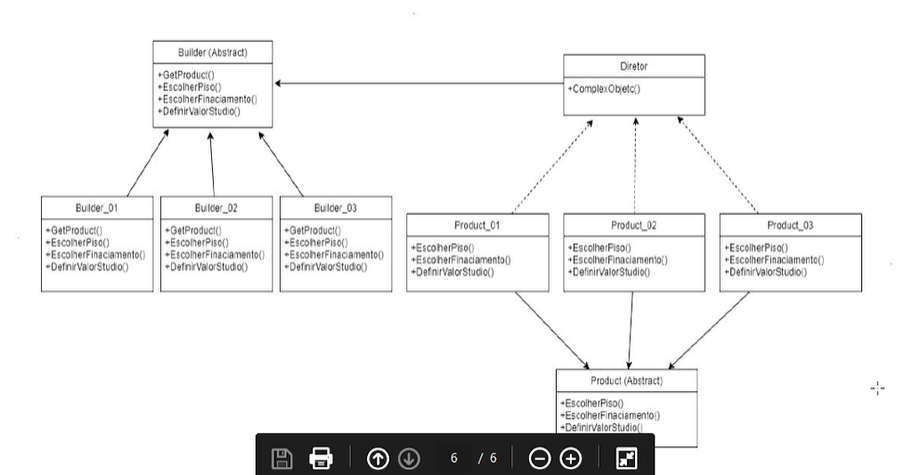
Factory Method

* Busca solucionar o problema de criação de objetos
* Define uma interface com o objetivo de criar objetos, essa interface é implementada por classes concretas, que irão criar através da fábrica ou criador, os objetos criados são chamados de products, esse padrão desacopla o código que cria o objeto do código que utiliza o objeto e torna o código flexível podendo criar novas factories sem alterar as existentes.
* A Estrutura consiste em duas partes que são:
  + Produtos: possui uma interface e as classes products que irão implementar a interface
  + Criação: Possui uma classe abstrata com um método abstrato criador e um método de operação, e uma classe concreta que implementa a classe abstrata e aplica a lógica de seleção do objeto.
* 

Abstract Factory

* É um padrão com escopo na criação de objetos, agrupa varias factorys que estão relacionadas entre si, o Abstract Factory não possui a necessidade de especificar classes concretas, pois essa função está encapsulada nas Factories
* a Estrutura
  + Parte criadora é composta por uma interface e as classes concretas (as fábricas) que implementam essa interface
  + Parte dos produtos é composta por interfaces e classes concretas (products) que implementam essas interfaces que são separadas pelos seus respectivos grupos de factories
* 
* Exemplo Real:
* 

Singleton

* É um padrão de projeto Singleton é um design pattner do tipo criacional, que visa criar um objeto (uma instancia de uma classe), e fornece um ponto global de acesso a essa instancia e não permitindo que o mesmo objeto seja instanciado novamente(instância única)
* Builder
* É um padrão que possibilita separar a construção de partes de um objeto complexo de sua representação. Esse padrão permite produzir diferentes tipos e apresentações de um objeto utilizando o mesmo código de construção
* O principal objetivo é encapsular a construção de um objeto complexo de sua representação e visa solucionar o problema na criação de objetos complexos em partes dentro de um builder. O que não torna necessário usar várias sub-classes ou uma classe único com muitos parâmetros a serem passados através de seu construtor.
* A Estrutura:
  + Builders: são responsáveis pela construção e montagem das partes das classes produtos.
  + Products são as representações dos objetos que estão sendo construídos pelos builders
  + Director: Define a ordem que é executada a construção dos products
* 
* 

Prototype