

Universidade Presbiteriana Mackenzie



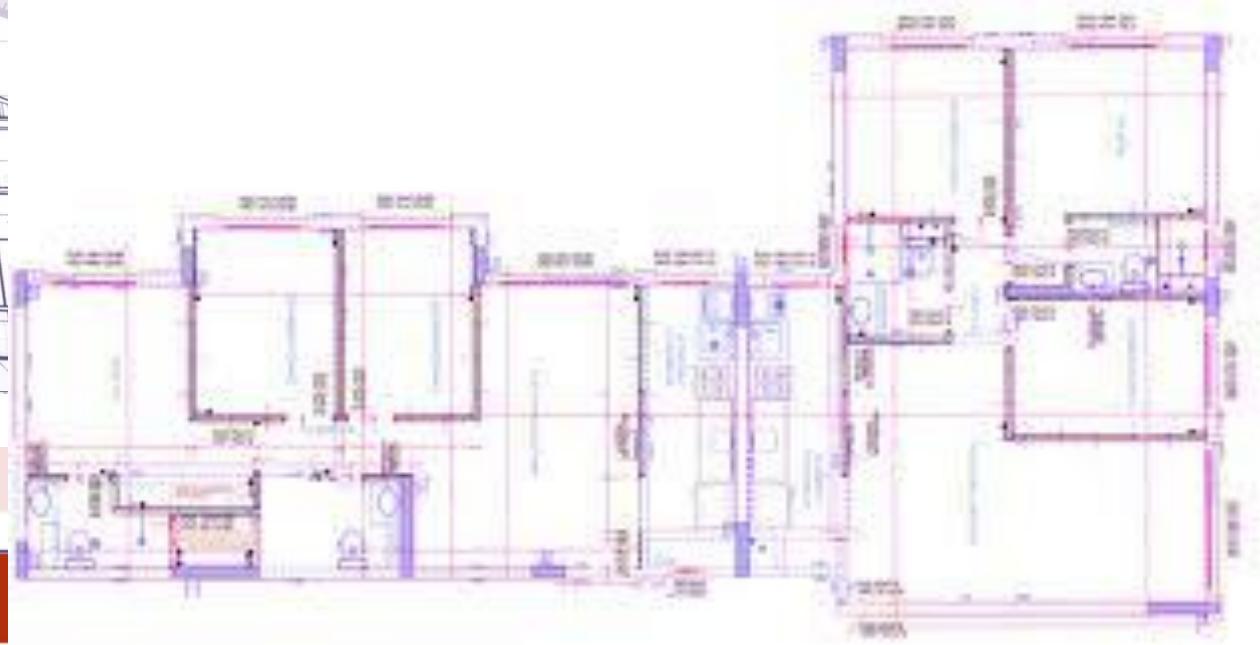
Análise, Projeto e Desenvolvimento III

Profa. Ana Claudia Rossi

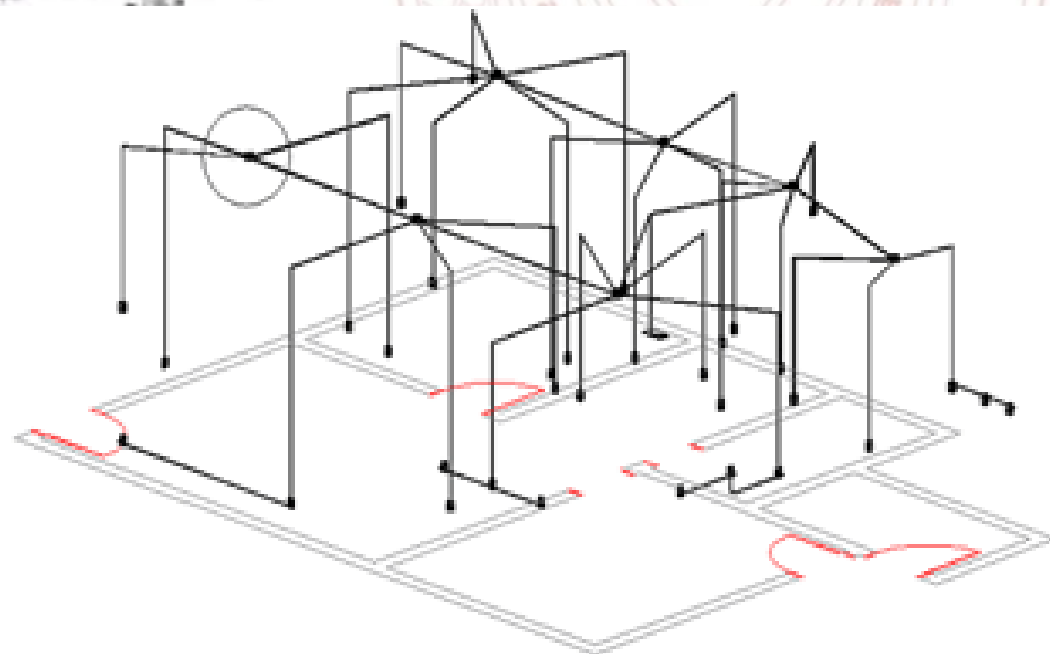
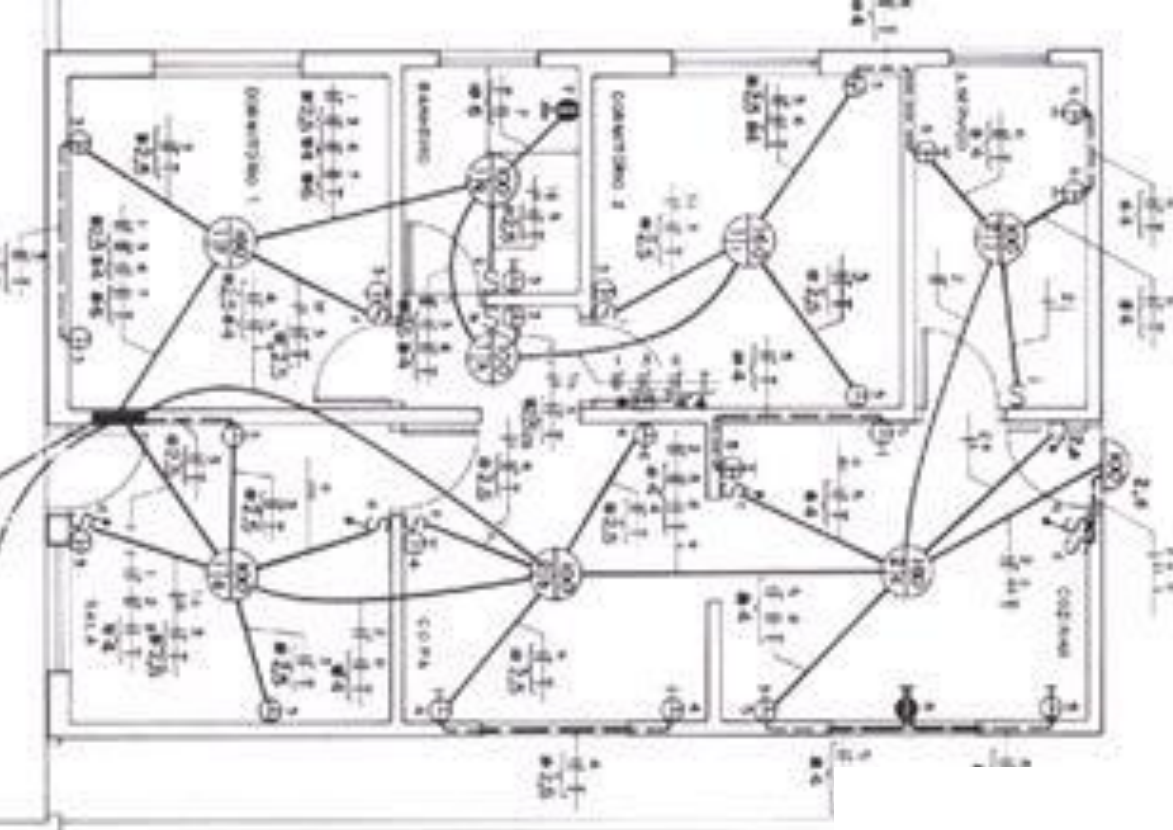
Faculdade de Computação e Informática



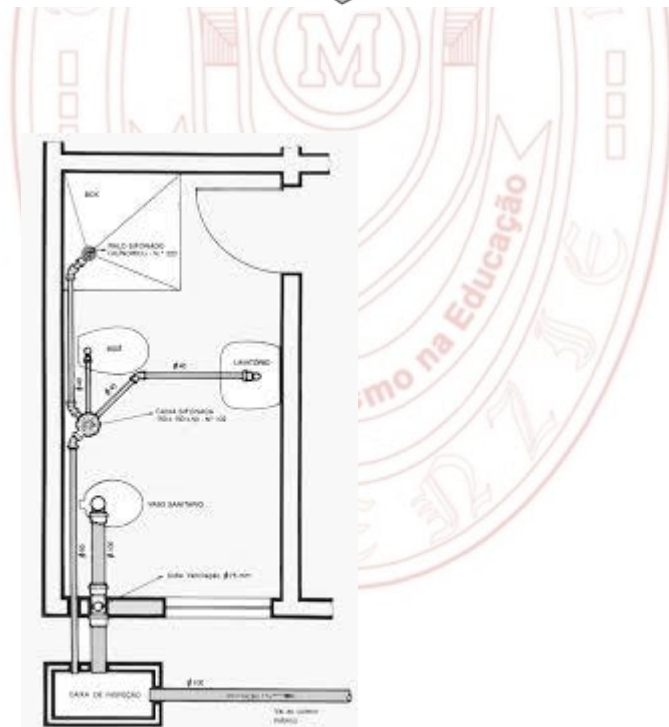
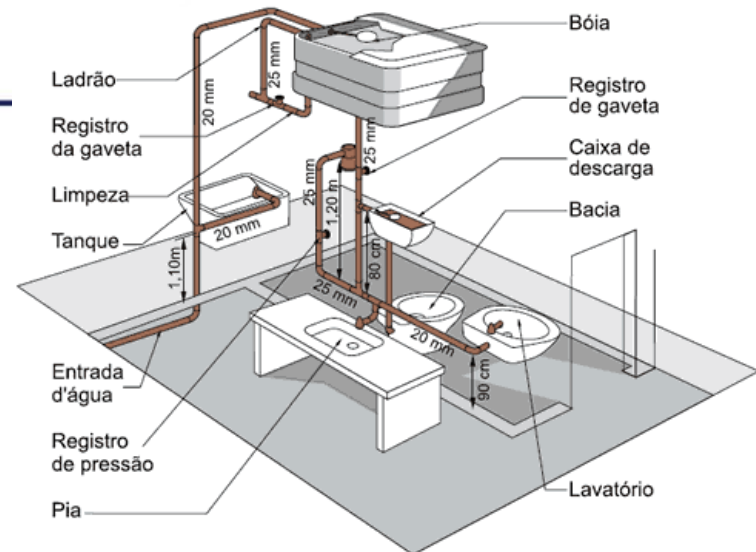
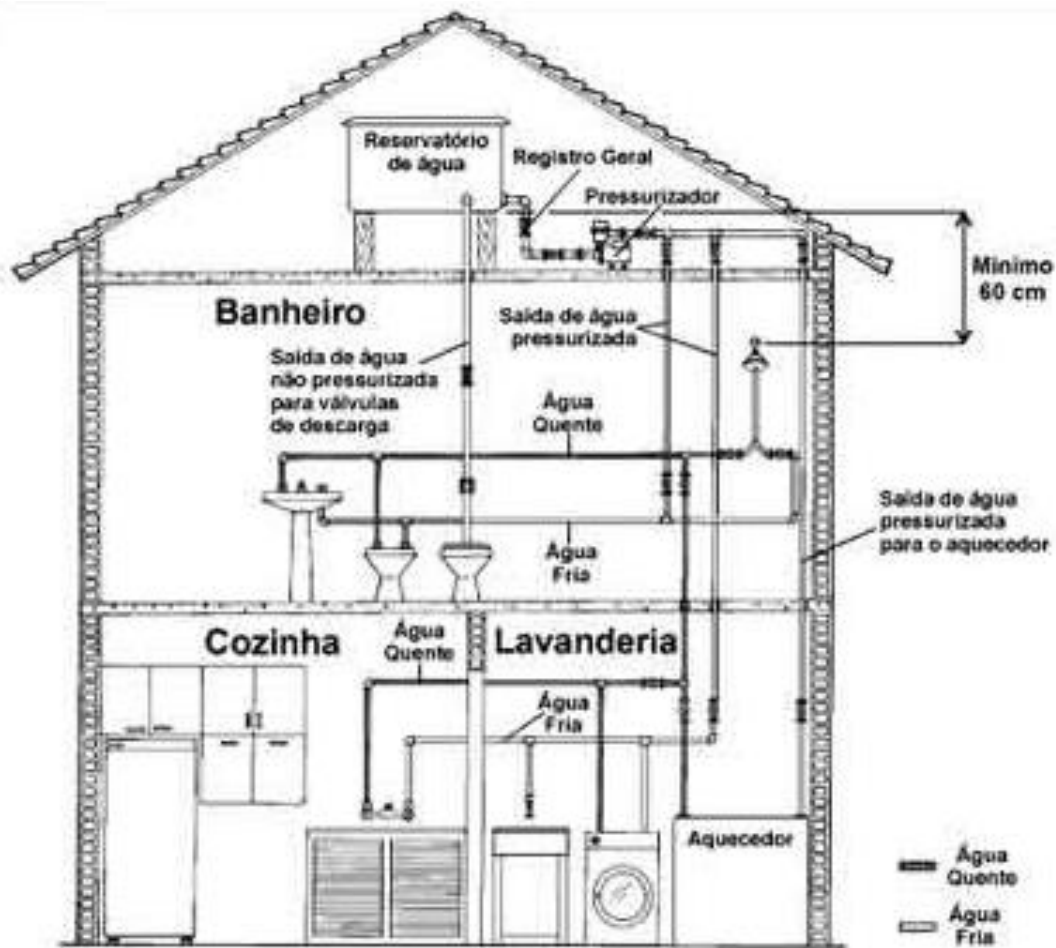
Visão da Planta Baixa



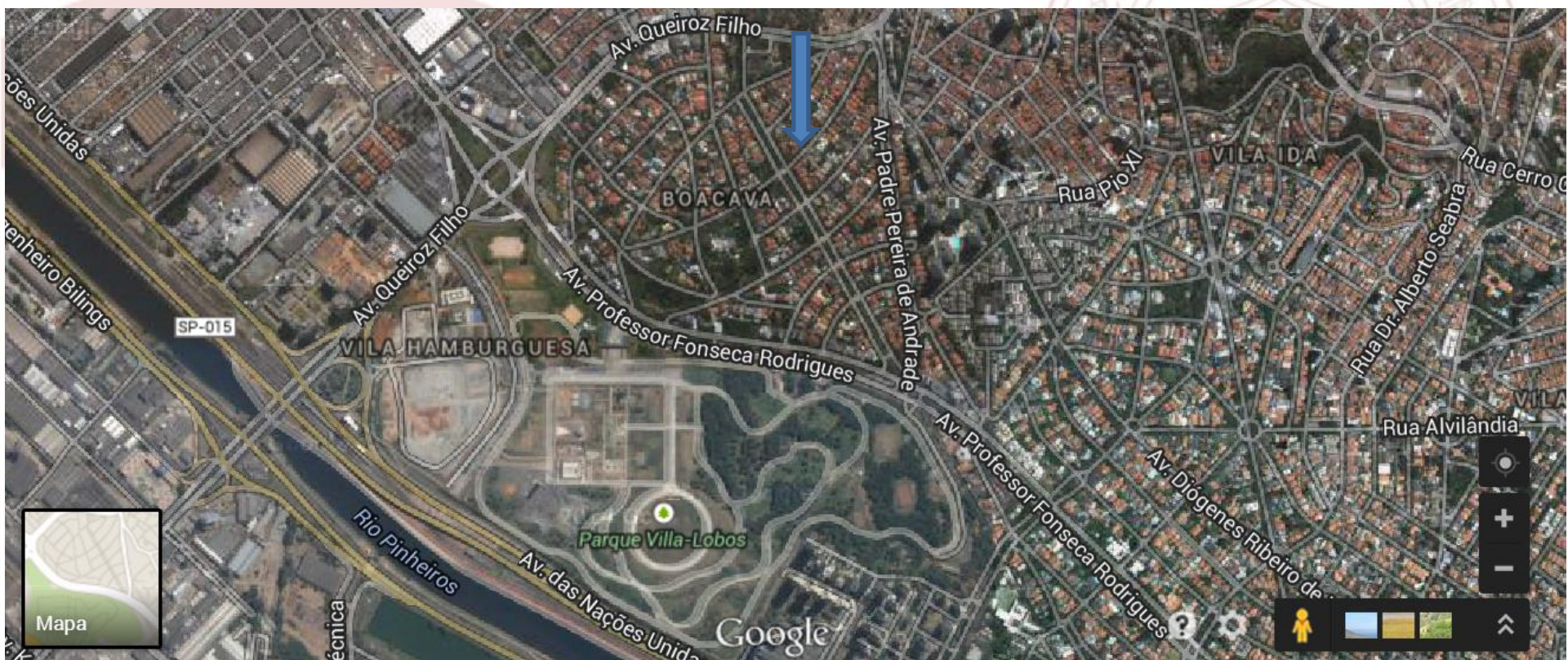
Visão Planta Elétrica



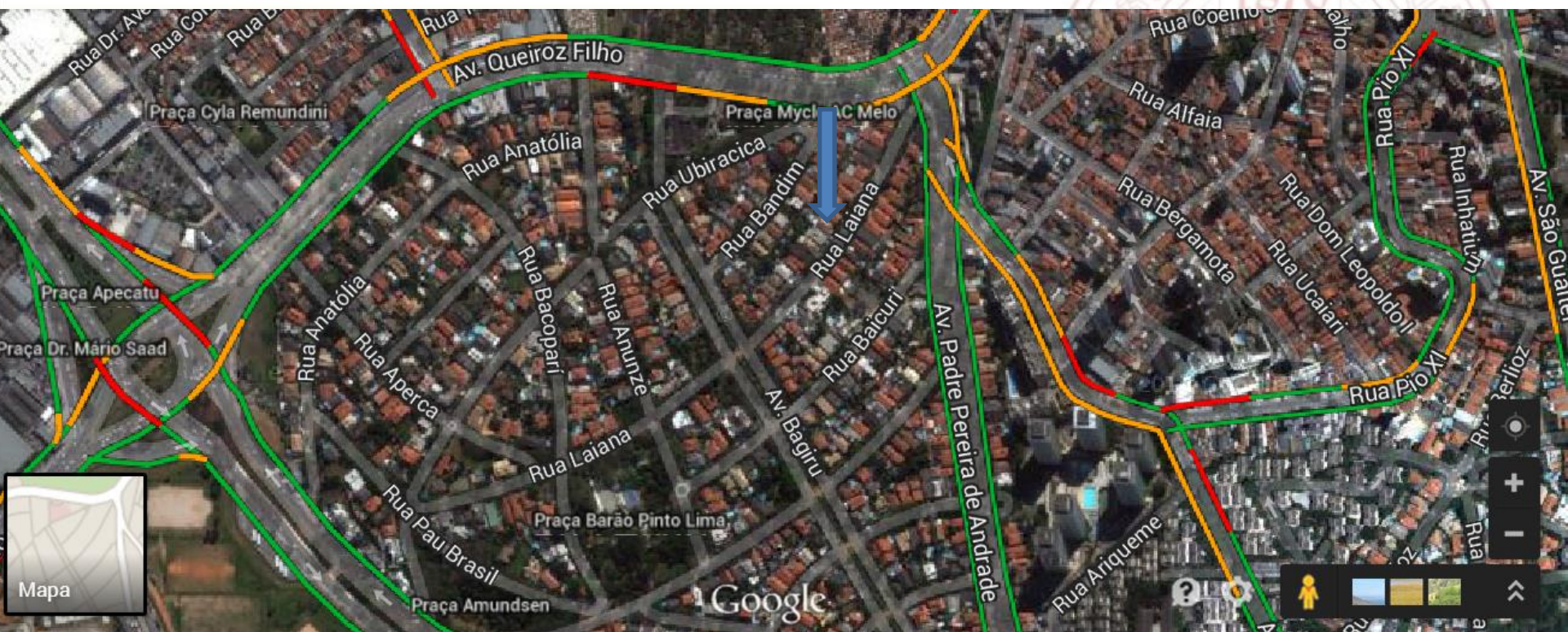
Visão Planta Hidraulica



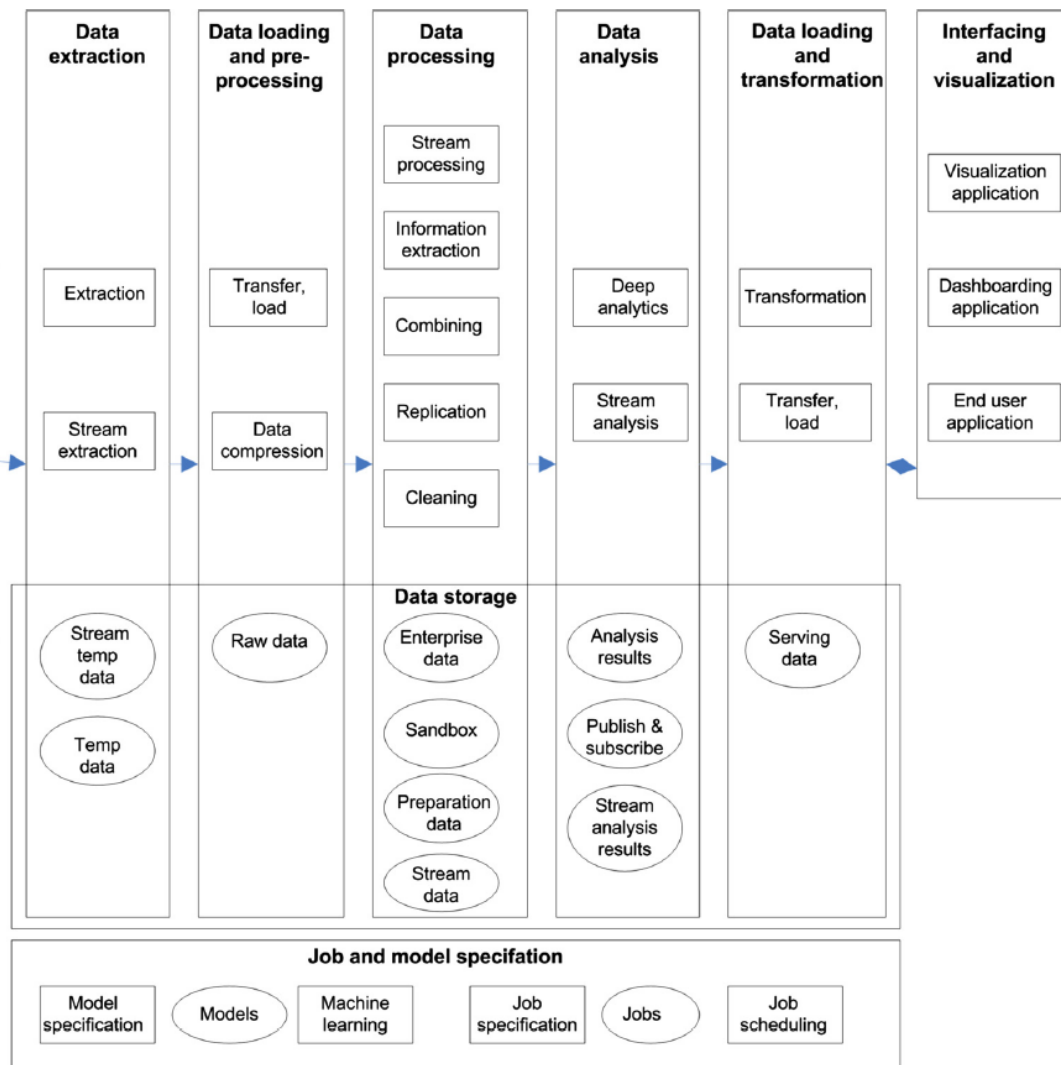
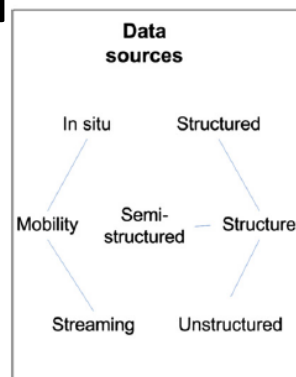
Contexto



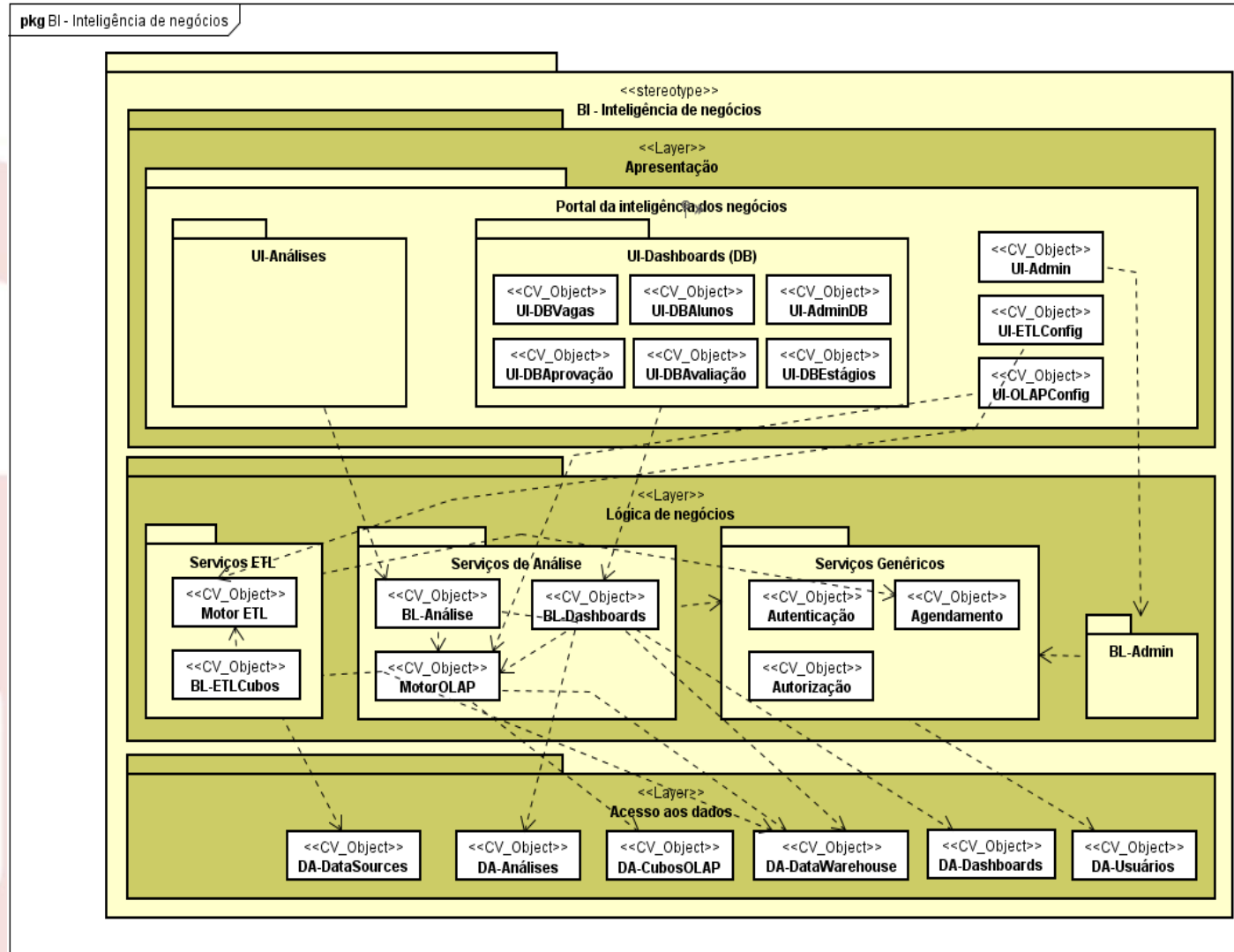
Contexto



Arquitetura de Referência - BigData



Portal de Estágios



Arquitetura de software

- Existem três elementos importantes nos sistemas computacionais
 - Hardware, software, dados
- A arquitetura de um sistema intensive-software é a estrutura ou as estruturas do sistema composta pelos elementos de software, as propriedades externas e visíveis destes elementos e os relacionamentos entre estes elementos

Estruturas do sistema

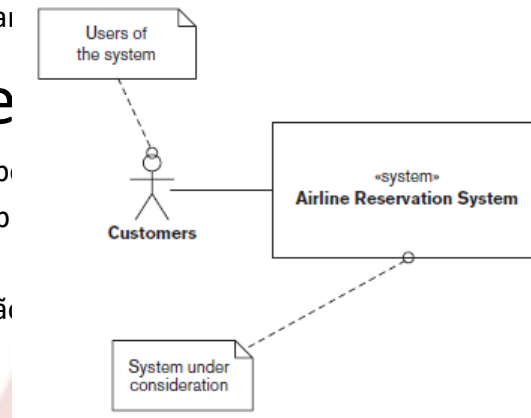
- Estrutura estática
 - Define os elementos internos do software na concepção do projeto (design-time)
 - Módulos, objetos, classes, pacotes, serviços, stored-procedure, outros
 - Formas de agrupamento: Associações, relacionamentos ou conexões
 - Depende do contexto
- Estrutura dinâmica
 - Elementos (“runtime”) e sua iterações
 - Como o sistema trabalha, que acontece “runtime” e como responde aos estímulos externos ou internos
 - Exemplo: fluxo de informações entre elementos

Propriedades externas

- Comportamento externamente visível
 - Iterações funcionais entre o sistema e o seu ambiente
 - Fluxo de informação de entrada e saída, como o sistema responde aos estímulos externos, o contrato publicado ou API que a arquitetura tem com o mundo exterior
 - O comportamento externo deve ser modelado observando o sistema como caixa preta
- Propriedades de qualidade
 - Propriedades não funcionais, externas e visíveis
 - Desempenho, segurança, outros
 - Como o sistema se comporta do ponto de vista de um observador externo

Exemplo

- Sistema de reserva de viagem
 - Processa diferentes tipos de transações
 - Check in, update e cancel
 - Comportamento externo e visível
 - Resposta a tra
 - Propriedade
 - Tempo médio de resp
 - Número máximo de p
 - Disponibilidade
 - Tempo de recuperaçã

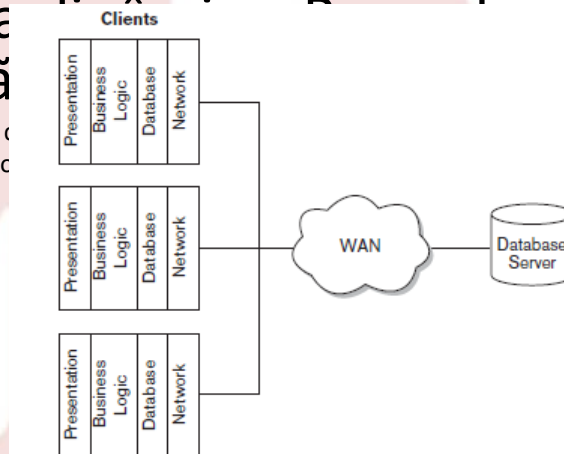


Solução 1

- Client-Server

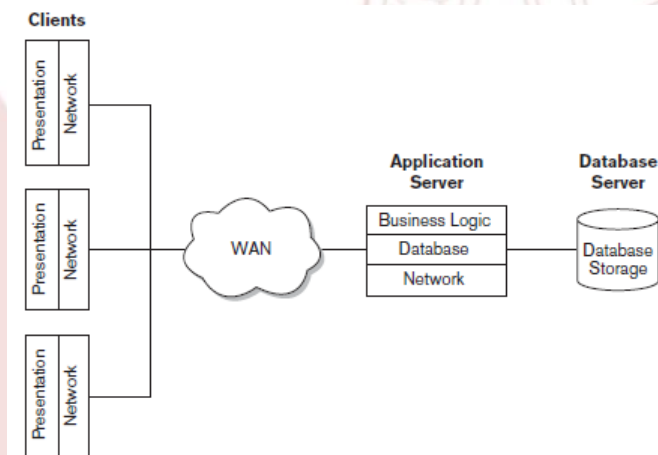
- Clientes comunicam-se com o servidor central de BD via rede Wan
- Estrutura estática: Programa do cliente, servidor de BD, e a conexão entre eles
- Estrutura dinâmica: Cliente, servidor e modelo de requisição

- A requisição do cliente
- A resposta do servidor



Solução 2

- Thin- Client ou three - tier
 - A apresentação é processada no cliente e a logica de negócios e a base de dados roda no servidor de aplicação
- Estrutura estática
 - Programa do cliente: Dividido em camada de apresentação e rede
 - Servidor de aplicação: Dividido em camada de lógica de negócios, banco de dados e rede
 - Servidor de banco e conexões
- Estrutura dinâmica
 - Modelo deve ter camadas de requisição e resposta



Arquitetura candidata

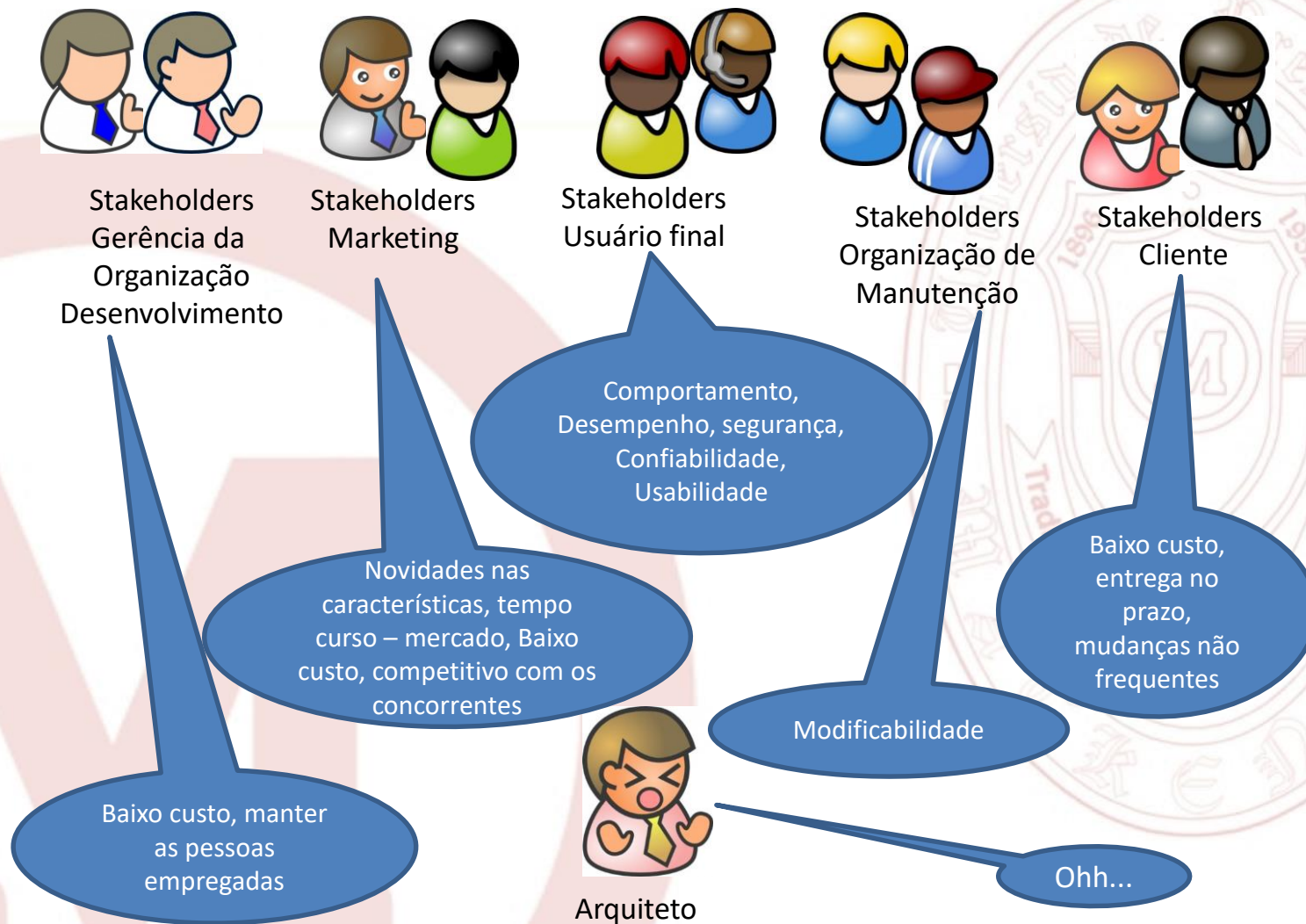
- A arquitetura candidata de um sistema é um conjunto de estruturas estáticas e dinâmicas que tem o potencial de mostrar o comportamento externo e visível e propriedades de qualidade
- As propriedades externas e visíveis do sistemas são determinadas pela combinação de comportamentos funcionais dos elementos internos
- As propriedades de qualidade do sistema se originam das propriedades de qualidade de seus elementos internos

Elementos arquiteturais

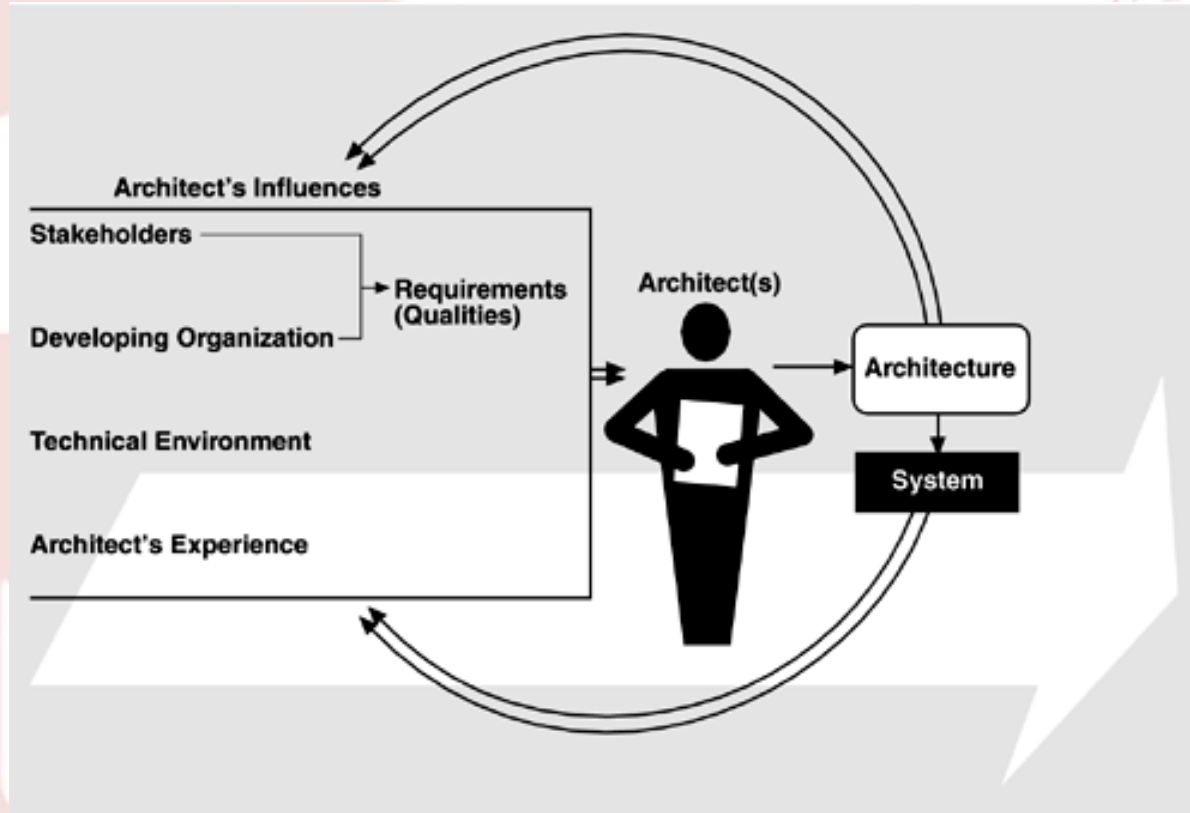
- É o elemento ou elementos básicos que se utilizam para construir o sistema
- A natureza do elemento arquitetural depende do tipo de sistema e de seu contexto
 - Bibliotecas de programas, subsistemas, unidades distribuídas (Enterprise java beans e active X control), aplicações
 - Componentes ou módulos
- Deve possuir um conjunto de atributos principais
 - Conjunto de responsabilidades
 - Restrições
 - Interfaces
 - Define os serviços que oferece aos outros elementos arquiteturais

Stakeholders

- Sistema
 - Usado, construído, testado, operado, mantido, melhorado, pago
- Pessoa, grupo, ou instituição com interesses ou “Concern” com a realização da arquitetura
- Concern
 - É um requisito, um objetivo, uma intenção ou aspiração que o stakeholder tem com sua arquitetura
- Uma boa arquitetura é aquela que atinge os objetivos, metas e necessidades dos stakeholders

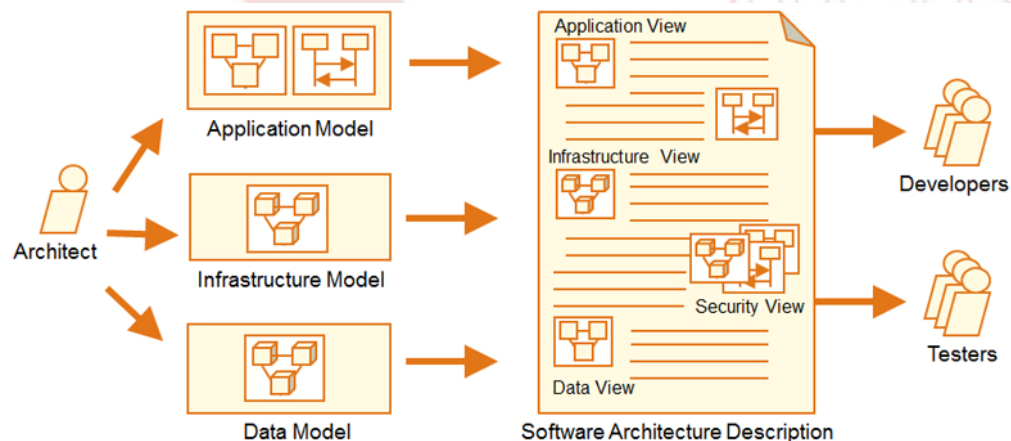


The Architecture Business Cycle



Descrição arquitetural

- Conjunto de produtos que documentam uma arquitetura de forma que os stakeholders consigam entender e demonstrar que seus “Concerns” foram atingidos



Processo de Desenvolvimento de Software

Orientação a Objetos

Por que modelar?

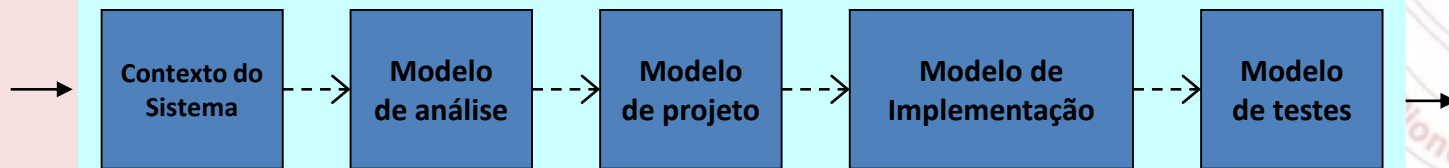
- Gerenciamento da complexidade
- Comunicação entre as pessoas envolvidas
- Ajuda a gerenciar custos de desenvolvimento
- Previsão do comportamento futuro do sistema.

Modelagem de Sistemas de Software consiste na utilização de notação gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema, considerando as várias perspectivas diferentes e complementares.

Principais atividades do desenvolvimento de software



Abstrato
Requisitos



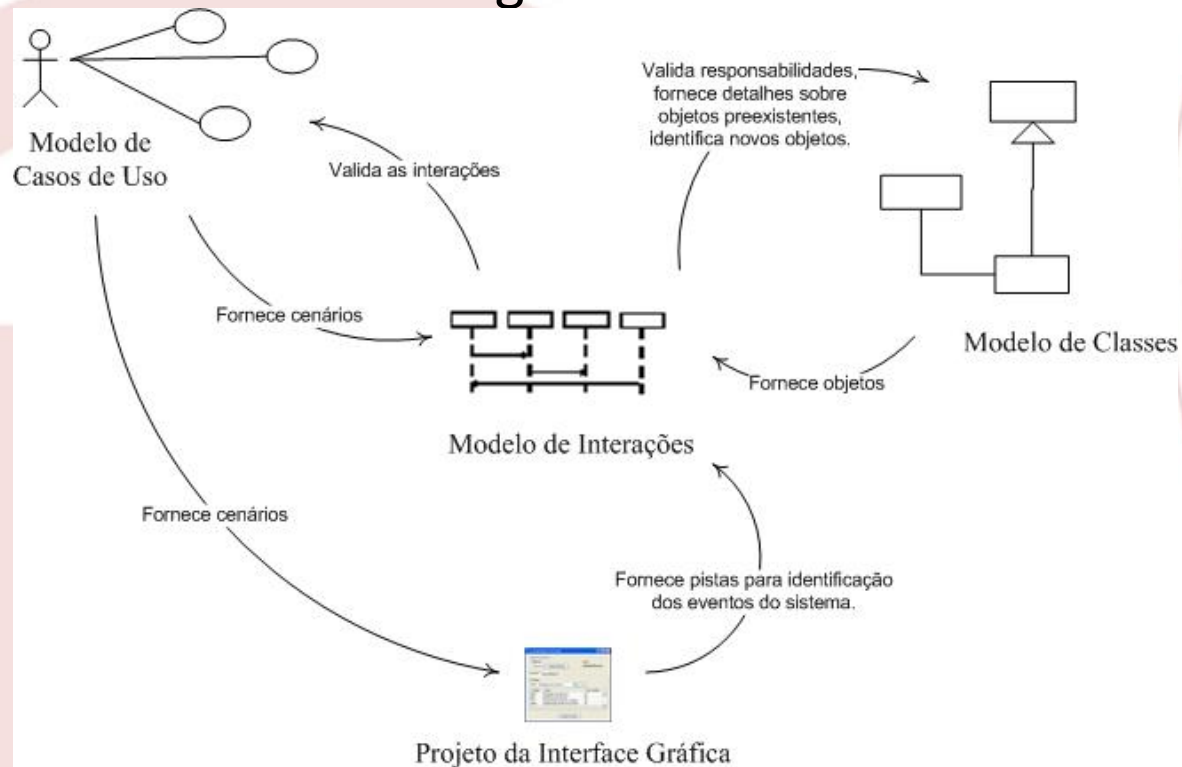
Concreta
Sistema

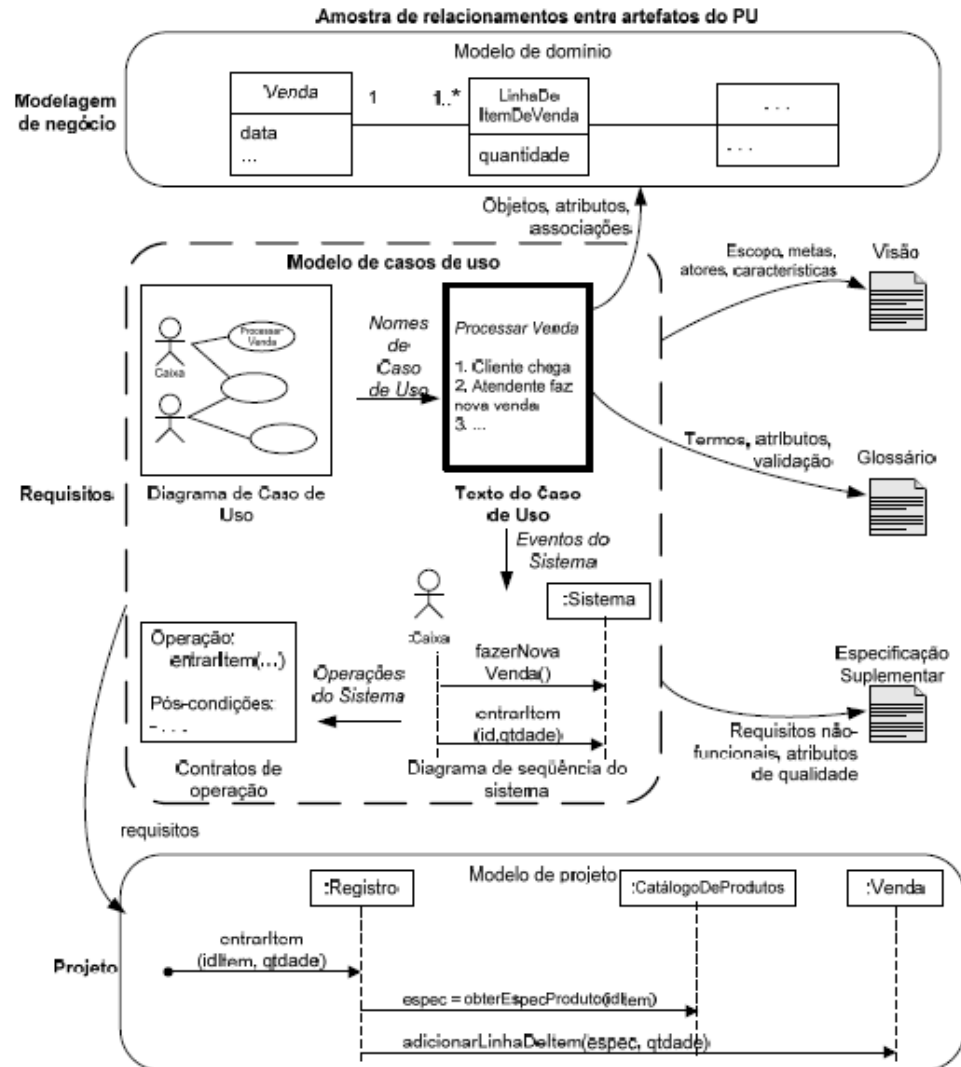
Captura de
Requisitos

Construção

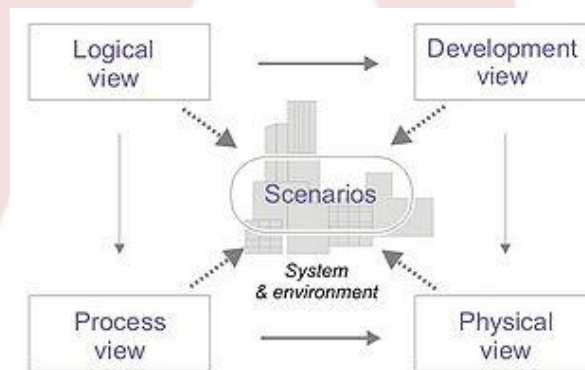
Testes Transição,
Instalação,
Entrega

Processo de Desenvolvimento – Abordagem Iterativa



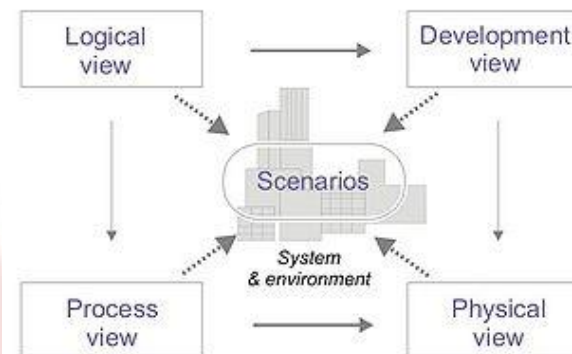


Modelos – Visões 4 + 1



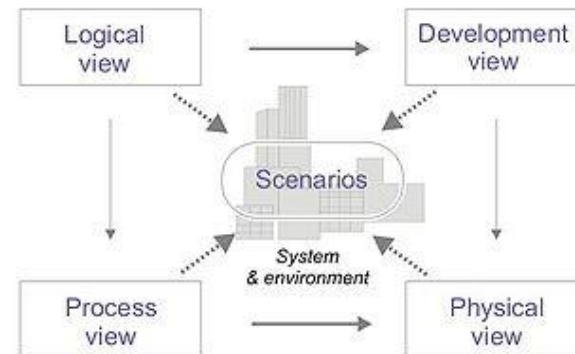
4+1 Visões

- Cenários
- Visão Lógica
- Visão de Processos
- Visão da Implementação
- Visão da Implantação



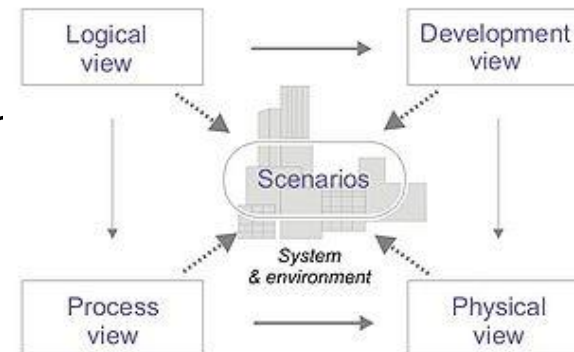
4+1 Visões

- Cenários
 - Descreve a funcionalidade do sistema.
 - Casos de uso e cenários são considerados o ponto de partida e também de consolidação das outras visões
 - Detalha ações e condições em cada caso de uso/cenário



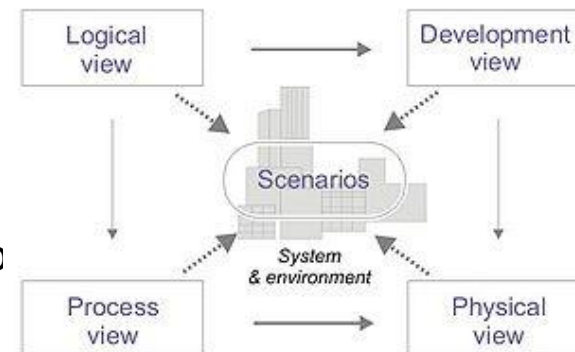
4+1 Visões

- Visão Lógica
 - Descreve requisitos comportamentais e a decomposição do sistema em um conjunto de abstrações
 - Classes e objetos são os principais elementos nesta visão
 - Diagramas de classes, sequência e colaboração mostram os relacionamentos entre estes elementos



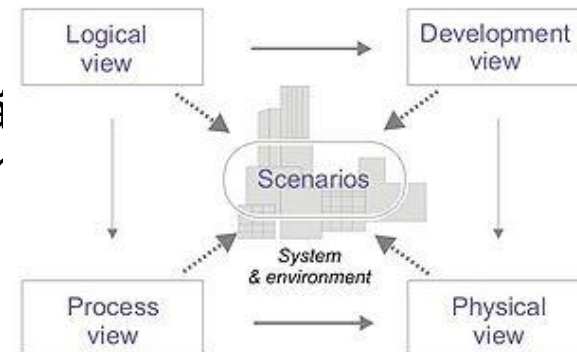
4+1 Visões

- Visão de Processos
 - Descreve os processos do sistema e como eles se comunicam
 - Útil quando se tem múltiplos processos ou threads concorrentes
 - Permite avaliar requisitos não funcionais relacionados à execução e comunicação:
 - Desempenho, disponibilidade
 - Diagramas de atividades são úteis para descrever esta visão



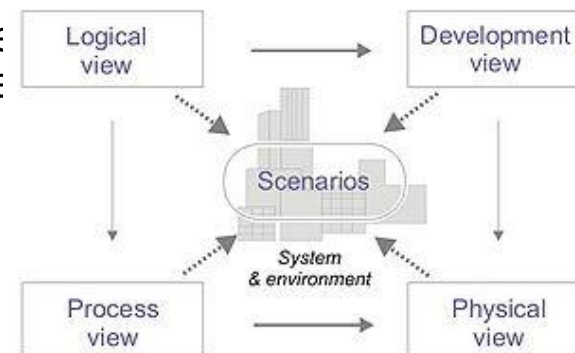
4+1 Visões

- Visão da Implementação
 - Usada para descrever os módulos do sistema
 - Módulos são elementos mais abstratos que classes e objetos
 - Pacotes e biblioteca de classes, componentes são exemplos de módulos e alguns ambientes de programação.



4+1 Visões

- Visão da Implantação
 - Descreve como a aplicação é instalada e como executa em uma rede de computadores.
 - Componentes executáveis são alocados a nós processadores.
 - Esta visão permite avaliar requisitos não-funcionais - desempenho, disponibilidade, confiabilidade, escalabilidade



Obrigado

Ana Claudia Rossi

ana.rossi@mackenzie.br

