



Análise e Desenvolvimento de Sistemas

ENGENHARIA DE SOFTWARE ÁGIL

RELATÓRIO DE AULAS PRÁTICAS

Nome: Vitória Freitas

RA: 2523279

Polo de matrícula: Carapicuíba - Vila Dirce

Local da realização da Aula Prática: Santana de Parnaíba

Ano da postagem: 2025

Instituto de Ciências Exatas e Tecnologia	Disciplina: Engenharia de Software Ágil Título da Aula: Engenharia de Software	Relatório 1
--	---	--------------------

1. Resumo Teórico

O que é engenharia de Software? É uma área que trabalha com a tecnologia, utilizando seus conhecimentos de engenharia, arquitetura e desenvolvimento para criação, projeção e gerenciamento de sistemas de diversos tipos para diversas áreas do mercado de trabalho.

Diferença de análise de software e de um sistema de software. Análise de software é buscar e entender as necessidades de um cliente em relação a construção de um sistema, o que ele quer que o sistema faça e a performance dele. Um sistema de software é um programa, feito para executar o hardware e os aplicativos do computador, é como se fosse a interface entre o hardware e os aplicativos do usuário, fornecendo uma plataforma para que o software possa ser executado. Um exemplo de sistema de software é o Sistema Operacional, como Windows ou Linux.

2. Estudo de Caso

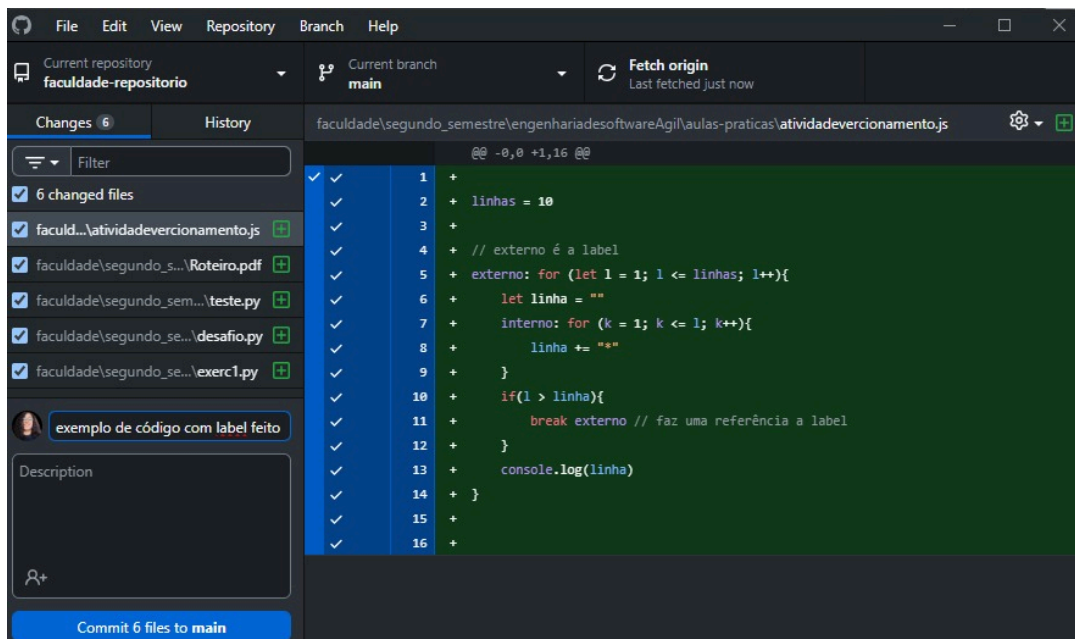
O Linux, é um Sistema Operacional criado por Linus Torvalds em 1991. Hoje está presente em diversos tipos de sistemas e dispositivos. Ele é open source (dar o acesso para pessoas fazerem mudanças e estudar o seu código-fonte). Foi desenvolvido para ser parecido com o Unix. Todo Linux possui o seu Kernel, que tem como função fazer o gerenciamento dos recursos de hardware.

Mudanças que ocorreram durante sua história foram a criação de 321 distribuições, que hoje são monitoradas pelo DistroWatch, como Mint, Ubuntu e Fedora. Tendo uma grande importância na computação nos dias de hoje, grandes servidores rodam em Sistema Operacional Linux.

3. Diagrama do versionamento

Utilizei o GitHub Desktop que é um aplicativo de código aberto, que facilita o trabalho com códigos hospedados no GitHub, e outros serviços de Hospedagem Git. Abaixo está uma

imagem de um commit dentro do GitHub Desktop



Labels:

```
aulas-praticas > JS atividadevercionamento.js > ...
1
2  linhas = 10
3
4  // externo é a label
5  externo: for (let l = 1; l <= linhas; l++){
6    let linha = ""
7    interno: for (k = 1; k <= l; k++){
8      linha += "*"
9    }
10   if(l > linha){
11     break externo // faz uma referência a label
12   }
13   console.log(linha)
14 }
15
16
17
```

4. Reflexões Finais

A importância da engenharia de software hoje, dentro da minha perspectiva, está na construção de diversos tipos sistemas, tendo seu funcionamento funcional e com boas performances, sendo muito bem planejados. As principais reclamações de usuários podem estar relacionadas a sua velocidade de resposta, por exemplo quando um site demora muito para mostrar os produtos em seu catálogo, e quando o sistema não possui uma interface intuitiva, deixando o usuário perdido na hora de utilizar os serviços oferecidos pelo sistema.

Nessa aula aprendemos sobre a engenharia de software, sobre sistemas de software, análises que têm que serem feitas antes da construção de um sistema.

5.Referências

<https://faculdade.grancursosonline.com.br/blog/engenharia-de-software/#o-que>

<https://blog.casadodesenvolvedor.com.br/analise-de-requisitos-de-software/>

<https://www.redhat.com/pt-br/topics/linux/what-is-linux>

<https://www.techtudo.com.br/noticias/2012/05/a-evolucao-do-linux.ghm>

<https://docs.github.com/pt/desktop/overview/getting-started-with-github-desktop>

Instituto de Ciências Exatas e Tecnologia	Disciplina: Engenharia de Software Ágil Título da Aula: Engenharia de Requisitos	Relatório 2
--	---	--------------------

1. Resumo Teórico

Engenharia de requisitos, disciplina que vai definir, documentar e manter os requisitos de um sistema de software. Entender as necessidades do cliente para a construção de um sistema com uma boa qualidade.

RS engloba todos os requisitos do sistema, RU são os requisitos do usuário, os desejos dele, como sistema deve ser, o que ele irá fazer, suas funcionalidades, design. Depois vão virar RF e RNF. RF são os requisitos funcionais, o que o sistema deve fazer, por exemplo o sistema deve ter um painel administrativo, deve mostrar o estoque. RNF são critérios que qualificam os requisitos funcionais, como performance, usabilidades, robustez, segurança.

Esses requisitos são extremamente importantes para o desenvolvimento de sistemas em toda área da TI.

2. Estudo de Caso

Sistema de software ERP, um sistema que tem como objetivo a otimização dos principais processos de negócio de uma empresa, como finanças, RH, produção, cadeia de suprimentos e vendas.

Nas reuniões com os clientes e desenvolvedores foram definidos os requisitos funcionais e não funcionais, abaixo estão eles.

Requisitos funcionais: o sistema deve ter uma área exclusiva para finanças, outra para o RH, uma para mostrar o estoque disponível e as vendas da empresa.

Requisitos não funcionais: o sistema terá criptografia para segurança dos dados, tem que ter uma boa velocidade para mostrar os produtos, no máximo 1 segundo de espera.

Ferramenta para o desenvolvimento, backend vai ser desenvolvido utilizando a

linguagem de programação Java, por sua robustez e segurança, framework Spring Boot. Front end irá usar React Js, para facilitar a navegação entre páginas, banco de dados o MySql.

3. Diagrama de casos de uso

Logo abaixo estão diagrama de caso de uso e uma tabela dos requisitos que foram feitos em grupo.



(Créditos: autoria própria)

Código	Tipo	Descrição	Prioridade	Status
RU01	Usuário	usuário deve acessar o sistema com login e senha	Alta	Em análise
RU02	Funcional	sistema deve registrar vendas e atualizar o estoque automaticamente	Alta	Validado
RU03	Não Funcional	O sistema deve processar consultas em até 2 segundos	Média	Validado

RU04	Sistema	O sistema deve armazenar dados em banco de dados relacional (MySQL)	Alta	Em Análise
------	---------	---	------	------------

4. Diagrama dos requisitos com mapa mental



(Créditos: autoria própria)

5. Reflexões finais

A importância de casos de uso e mapa mental, está na melhora do entendimento do que é para ser feito no desenvolvimento do sistema, assim deixando claro o que o sistema deve fazer e como deve se comportar.

Validação dos requisitos deve ser feita, se aquele requisito é viável para o desenvolvimento daquele sistema, por exemplo, se ele se encaixa no orçamento do sistema

que está sendo feito.

6. Referências

<https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>

<https://www.sap.com/brazil/products/erp/what-is-erp.html>

Instituto de Ciências Exatas e Tecnologia	Disciplina: Engenharia de Software Ágil Título da Aula: Processos de Software	Relatório 3
--	--	--------------------

1. Resumo Teórico

Processo de software, conjunto de atividades que vão levar a produção de um produto de software, são procedimentos com um propósito de arquitetar, desenvolver, testar e implantar o software.

Modelos de processos de software. Modelos Prescritivos, foram feitos para trazer uma estrutura inicial aos processos e a maneira como eles se inter-relacionam, dentro dele há modelo cascata, Modelo Incremental, Modelo Evolucionário e Modelos Concorrentes.

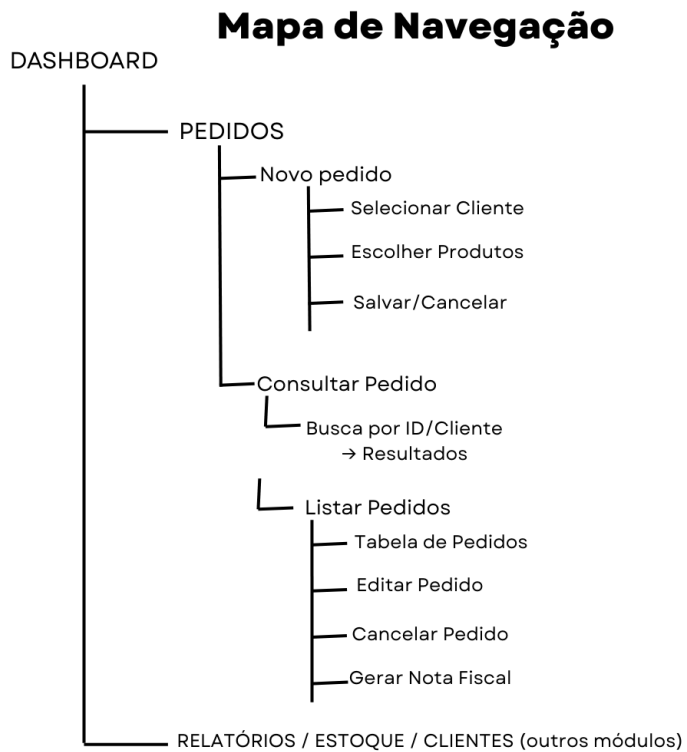
Dentro do sistema de software ERP, as principais operações do usuário seriam: o acesso a finanças da empresa, acesso ao RH e ao estoque de produtos.

2. Estudo de caso

Funcionalidade escolhida: Estoque de produtos do sistema ERP. Modelo escolhido: Incremental

Para criar as funcionalidades financeiras, iremos deixar claro os requisitos iniciais bem definidos, o que o cliente quer que tenha na parte estoque, isso inclui suas ações e como o sistema irá agir em relação a sua performance.

3. Mapa de navegação de funcionalidades



4. Diagrama de atividades

[Início]



(Planejar objetivos e requisitos)



(Analisar riscos e alternativas)



[Decisão: riscos aceitáveis?]

├─ Não → (Revisar requisitos/estratégia) → volta para "Planejar"

└─ Sim → segue



(Desenvolver protótipo / incremento)

5. Reflexões finais

Para construção de uma funcionalidade precisa entender o que o cliente precisa em seu sistema, suas ações e como ele deve agir.

Mapa de navegação importante para um melhor entendimento de como o sistema irá funcionar.

O que determinou a escolha de um modelo de processo, foi para ter um melhor entendimento de como irá funcionar o desenvolvimento do sistema, assim deixando bem documentado desde do início.

6. Referências

<https://portal.ifto.edu.br/iftoreitoria/diretoria-sistemica/tecnologia-da-informacao/documentos/processos/processo-de-gestao-de-software>

<https://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839>

Instituto de Ciências Exatas e Tecnologia	Disciplina: Engenharia de Software Ágil Título da Aula: Planejamento do Processo de Software	Relatório 4
--	---	--------------------

1. Resumo Teórico

Na componentização de software, a base do código é dividida em pequenas porções, assim incentiva o reuso e boas propriedades arquiteturais.

Diagramas da UML, UML é uma linguagem de modelagem unificada, a principal forma de sua representação é por meio de diagramas.

No total são 14 diagramas da UML, 7 estruturais e 7 comportamentais, estruturais lidam com aspectos estáticos do sistema, comportamentais com aspectos dinâmicos do sistema.

Um sistema distribuído é uma coleção de programas de computador que iram utilizar recursos computacionais, tem objetivo de remover gargalos e falhas no sistema.

Ao meu ver, as principais operações dos usuários, vai depender do tipo de sistema que ele está utilizando, se for um site de compras, seria poder ver seus produtos no carrinho, efetivar a compra com sucesso e visualizar com rapidez os produtos disponíveis.

2. Estudos de Caso

Desenvolvimento de um portal web, consiste na criação de um software pensado para internet, que possui distribuição de diversos tipos de conteúdos de fontes diferentes. Através dele o usuários podem ter acesso aos conteúdos postados pela empresa, aos seus canais e outras informações oferecidas pelo portal.

Componentes deste portal podem possuir uma navbar para a navegação entre os portais, o componente central, com o conteúdo principal e um footer com informações sobre a empresa, como e-mail ou telefone para contato. Módulos de um portal web fazem o gerenciamento de conteúdo ou de vendas.

3. Requisitos do sistema (RS)

Portal web escolhido: Broadcom Inc., Portal do Cliente. Líder global em tecnologia de semicondutores e soluções de software de infraestrutura, criou o portal para atender sua clientela em seus diversos setores, assim atuando como um centro de suporte universal para eles.

Nº	Requisito	Descrição	Tipo
1	Ajudar os clientes com eficiência	O sistema deve fornecer recursos que permitam solucionar dúvidas e problemas de forma ágil.	Funcional
2	Documentação	O sistema deve disponibilizar documentação clara e acessível para referência.	Funcional
3	Licenciamento	O sistema deve possibilitar controle de licenças de uso do software.	Funcional
4	Bases de conhecimento	O sistema deve conter uma base de conhecimento para consulta de informações.	Funcional
5	Blogs	O sistema deve permitir a publicação e acesso a artigos informativos em	Funcional

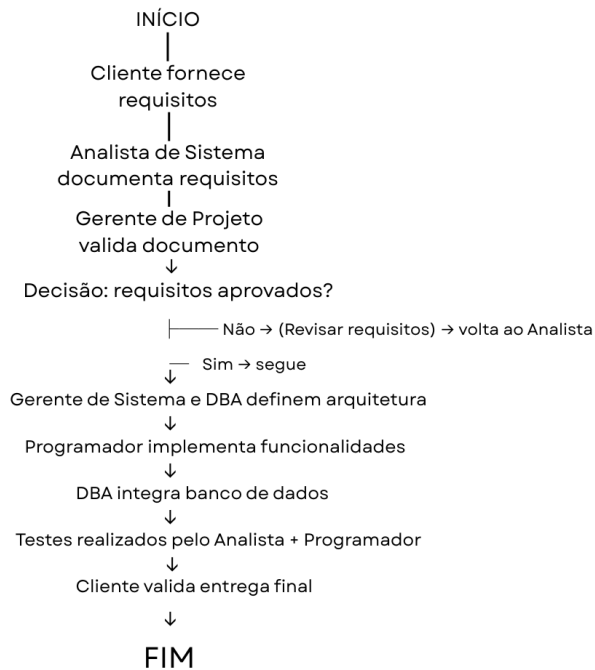
		formato de blog.	
6	Suporte por meio de agentes virtuais	O sistema deve fornecer atendimento automático por meio de chatbots ou IA.	Funcional
7	Atendimento ao vivo	O sistema deve permitir interação em tempo real com atendentes humanos.	Funcional
8	Rapidez na resposta	O sistema deve apresentar baixo tempo de resposta às solicitações.	Não Funcional
9	Intuitivo	A interface deve ser de fácil compreensão, reduzindo curva de aprendizado.	Não Funcional
10	Design agradável	O sistema deve ter uma interface visual atrativa e moderna.	Não Funcional
11	Seguro para acesso	O sistema deve garantir autenticação, criptografia e proteção de dados.	Não Funcional
12	Fácil navegação	O sistema deve oferecer menus e fluxos claros, permitindo localizar funções sem esforço.	Não Funcional

4. Diagrama de componente

Componentes vão atender os requisitos fornecidos pelo cliente, e foram aprovados pelo analista de sistema, para ver se aqueles requisitos se encaixam no desenvolvimento daquele software.

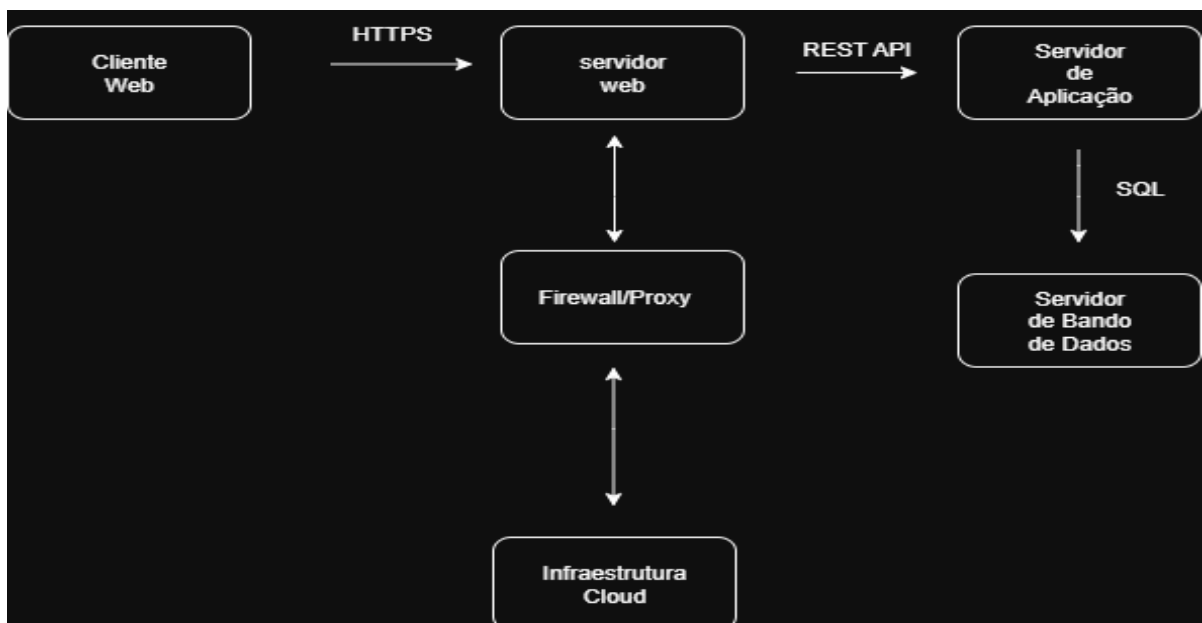
Abaixo está uma imagem de um diagrama:

Diagrama



5. Diagrama de implantação

São usados para visualizar os processadores, nós ou dispositivos de um sistema. Mostrando a execução da arquitetura de um sistema. Ajudando a modelar a topologia de hardware, esboçando os componentes lógicos de um sistema. Exemplo:



(Créditos: autoria própria)

6. Reflexões finais

Componentização de um sistema de software, consiste em separar as porções de código em pequenas partes. Componentes de sistema devem trabalhar para implementar e suportar o ambiente do software.

Diagramas UML ajudam a implementar a arquitetura de um sistema na área da TI, pois melhoram a visualização desse sistema, o acompanhamento e o suporte.

7. Referências:

<https://hnz.com.br/o-que-e-componentizacao/>

<https://www.estrategiaconcursos.com.br/blog/engenharia-software-diagramas-estruturais-uml/>

<https://www.atlassian.com/br/microservices/microservices-architecture/distributed-architecture>

<https://tradeconsulting.com.br/desenvolvimento-portal-web.html>

<https://www.liferay.com/pt/blog/current-experiences/exemplos-de-portais-excelentes>

<https://creately.com/blog/pt/diagrama/tutorial-do-diagrama-de-implantacao/>

<https://www.ibm.com/docs/pt-br/aix/7.3.0?topic=partitioning-system-components>

Instituto de Ciências Exatas e Tecnologia	Disciplina: Engenharia de Software Ágil Título da Aula: Fusão do Produto e do Processo de Software	Relatório 5
--	---	--------------------

1. Resumo Teórico

Matriz responsabilidade, tem a função de mostrar para cada pessoa envolvida no projeto sua função dentro dele e o papel que está ocupando. É uma das principais ferramentas na gestão de projetos, com ela é mais fácil a identificação com clareza das necessidades pessoais durante a fase de implementação.

Principais características da MR Raci: o acrônimo de RACI é formado por 4 palavras, que são os 4 atores fundamentais na gestão de projetos, que são eles. O responsável (Responsável), assume a atribuição de realizar tarefas e entregas. O accountable (Aprovador), faz a aprovação final do projeto. O consulted (Consultado) possui informações para continuidade do projeto. Por fim, Informed (Informado), podem ser clientes, stakeholders ou quaisquer pessoas que são atualizadas sobre o projeto.

Para ter mais sucesso comercial precisa-se de um bom alinhamento entre o processo, produto e pessoa. Isso inclui mapear, atender as necessidades dos consumidores, e entender as tendências do mercado. Um bom treinamento para equipe, com bons profissionais e uma cultura de inovação.

Principais medidas a serem feitas no controle e esforços de serviços pelos stakeholders, seria uma boa gestão de stakeholders, no qual identifica e gerencia relacionamentos com as partes interessadas pelo projeto, assim fazendo o alinhamento de expectativas e metas, promovendo uma comunicação clara.

2. Estudo de Caso

Principais perfis que compõem uma equipe de software, desenvolvedores de software: codificam o produto, engenheiro de garantia de qualidade: testa o produto, gerente de projeto: coordena o planejamento, designer de UX/UI: projeta interações do usuário, liderança empresarial: define metas, analista de negócios: traduz requisitos de negócios.

O modelo incremental. Esse modelo trabalha com pequenos pedaços de software entregue de cada vez, processo melhorado a cada iteração.

JAD é uma metodologia que permite extrair informações dos usuários, para isso se faz reuniões para tomar decisões sobre o desenvolvimento, determinando os objetivos e requisitos do sistema.

Distribuição de tarefas pela matriz responsabilidade, Responsável (atribui tarefas), Aprovador (aprova o projeto), Consultado (Informações para continuidade do projeto), Informado (Stakeholders, pessoas que são informadas sobre o projeto).

Artefato de software é um item produzido durante o desenvolvimento, como um modelo de dados, um diagrama de fluxo de trabalho, algum protótipo, um documento de design ou script de configuração.

3. Modelo de processo de software

O modelo escolhido foi o Iterativo, que é baseado em ciclos de desenvolvimento repetitivos, sua versão inicial é criada de maneira rápida, depois é feito refinamentos de forma contínua em iterações, cada uma delas inclui as fases de desenvolvimento.

4. Definição dos cargos e funções dos stakeholders

Stakeholders são partes interessadas em uma organização, projeto ou iniciativa, tendo influência sobre ela. Assim são afetadas pelas suas ações, podem ser grupos, indivíduos ou instituições. Sua tradução é partes interessadas ou envolvidas, são aqueles que têm influência e impacto sobre aquele projeto.

5. Matriz de responsabilidades

Matriz de responsabilidades com o objetivo de mostrar para os envolvidos dentro de um projeto sua função dentro e o papel que está ocupando. Também é chamada de MR RACI.

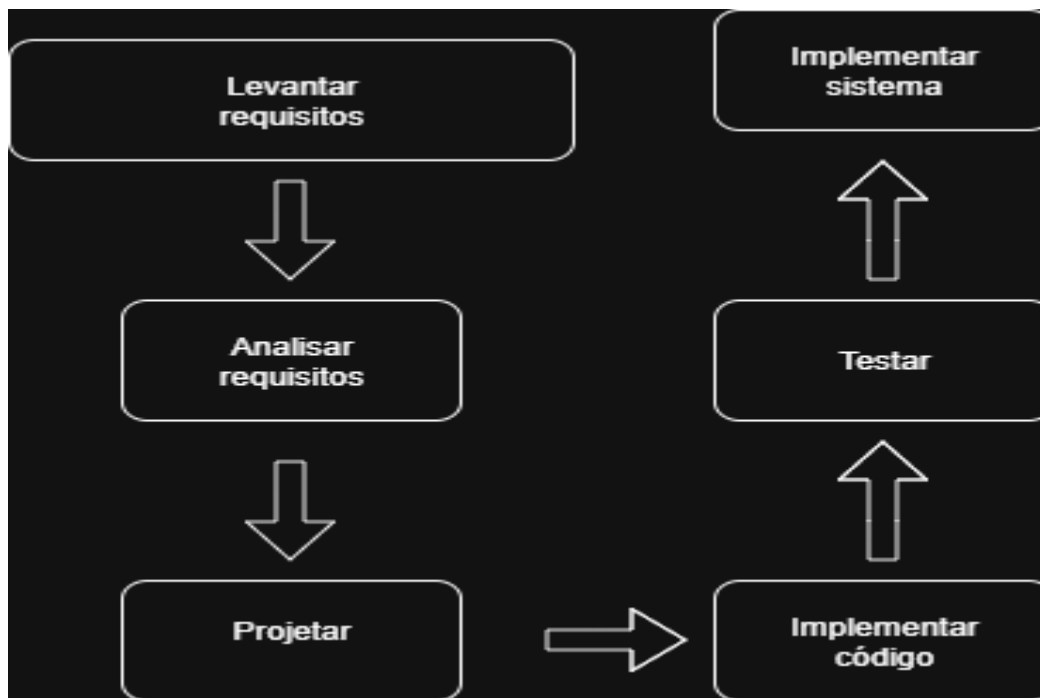
Um exemplo de Matriz de responsabilidades:

Atividade	Analista de Sistemas	Desenvolvedor	Usuário-chave	Facilitador JAD	Gerente de Projeto
Levantamento de requisitos	Responsável	Consultado	Responsável	Aprovado	Informado
Diagramas	Responsável	Consultado	Consultado	Aprovado	Informado
Requisitos Funcionais	Responsável	Consultado	Consultado	Aprovado	Informado
Prototipação	Consultado	Consultado	Consultado	Aprovado	Informado
Desenvolvimento da funcionalidade	Consultado	Responsável	Informado	Informado	Aprovado
Teste	Consultado	Responsável	Responsável	Aprovado	Informado
Validação	Consultado	Aprovado	Responsável	Aprovado	Aprovado

O esforço de cada participante está relacionado à importância do papel de cada um deles no desenvolvimento do sistema, sendo fundamental para o funcionamento do sistema, a colaboração de todos.

6. Diagrama de atividades.

Escolhi fazer pela plataforma [Draw.io](https://draw.io) que é gratuita, e ótima para construção de diagramas, pelo seu sistema ser feito de maneira intuitiva facilitando o uso para novos usuários.



(Créditos: autoria própria)

7. Reflexões Finais

A divisão de tarefas para equipe, é feita analisando as características, habilidades de cada membro da equipe envolvido no desenvolvimento do sistema, assim selecionando a pessoa mais qualificada para aquele determinado cargo.

Para criação de uma matriz de responsabilidade deve ser feito uma tabela com as atividades do projeto nas linhas e os membros da equipe nas colunas.

Uma melhoria a ser feita na distribuição de tarefas dos desenvolvedores, deve como diz anteriormente, selecionar a pessoa mais qualificada para aquele cargo, tendo em vista suas características e habilidades.

8. Referências

<https://robsoncamargo.com.br/blog/Matriz-de-responsabilidades-no-gerenciamento-de-projetos-saiba-tudo>

<https://blog.runrun.it/matriz-raci/>

<https://www.pipify.com/pt-br/blog/pessoas-processos-e-produtos/>

https://www.wrike-com.translate.goog/blog/4-strategies-dealing-difficult-stakeholders/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc

https://www-softkraft-co.translate.goog/software-development-team-structure/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc

<https://medium.com/contexto-delimitado/o-modelo-incremental-b41fc06cac04>

<https://www.devmedia.com.br/a-contribuicao-do-jad-para-o-levantamento-de-requisitos/12145>

https://www-leanix-net.translate.goog/en/wiki/trm/software-artifacts?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc

<https://www.linkedin.com/pulse/modelos-de-processo-desenvolvimento-software-vinicius-garcia-gvqaf>

<https://exame.com/carreira/guia-de-carreira/stakeholders-o-que-sao-e-quais-sao-os-principais-tipos/>

Instituto de Ciências Exatas e Tecnologia	Disciplina: Engenharia de Software Ágil Título da Aula: Processo Unificado	Relatório 6
--	---	--------------------

1. Resumo Teórico

O processo Unificado (PU), é um processo popular para o desenvolvimento de software visando à construção de sistemas orientado a objetos.

Metodologia PERT, é uma técnica utilizada para gestão de projetos, criada pela NASA em 1958, apontava a duração de um projeto em estimativa de 3 pontos. Otimista, Pessimista, Mais provável. Serve para o descobrimento da duração de uma atividade.

O quadro kanban é uma ferramenta para a gestão de projetos, feita para projeção do trabalho, ela ajuda as equipes ágeis e DevOps. Utilizam cartões, colunas, assim ajudando a equipe de desenvolvimento na conclusão do trabalho.

RUP tem como suas principais características ser incremental e iterativo, foi criado pela Rational Software Corporation, em 2003 adquirido pela IBM.

Com a divisão de etapas a empresa consegue fazer a integração da metodologia PERT com o RUP.

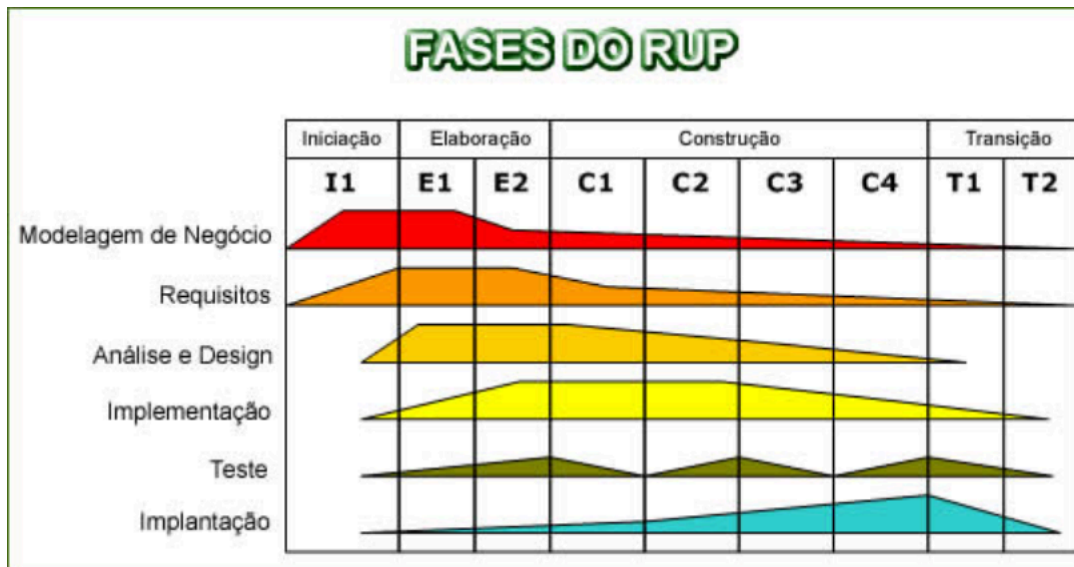
Principais medidas a serem feitas no cronograma e rede de cartões seria lembrar cada atividade que deve ser feita no dia, para a conclusão do sistema dentro do prazo esperado.

2. RUP

Nesta metodologia, o projeto passa por 4 fases básicas, seriam: Inception, Elaboration, Construction, Transition. Cada fase possui iterações, duram em torno de 1 a 2 semanas. Existem workflows que são uma sequência de tarefas encadeadas, que estão relacionadas a algum aspecto do projeto.

Estruturas estáticas possuem um tamanho fixo e não podem ser alteradas durante a

execução do algoritmo. Dinâmica pode crescer ou diminuir dependendo da necessidade, alocando memória.



(Créditos: <https://www.infoescola.com/engenharia-de-software/rup/>)

Para a definição de tarefas e tempos distribuídos dentro de 30 dias do processo unificado, precisa determinar a fase, se é concepção, elaboração, construção ou transição, fazer a divisão de tarefas e mapear as dependências do projeto e fazer uma boa distribuição de tempo para a realização de tarefas.

3. Tabela PERT

Metodologia PERT utilizada na área de Gestão de Projetos. Facilita a estimar prazo da duração dos projetos, assim evitando atrasos em sua entrega. Levando em consideração o conhecimento prévio obtido, ela consegue descobrir a duração da atividade.

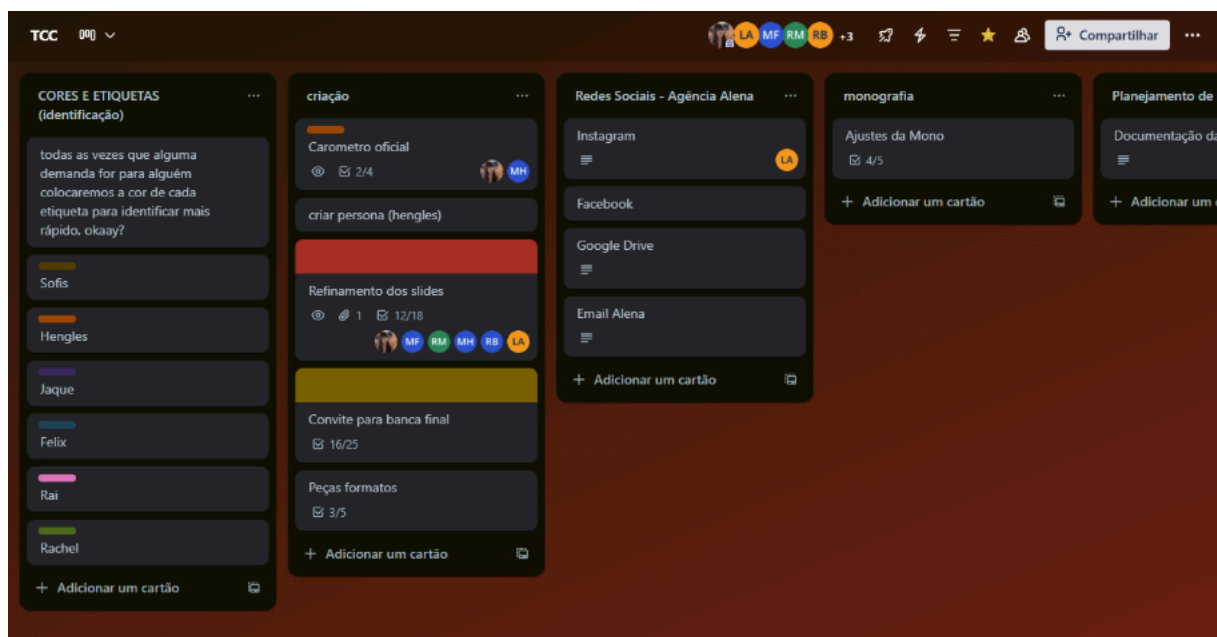
Tabela construída em grupo, utilizamos o Google Planilhas:

ID	Fase RUP	Disciplina	Tarefa	# O(d)	# M(d)	# P(d)	TE(d)	Predecessoras
T1	Iniciação	Engenharia de Requisitos	Elicitar requisitos iniciais com stakeholders	2	4	6	4.0	
T2	Iniciação	Gerenciamento de Projeto	Definir visão do produto e escopo de alto nível	1	3	5	3.0	T1
T3	Iniciação	Modelagem do Negócio	Mapear processos-chave do negócio (AS-IS/TO-B)	2	3	6	3.3	T1
T4	Elaboração	Análise e Design	Definir arquitetura base (camadas, componentes)	3	5	8	5.2	T2,T3
T5	Elaboração	Engenharia de Requisitos	Especificar casos de uso prioritários	2	4	7	4.2	T2
T6	Elaboração	Gerenciamento de Configuração e Mudanças	Configurar repositório e pipeline CI/CD	1	2	4	2.2	T4
T7	Construção	Implementação	Implementar caso de uso principal (Iteração 1)	3	6	9	6.0	T4,T5,T6
T8	Construção	Testes	Testes unitários e de integração (Iteração 1)	2	4	7	4.2	T7
T9	Construção	Implementação	Implementar incrementos secundários (Iteração 2)	3	5	8	5.2	T8
T10	Construção	Testes	Testes funcionais e não funcionais (Iteração 2)	2	4	6	4.0	T9
T11	Transição	Implantação	Preparar ambiente e realizar implantação piloto	2	3	6	3.3	T10
T12	Transição	Implantação	Treinar usuários e realizar Go-Live	1	2	4	2.2	T11

(créditos: autoria própria)

4. Construção do cronograma e rede de cartões (Kanban)

Cronogramas são formas de visualizar o planejamento de um projeto, assim facilitando o seu desenvolvimento. Quadro Kanban é um sistema visual da organização de um projeto, cada tarefa é representada por um cartão. Exemplo feito em grupo:



(créditos: Autoria própria)

5. Reflexões finais

Com o RUP utilizando das suas 4 fases (Inception, Elaboration, Construction, Transition), a empresa consegue uma boa distribuição de atividades e períodos.

As tabelas PERT fornecem uma perspectiva visual sobre requisitos de tempo do projeto, assim melhorando a tomada de decisões. Por tanto a construção de uma tem sua importância dentro do desenvolvimento de software.

As principais funcionalidades de uma aplicação específica para a construção de cronogramas e rede de cartões, seria cronogramas visual e dinâmico, assim facilita atualização e acompanhamento de tarefas e redes de cartões com informações claras e objetivas sobre a projeção do sistema.

6. Referências

<https://www.devmedia.com.br/introducao-ao-processo-unificado/3931>

<https://www.hostgator.com.br/blog/metodologia-pert-o-que-e-e-como-funciona/>

[https://www-atlassian-com.translate.goog/agile/kanban/boards?_x_tr_sl=en&_x_tr_tl=pt
&_x_tr_hl=pt&_x_tr_pto=tc](https://www-atlassian-com.translate.goog/agile/kanban/boards?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc)

<https://www.treinaweb.com.br/blog/o-que-e-rup-rational-unified-process>

<https://www.devmedia.com.br/rup-rational-unified-process/4574>

<https://medium.com/@pedro.vaf/estruturas-de-dados-est%C3%A1ticas-vs-estruturas-de-dados-din%C3%A2micas-e92e1d29b8f5>

<https://www.educamaisbrasil.com.br/educacao/dicas/metodologia-pert-o-que-e-e-como-funciona>

<https://asana.com/pt/resources/visual-project-management-kanban-timeline-calendar>

<https://www.lucidchart.com/pages/pt/guia/diagrama-pert>

<https://sienge.com.br/blog/cronograma-gantt/>

Instituto de Ciências Exatas e Tecnologia	Disciplina: Análise e Projeto de Sistemas Título da Aula: Metodologias Ágeis I	Relatório 7
--	---	--------------------

Instituto de Ciências Exatas e Tecnologia	Disciplina: Análise e Projeto de Sistemas Título da Aula: Metodologias Ágeis II	Relatório 8
--	--	--------------------

1. Resumo Teórico

(Escreva aqui um texto de 8 a 10 linhas abordando:)

- O que é a ferramenta Trello?
- Principais características da metodologia ágil Scrum.
- Como funciona a integração das atividades do Scrum em um quadro kanban.
- Principais medidas e acompanhamentos a serem feitas com o uso da ferramenta Trello.

2. Metodologia ágil Scrum

Descrição do framework XP.

- Breve descritivo sobre as atividades em pares e a distribuição das atividades-chave entre os programadores e (ou) analistas da XP.
- Breve descritivo sobre as atividades-chave da XP.

3. Quadro kanban com a ferramenta Trello

Descrição sobre a aplicação Trello e construção do quadro kanban.

Descrição:

- Definir o modelo de cartão, que deverá ter as informações de descrição do backlog e das sprints.

-
-
-
-
-
- Construir um quadro kanban (ferramentas: Trello) dimensionado para entrega da funcionalidade em 15 dias.

- (Inserir aqui imagens de tela do Trello do quadro de cartões e do painel de acompanhamento.)

4. Reflexões Finais

- Como funciona a metodologia ágil Scrum e os parâmetros associados da ferramenta Trello?
- Por que existe uma flexibilidade maior na gestão da equipe de desenvolvimento com o uso da ferramenta Trello em relação a outras ferramentas?
- Quais as melhorias associadas ao Trello que podem ser adotadas com o uso da função Jira?

5. Referências

(Inclua pelo menos 1 referência do livro-texto e outras fontes utilizadas.)