MECANISMOS DE VARIABILIDADE

Prof. Dra. Rosana Teresinha Vaccare Braga

Disciplina de Reúso de Software.

ALGUNS MECANISMOS:

- **Bibliotecas Estáticas e Dinâmicas**:
  - **Estáticas:** contêm um conjunto de funções externas que podem ser ligadas à aplicação depois da compilação;
    - A forma como as funções são chamadas é conhecida pelo código compilado e, portanto, não precisa ser alterada;
    - A implementação pode mudar de acordo com a seleção de diferentes bibliotecas e isso gera a variabilidade.
  - **Dinâmicas**: são carregadas em aplicações em tempo de execução.

- **Compilação Condicional**: utiliza diretivas que habilitam o controle sobre partes do código que deverão ser incluídos ou excluídos durante a compilação do programa.

- **Configuração**: usada para selecionar componentes do repositório e conectá-los para incluí-los no produto.

ALGUNS MECANISMOS (CONT.):

- **Encapsulamento/Informação Escondida**: é um conceito básico de OO;
  - A informação que pode ser alterada está escondida dentro do objeto;
  - Várias versões do mesmo componente podem ser construídas com a mesma interface;
  - A variabilidade está contida dentro de cada versão do componente/objeto.

- **Extensão e Pontos de Extensão**: usado quanto partes de um componente pode ser estendido com comportamento adicional, selecionado de um conjunto de possíveis variantes.
  - Para o padrão de projeto Strategy.

- **Geração**: um sistema denominado gerador traduz uma especificação, escrita em linguagem específica, para um componente em nível de código fonte que fará parte do produto ou aplicação.

- **Herança**: uma subclasse é derivada de uma superclasse.
  - Uma classe derivada é especializada para um contexto específico;
  - Herança é usada para modelar diferentes subclasses de variantes de uma classe abstrata;
  - Número de variantes limitado (a partir de um conjunto pré-definido).

ALGUNS MECANISMOS:

- **Padrão de Projeto**:
  - Muitos deles identificam características do Sistema que podem variar e fornecem soluções reusáveis:
    - Builder: pode ser usado para carregar um código variante em tempo de execução.
    - Modelar a variabilidade da LPS.

- **Parametrização**: mecanismo em que os atributos de um componente do core assets devem ser preenchidos.
  - O comportamento é determinado em função dos valores atribuídos aos parâmetros
  - Pode aumentar o reúso, mas a manutenção e seleção entre funcionalidades alternativas são consideradas tarefas complicadas e sujeitas a erros.

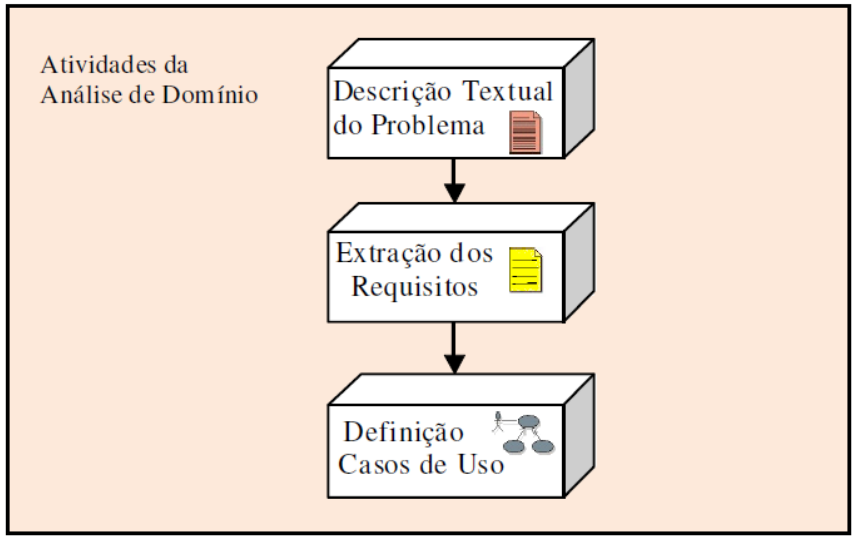
- **Programação Orientada a Aspectos**: separação das responsabilidades espalhadas em módulos distintos.
  - Esta separação habilita alteração, inserção ou remoção de comportamentos do Sistema em tempo de compilação, sem afetar as demais partes do sistema.
  - A variabilidade pode ser implementada a partir destas alterações no comportamento do sistema.

ALGUNS MECANISMOS:

- **Reflexão**: sistemas reflexivos, incorporam estruturas que representam a si próprio, são aptos a examinar e alterar suas próprias propriedades, e a controlar sua implementação em tempo de execução.
  - É possível que o sistema obtenha informações sobre seu tipo de objeto, seus atributos, métodos, construtores e suas estruturas de herança.
  - É possível ainda, dinamicamente, carregar uma nova classe, criar novas instâncias e chamar métodos.

- **Templates**:
  - Componentes não configurados de forma específica para a aplicação, um corpo genérico é preenchido com partes de um produto específico.
    - Extensões com comportamentos adicionais são são possíveis neste mecanismo, já que se deve seguir um modelo genérico pré-estabelecido.

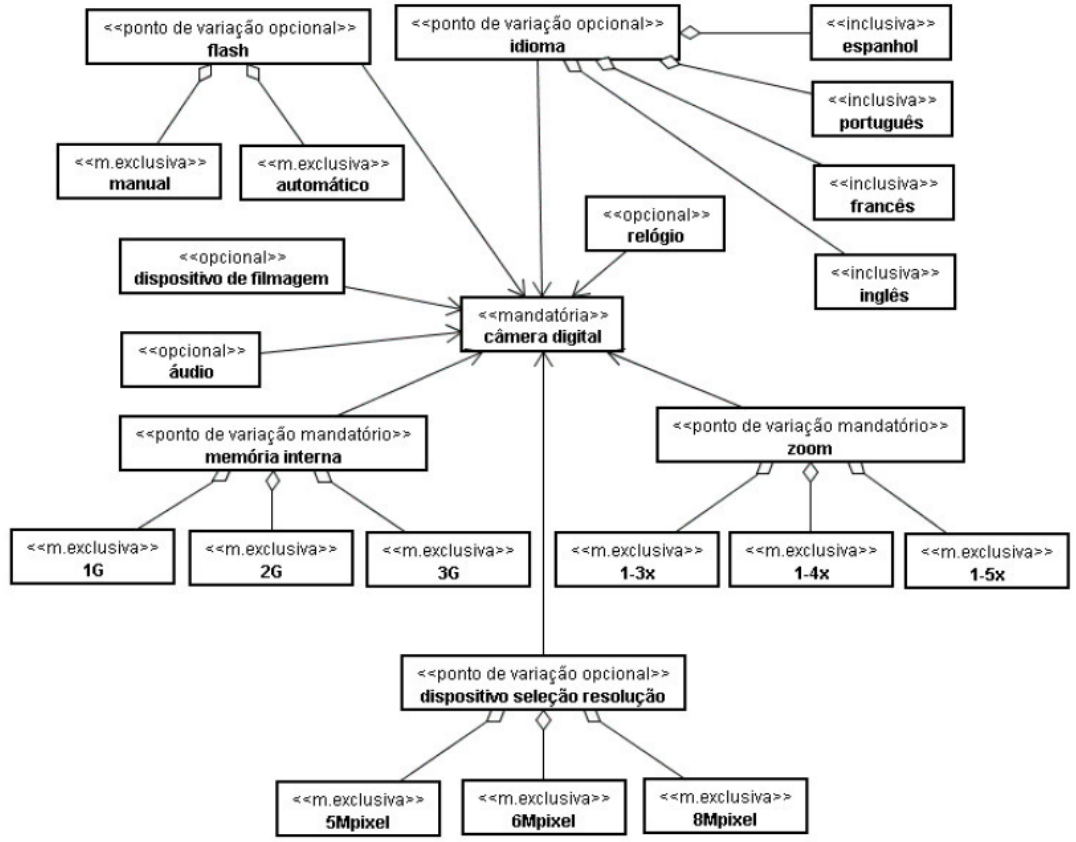
EXEMPLOS DE DIAGRAMAS:



TIPOS/CLASSIFICAÇÃO DE FEATURES:

- Mandatórias;
- Opcionais;
- Pontos de variação mandatória;
- Pontos de variação opcional;
- Variante Inclusiva;
- Variamente mutuamente inclusiva;
- Variante mutuamente exclusiva;

MODELO DE FEATURES



Legenda:

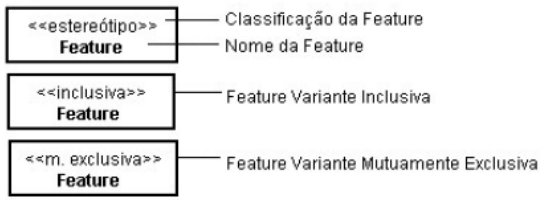
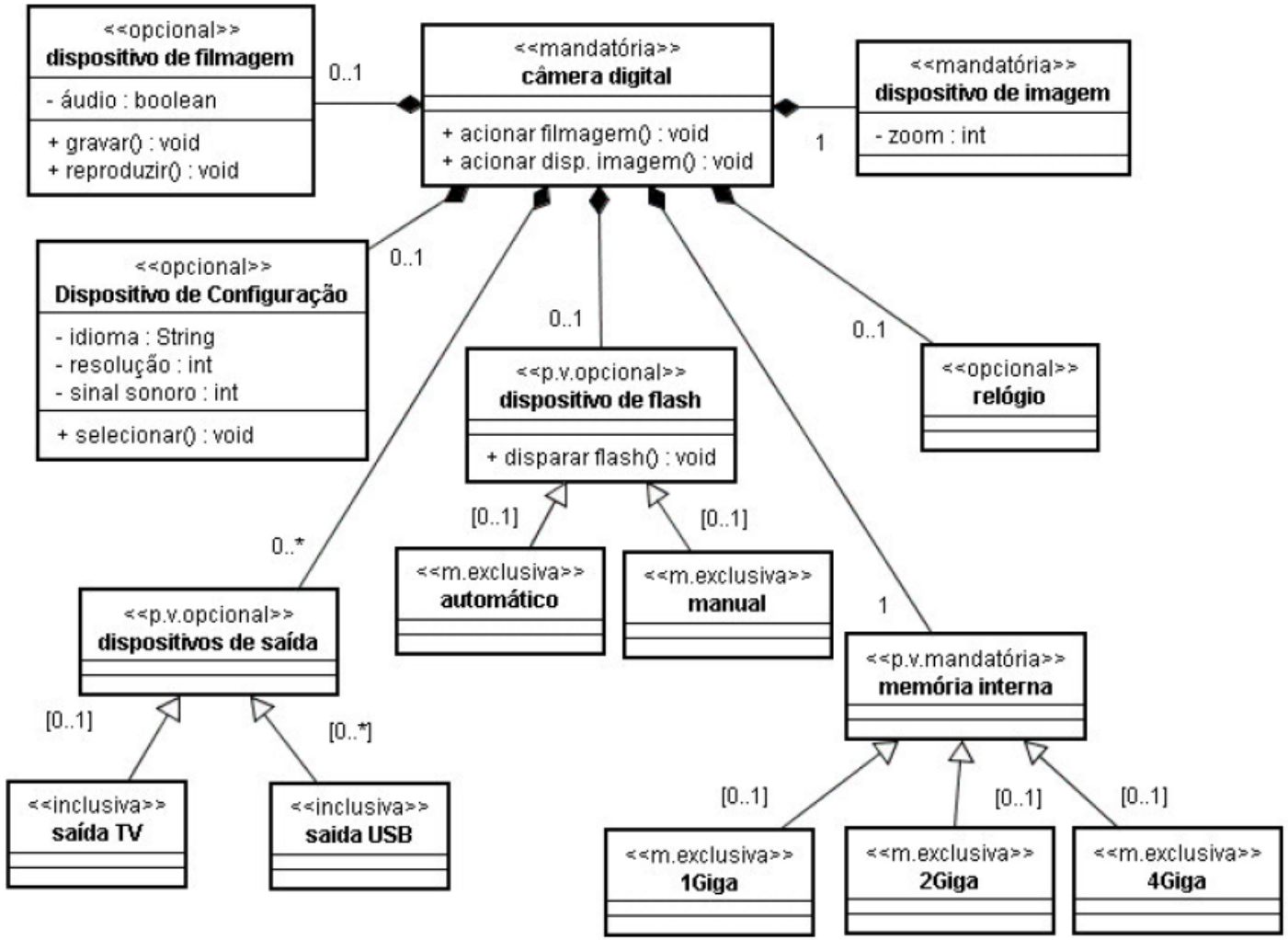


DIAGRAMA DE CLASSES



REFERÊNCIAS:

- Anastasopoulos, M.; Gacek, C (2001). **Implementing Product Line Variabilities.** In: Symposium on Software Reusability – SSR – Putting Software Reuse Context. 2001 Toronto, Ontario, Canada, p. 109-117.
  - http://www.cs.ncl.ac.uk/research/pubs/inproceedings/papers/385.pdf
- Burgareli, L. A. (2009). Gerenciamento de Variabilidade de Linha de Produtos de Software com utilização de Objetos adaptáveis e reflexão. *Universidade de São Paulo. São Paulo*.
  - http://www.teses.usp.br/teses/disponiveis/3/3141/tde-28102009-103948/pt-br.php