



Padrões de criação

Padrões de criação



Padrões de projeto focados em controlar a criação de objetos



Torna mais acessível, flexível e eficiente



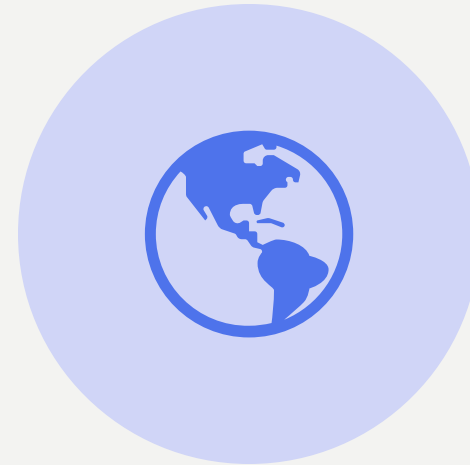
Evitam problemas como acoplamento excessivo, dependências desnecessárias e dificuldades na manutenção do código

Singleton

Singleton



A CLASSE TERÁ SOMENTE
UM OBJETO



PROVE UM PONTO DE
ACESSO GLOBAL PARA ELA

Singleton

1

Existem situações onde é necessário que a classe possua somente um objeto

2

O padrão permite que somente um objeto da classe seja instanciado

3

Não existe perigo de sobreposição desse objeto

Singleton

É utilizado quando o sistema deve ter somente uma instancia de uma classe

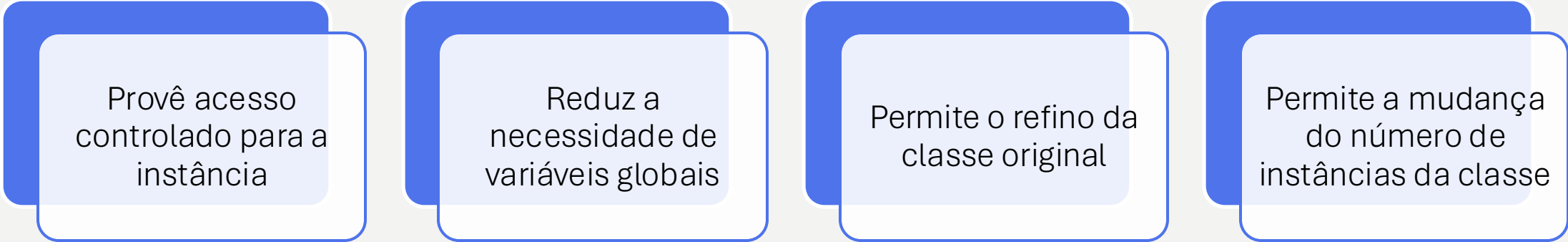
Também quando é necessário que a instancia seja extensível por herança

Singleton

ConnectionManager

- cm : ConnectionManager
- conexao : Connection
- ConnectionManager()
- + getConnection() : ConnectionManager

Singleton



Provê acesso controlado para a instância

Reduz a necessidade de variáveis globais

Permite o refino da classe original

Permite a mudança do número de instâncias da classe


```

public class ConnectionManager {

    private static ConnectionManager cm = null;
    private Connection conexao;

    private ConnectionManager() {
        try {
            Class.forName("org.mariadb.jdbc.Driver");
            Connection cn = DriverManager.getConnection(
                "jdbc:mariadb://localhost:3306/lojaDeJogos",
                "jbcnrlz",
                "a12b25c54");
            this.setConexao(cn);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static ConnectionManager getConnection() {
        if (ConnectionManager.cm == null) {
            ConnectionManager.cm = new ConnectionManager();
        }
        return ConnectionManager.cm;
    }
}

```

Factory method

Factory Method



DEFINE UMA INTERFACE PARA A
CRIAÇÃO DE OBJETOS



DEIXA AS SUBCLASSES DECIDIREM
QUAIS CLASSES INSTANCIAR

Factory Method



Existem situações onde não é possível saber qual objeto deve ser criado

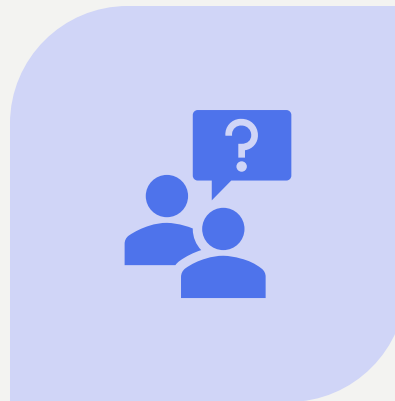


Delegar essa atuação para uma subclasse torna mais simples o processo de decisão

Factory Method



USAR QUANDO UMA CLASSE NÃO CONSEGUE
PREVER QUAL OBJETO DEVE CRIAR

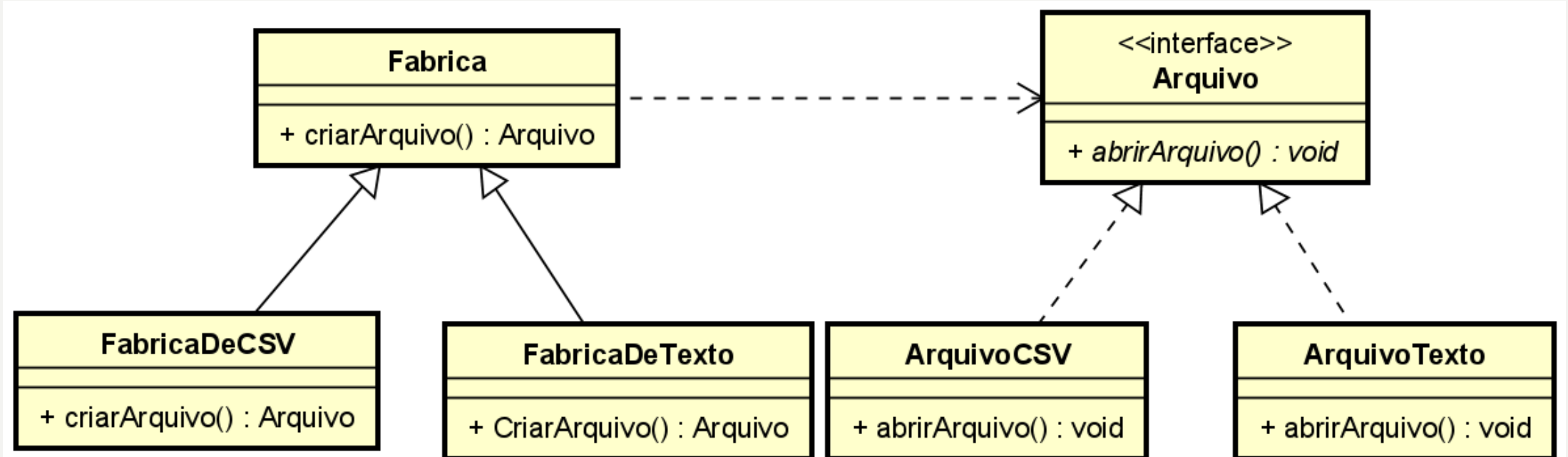


UMA CLASSE QUER QUE SUAS SUBCLASSES
ESPECIFIQUE QUAIS OBJETOS QUE DEVEM SER
CRIADOS

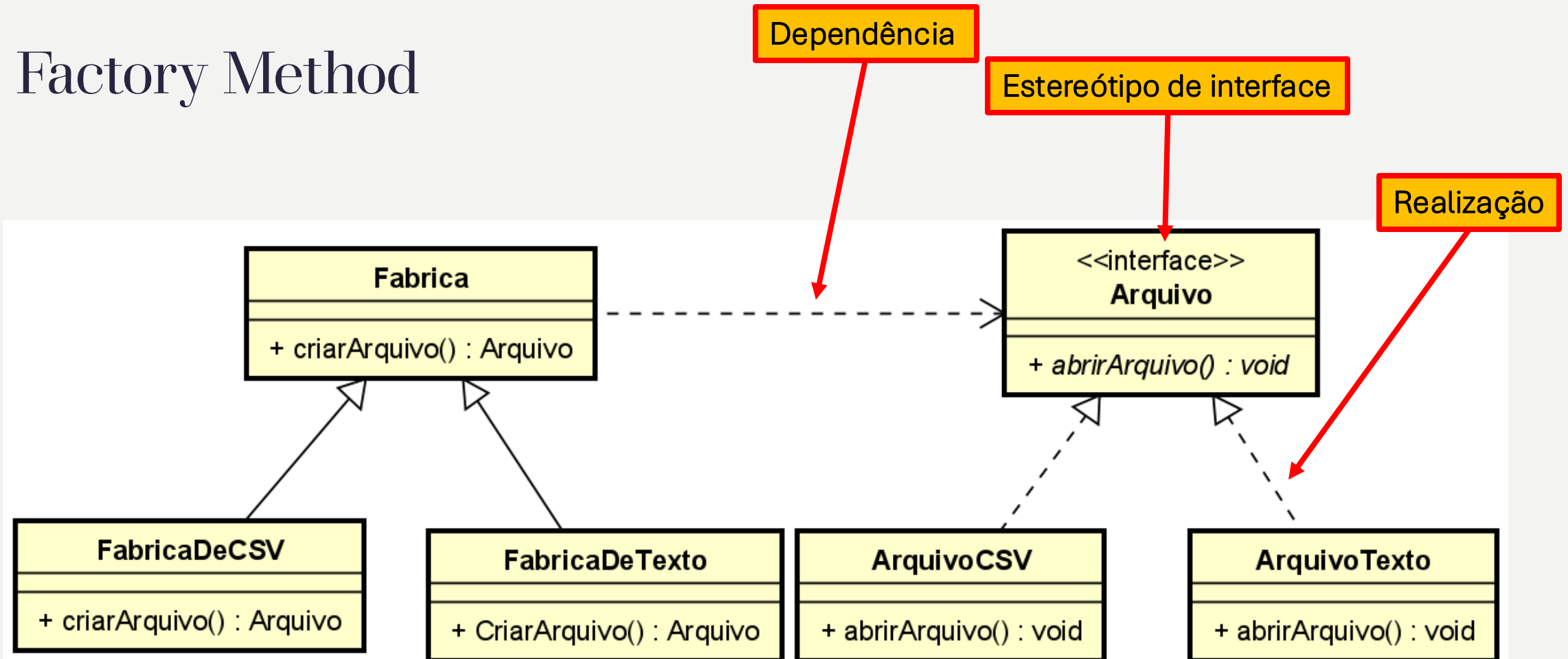


UMA CLASSE DELEGA A RESPONSABILIDADE
PARA UM CONJUNTO DE SUBCLASSES E É
NECESSÁRIO LOCALIZAR QUAL SUBCLASSE
QUE FOI DELEGADO

Factory Method



Factory Method



```

3 public interface Arquivo {
4
5     public void abrirArquivo();
6
7 }
3 public class ArquivoCSV implements Arquivo{
4
5     @Override
6     public void abrirArquivo() {
7         // TODO Auto-generated method stub
8     }
9
10 }
3 public class ArquivoTexto implements Arquivo{
4
5     @Override
6     public void abrirArquivo() {
7         // TODO Auto-generated method stub
8     }
9
10 }
11 }

```

```

3 public class Fabrica {
4
5     public Arquivo criarArquivo() {
6         return null;
7     }
8
9 }
3 public class FabricaDeCSV extends Fabrica{
4
5     @Override
6     public Arquivo criarArquivo() {
7         return new ArquivoCSV();
8     }
9
10 }
3 public class ArquivoDeTexto extends Fabrica{
4
5     @Override
6     public Arquivo criarArquivo() {
7         return new ArquivoTexto();
8     }
9
10 }

```


Abstract Factory

Abstract Factory

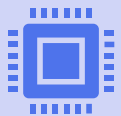
Cria uma interface
para criação de
famílias de objetos
dependentes ou
relacionados

Não é necessário
especificar as classes
concretas

Abstract Factory



Em situações onde é necessário criar objetos que tenha comportamentos próximos às vezes é necessário desacoplar sua parte concreta da implementação de sua criação



A criação de aplicações que devem mimetizar o visual de diferentes sistemas operacionais passa por isso

Abstract Factory

É necessário utilizar quando um sistema é independente de como seus produtos são criados, compostos e representados

Quando um Sistema deve ser configurado com uma de múltiplas famílias de produtos

A família de objetos deve ser utilizada conjuntamente

É necessário prover uma biblioteca de produtos e é desejado mostrar só suas interfaces e não as implementações

Abstract Factory

