

RESPOSTA DA QUESTÃO - PROTOTYPE

Solução com Padrão Prototype:

A estrutura básica seria:

1. Interface Prototype: `ElementoDesign` com método `clone()`
2. Protótipos Concretos: `BotaoModerno`, `CabecalhoCorporativo`, etc.
3. Registry: `BibliotecaElementos` com protótipos pré-configurados

Código simplificado:

```
java
// Interface Prototype
public interface ElementoDesign {
    ElementoDesign clone();
    void renderizar();
}

// Protótipo Concreto
public class BotaoModerno implements ElementoDesign {
    private String corFundo, corTexto, fonte;
    // ... dezenas de propriedades

    public BotaoModerno() {
        // Configuração padrão do botão moderno
        this.corFundo = "#3498db";
        this.corTexto = "#ffffff";
        this.fonte = "Arial, 14px, bold";
        // ... configura todas as propriedades
    }

    public ElementoDesign clone() {
        BotaoModerno clone = new BotaoModerno();
        // Copia todas as propriedades
        clone.corFundo = this.corFundo;
        clone.corTexto = this.corTexto;
        // ... copia as outras propriedades
        return clone;
    }
}
```

Como usar:

```
java
// Registra protótipos pré-configurados
BibliotecaElementos biblioteca = new BibliotecaElementos();
biblioteca.registrar("botao-moderno", new BotaoModerno());

// Clona quando precisar

ElementoDesign novoBotao = biblioteca.obter("botao-moderno").clone();
```

Vantagens:

- Evita repetição de código de configuração
- Usuário não precisa conhecer todas as propriedades
- Fácil criar variações (clona e modifica apenas o necessário)
- Performance melhor que criar objetos complexos do zero