

# AULA 13 – PADRÃO FACADE (FACHADA)

GS1020 – Programação Orientada a Objetos II

Prof. Dr. Murillo G. Carneiro  
*mgcarneiro@ufu.br*

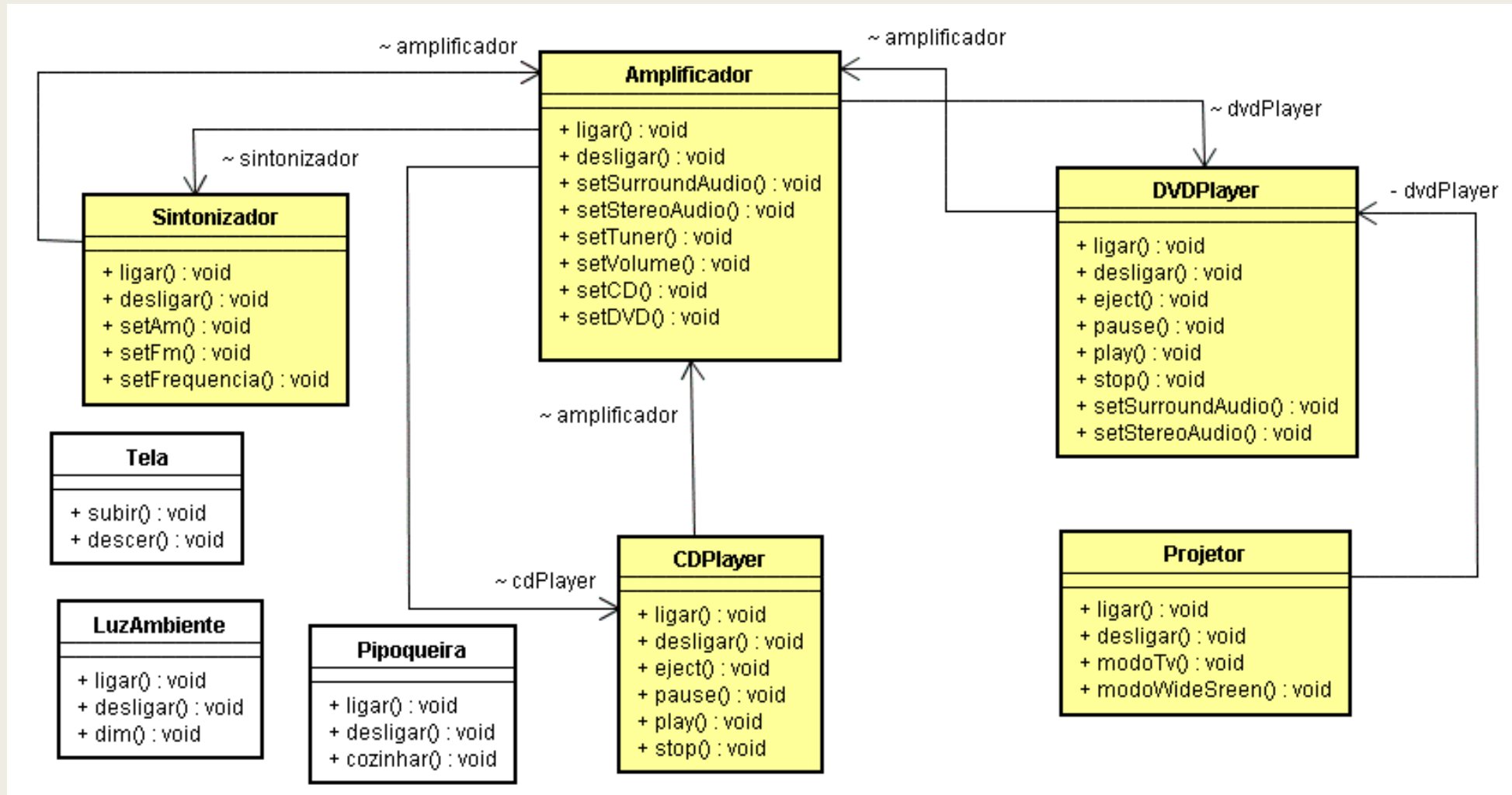


*Material baseado nos slides cedidos pelo Prof. Rafael D. Araújo (FACOM/UFU)*

# Objetivo da aula

- Entender o funcionamento do padrão de projeto *Facade (Fachada)*.

# Imagine um Home Theater



# Imagine um Home Theater

- Muitas classes
- Muitas interações
- Você precisa saber o funcionamento de tudo para saber usar corretamente

# Para assistir um filme...

- Ligar a pipoqueira
- Preparar a pipoca
- Reduzir as luzes
- Baixar a tela
- Ligar o projetor
- Configurar a entrada do projeto para DVD
- Colocar o projetor em modo wide-screen
- Ligar o amplificador de áudio
- Configurar a entrada do amplificador para DVD
- Configurar o amplificador para surround
- Ajustar o amplificador para o volume médio (5)
- Ligar o DVD Player
- Acionar o play no DVD Player

# Para assistir um filme... ... em Java

- Ligar a pipoqueira
- Preparar a pipoca
- Reduzir as luzes
- Baixar a tela
- Ligar o projetor
- Configurar a entrada do projeto para DVD
- Colocar o projetor em modo wide-screen
- Ligar o amplificador de áudio
- Configurar a entrada do amplificador para DVD
- Configurar o amplificador para surround
- Ajustar o amplificador para o volume médio (5)
- Ligar o DVD Player
- Acionar o play no DVD Player

- `pipoqueira.ligar();`
- `pipoqueira.preparar();`
- `luz.dim(10);`
- `tela.descer();`
- `projetor.ligar();`
- `projetor.setInput(dvdPlayer);`
- `projetor.modWideScreen();`
- `amplificador.ligar();`
- `amplificador.setDVD(dvdPlayer);`
- `amplificador.setSurroundAudio();`
- `amplificador.setVolume(5);`
- `dvdPlayer.ligar();`
- `dvdPlayer.play(filme);`

# Problemas

- Neste tipo de situação ainda há uma ordem para fazer tudo
- Ao final pode-se precisar fazer as ações em ordem inversa
- Se precisar atualizar o sistema terá que aprender um novo algoritmo
- E para ouvir CD?

# Fachada

- É um padrão de projeto de propósito **estrutural** com escopo de **objetos**
- Propõe uma solução para ocultar a complexidade de um conjunto de classes por meio de uma interface simples
  - *Isola as classes internas (partes do sistema) das externas*



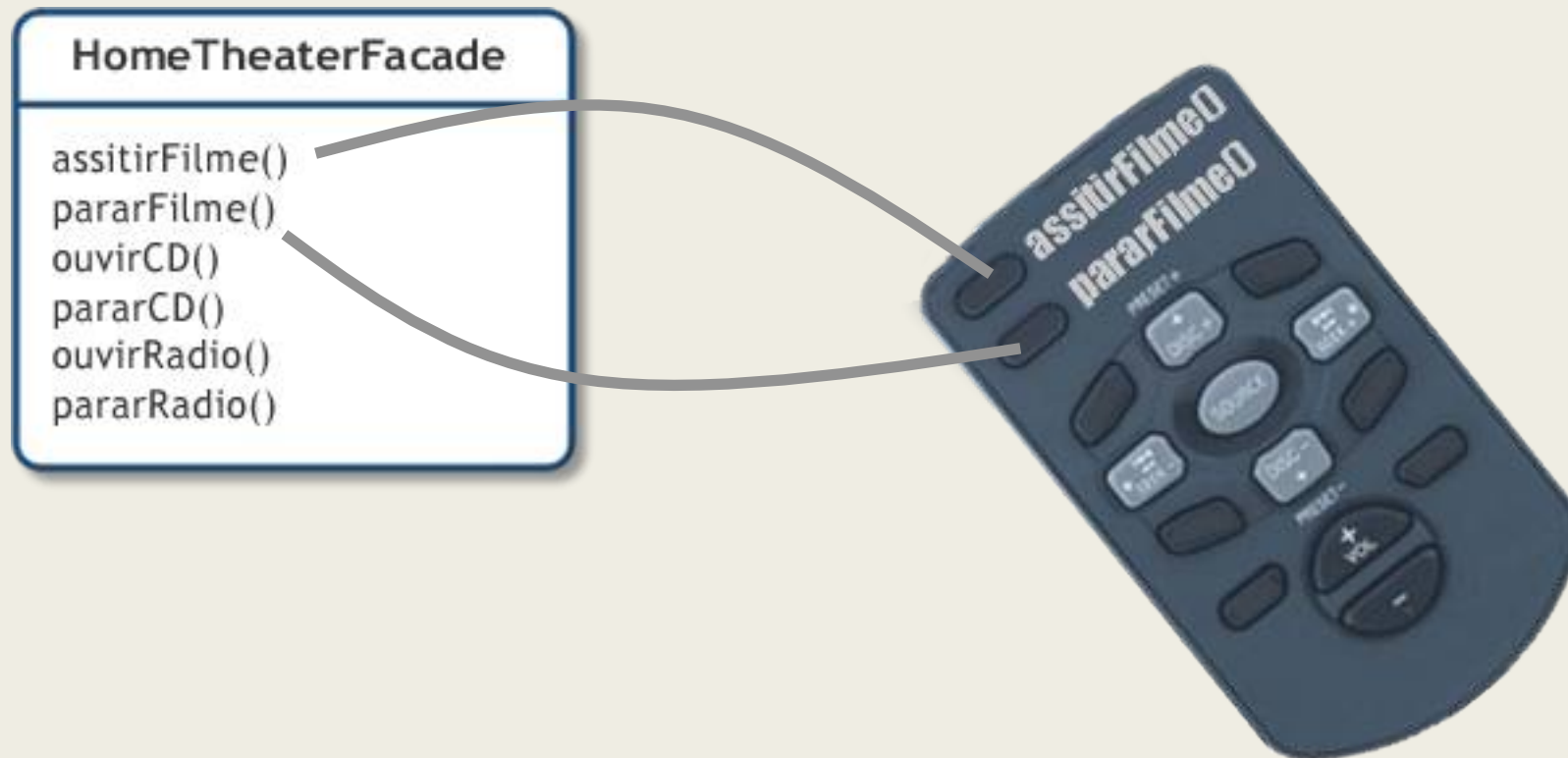
# Intenção

- Fornecer uma interface unificada para um conjunto de interfaces em um subsistema.
- Definir uma interface de nível mais alto para tornar o subsistema mais fácil de usar.

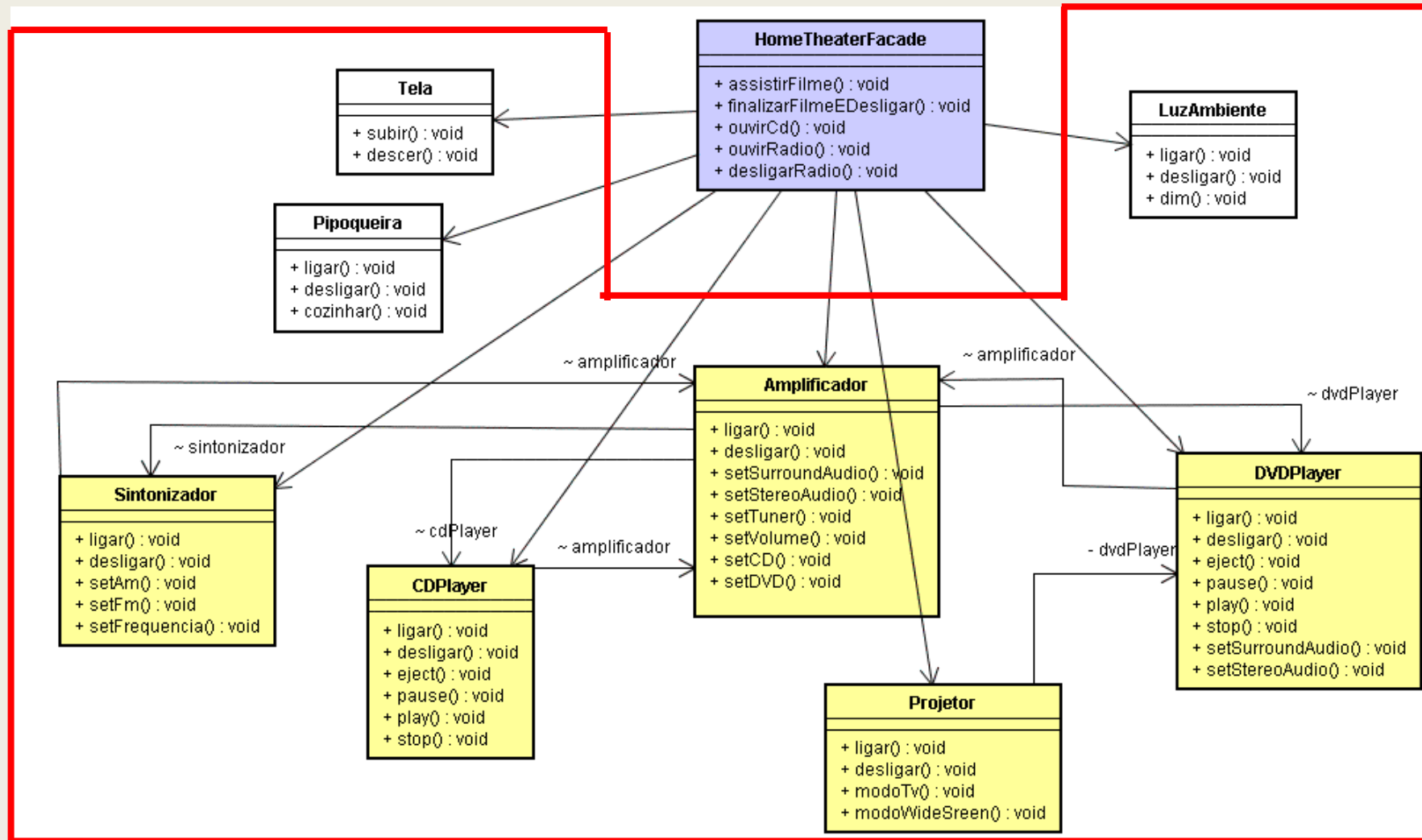
# Solução para o HomeTheater

- Uma classe que expõe somente alguns métodos simples
  - *assistirFilme()*
  - *pararFilme()*
- Trata os componentes do Home Theater como um subsistema
- Aciona as diversas partes necessárias para implementar os métodos *assistirFilme()* e *pararFilme()*

# Solução para a complexidade



# Solução para a complexidade



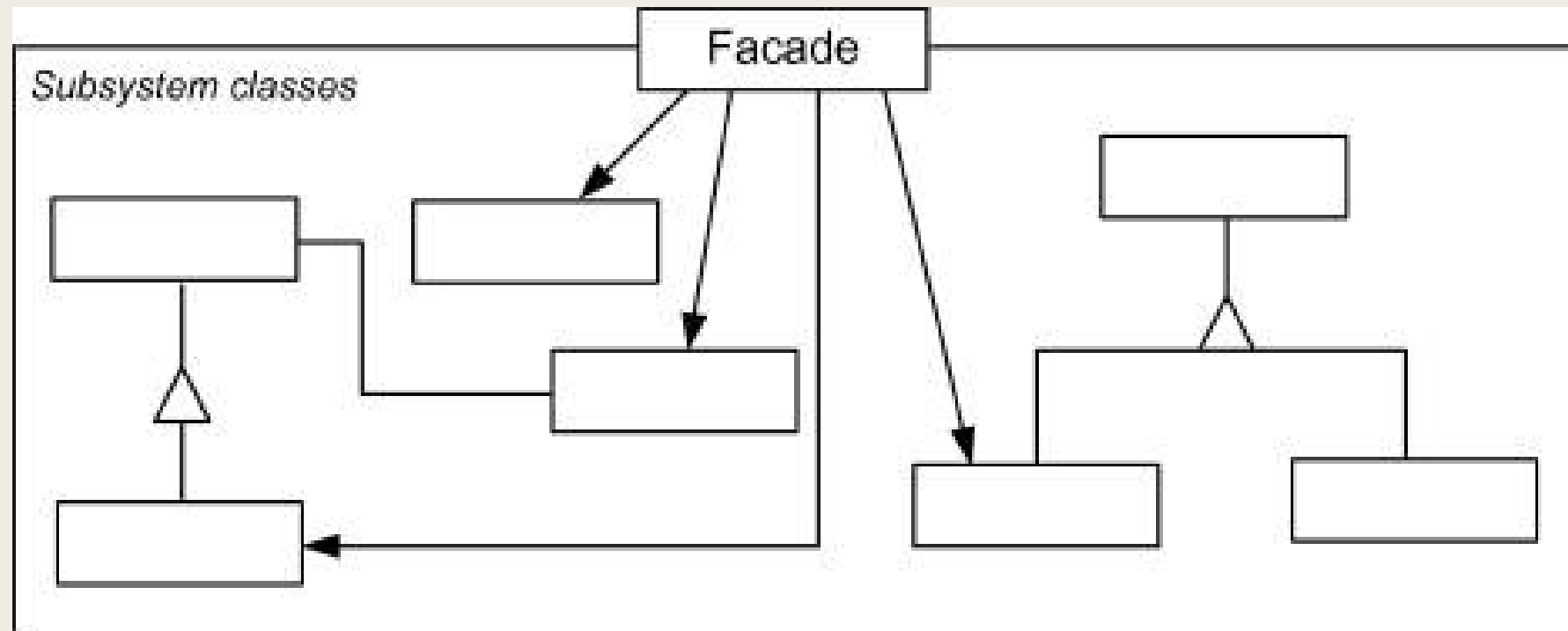
# Quando usar

- Quando deseja-se fornecer uma interface simples para um subsistema complexo
- Quando existirem muitas dependências entre clientes e classes de implementação de uma abstração
- Quando deseja-se estruturar seus subsistemas em camadas
  - *A fachada seria o ponto de entrada de cada nível*

# Elementos participantes

- **Facade (Fachada)** é uma classe que delega solicitações de clientes a objetos apropriados de um subsistema
  - *Por isso, conhece quais as classes do subsistema que são responsáveis pelo atendimento de uma solicitação*
- **Classes de subsistema** é um conjunto de classes que implementam a funcionalidade do subsistema e não tem conhecimento da Fachada

# Estrutura



# Benefícios

- Isola os clientes dos componentes do sistema, o que reduz o número de objetos com que os clientes têm que lidar
- Promove o acoplamento fraco entre o subsistema e seus clientes
- Pode simplificar a migração de sistemas para diferentes plataformas
- Ainda permite que os clientes possam definir suas operações de forma customizada



# Desvantagens

- Não impede que os clientes ainda utilizem as classes internas dos subsistemas, o que pode causar problemas arquiteturais
- A classe fachada pode crescer descontroladamente
- A classe fachada deve ser completa e sem defeitos

# Fachada vs Adaptador

- Uma Fachada simplifica uma interface enquanto um Adaptador converte uma nova interface em uma interface pré-existente
  - *Simplicidade vs Interoperabilidade*

# Problema prático

- Implemente o cenário do Home Theater.

# Referências

- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1995. Capítulo 4.