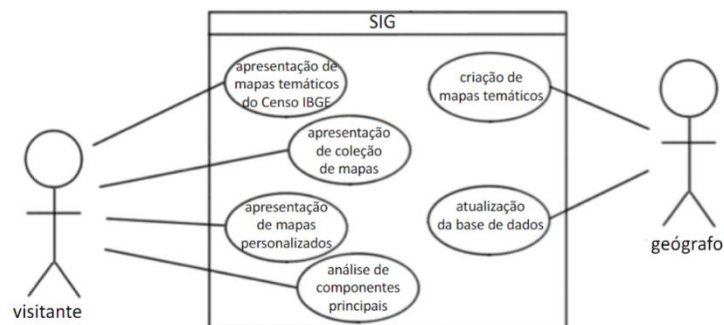


Introdução e conceitos:

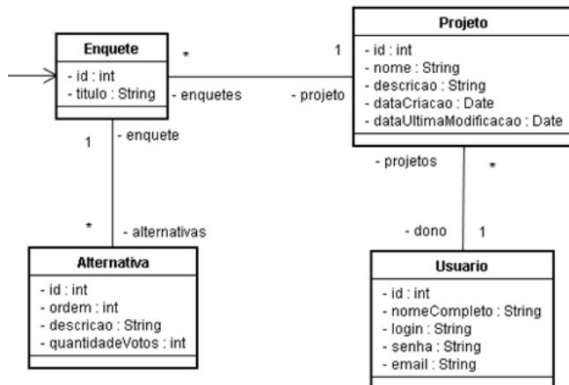
1. Qualidades em projeto de software:
 - a. Estimativas -> métricas
 - b. Reuso
 - c. Padrões, padrões arquiteturais e frameworks, design patterns
 - d. Testes
 - e. Integração contínua
2. Projeto Estruturado X Projeto Orientado a Objetos
 - Estruturado - 1971:**
 - Top- down
 - Modular
 - Módulo caixa preta
 - Refinamento sucessivo
 - OO - 1986:**
 - Abstração, a modularização e a reutilização de código
3. Projeto preliminar: Caso de Uso , Interface e Robustez
4. Projeto detalhado: Interação de objetos(Sequencia e colaboração) Pacotes e componentes e execução

Casos de uso:



Identificador	CSU01
Lista de atores	Visitante
Descrição	Apresentação de mapas personalizados
Pré-condições	Página de mapas personalizados ativa
Fluxo básico de eventos	1 - Usuário seleciona a variável (cursos, matrículas, ingressos, matrículas por mil) 2 - Sistema atualiza o mapa e estatísticas 3 - Usuário seleciona a área de conhecimento (total, grande área, área, subárea) 4 - Sistema atualiza o mapa e estatísticas 5 - Usuário seleciona o grau dos cursos (total, bacharelado, licenciatura, tecnólogo) 6 - Sistema atualiza o mapa e estatísticas 7 - Usuário seleciona a modalidade (total, presencial, a distância) 8 - Sistema atualiza o mapa e estatísticas
Fluxo alternativo	
Fluxo de exceção	
Pós-condições	Dados apresentados no mapa de acordo com as seleções nos campos

Classe:



Robustez:

Consiste na produção incremental e em paralelo de um conjunto de artefatos que retratam as visões dinâmica e estática de um sistema, privilegiando a “rastreadibilidade” e a robustez.

Fazer a análise de robustez:

Criar Diagramas de Robustez usando os estereótipos de classes boundary, control, e entity – Atualizar o modelo do domínio, com os novos Objetos e atributos descobertos:



Regras:

- Os actores podem comunicar com o sistema através de Objetos fronteira.
- Os Objetos fronteira comunicam apenas com actores e Objetos de controle.
- Os Objetos entidade comunicam apenas com Objetos de controle.
- Os Objetos de controle comunicam apenas com Objetos de fronteira e de entidade

Exemplo:

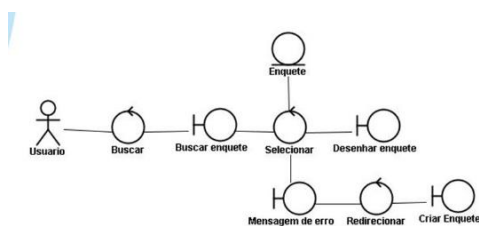
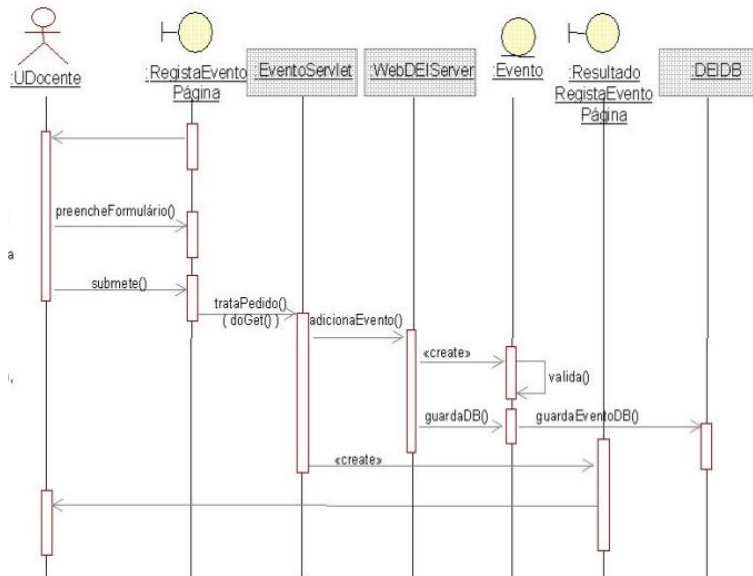
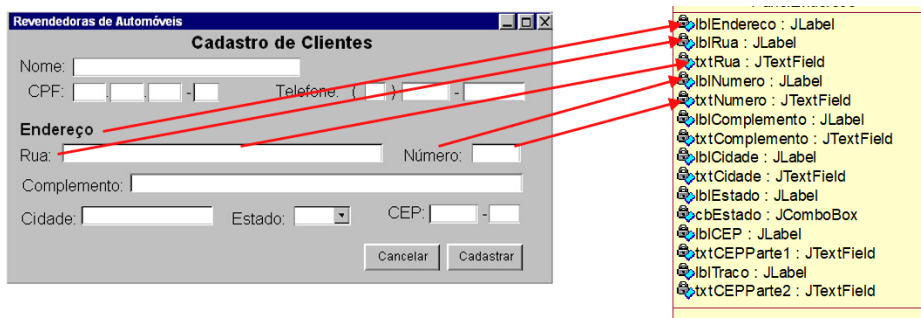


Diagram : Sequencia:



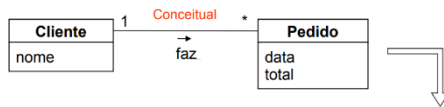
Projeto de Interface – Projeto detalhado(inclui tudo até aqui)

- Interface é uma área cinza com diversas possibilidades de comportamento
 - **Forma** que possibilita informação
 - **Estrutura** que possibilita interação
 - **Função** possibilita a experiencia
- Design:
 - Conformidade
 - Articulação
 - Referência
- Padrões de design → organização e estrutura
- Regras:
 - Cada tela é considerada uma classe
 - Criar um relacionamento de dependência entre a classe da tela e as classes de negócio que possuem os valores a serem exibidos
 - Agrupamentos podem ser considerados como uma classe
 - Analise as ações que o usuário pode fazer com a janela
 - Verifique quais deles precisam de especificar a ação
- Exemplo:

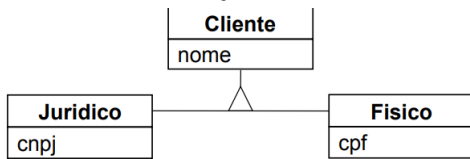


Projeto de Persistência:

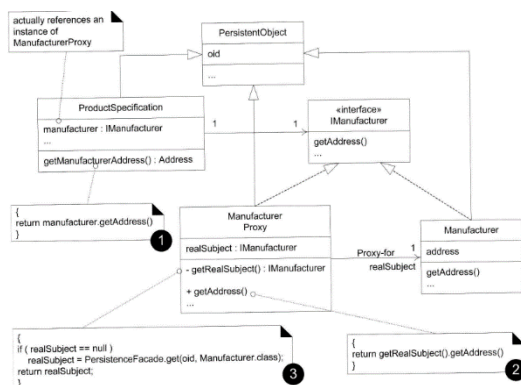
- Projeto do banco de dados:
 - Mapeamento OO/Relacional (MOR) ou Object-relational mapping(ORM)
 - Linguagem de Consulta
 - API de acesso aos dados
- Mapeamento Objeto Relacional(MOR):
- Regras:
 - Associações – losango no meio :



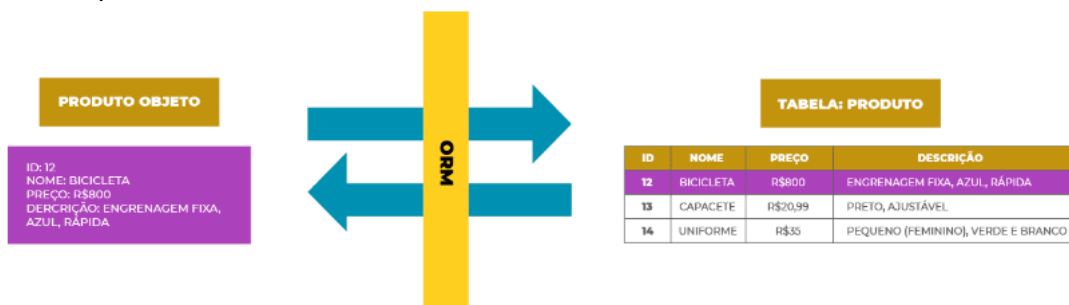
- Herança:



- Exemplo:



- Mapeamento ORM:



- Hibernate para java
 - Django para Python
 - Sequelize para JS
- Projeto Arquitetural (Arquitetura Lógica X Arquitetura Física)
- Arquitetura Lógica: especifica as propriedades funcionais do sistema (objetos de negócio)
- Arquitetura Física: aborda os aspectos não funcionais do sistema, como:
 - Segurança
 - Compatibilidade (Portabilidade)

- Ambiente de execução e acesso a recursos
- Física:
 - Visão de Componentes - Diagramas de Componentes
 - Visão de Concorrência - Diagramas de Implementação (Diagramas de Componentes; Diagramas de Execução)

Diagrama de Componentes:

Componentes de Código-fonte

Arquivos contendo código-fonte que implementam uma ou mais classes do sistema

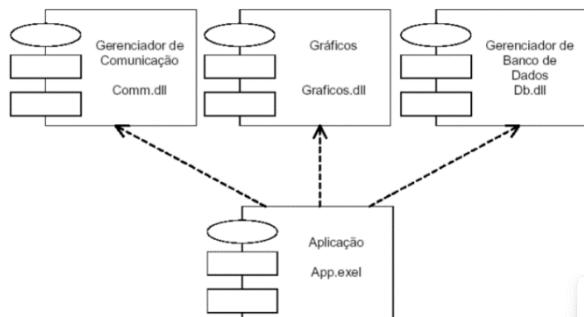
Componentes Binários

- Arquivos com código objeto, resultante da compilação de um ou mais componentes de código-fonte

Componentes Executáveis

- Arquivos de programa executável resultante da ligação dos componentes binários
- Representam unidades de software que podem ser executadas por um computador

Exemplo:



Exemplo de diagrama de componentes

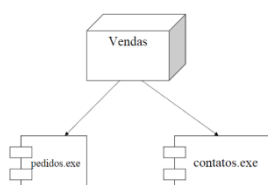
17/11

Diagramas de Execução:

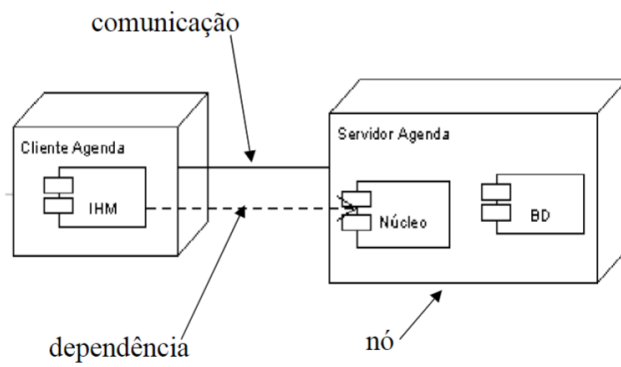
É composto por:

- Componentes de software
- Nós
- Conexões

Nós e componentes:



Exemplo:



Stack proposto:

8 Stack proposto:

Front-end: Framework Svelte (JavaScript/HTML)
sobre Node + astro (baseado em:)

Back-end: SGBD postgres SQL
com atualização das tab
+ API C# + Sequelize

Execução: Vercel sobre AWS