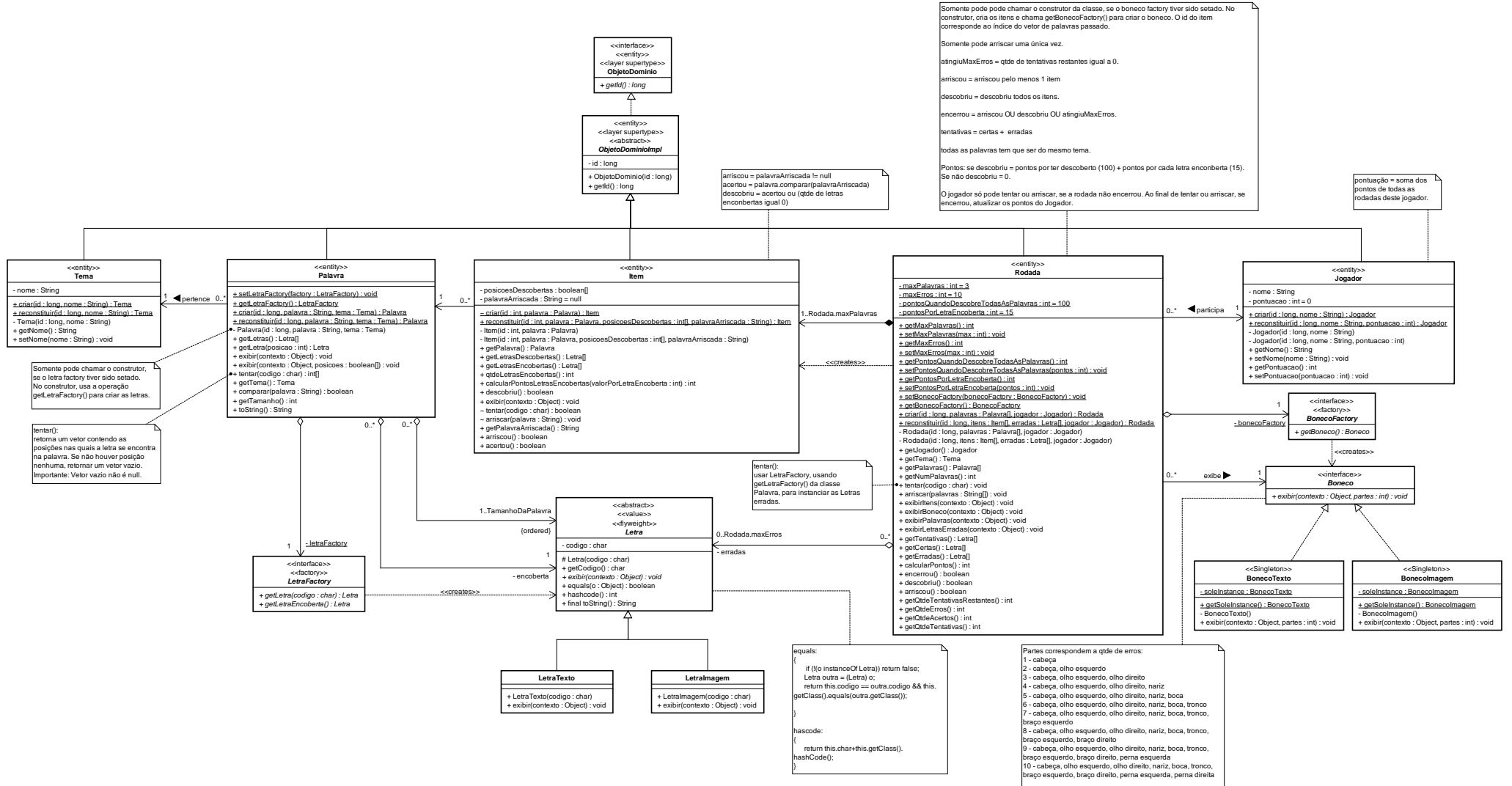
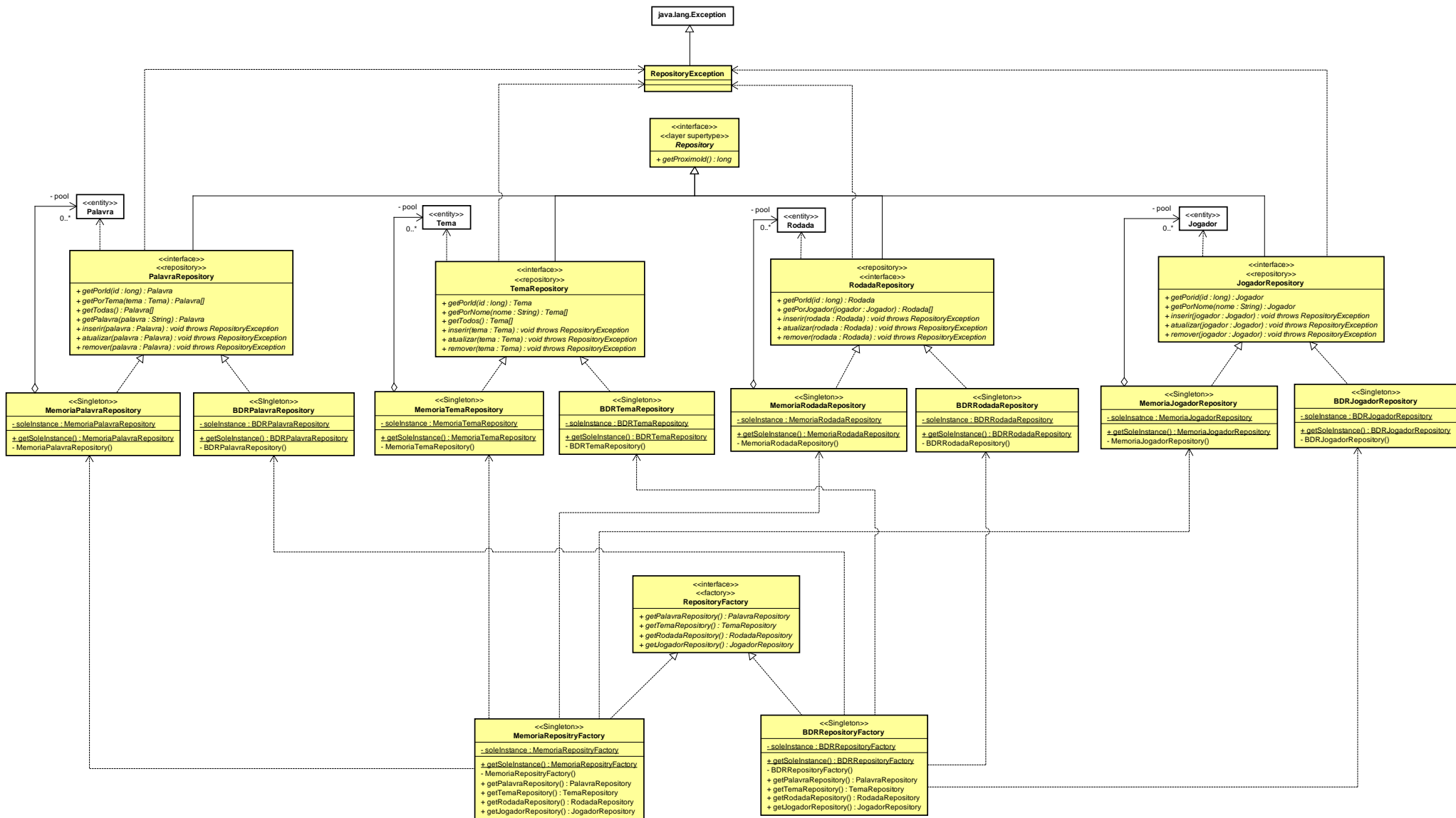


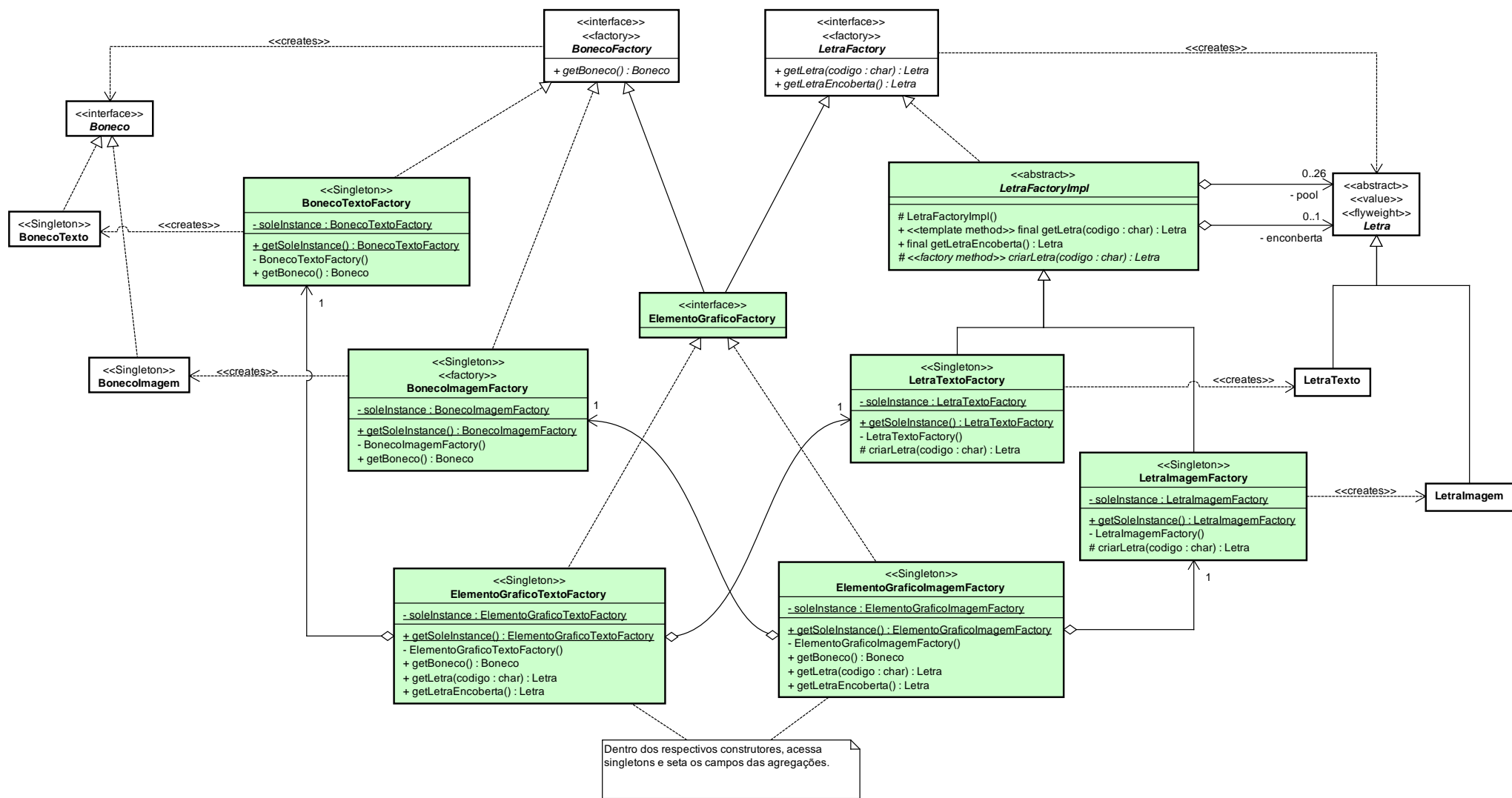
Modelo de domínio - classes de negócio (classes sem cor):



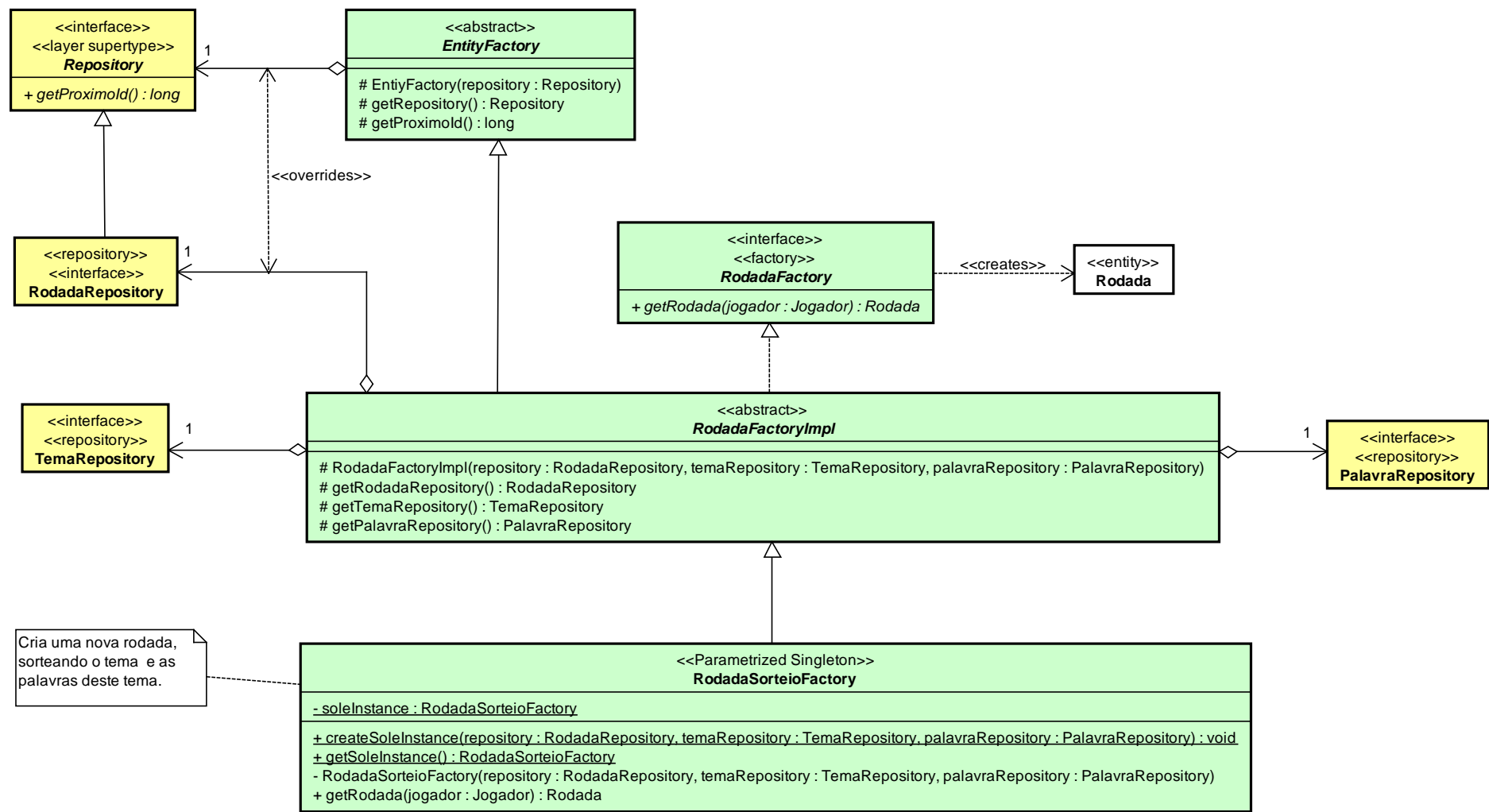
Repositórios (classe amarelas):



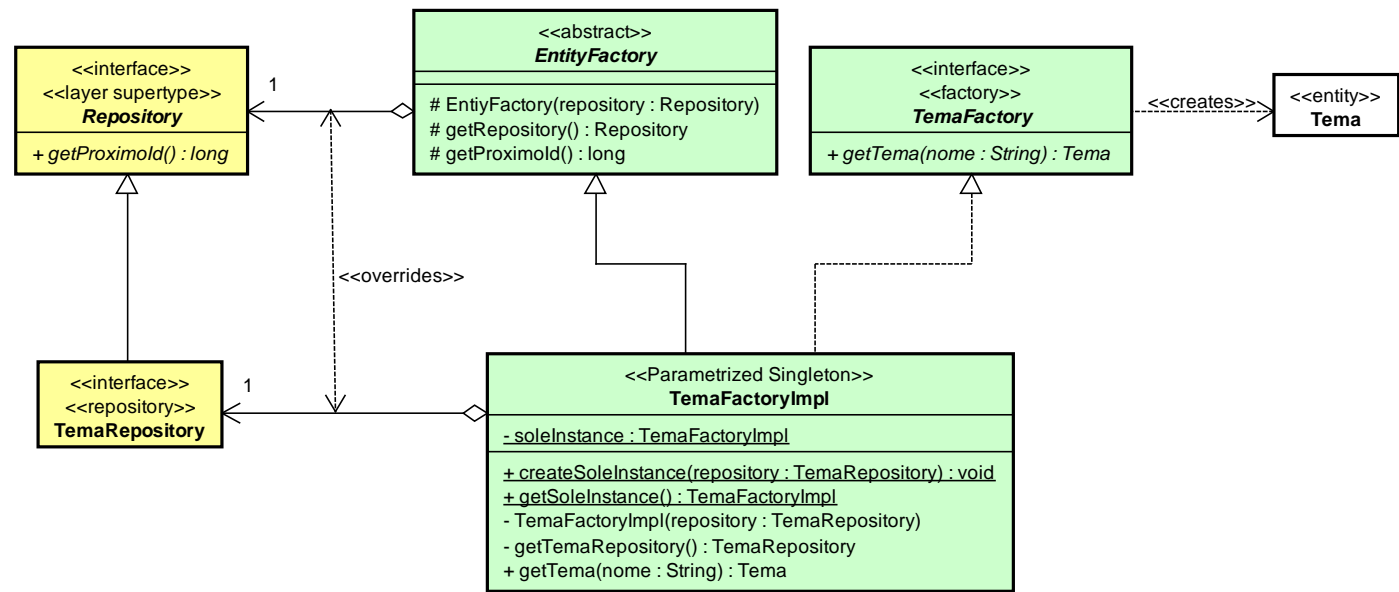
Fábricas (classes verdes):



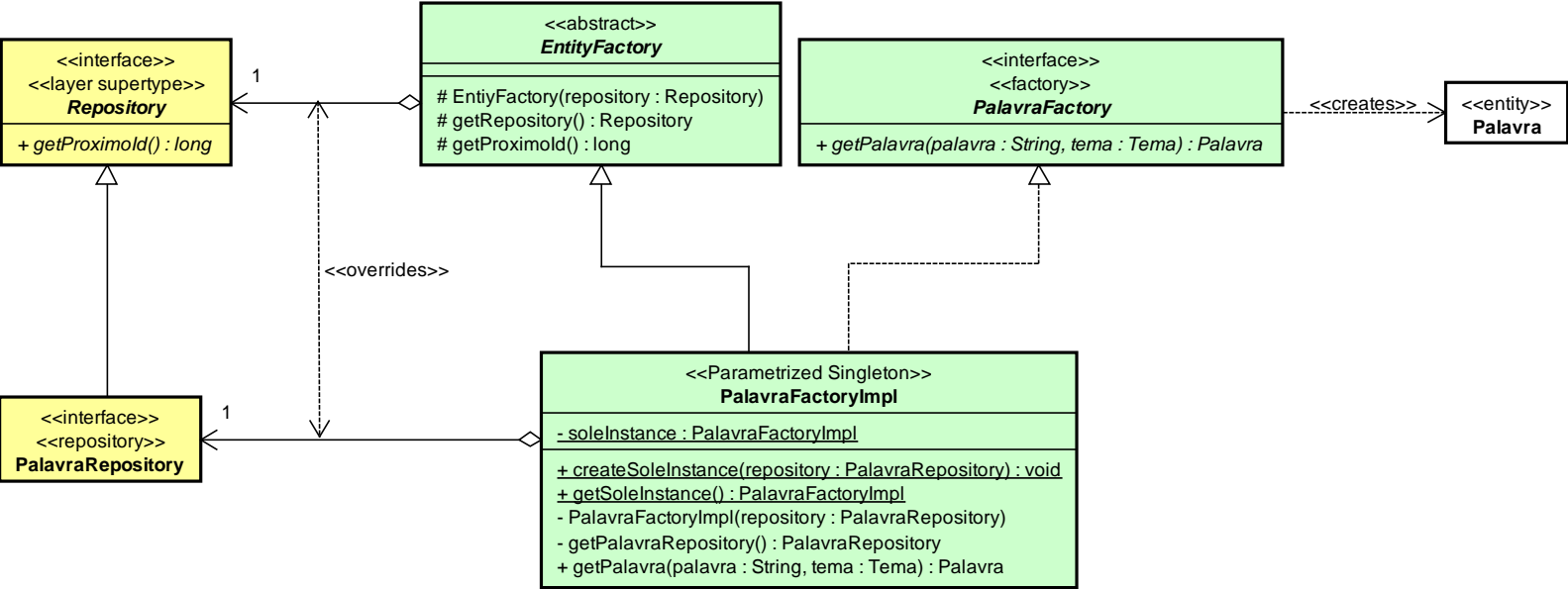
Fábricas (classes verdes):



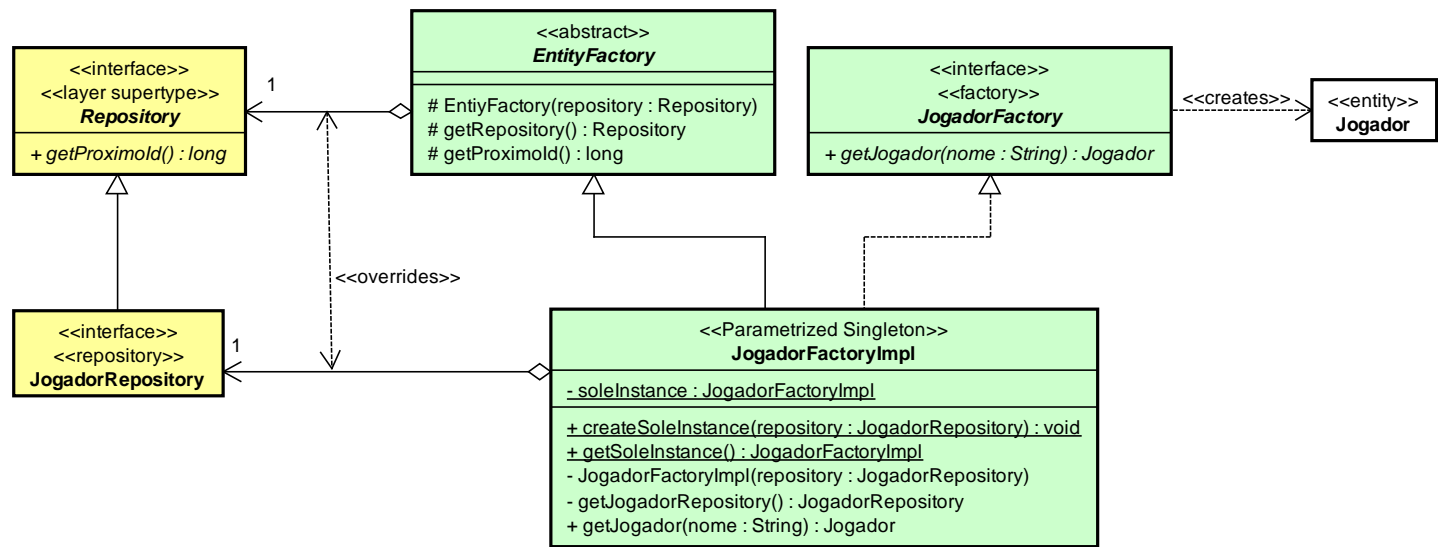
Fábricas (classes verdes):



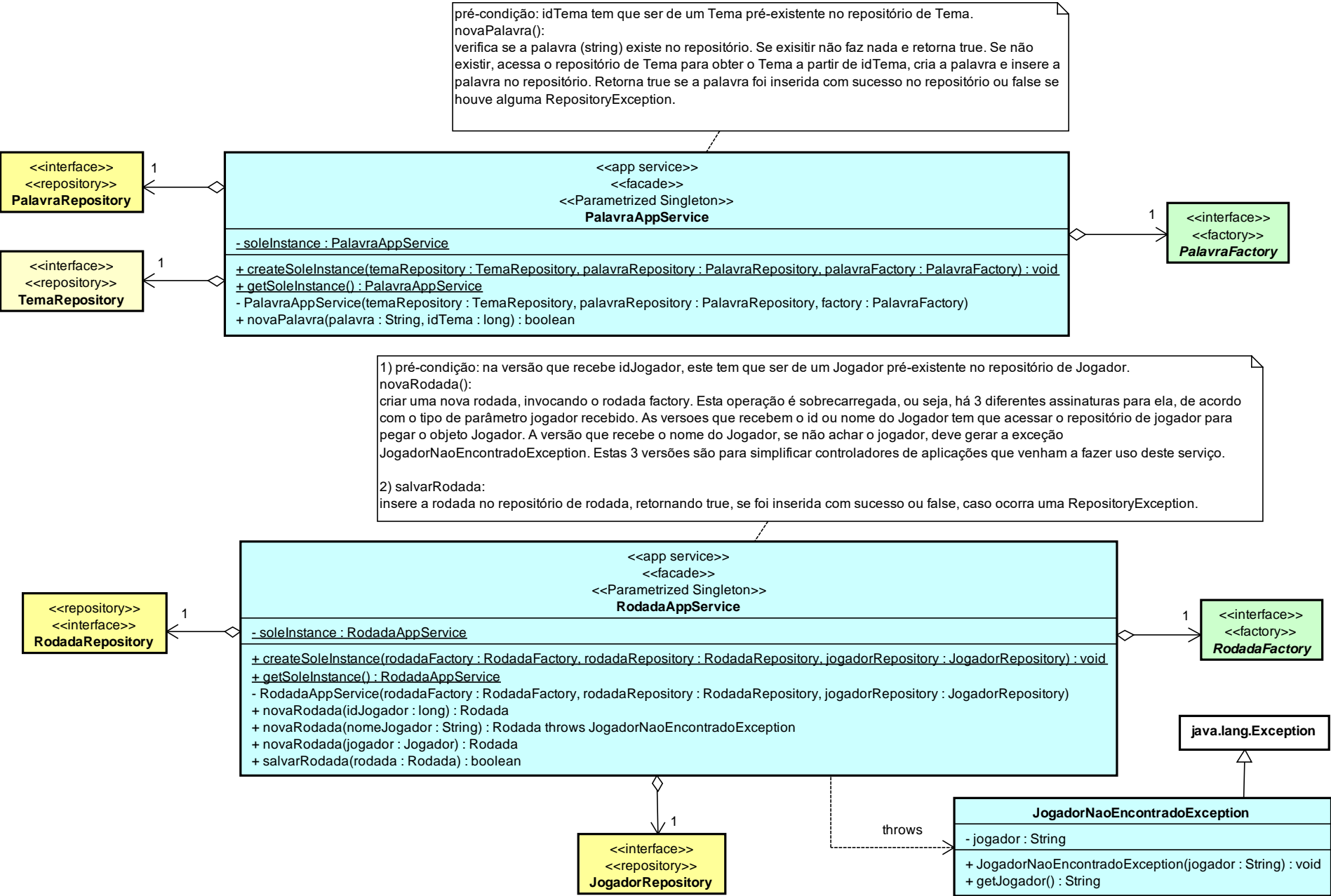
Fábricas (classes verdes):



Fábricas (classes verdes):



Serviços de aplicação - para apoiar passos de uses cases (classes azuis):



Classe de Configuração da Aplicação (classe roxa):

configurar():
1) cria os singletons parametrizados da aplicação
2) Setta os factories estáticos de classes de domínio

Todas vez que alterar parâmetros desta classe, deve chamar configurar() para reconfigurar as classes da aplicação.

As operações públicas getBonecoFactory() e getLetraFactory() ambas são implementadas chamando a operação privada getElementoGraficoFactory(), assim:

```
{  
    return this.getElementoGraficoFactory();  
}
```



Esta classe é um "Parametrized Factory". Dentro de seus métodos (getRepositoryFactory(), getElementoGraficoFactory() e getRodadaFactory()), define os Singletons concretos a serem retornados, de acordo com tipos que foram setados, nos respectivos campos. É o único lugar da aplicação que conhece os Singletons concretos. Mas o retorno das operação são interfaces (abstrações), ou seja, desta classe para fora, todo o resto da aplicação, só conhece interfaces (abstrações), respeitando o principal princípio de OO: Open-Closed Principle (OCP).

Os métodos getTemaFactory(), getPalavraFactory() e getJogadorFactory(), simplesmente, retornam o respectivo Singleton concreto.