MAS

1. O que é que o SDLC inclui?

O trabalho do analista na equipa de desenvolvimento

sistema ~~[illegible]~~ conjunto de componentes que interagem por forma a atingir uma finalidade. Tem um sistema interno, que transforma inputs em outputs.

exemplo | corpo humano — input: comida
partes processadoras: boca, estômago, ...
output: nutrientes

sistema de informação | conjunto de recursos interligados / relacionados (humanos ou tecnológicos) para satisfazer as necessidades de informação de uma organização e dos seus processos de negócio

exemplo | componentes: hardware, utilizadores, software, bases de dados e procedimentos

SDLC | System development lifecycle
↳ consiste em entender o PADI junto dos usuários de um SI e como este pode suportar as necessidades de um negócio

- Planeamento | porquê? como?
  - ↳ vai ser construído - determinar equipa
  - ↳ valor de negócio
- Análise | quem vai utilizar?
  o que vai fazer?
  onde e quando vai ser utilizado?
- Design | definição de hardware, interface, estrutura de rede
- Implementação | construção do SI (ou compra)
  transição para o novo ambiente

analista de sistemas
- analisar a situação do negócio
- identificar oportunidades de melhoria
- desenho de SI para as implementar

OpenUp (método ágil)

Processo de trabalho que procura definir o processo de engenharia de software

divide-se em 4 fases que por sua vez são compostas por 1 ou + iterações

- **Conceção** (inception) — definir objetivos e decidir se não executar (custo/benefício) — 1 iteração
- **Elaboração** (elaboration) — produzir e validar arquitetura; valor obtido e risco remanescente são aceitáveis para continuar — n iterações
- **Construção** (construction) — definir, desenhar, implementar e testar cenários sucessivos — >n iterações
- **Transição** (transition) — estabilização do que foi feito — <n iterações

O unified process / open up

processos sequenciais

abordagem "tradicional"
- fases sequenciais (seguinte só começa qnd anterior termina)
- em caso de erros é preciso repetir passos anteriores

vantagens: abordagem e passos simples, disciplina

problemas:
- confirmação tardia do controlo dos riscos
- integração e teste tardios
- pressupõe que requisitos são estáveis

principais características dos métodos ágeis

métodos ágeis

resposta: aceitar mudanças e adaptação como inevitáveis e fundamentais

como? práticas que equilibrem disciplina e feedback
princípios de desenho que favorecem construção flexível e evolutiva

objetivo: resposta rápida às alterações!

métodos ágeis
- abordagem...
  - incremental | ir juntando peças
  - iterativa | esboçar e ir aperfeiçoando
    - orientações em ciclos curtos, havendo em cada um um para de requisitos, desenho, implementação e teste

o dois métodos

ágil		vs	"tradicional"
indivíduos e interações		"	processos e ferramentas
software funcional		"	documentação abrangente
colaboração c/ o cliente		"	negociação contratual
responder à mudança		"	seguir um plano

o papel de modelação visual

modelos
importância
- ajudam a gerir a complexidade, permitindo visualizar um sistema como se pretende que venha a ser.
- especifica | a estrutura e o comportamento do SI antes de ser implementado
- referência → para a construção (como uma "planta")
- documenta → as decisões feitas

tipos de modelos
- funcionais | O que o sistema faz?
  - comportamento externo do sistema
- estrutural → quais são as partes do sistema?
- comportamental → quais as operações que o sistema faz?

modelação visual
vantagens
- comunicação clara e sucinta
- mantém o desenho (planeamento)
- mostra/esconde níveis de detalhe (conforme pretendido)

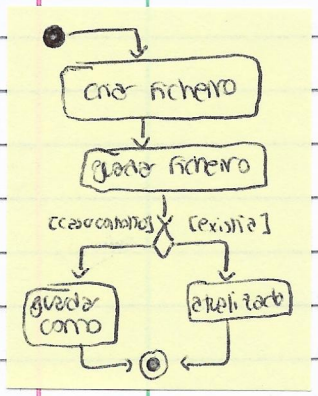
UML

diagramas de atividades

comportamental

mostram fluxo de ações

quando? modelar fluxos de trabalho / negócio
descrever algoritmos
descrever sequência de iterações entre ators e sistema

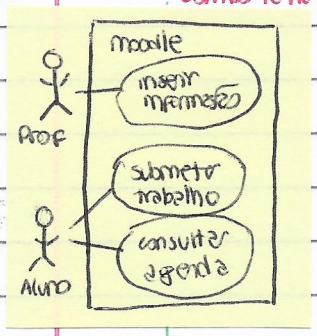


- initial node
- final node
- atividade
- decisão

casos de utilização

comportamental

captem o que o ator faz (interage) com o sistema e com que fim (objetivo)



ator qualquer entidade externa que interage com o sistema
cenário situação particular de utilização do sistema
caso de utilização conjunto de cenários c/ um objetivo

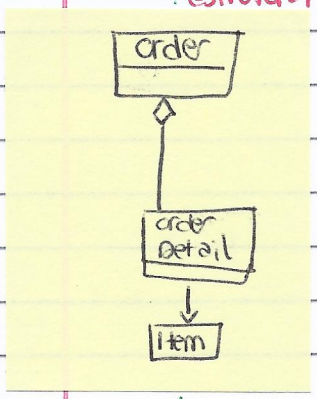
fornece os requisitos do sistema (identifica-los)

CaU descreve diálogo entre o ator e sistema
↳ narrativa
sequência de ações de um ator particular
face menor grão de atividade possível
conjunto de cenários relacionados c/ um objetivo

classes

estrutural

objeto tem um propósito, estado e comportamento
classe categoria de objetos semelhantes, que partilham atributos, operações, etc

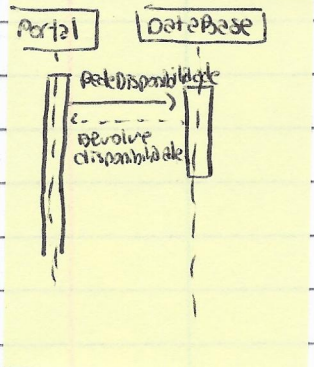


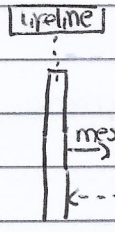
- classe c/ atributos
- esq é parte da dir — agregação
- esq é composta pela dir — composição
- esq depende da dir — dependência
- associação

sequências

comportamental

modelam interações e detalham como estas se processam

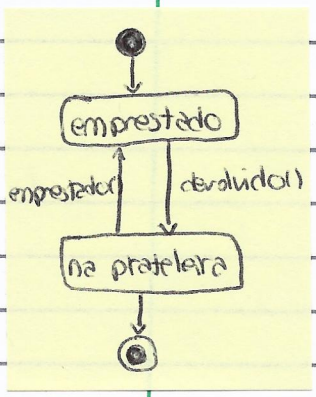




lifeline representa um participante individual na interação
ativação período durante o qual um elemento executa 1 operação
define comunicação particular entre 2 lifelines

estado
comportamental

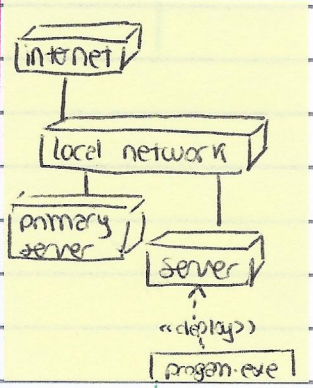
descrevem comportamentos de um objeto dependendo do seu estado



- estado inicial
- estado final
- estado
- → transições

implementações
estrutural

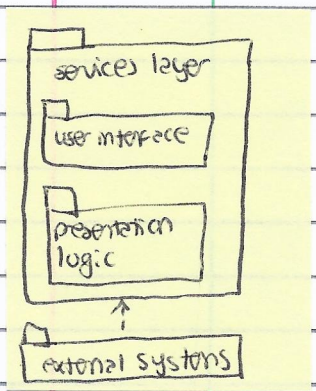
modelações os aspetos físicos de um sistema orientado a objetos



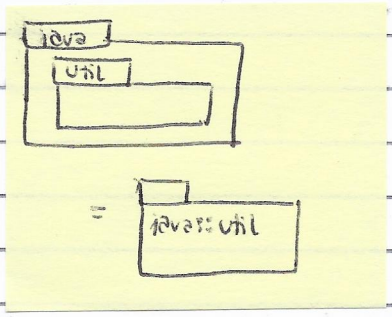
- node
- — associações

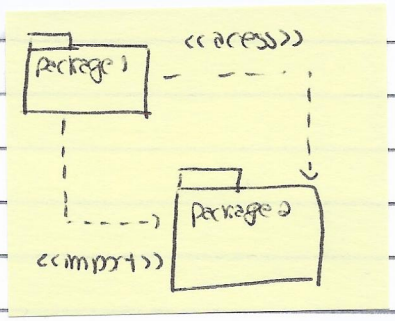
pacotes
estrutural

modelações de elementos, em média a gente esala simplifica diagramas de classes



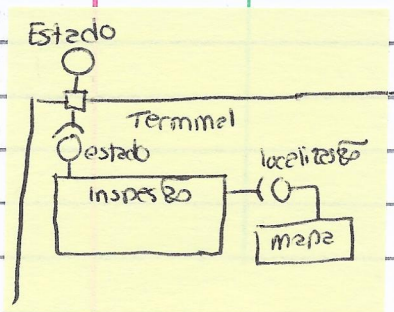
- pacote





componentes
estrutural

detalha interações entre vários componentes



- componente
- porta
- interface fornecida
- interface necessária