



Projeto de Software

Profa. Vanessa Cristina Lourenço



Introdução

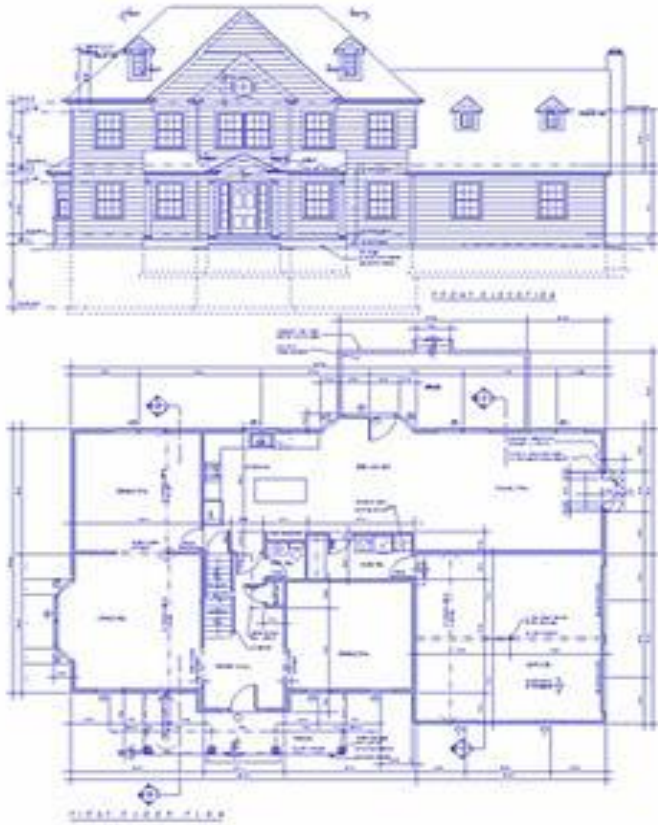
- Engenharia de projeto:
 - Engloba princípios, conceitos e práticas que levam ao desenvolvimento de um sistema ou produto de alta qualidade.



Introdução

- Modelo de projeto de software é equivalente aos modelos de plantas de arquitetura de uma casa:
 - Começa com a representação da totalidade da casa a ser construída
 - Ocorrem refinamentos para se obter os detalhes do que vai ser feito (encanamentos, elétrica, etc.)

Introdução



Assim como é preciso um projeto bem feito antes de se construir uma casa, é preciso um bom projeto antes se construir o software.

A planta de uma casa facilita a comunicação entre os envolvidos na construção, permite uma avaliação do cliente e assim por diante.



Conceito

- Projeto pode ser entendido como um processo:
 - criativo de transformar o problema em uma solução
 - de detalhamento de uma solução
 - de transformação da especificação de requisitos em especificação do software
 - para definir COMO o sistema será implementado



Conceito

- Projeto consiste basicamente:
 - Decompor o sistema em módulo;
 - Descrever o que cada módulo deve fazer (arquitetura do software);
 - Definir como relacionamentos entre os módulos.



Resultado do Projeto de Software

- Produzir um modelo de projeto que contemple:
 - Estrutura dos dados;
 - Arquitetura do sistema;
 - Interfaces com os usuários finais, outros sistemas, dispositivos, etc;
 - Componentes do software.
- Resultado: Documento de Especificação do Projeto.



Resultado do Projeto de Software

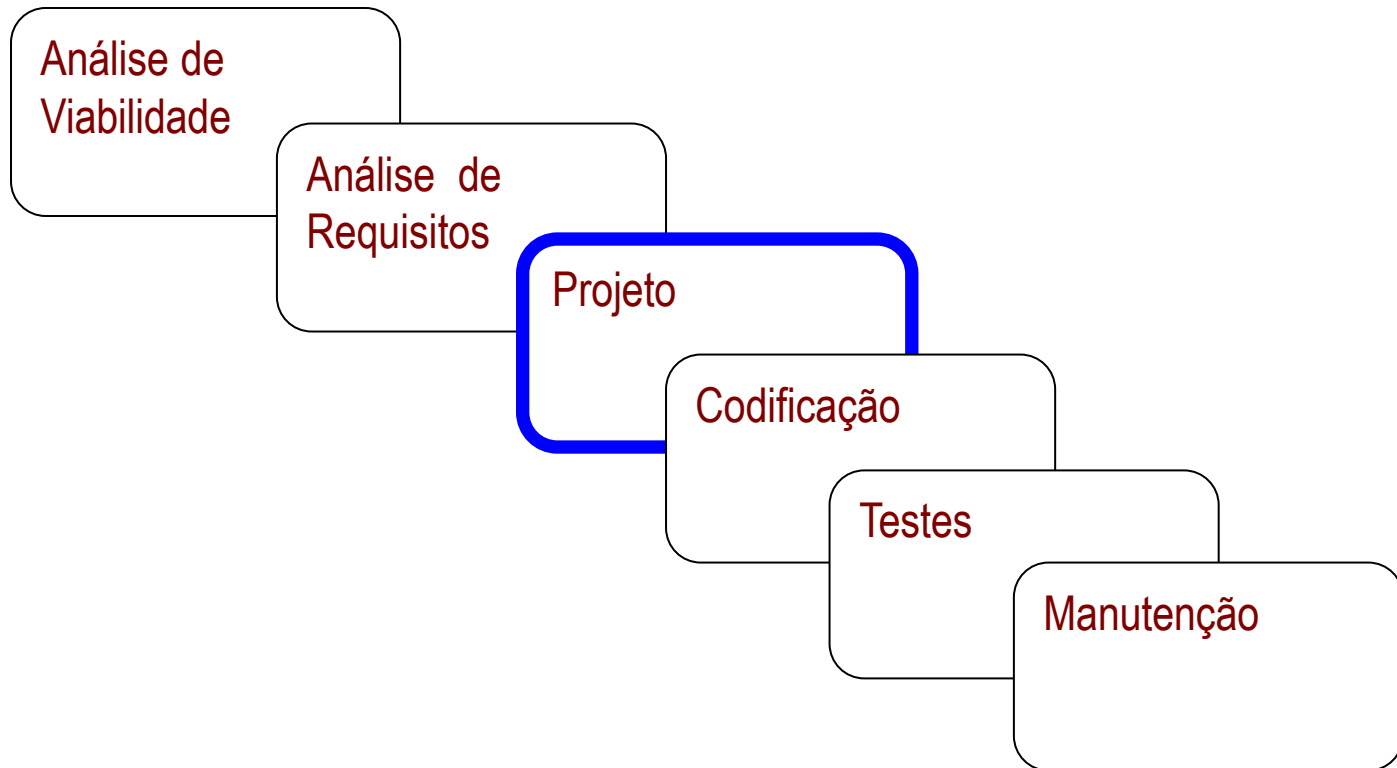
- O projeto de software compreende concepção, especificação e prototipação de partes "externas" e "internas" do software.
- A parte externa compreende:
 - **O modelo conceitual da aplicação e a interface de usuário.**
 - comportamento do sistema de acordo com as funções oferecidas e interação com usuário
- A parte interna compreende:
 - **A arquitetura de componentes de software e os algoritmos e estruturas de dados que implementam estes componentes.**



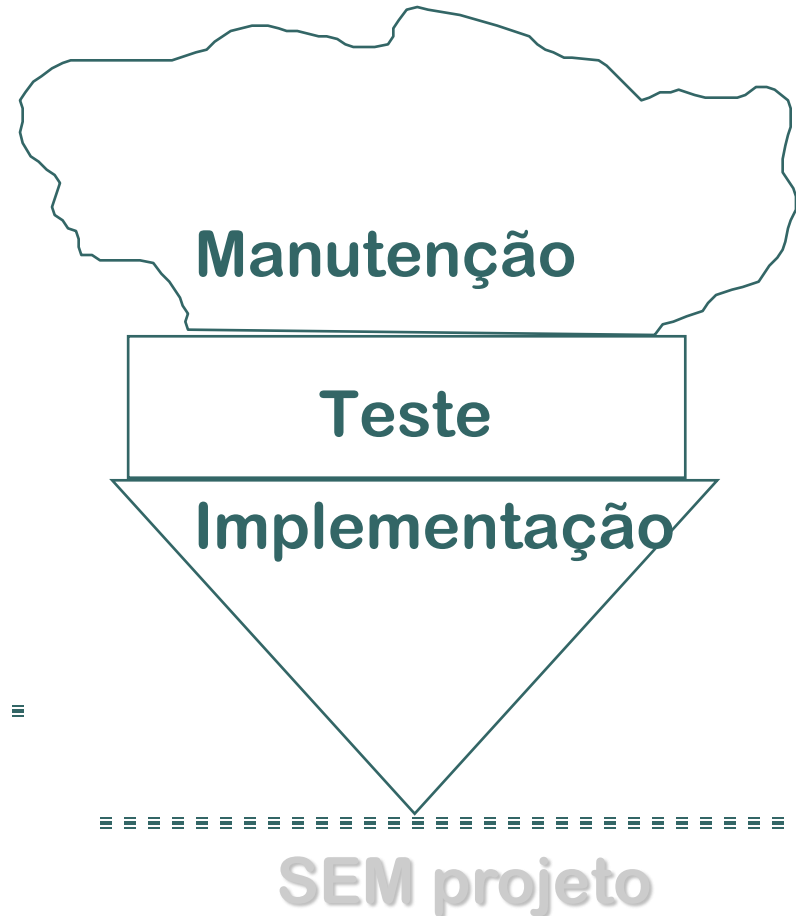
Quando é realizado o projeto?

- Assim que os requisitos do sistemas estiverem identificados, analisados e modelados.
- Deve ocorrer antes do início da codificação, testes e implantação.

Independente do ciclo de vida adotado - sempre antes do início da codificação, testes e manutenção



Importância do Projeto - QUALIDADE - fundação



Importância do Projeto

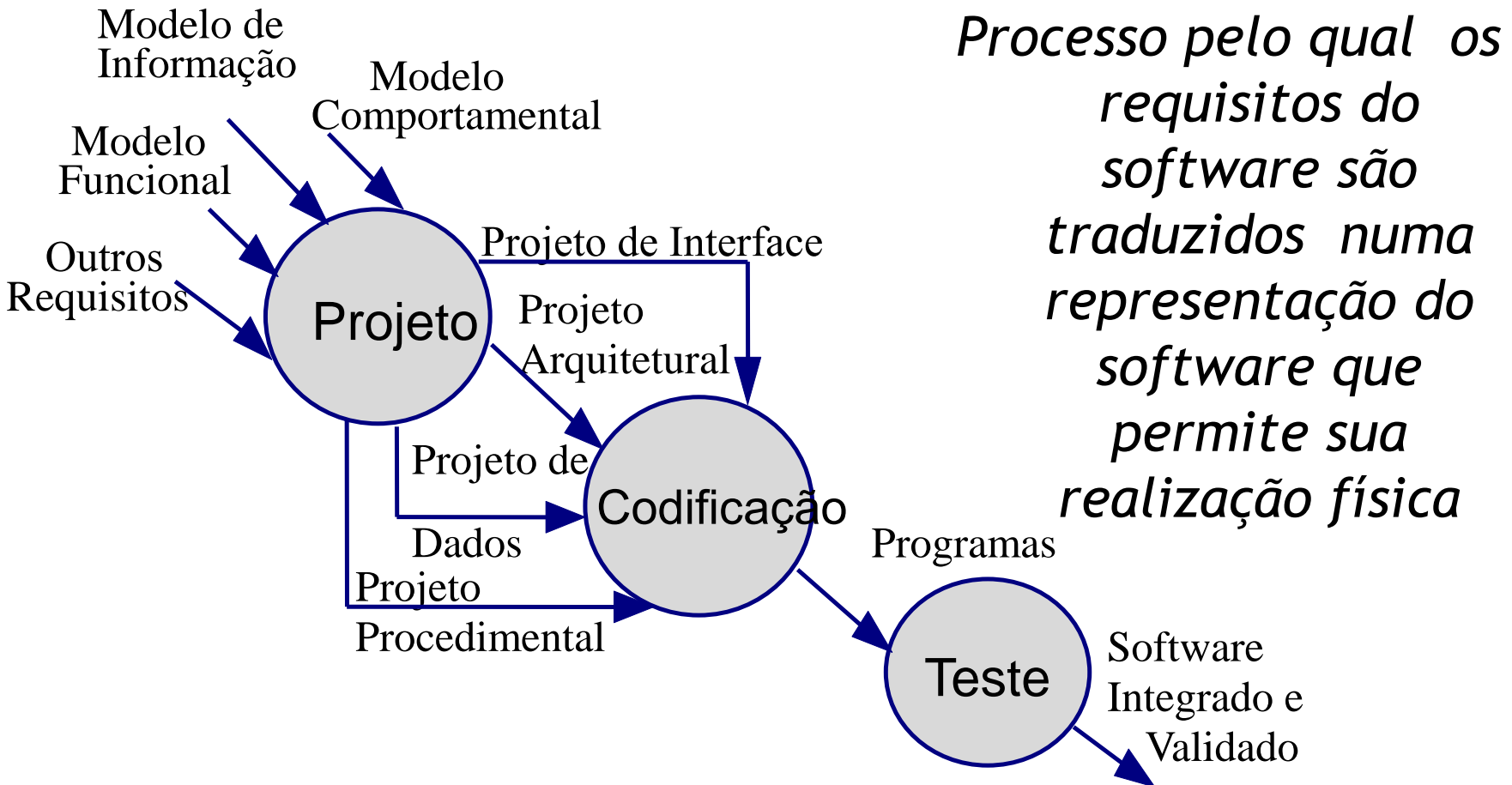
- Projeto permite **modelar o sistema** ou produto a ser construído.
- **Modelo** gerado **pode ser avaliado** quanto a qualidade e aperfeiçoado **antes que**:
 - o código seja gerado,
 - os testes realizados e
 - usuários finais sejam envolvidos.
- Avaliações são feitas através de revisões técnicas formais.
 - reuniões de aproximadamente 90 min a duas horas, onde os envolvidos após terem lido previamente uma cópia do projeto fazem seus comentários. Procura por: Erros, omissões ou ambiguidades.



Exemplos de Diagramas UML gerados nesta fase

- Diagramas UML de Implementação (Projeto arquitetura),
- Diagrama de classes,
- Diagrama de casos de uso,
- Diagrama de interação,
- Diagrama de estados,
- Diagrama de atividades,
- Diagrama de sequência, etc.

Projeto de Software





Modelos de Especificação de Projeto de Software

1. **MODELO DE DADOS:** selecionar representações lógicas dos dados identificados na fase de especificação dos requisitos (Diagrama de classes e dicionário de dados)
2. **MODELO ARQUITETURAL:** desenvolver uma estrutura modular do sistema e representar as relações de controle entre os módulos (elementos estruturais do software - DFD)
3. **MODELO DE INTERFACE:** estabelecer os mecanismos de interação e layout para a interação homem-máquina e com outros sistemas (diagrama de casos de uso)
4. **MODELO EM NÍVEL DE COMPONENTES** (procedural): especificar os detalhes dos algoritmos



Guidelines para o Projeto de Software

- Exibir boa estrutura arquitetural
- Ser modular
- Conter representações distintas de dados, arquitetura, interface e componentes
- Conduzir a componentes que tenham características funcionais independentes



Princípios de Projeto

- Utilizar abordagens alternativas
- Acompanhar os requisitos do modelo de análise
- Não reinventar (utilizar padrões)
- Projetar não é codificar
- Revisar para minimizar erros semânticos -
Revisões Formais



Atributo Coesão

- Medida da proximidade das partes (elementos) de um sub-componente
- Um componente deve implementar uma única entidade lógica ou função
- Importância
 - Quando uma mudança tiver que ser feita, ela será facilmente localizada

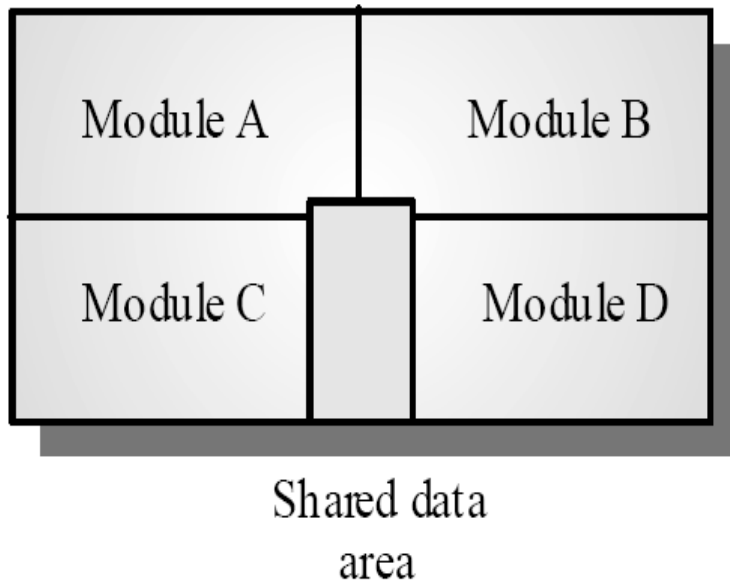


Atributo Acoplamento

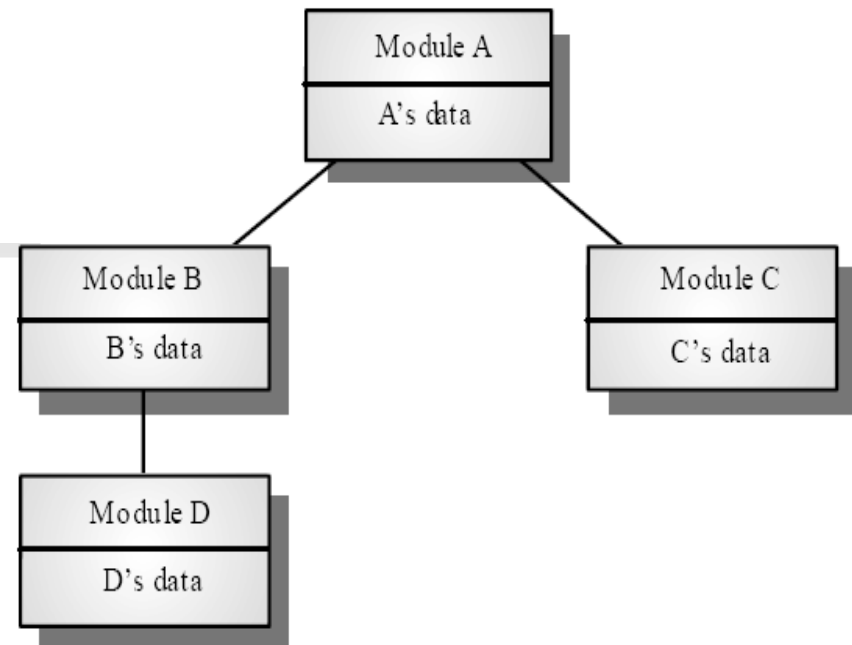
- Medida da intensidade das interconexões entre componentes (módulos) do sistema
- Grau de dependência
- Importância
 - Baixo acoplamento implica que mudanças em um componente tendem a não afetar outros componentes

Atributo Acoplamento

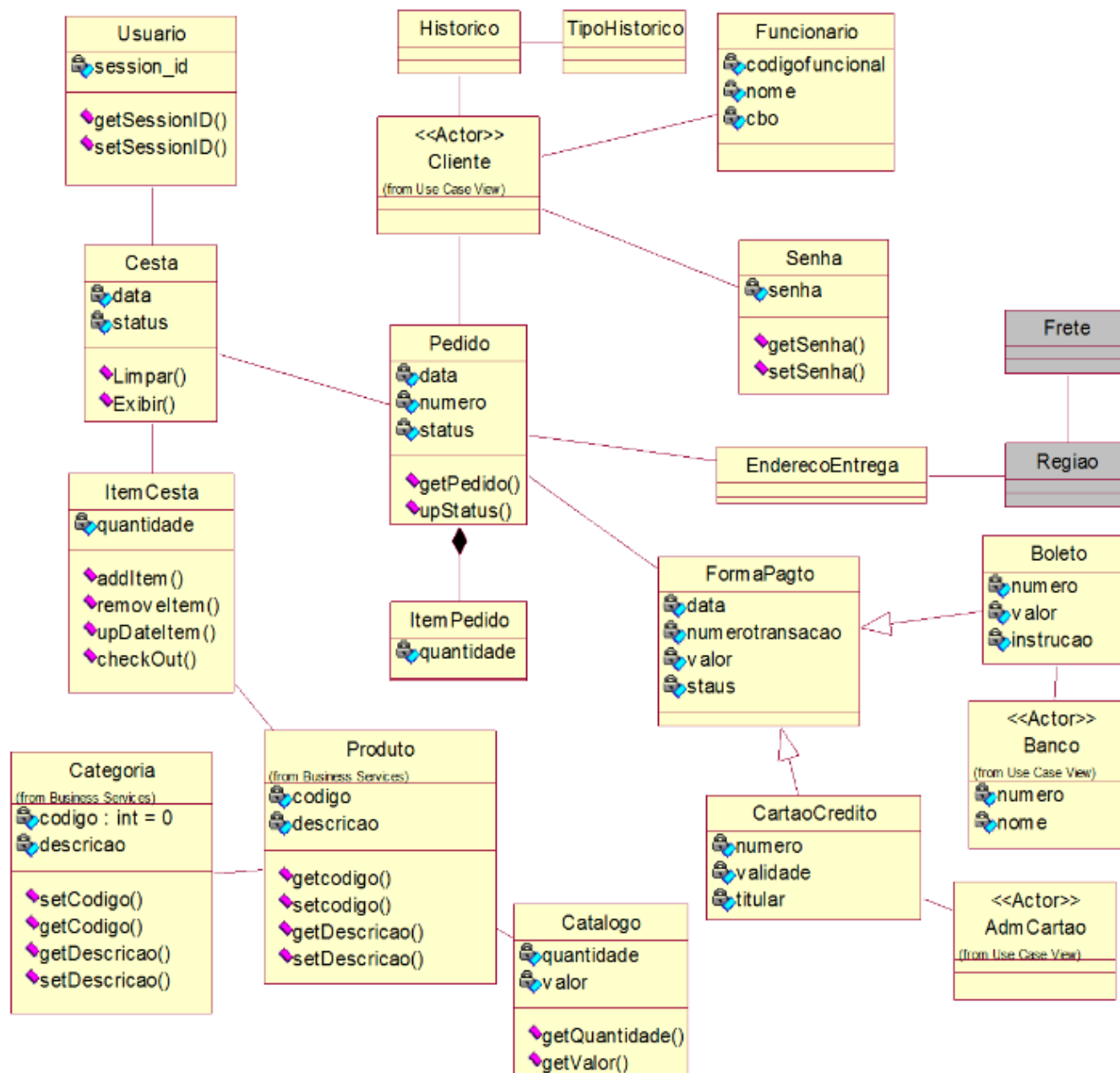
Uma boa arquitetura deve ter alta coesão e baixo acoplamento



Acoplamento Forte



Acoplamento Fraco



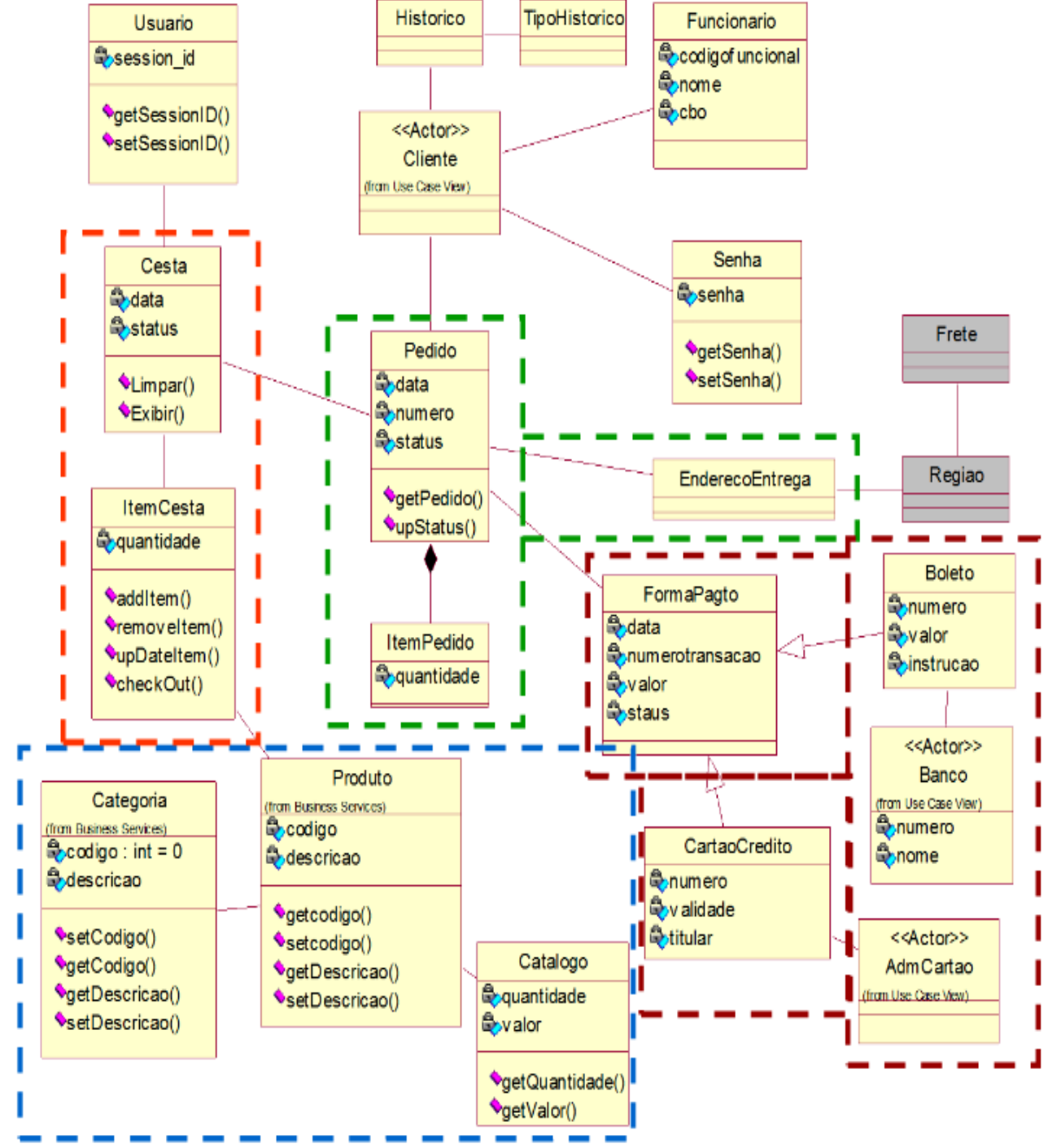
A partir do diagrama de classe, agrupar classes usando os conceitos de coesão e acoplamento.

Pedido

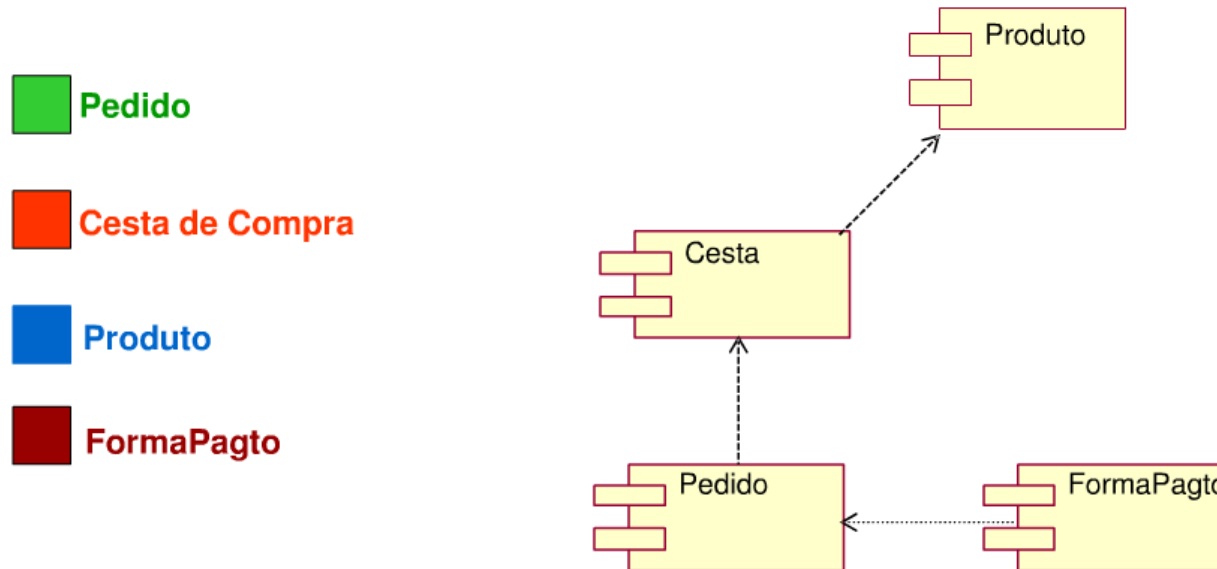
Cesta de Compra

Produto

FormaPagto



Resultado - Diagrama de Componentes





Fatores que influenciam no Projeto

- Funcionalidade
- Performance - tempo de resposta, tempo total de execução
- Eficiência - uso eficiente do dispositivo de armazenamento e do hardware
- Portabilidade - funcionar em novas plataformas
- Segurança
- Disponibilidade - disponível quando requerido
- Robustez
- Usabilidade
- Reusabilidade



Referências

- Sommerville, I. Engenharia de Software.
- Pressman, R. Engenharia de Software.