**DIAGRAMA DE SEQUENCIA(WARNER)**

-Se baseia no de caso de uso

* Linha de vida
  + representa objeto ou ator, participante da realização do caso de uso em um período de tempo.
    - Representq toda linha de tempo de todas as funções da aplicação, o diagrama de sequencia representa essa relação dentro de uma sequencia lógica. Essa é a representação que devemos apresentar.
  + Tempo de existencia do processo, termina com a destruição dele.
* Ator
  + Atua no sistema, faz entrada, interage com a aplicação. Apoiados por uma linha de vida, enviam mensagens para os objetos
  + Job é representado por uma classe, no caso, uma função autômoma, acionada por gatilho.
* Stakeholder
  + Interessado, não necessariamente um ator
* OBJETO
  + Representa os dados que participam da realização do caso de uso.
* Foco de controle
  + Tem o ator, o objeto, dentro desse foco de controle está acontecendo aquela ação dentro da aplicação. Pode ser redimensionado de acordo com as interações com os outros elementos.
* Mensagem síncrona seta fechada
  + Mando uma pergunta pro ator, não falo mais nada, enquanto ele não devolve, não dou mais continuidade.
* Mensagem assíncrona seta aberta
  + Conseguimos ir mandando mensagem independente da resposta do ator ou classe.
* Em engenharia de software, tudo é subjetivo, tudo vem de acordo com a necessidade.
* Mensagem de retorno seta tracejada
  + É uma mensagem devolvida por um objeto a outro, depois da mensagem enviada, a resposta.
* Mensagem reflexiva
  + Mesma classe se comunica com ela
* Mensagem construtora
  + Quando estou instanciando um objeto, faço isso no diagrama de sequencia.
* Mensagem destrutora
  + Destruição do objeto no decorrer, quer dizer, o objeto não é mais necessário, então eu mato ele.
* Quadro de interação
  + Facilita, quando criamos outras subfunções dentro do diagrama de sequencia, evitando que o diagrama fique muito grande, isolando um diagrama de sequencia com contexto específico.
* Nota ou comentário
* Ver exemplo do Warner
* Opt
  + Opcional, ex, opt em manter cliente, pode ser que seja salvo ou não.
* Alt
  + Multiplos fluxos alternativos, onde só um será executado.
* Não ter a visão de usuário, mas de analista, para entendermos a forma como as comunicações acontecerão.
* Muitas vezes tem a rixa entre quem projeta e quem desenvolve.
* Só vou saber que comunicação terei quando souber que aplicação estou desenvolvendo.
* Restrições de tempo
  + Temporizador.
* Mensagem perdida
  + - Não chegou ao destino ou não está no diagrama.
  + Encontrada
    - Remetente desconhecido ou não representado no diagrama.
  + Forma de tratar excessões nas comunicações entre os objetos.
* WorkFlow
  + Facilita entendimento entre condições funcionais de processos.
* Portas
  + Objeto poderá ter mais de uma linha de vida, cada uma representando uma poprta de comunicação.
* Fragmento de interação
  + Minimizar ou organizar melhor o diagrama.
  + É o ref, Opt ou Alt.
* Fragmentos combinados
  + Ref
    - Representa uma chamada ou diagrama de sequencia
      * Indica que a sequencia de mensagens é representada em outro diagrama.
  + Opt
    - Pode ser executado ou não.
  + Alt
    - Multiplos fluxos alternativos, só um vai ser executado.
      * Tipo If Else.
  + Par
    - Ações que acontecem de forma paralela.
      * A partir de um gatilho podem ocorrer múltiplas ações.
  + Critical
    - Apenas uma execução pode ocorrer por vez
      * Ex. Numa aplicação estou editando um registro de cliente, ao mesmo tempo outra pessoa não pode fazer isso ao mesmo tempo, porque senão sobrepõe.
      * Banco de dados trabalha com concorrência(critical), ele ignora coisas que vem em sequencia. Isso garante integridade da informação.
  + Loop
    - Tipo de rotina de loop
  + Break
    - Interrompe a execução quando uma execução é atendida.
    - Todo loop deve parar, senão é eterno.
  + Assertion
    - Condição verdadeira para que o sistema continue funcionando
  + Ignore
    - Certas mensagens podem ser ignoradas
  + Negative
    - Comportamento que não deve ocorrer no sistema.
    - ex: Não deixar acontecer pagamento duplicado.
  + Consider
    - Especifica as mensagens que devem ser especificadas, ignorando as outras.
  + Seq
    - As mensagens devem ser processadas uma após a outra
  + Strict
    - Se fugir da ordem de processamento pré definida, é entendida como excessão.
* Tem exercício.
  + Construir o diagrama