

- Usar métodos de ação
- Padrões → Escalabilidade
- Não "instanciar classe base"
- **abstract**: não pode sofrer instanciação } protege Base Class
  - ↳ precisa ter herança
- Métodos abstract não têm implementação
  - ↳ obrigam cada uma das suas derivadas classes tenham a sua própria implementação
- Classe abstracta é um modelo
- Criar propriedades e métodos abstract informam os herdeiros porque forçamos a criar implementações
  - ↳ usar show potential fixes

### Interfaces

- Como se fosse uma classe abstracta com tudo abstracto
  - ↳ Não tem implementação
    - ↳ As classes derivadas têm de implementar
- Não pode ter métodos static
  - ↳ Há formas de contornar
- Add / New item / Interface
  - ↳ Nome: começa por "I"
  - ↳ ICourse.cs
- Têm Propriedades e Métodos
  - ↳ Não têm Construtores → não fazem implementação
- Não pode haver multi-herança de classes (mais do que 1 classe base) mas há multi-herança de interfaces
- Interfaces têm de ser public (os seus membros)
- Suporta herança de 1 classe e vários Interfaces