01/02/2024

* Histórico:
  + 1971 – Projeto Estruturado de Sistemas (Tom Marcos)
    - Top- down
    - Modular
    - Módulo caixa preta
    - Refinamento sucessivo
  + Devido a grande complexidade e tamanho do projeto estruturado ele caiu e parou de ser usado
  + 1986 – Projeto Orientado a Objetos
    - Rational Unified Process
  + 2006/2007 – Desenvolvimento ágil
    - Projeto de interface (front end)
    - Projeto de dados/persistência(back end)
* POO
  + RUP
  + Arquitetura logica
  + Arquitetura física
* Qualidade em projeto de software
  + Estimativas
    - Pontos de função
    - Pontos de casos de uso
    - Estimativas ágeis
    - Tamanho/prazo/ custo
  + Padrões em projeto
    - Criacionais
    - Estruturais
    - Comportamentais
  + Testes e CI/CD
    - Unitário
    - Integração
    - Sistema

06/02/2024 – Projeto de software

Projeto Orientado a Objeto:

* Projeto preliminar:
  + Especificações de requisitos 🡪 Casos de Uso, Classe/Pacote , Interface(revisão ES I)
  + Modelo de projeto 🡪 Diagrama de Robustez
* Engenharia de Software:
  + O que ? Arquitetura
  + Como? Engenharia- media , planejar , desenhar
* O projeto ágil é bem mais simples e rápido, entretanto não é tão detalhado como o POO, assim as vezes, tendo falhas de planejamento e erros no projeto

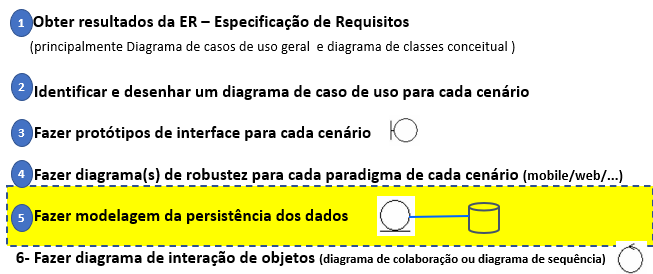
08/02/2024

* FOTO

20/02/2024 – Projeto de Interface

* 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
* Sai do pequeno para chegar no maior

Projeto de persistência

* Projeto do backend do projeto
* Passo a passo:
* Projeto do BD

Objetos transisente x persistente:

* Transiente 🡪 existem somente em memoria principal – Objetos de controle e fronteira
* Persistente 🡪 Tem uma existência que irá ser chamada varias vezes. Cliente , Conta,Pedido
* Estratégias de Persistência:
  + Vantagens:
    - Objetos do negócio podem ser reutilizados em aplicações que utilizam diferentes SGBD’s.
    - Objetos de domínio se tornam mais fáceis de entender, já que não implementam acesso a banco de dados.
    - Melhores manutenibilidade e no reusabilidade
  + Serialização • SGBD Orientado a Objetos • Frameworks MOR • Padrões (Data Source Architectural Patterns): Active Record, Row Data Gateway, Table Data Gateway, DAO, Data Mapper
* Estratégias de MOR:
  + A OO é a forma usual de desenvolver sistemas de software. • Por outro lado, os SGBD relacionais dominam o mercado
  + Problema:
    - O descasamento de informações (impedance mismatch) corresponde ao problema de mapear as representações do modelo de objetos e do modelo relacional.
* Componentes de um mecanismo de persistência:
  + Mapeamento OO / Relacional (MOR)
  + Linguagem de consulta
  + API de acesso a dados
* Framework MOR
  + Facilidade para permitir a definição do mapeamento de objetos para relações no banco de dados via arquivos de configuração.

21/03/2024 – Revisão prova

* Prova:
  + Focar no Design System
  + Diagrama de componente