Universidade Tecnológica Federal do Paraná  
Engenharia de Software - Curso de Arquitetura de Software (AS27S)

INSTRUTOR: Prof. Dr. Gustavo Santos

Felippe Negrão de Oliveira, 2372517

linha horizontal

# CCH - Design Patterns Prototype

## Problema

## Consideremos um sistema de geração de relatórios onde os relatórios podem ter várias configurações, como tipo de gráfico, intervalo de datas, filtros aplicados e estilo de formatação.

## Sem o padrão Prototype, cada vez que um usuário solicitar um novo relatório com diferentes configurações, o sistema precisará criar um novo objeto relatório e configurá-lo de acordo com as especificações do usuário.

## Descrição da Solução

A solução envolve a criação de uma classe base Prototype, que define o método clone() para criar cópias profundas dos objetos. Em seguida, são implementadas classes concretas que representam os diferentes tipos de relatórios, como SalesReport e InventoryReport, cada uma contendo configurações específicas, como tipo de gráfico, intervalo de datas, filtros e estilo de formatação.

Além disso, é criada uma classe ReportManager responsável por gerenciar os protótipos de relatórios. Essa classe registra os protótipos de relatórios disponíveis e fornece um método para obter clones desses protótipos quando necessário.

Com essa estrutura, quando um novo relatório é solicitado, o ReportManager clona o protótipo relevante, ajusta as configurações conforme necessário e retorna o relatório personalizado.

## Visão Geral

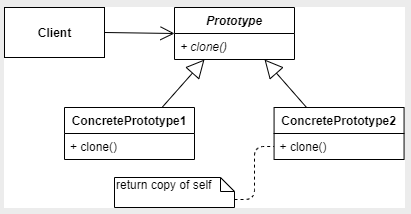


Diagrama de Classe do Prototype

## Exemplo de Código - Dart

O código pode ser encontrado no seguinte link: <https://github.com/FelippeNO/design_patterns_utfpr/blob/main/prototype_example.dart>

## Consequências

* Vantagens
  + Redução da sobrecarga de criação de objetos.
  + Flexibilidade na criação de objetos.
  + Manutenção simplificada.
* Desvantagens
  + Complexidade na gestão de protótipos.
  + Cuidados com a clonagem profunda.
  + Potencial para aumento do uso de memória.