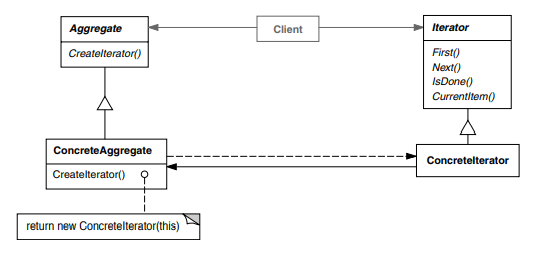
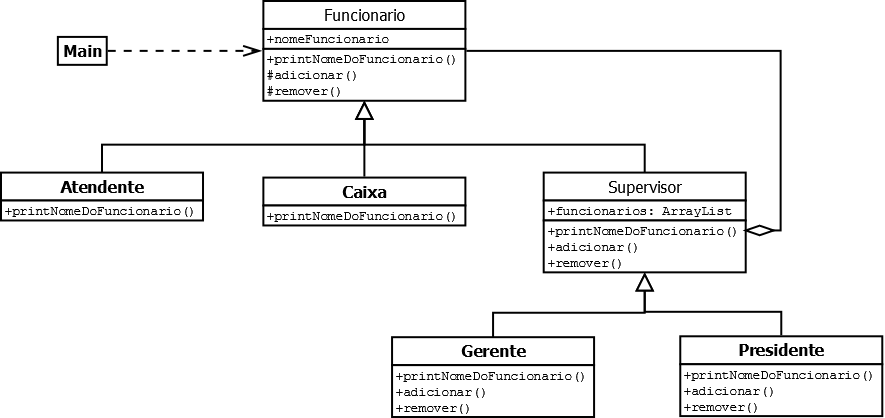
* Padrão **Iterator**

O Padrão Iterator é definido como: “O Padrão Iterator fornece uma maneira de acessar sequencialmente os elementos de um objeto agregado sem expor a sua representação subjacente”. Portanto, temos que o padrão Iterator permite acessarmos um a um os elementos de um agregado mesmo sem saber como eles estão sendo representados, assim torna-se irrelevante se a coleção de objetos está num ArrayList, HashTable ou que quer que seja. Além disso, o Padrão Iterator assume a responsabilidade de acessar sequencialmente os elementos e transfere essa tarefa para o objeto Iterador, dessa forma o objeto agregador tem a sua interface e implementação simplificadas, não sendo mais o responsável pela iteração.



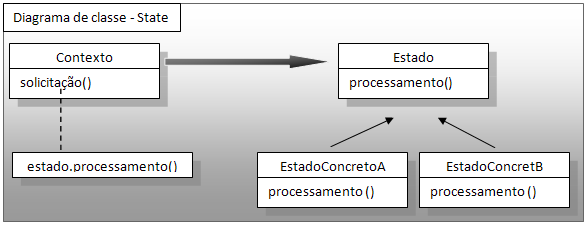
* Padrão **Composite**

O **Composite** é um padrão de projeto estrutural que permite compor objetos em uma estrutura semelhante a uma árvore e trabalhar com eles como se fosse um objeto singular.



* Padrão de Projeto **State**

O padrão state permite que um objeto altere o seu comportamento quando o seu estado interno muda. O objeto parecerá ter mudado de classe. O padrão encapsula os estados em classes separadas e delega as tarefas para o objeto que representa o estado atual, nós sabemos que os comportamentos mudam juntamente com o estado interno.



* Padrão **Proxy**

Proxy é um padrão de design estrutural que permite fornecer um substituto ou espaço reservado para outro objeto. Um proxy controla o acesso ao objeto original, permitindo que você execute algo antes ou depois que a solicitação chega ao objeto original.

