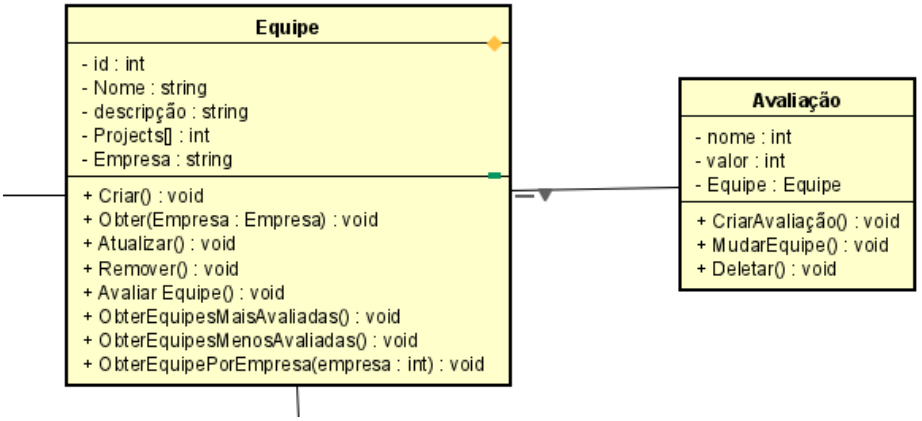
Princípios SOLID no Projeto

Princípios Violados

1 - Princípio Violado: Princípio de responsabilidade única

Descrição: Na tabela equipe (Team) é possível ver uma propriedade que se chama rating(avaliação), isso quebra o princípio de responsabilidade única porque existe mais de um motivo para mudá-la, porque o formato da equipe pode ser mudada assim como sua avaliação.

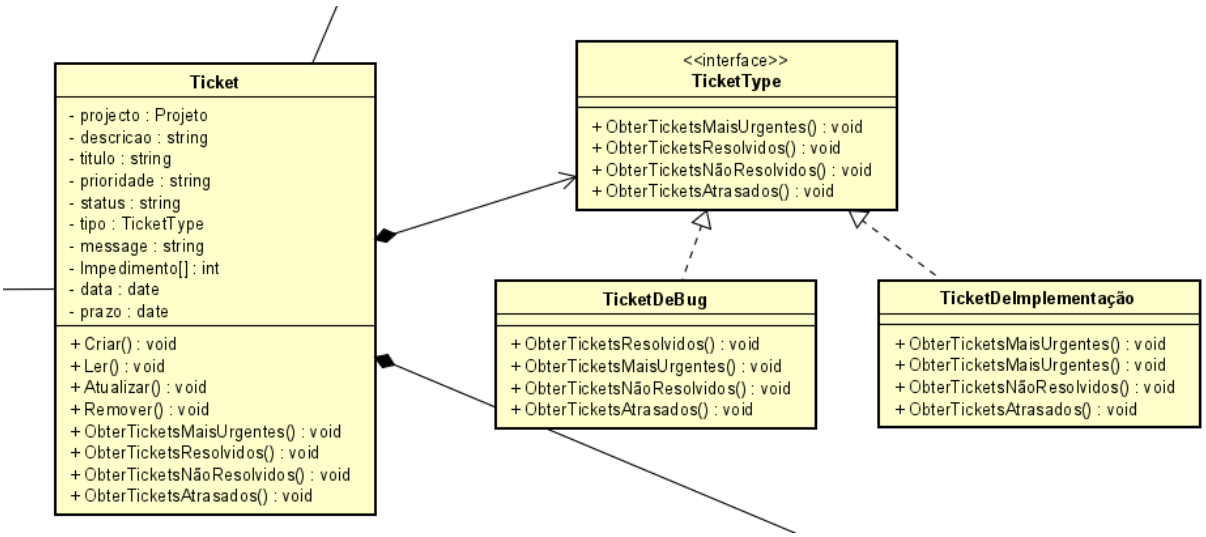
Refatoração:



2 e 3 - Princípios Violados: Princípio aberto-fechado e princípio da inversão de dependência.

Descrição: Na tabela Ticket é possível encontrar propriedades como tickets mais urgentes, tickets não resolvidos, tickets resolvidos, ou seja, há necessidade de mudança sempre que há um novo método de se obter tickets. Com a refatoração, a propriedade "tipo" passa a implementar uma interface TicketType, fazendo com que a entidade dependa de uma abstração. Além disso, a interface passa a ser extensível e flexível pois na refatoração passamos a ter dois novos tipos de Tickets, TicketDeBug e TicketDeImplementação.

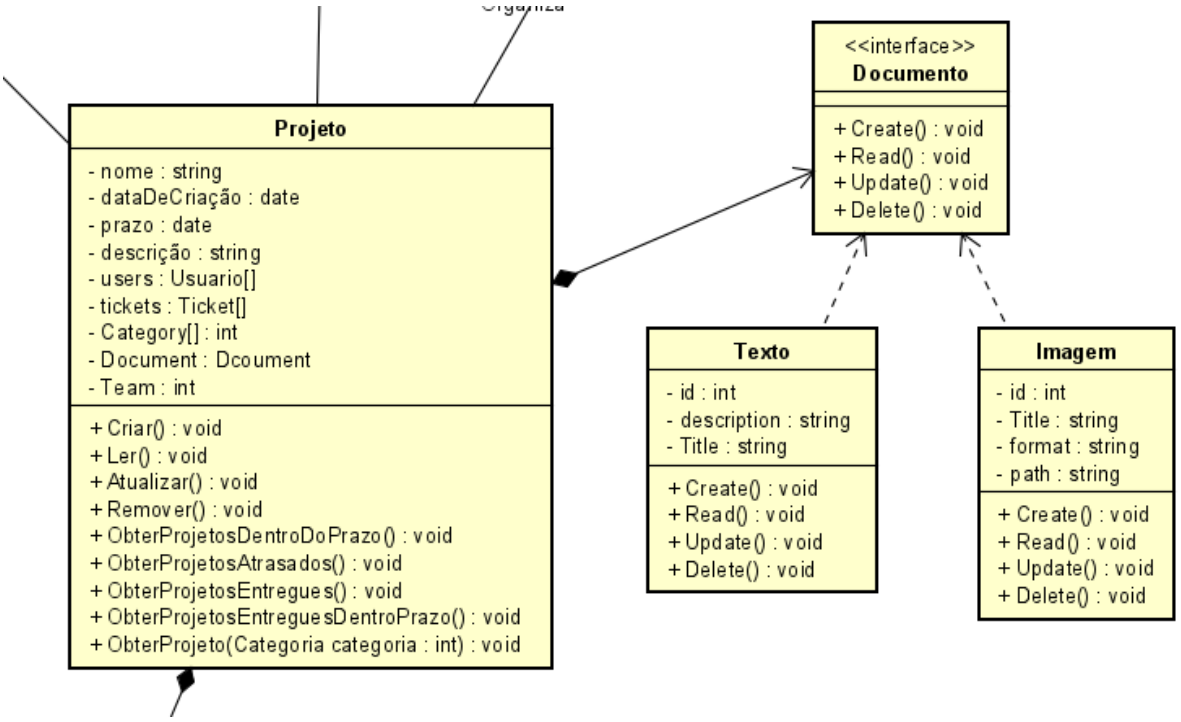
Refatoração:



4 e 5 - Princípios Violados: Princípio da inversão de dependência e princípio aberto-fechado.

Descrição: Na classe projeto é possível verificar uma implementação concreta de documento, não utilizando-se de uma interface para descrever as implementações de leitura e gravação dos tipos de documentos. Após a refatoração, a classe projeto (de alto nível) não dependerá da classe Documento (de baixo nível), mas sim de abstrações. Além disso, com a refatoração não será necessária mais nenhuma outra mudança sempre que um novo formato de documento for adicionado.

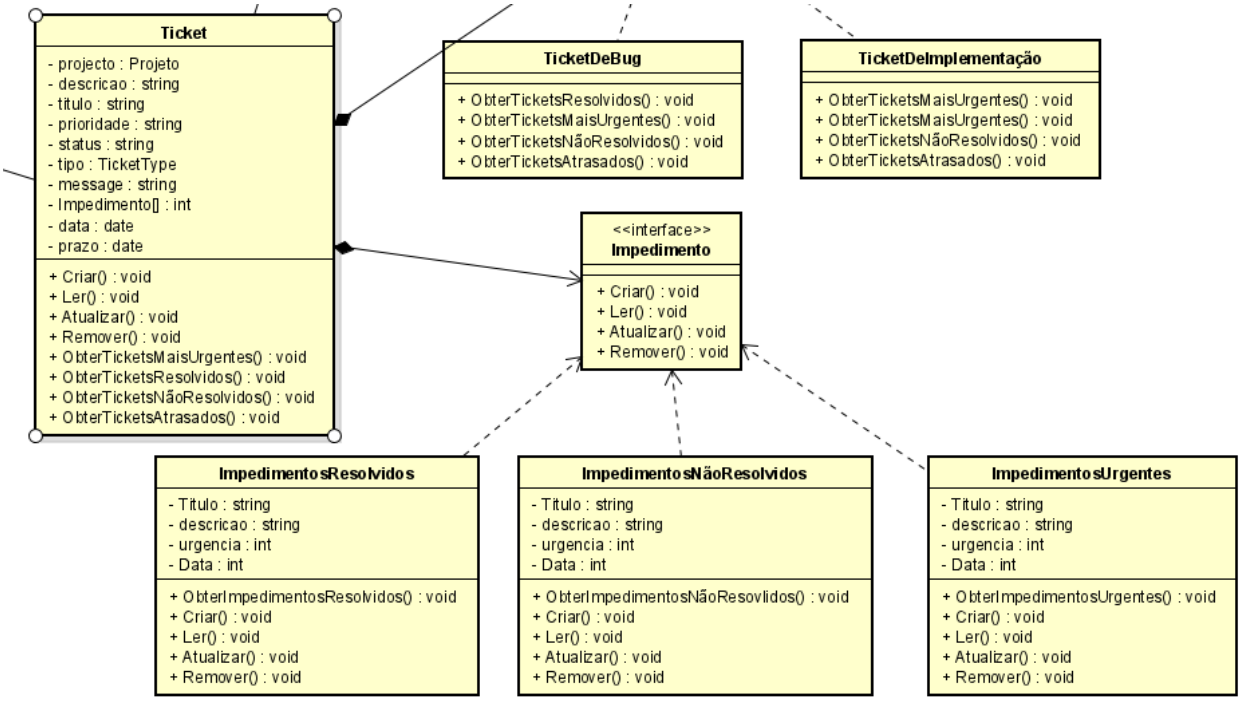
Refatoração:



6 e 7 - Princípios Violados: Princípio da inversão de dependência e princípio aberto-fechado.

Descrição: Na classe ticket é possível verificar uma implementação concreta de impedimento, não utilizando-se de uma interface para descrever as implementações dos impedimentos resolvidos, não resolvidos e urgentes. Após a refatoração, a classe Ticket (de alto nível) não dependerá da classe Impedimento (de baixo nível), mas sim de abstrações. Além disso, com a refatoração não será necessária mais nenhuma outra mudança sempre que um novo formato de impedimento for adicionado.

Refatoração:



Alguns dos princípios seguidos:

Princípio seguido: Princípio da Segregação de Interfaces

Descrição: Foi criado interfaces específicas (Colaborador e Manager) que estendem a interface genérica (User). Pois a interface Colaborador possui métodos que não são utilizados pela interface Manager e vice-versa.

Princípio seguido: Princípio Prefira Composição a Herança

Descrição: A maior parte dos relacionamentos entre as classes é de composição.