### **Abstract Factory**

A intenção deste é fornecer uma interface para criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas. Também é conhecido como Kit.

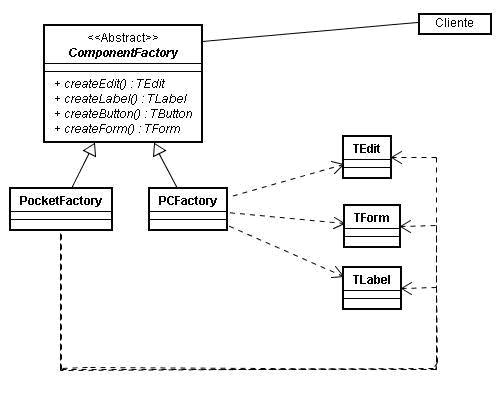
Este padrão deve ser aplicado quando se deseja isolar a aplicação da implementação da classe concreta, que poderia ser um componente e ou framework específico no qual a aplicação conheceria apenas uma interface e a implementação concreta seria conhecida apenas em tempo de execução ou compilação.

Imagine que em uma aplicação houvesse a necessidade de que ela fosse implementada para oferecer suporte a plataformas e características distintas. Por exemplo: Uma visão desktop e uma móvel (celular Pocket PC). A maneira de constituí-la, seria definindo uma família de componentes para cada plataforma e uma fábrica que os instancia de acordo com a plataforma alvo na qual a aplicação será executada.

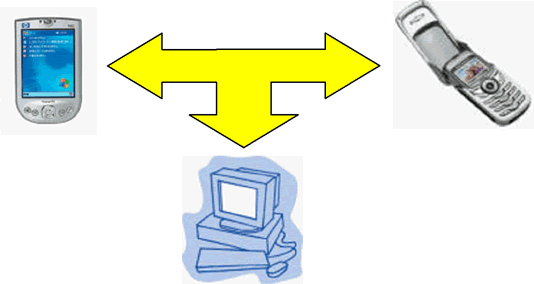
De acordo com o exposto pelos quatro amigos, o uso do padrão Abstract Factory deve estar restrito as seguintes situações:

* Um sistema deve ser independente de como seus produtos são criados, compostos ou representados;
* Um sistema deve ser configurado como um produto de uma família de múltiplos produtos;
* Uma família de objetos for projetada para ser usada em conjunto, e você necessita garantir esta restrição;
* Você quer fornecer uma biblioteca de classes de produtos e quer revelar somente suas interfaces, não suas implementações.

A estrutura arquitetural do padrão definido segundo GoF é de acordo com o apresentado na **Figura 1**.

**Figura 1**. Estrutura arquitetural do padrão

A estrutura de um exemplo mais de acordo com a realidade do desenvolvedor é apresentada na **Figura 2**. A ideia básica apresentada pela figura é a de oferecer ao usuário (desenvolvedor) a possibilidade de executar uma aplicação sobre diferentes plataformas.

**Figura 2**. Diversas plataformas

Os participantes são:

* ComponentFactory-declara uma interface para operações que criam objetos dos componentes utilizados na aplicação;
* PocketFactory– classe concreta que implementa as operações que criam os objetos no formato do dispositivo cliente;
* PCFactory -classe concreta que implementa as operações responsáveis por criar os objetos no formato do PC.

O padrão Abstract Factory possui os seguintes benefícios e desvantagens:

* Ele isola as classes concretas.
* Ele torna fácil a troca de famílias de produtos.
* Ela promove a harmonia entre produtos.
* É difícil de suportar novos tipos de produtos.