

Módulo

Nome do módulo / Descrição / Prioridade (Must have/Should have/Could have/ Won't have)

Requisitos Funcionais

ID / Módulo a que pertence / Descrição (O sistema deverá permitir que o utilizador...)/ Prioridade

Requisitos de Qualidade

ID/ Categoria / Tipo / Descrição (O sistema deverá funcionar...)/ Prioridade

ISO 9126

- **Portabilidade**
 - **Adaptabilidade** (Ex: O sistema deve funcionar em qualquer tipo de sistema operativo);
 - **Instalabilidade** (Ex: As funcionalidades bases do sistema devem de estar disponíveis sem ser necessária configuração previa);
 - **Conformidade** (Ex: O sistema deve funcionar segundo a regulamentação da portabilidade europeia);
 - **Substituibilidade** (Ex: O Sistema deve permitir resolver o problema de gestão de equipas de *esports* melhor que as ferramentas atuais).
- **Funcionalidade**
 - **Adequação** (Ex: O sistema deve garantir que o utilizador tenha uma ferramenta de calendarização);
 - **Precisão** (Ex: O sistema deve garantir que a funcionalidade de procura de *scrims* automática encontre a equipa mais adequada);
 - **Interoperabilidade** (Ex: O Sistema deve permitir ver o perfil de equipas através de uma API);
 - **Segurança** (Ex: O sistema deve encriptar todos os dados sensíveis pertencentes às equipas e utilizadores);
 - **Conformidade de usabilidade** (Ex: O sistema deve garantir a segurança na gestão de dados do cliente, segundo o regulamento SOC).
- **Manutenção**
 - **Capacidade de análise** (Ex: O sistema deve garantir que todas as alterações dos dados do sistema fiquem registadas na base dados);
 - **Mutabilidade** (Ex: O sistema deve garantir que o código do software possa ser modificado facilmente);
 - **Testabilidade** (Ex: O sistema deve permitir verificar onde ocorreram erros no *front-end* através da consola do browser);
 - **Estabilidade** (Ex: O sistema deve garantir a responsividade em horas de maior afluência).
- **Confiabilidade**
 - **Maturidade** (Ex: O sistema deve verificar a introdução de caracteres inválidos nos respetivos campos);
 - **Tolerância ao erro** (Ex: O sistema deve se manter estável mesmo que os utilizadores introduzam informação ou tipos de dados errados nos campos de "input" da aplicação);

- **Recuperabilidade** (Ex: O sistema deve permitir que o utilizador consiga recuperar a sua conta caso se tenha esquecido da password).
- **Eficiência**
 - **Comportamento do tempo** (Ex: O sistema deve garantir que o *login* demore menos de 15 segundos a ser concluído);
 - **Utilização de recursos** (Ex: O sistema deve apagar as variáveis temporárias utilizadas no browser após a utilização das mesmas).
- **Usabilidade**
 - **Apreensibilidade** (Ex: O sistema deve garantir que os utilizadores aprendam a utilizar o sistema de forma rápida);
 - **Inteligibilidade** (Ex: O sistema deve garantir que o utilizador rapidamente compreenda as capacidades do sistema);
 - **Operabilidade** (Ex: O sistema deve garantir que o utilizador consiga alterar que campos estão visíveis no seu perfil).

RQ33	Descrição
Pior caso:	O login demorar mais de 1 minuto a ser concluído
Planeado:	O login demorar menos de 15 segundos a ser concluído
Teste:	Colocar um utilizador, registado no sistema, a fazer o <i>login</i>
Unidades:	Quantidade de tempo

Requisitos Ambientais

ID/ Categoria/Descrição

Categorias: **Hardware** (PC), **Software** (Qualquer sistema que suporte o programa), Linguagem (JAVA), **Browser** (Qualquer browser lançado depois de 2012), **Ambiente de desenvolvimento** (Qualquer programa que dê para desenvolver JAVA, etc.), **Necessário para o cliente correr a aplicação** (Qualquer sistema que suporte o programa) e **Normas/Legislação a seguir** (Utilização do padrão de desenvolvimento de software MVC (Model-view-Controller)).

Modelos e metodologias

Etapas típicas do processo:

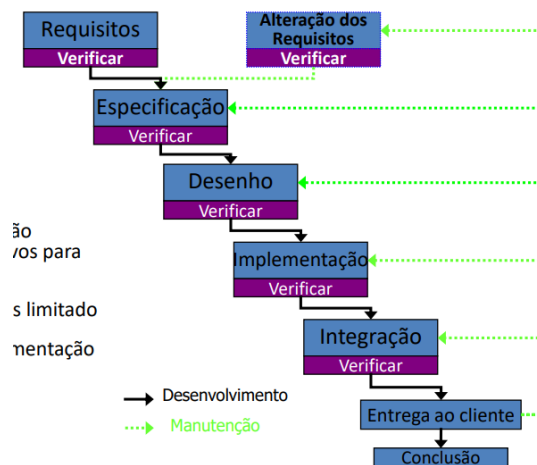
- **Análise e Especificação** (Levantamento de requisitos, descrição da solução e descrição rigorosa das funcionalidades do software a ser desenvolvido);
- **Desenho** (Representação abstrata do sistema e descrição detalhada do funcionamento e estrutura);
- **Implementação** (Pode incluir configuração e integração de componentes e recorrer a normas, convenções e padrões);
- **Documentação** (Suporte a futura manutenção e deve incidir em todos os passos do processo);
- **Testes e Integração** (Avaliação da integração em ambiente de produção e testes de sistema e de aceitação);

- **Manutenção** (Monitorização de problemas e definição de novos requisitos).

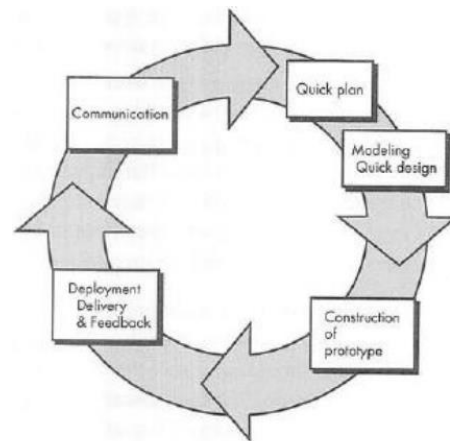
Metodologia tradicional

Modelo	Pontos Fortes	Pontos Fracos
Cascata	<ul style="list-style-type: none"> - Segurança na passagem entre fases - Linear - Controlo - Documentação 	<ul style="list-style-type: none"> - Rígido - Risco de insatisfação final
Prototipagem	<ul style="list-style-type: none"> - Permite ao cliente visualizar a ideia relativa ao produto final - Revisitar com facilidade fases anteriores - Assegura que os requisitos do cliente são satisfeitos: Construtivista 	<ul style="list-style-type: none"> - Pode ser complicado definir os protótipos - Custo e tempo associados
Espiral	<ul style="list-style-type: none"> - Ênfase na análise de riscos - Incorpora actividades dos restantes modelos - Construtivista 	<ul style="list-style-type: none"> - Adequado para projectos maiores (e longos!) - ... e com equipas experientes - Complexo de usar
Incremental	<ul style="list-style-type: none"> - Vai de encontro as necessidades do cliente progressivamente - Ciclos + rápidos com entregas operacionais - Divide complexidade (Modular/Escalavel) 	<ul style="list-style-type: none"> - Varias versões (controlo!!) - Obriga a uma formação continua dos utilizadores - Se mal usada pode degenerar em "construir e modificar"

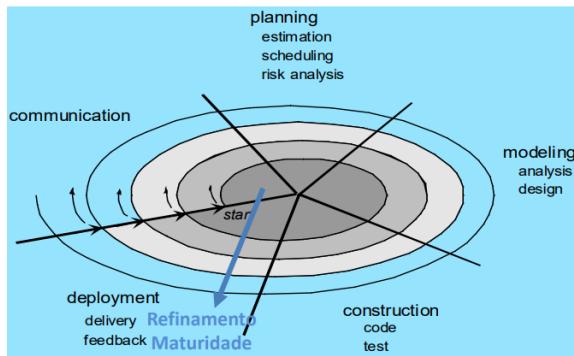
Cascata:



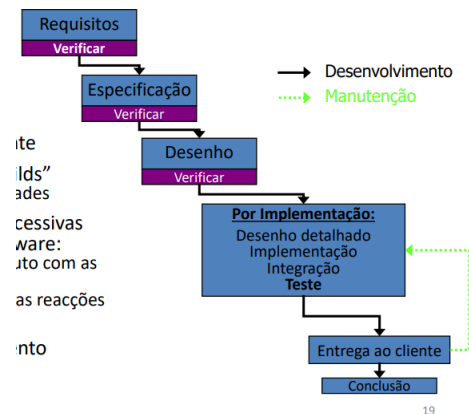
Prototipagem:



Espiral:



Incremental:



Metodologia ágil

SCRUM

Uma equipa que trabalha como uma unidade altamente integrada com cada membro desempenhando um papel bem definido e toda a equipa focando-se num único objetivo comum.

O SCRUM define um processo para construir software incrementalmente em contextos complexos, onde os requisitos não são claros ou mudam com muita frequência.

O SCRUM é um *framework* ágil de gestão de projetos usado para entregar aos clientes, de forma iterativa, incrementos de produto de alto valor.

O trabalho a realizar é registado no **Product Backlog**, que é uma lista de tarefas (incluindo associadas à implementação dos requisitos/funcionalidades) ou de user stories.

O projeto desenvolve-se numa série de iterações, chamadas incrementos (**sprints**).

No início de cada incremento é feita uma Reunião de Planeamento de Incremento (**Sprint Planning Meeting**) na qual o Dono do Produto (**Product Owner**) define as prioridades do **Product Backlog**.

A Equipa (**Scrum Team**) seleciona as tarefas que completará durante o próximo Incremento

Essas tarefas são então transferidas do **Product Backlog** para o **Sprint Backlog**.

Durante o incremento, são conduzidas curtas reuniões diárias chamadas de Scrum Diário (**Daily Scrum**), que ajudam a equipa a sincronizar-se e manter-se coesa e direcionada.

Ao final de cada incremento a equipa demonstra a funcionalidade concluída, na **Sprint Review Meeting**.

Tipicamente:

- O desenvolvimento é dividido em Sprints de 30 dias;
- Equipas são pequenas;

- O **Daily Scrum** segue o formato de uma reunião de 15 minutos onde a equipa reporta sobre o que fez e expõe o que será feito no próximo dia e identifica os potenciais fatores de impedimento ao progresso.

KANBAN

Um sistema de administração da produção que determina que deve ser feito somente o imprescindível para realização da etapa seguinte do processo, num fluxo de trabalho contínuo. Noutras palavras: **fazer apenas o que é necessário, quando necessário e na quantidade necessária.**

A metodologia Kanban propõe a utilização de cartões ou post-its num quadro para indicar e acompanhar, de maneira visual, prática e utilizando poucos recursos, o andamento dos fluxos de produção nas empresas. De um lado do quadro, ficam as tarefas que precisam ser executadas, o que pode ser chamado de “**Backlog**”. E, do outro, as etapas de execução: em andamento e entregue. Pode-se alterar o nome dessas etapas de acordo com seus processos internos. Conforme as tarefas são desempenhadas, o cartão ou post-it é colocado no campo correspondente ao status da tarefa.

Scrum vs Kanban

Kanban trata de visualizar seu trabalho, limitando o trabalho em andamento e maximizando a eficiência (ou fluxo). As equipas Kanban concentram-se em reduzir o tempo que leva para levar um projeto (ou história de usuário) do início ao fim. Eles fazem isso utilizando um quadro kanban e melhorando continuamente o seu fluxo de trabalho.

As equipas **Scrum** comprometem-se a enviar software funcional por meio de intervalos definidos chamados **sprints**. O seu objetivo é criar ciclos de aprendizagem para reunir e integrar rapidamente o feedback do cliente. As equipas Scrum adotam funções específicas, criam artefactos especiais e realizam cerimónias regulares para manter as coisas a avançar.

	Scrum	Kanban
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint	Continuous delivery
Roles	Product owner, scrum master, development team	No required roles
Key metrics	Velocity	Lead time, cycle time, WIP
Change philosophy	Teams should not make changes during the sprint.	Change can happen at any time

XP

O XP agrega uma coleção de práticas do desenvolvimento de software. As práticas recomendadas pela XP não são novas individualmente. A contribuição emerge na conjugação e no ênfase extremo que se coloca na sua utilização

Objetiva software de alta qualidade

- Redução de bugs
- Adaptável, modular e escalável

XP – Processo

- Integração contínua dos incrementos
- Reuniões em pé (stand up meetings: SUM)
- Ênfase no combate ao erro (abugging)
- “Releases curtas”

XP – Principais Práticas

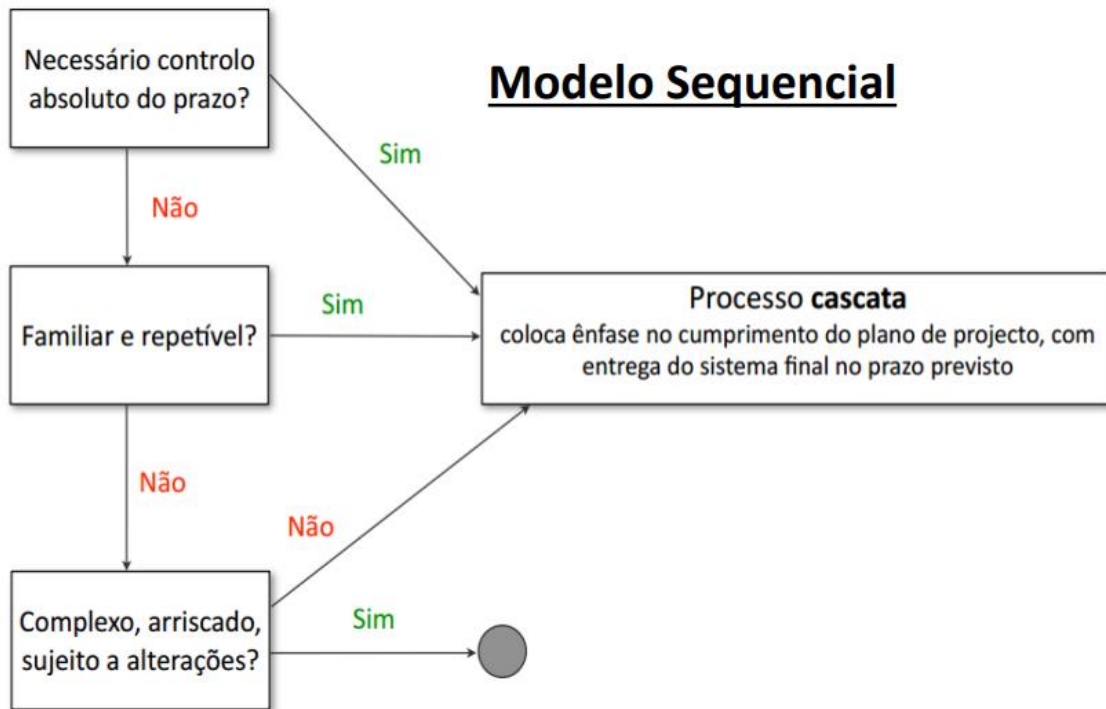
- Spiking
- Desenvolvimento orientado por testes (TDD)
- Refactoring
- Pair Programming

Seleção da Metodologia

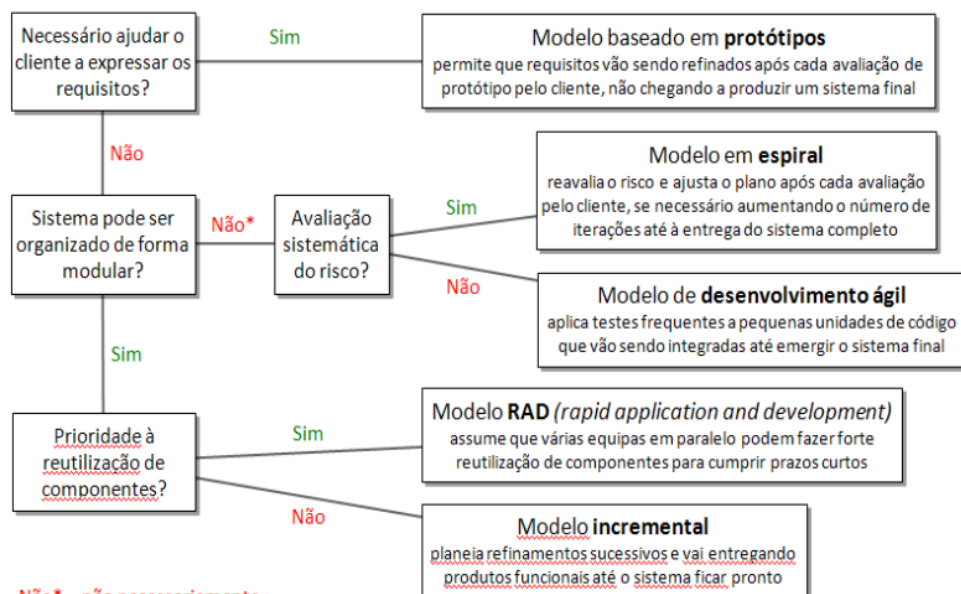
	Sequencial	Iterativa
Requisitos	Estáveis	Instáveis ou pouco entendidos
Desenho	Familiar e repetível	Complexo e desafiante
Equipa	Conhece o contexto	Estranha ao contexto
Risco	Reduzido	Elevado
Previsibilidade	Importante	Negligenciável
Alterações	Caras	Baratas

Modelo **sequencial linear**

coloca ênfase no cumprimento do plano de projeto, com entrega do sistema final no prazo previsto

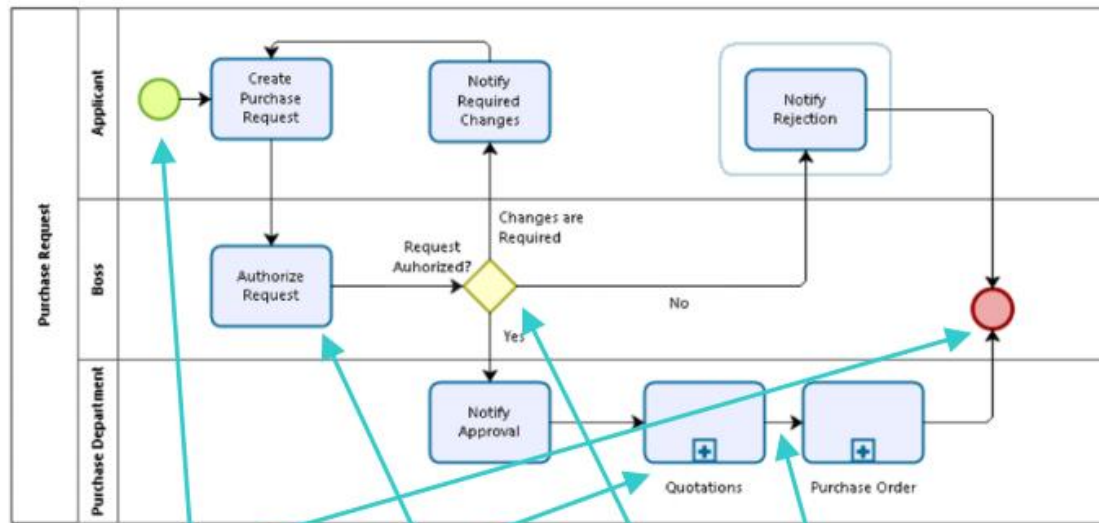


Modelos Iterativos



Não* = não necessariamente

BPDS



Eventos Tasks Gateway Sequence flows

Eventos

Três tipos:

- Start event;
- Intermediate event;
- End event.



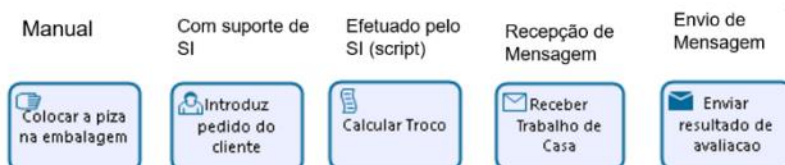
Atividades

Dois tipos diferentes:

- Tarefas;
- Subprocessos.



Tarefas



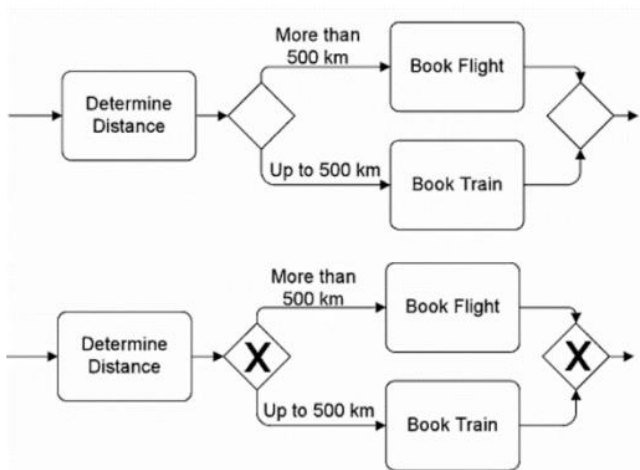
Gateway

Exemplos:

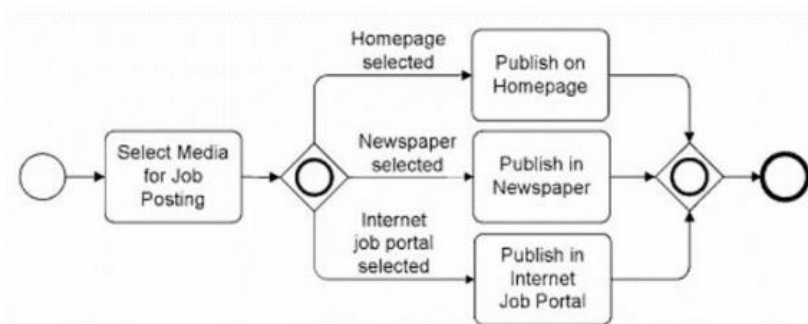
- Desvio exclusivo (XOR);
- Desvio inclusivo (OR);
- Desvio paralelo (AND);
- Desvio complexo;
- Desvios baseados em evento:
 - Desvio condicionado por evento (apenas usado para divergência e para eventos);
 - Desvio exclusivo baseado em eventos;
 - Desvio paralelo baseado em eventos.



Desvio exclusivo (XOR)



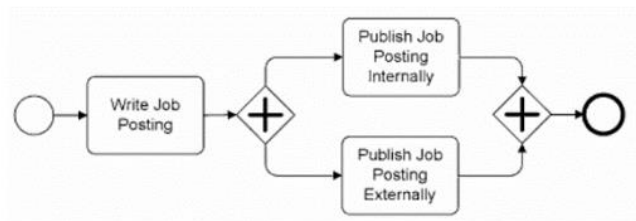
Desvio inclusivo (OR)



Podem ser selecionados um ou mais fluxos. Um desvio inclusivo aguarda por todos os “tokens” que foram iniciados.

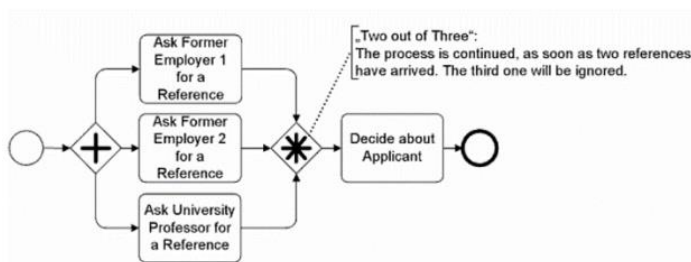
Exemplo: A entrega do cliente apenas é feita quando todos os fluxos que foram iniciados terminem.

Desvio paralelo (AND)

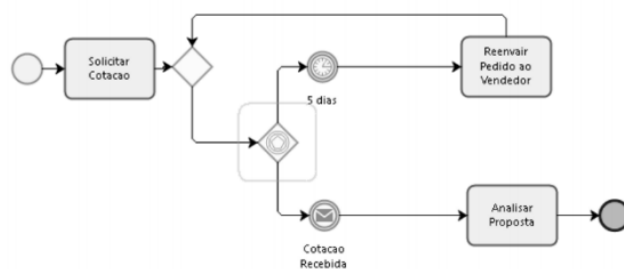
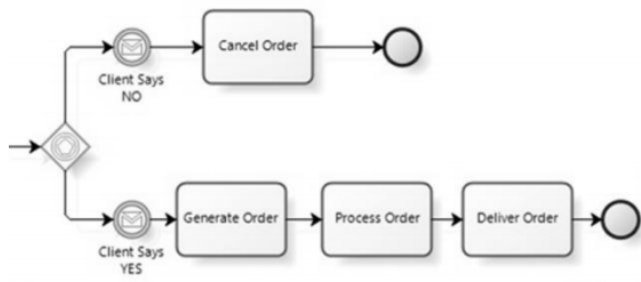


O processo só continua quando ambos os “tokens” tenham chegado.

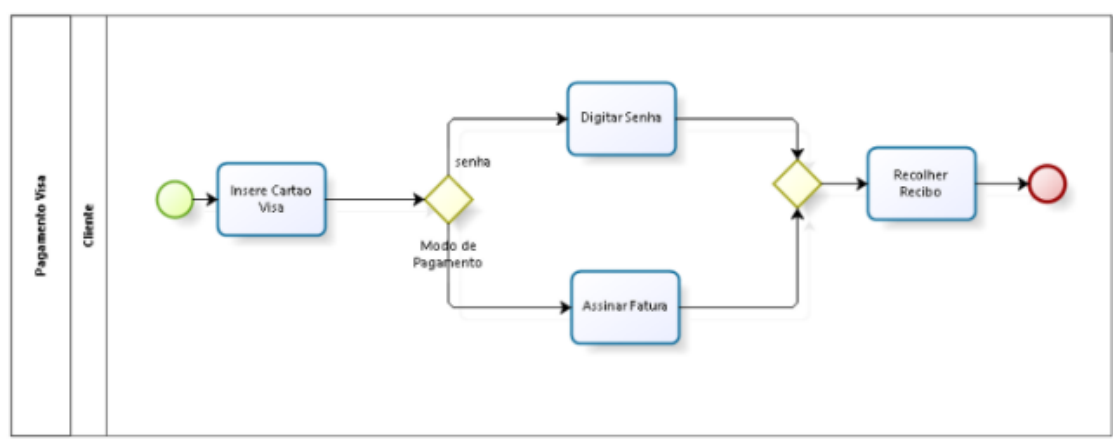
Desvios complexos



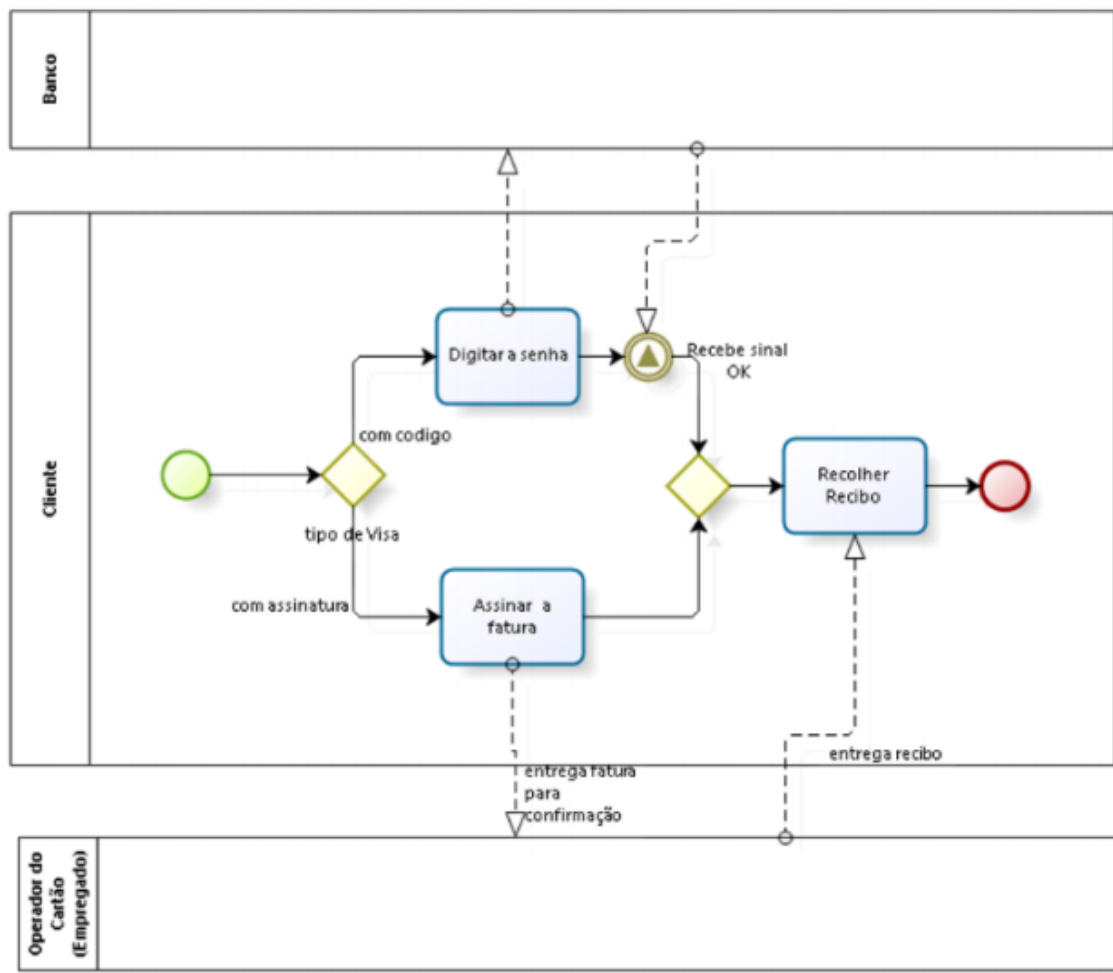
Desvios por eventos



Private (internal) business process



Abstract (public) process



Collaboration (global) process

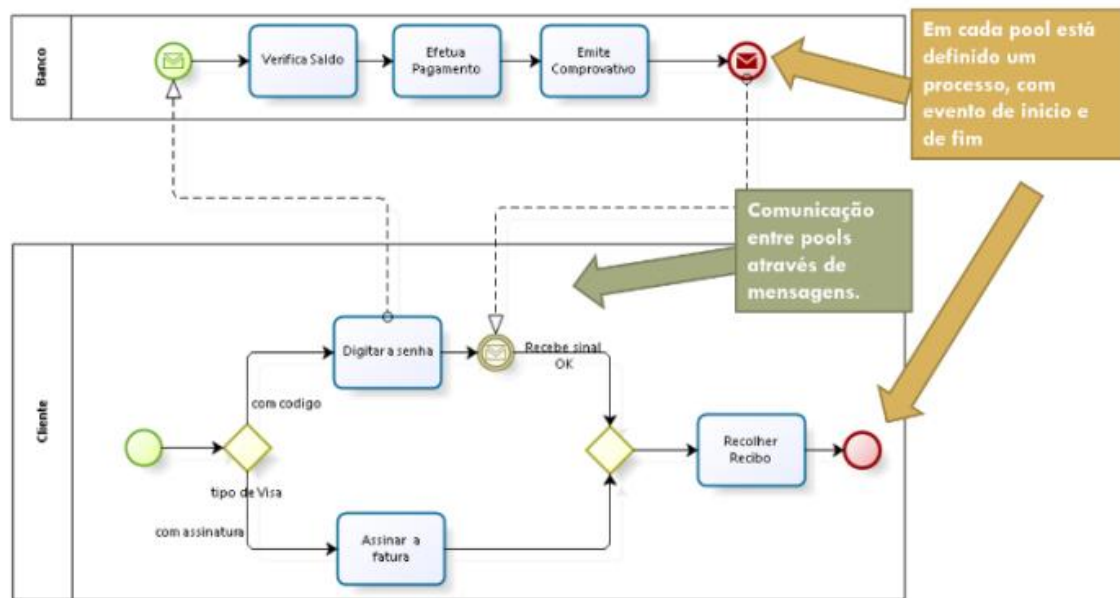


Diagrama de classes

DIAGRAMA DE CLASSES

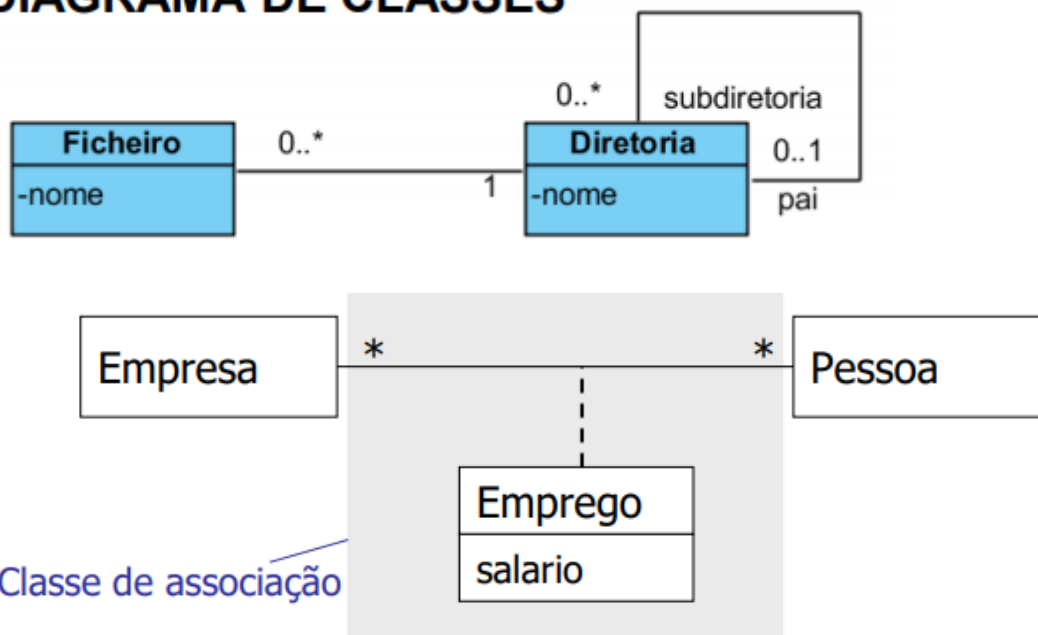


Diagrama de objetos

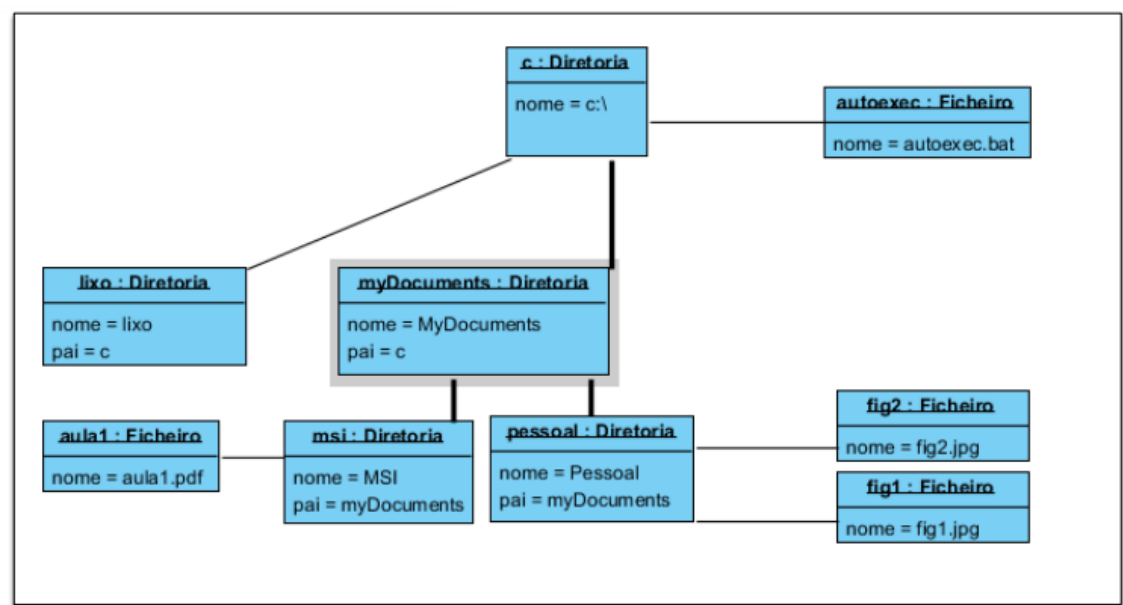


Diagrama de pacotes

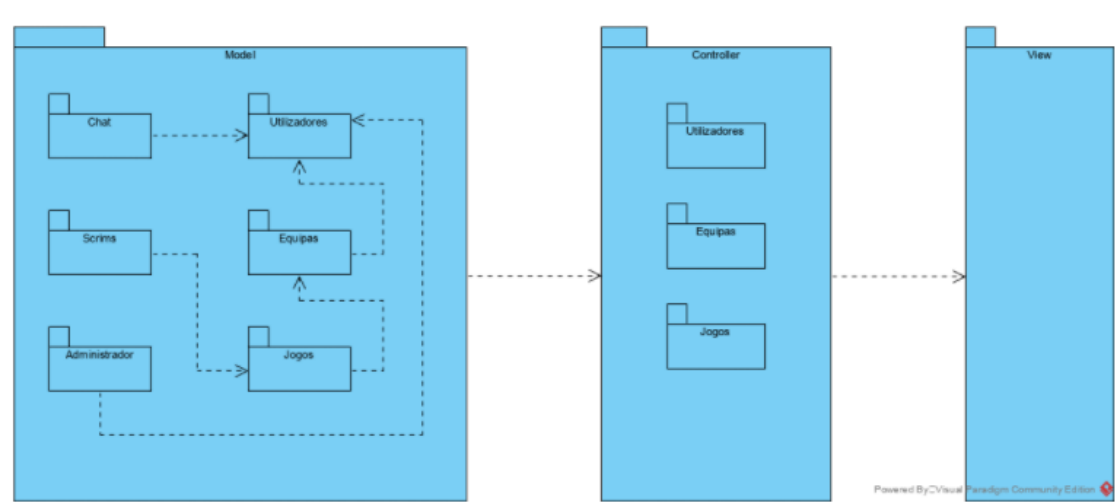


Diagrama de componentes

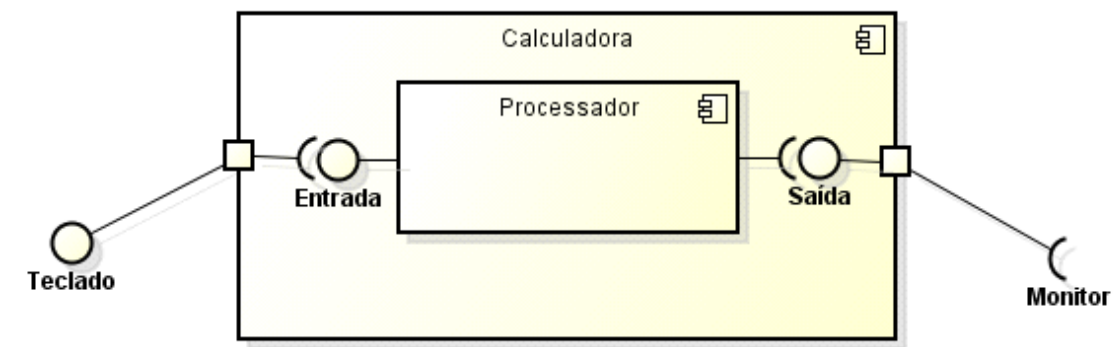
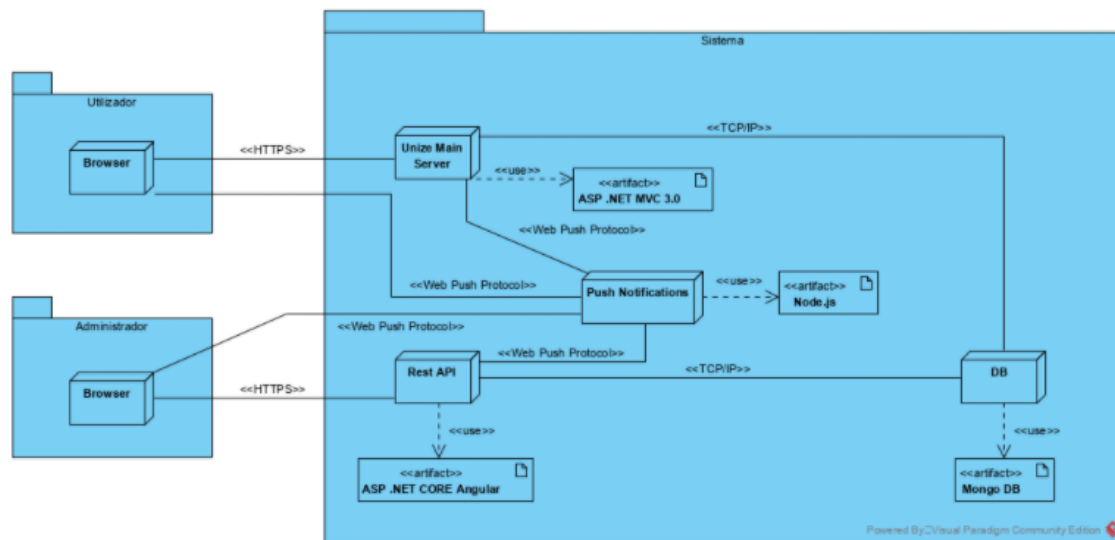


Diagrama de instalação



Exercício

“A maioria dos projetos têm tendência a deslizar”

Verdadeiro, a imprevistos na implementação da aplicação ou por exemplo desvios acidentais devido a comunicação falhada ou imperfeita entre um cliente e um *project manager* ou mau planeamento de tarefas entre a equipa e/ou cliente.