



*Faculdade de
Tecnologia SENAI
“Roberto Mange”*



*Engenharia de Software
Introdução*

Expectativa x Realidade

Engenharia de software é um assunto denso, extenso e por vezes abstrato, para isso, existem graduações e pós graduações específicas para o tema, dito isso, para o nosso curso ela será tratada obviamente como uma disciplina com a carga horária extremamente limitada e onde será abordado muitos contextos no semestre, cuja a implementação de um código (mão na massa, programando claro!) nem será a parte principal.



Graduação Tecnológica
Engenharia de Software

Literatura Básica

No PPC do ADS existem diversos livros sugeridos muito bons para o tema, porém, vou usar como base o seguinte livro, que tem uma versão gratuita em HTML →

<https://engsoftmoderna.info/>



Graduação Tecnológica
Engenharia de Software



Contextos Históricos

Leia a introdução desse livro com os aspectos históricos, fatos e a organização dessa área do conhecimento, como essa ponderação feita pelos engenheiros do primeiro encontro de discussão no século passado para demonstrar a dificuldade que os softwares já estavam impondo na época.

“

“O problema é que certas classes de sistemas estão colocando demandas sobre nós que estão além das nossas capacidades e das teorias e métodos de projeto que conhecemos no presente tempo. Em algumas aplicações não existe uma crise, como rotinas de ordenação e folhas de pagamento, por exemplo. Porém, estamos tendo dificuldades com grandes aplicações. Não podemos esperar que a produção de tais sistemas seja fácil.”



Graduação Tecnológica
Engenharia de Software

Temas da engenharia de software

Conforme o guia contendo o conjunto de conhecimentos SWEBOK (IEEE), é possível dividir engenharia de sistemas em 12 áreas:

- 1. Engenharia de Requisitos**
- 2. Projeto de Software**
- 3. Construção de Software**
- 4. Testes de Software**
- 5. Manutenção de Software**
- 6. Gerência de Configuração**

- 7. Gerência de projetos**
- 8. Processos de Software**
- 9. Modelos de Software**
- 10. Qualidade de Software**
- 11. Prática Profissional**
- 12. Aspectos Econômicos**



Graduação Tecnológica
Engenharia de Software

Conhecimentos exigidos no PPC de ADS

Vamos adequar esses temas aos conhecimentos exigidos no PPC de ADS.

- 1. Engenharia de Requisitos**
- 2. Projeto de Software**
- 3. Construção de Software**
- 4. Testes de Software**
- 5. Manutenção de Software**
- 6. Gerência de Configuração**

- 7. Gerência de projetos**
- 8. Processos de Software**
- 9. Modelos de Software**
- 10. Qualidade de Software**



Graduação Tecnológica
Engenharia de Software

Engenharia de Requisitos

Na engenharia de requisitos focamos “**no que fazer**” e “**como deve operar**” o sistema em questão.

Deve propor uma forma sistemática de definir, analisar, documentar e validar seu sistema. Mais comumente podem ser classificados como funcionais e não funcionais:

- **Funcionais** foca no que fazer. Exemplo num homebank seria algo como imprimir um extrato, fazer um saque ou pix;
- **Não funcionais** foca na sua operação. Para o mesmo exemplo a disponibilidade do sistema e sua segurança. Métricas são usadas para determinar os testes e qualidade do software também nos requisitos;



Graduação Tecnológica
Engenharia de Software

Projeto de Software

É através do projeto de software, que com base nos requisitos apresentados, o software deve ir tomando sua forma real e concreta.

Envolve decisões estratégicas e técnicas que impactarão diretamente na qualidade, manutenção e desenvolvimento.

Geralmente divide-se o problema em partes menores e mais gerenciáveis (interfaces) e sua integração, ainda não é a implementação do código, mas a sua estratégia de criação.



Graduação Tecnológica
Engenharia de Software

Projeto de Software

No exemplo de uma unidade de código em OO, novamente de um homebank, podemos dividir essas interfaces em:

Interfaces providas: são aqueles serviços que uma unidade de código torna público para uso pelo resto do sistema.

Interfaces requeridas ou dependentes: são aquelas interfaces das quais uma unidade de código depende para funcionar.

```
class ContaBancaria {  
    private Cliente cliente;  
    private double saldo;  
    public double getSaldo() { ... }  
    public String getNomeCliente() { ... }  
    public String getExtrato (Date inicio) { ... }  
    ...  
}
```



Graduação Tecnológica
Engenharia de Software

Construção do Software

A construção trata da implementação, isto é, da codificação do sistema;

- Definir os algoritmos e estruturas de dados que serão usados;
- Definir os frameworks e bibliotecas disponíveis que serão usados;
- Definir técnicas para tratamento de exceções;
- Definir padrões de nomes, leiaute e documentação de código, se existir alguma norma aderente seguir;
- Definir as ferramentas que serão usadas no desenvolvimento, incluindo compiladores, ambientes integrados de desenvolvimento (IDEs), depuradores, gerenciadores de bancos de dados, ferramentas para construção de interfaces, etc.



Graduação Tecnológica
Engenharia de Software

Testes de Software

Teste consiste na execução de um programa com um **conjunto finito de casos**, com o objetivo de verificar se ele possui o comportamento esperado.

“

“Testes de software mostram a presença de bugs, mas não a sua ausência.”

Os tipos de testes mais comuns são: **unitário, integração, usabilidade e performance.**

Os testes podem ser ainda para **verificação e validação.**



Graduação Tecnológica
Engenharia de Software

Testes de Software

Testes de unidade → quando se testa uma pequena unidade do código, como uma classe;

Testes de integração → quando se testa uma unidade de maior granularidade, como um conjunto de classes;

Testes de performance → quando se submete o sistema a uma carga de processamento para verificar seu desempenho;

Testes de usabilidade → quando o objetivo é verificar a usabilidade da interface do sistema;

O termo “**verificação**” geralmente é adotado para testes pela óptica do desenvolvedor identificando os requisitos inicialmente previstos, enquanto o conceito de “**validação**” seria para os testes com outputs provenientes de usuários, isto é, se o software realmente “**agrada**” o cliente.



Graduação Tecnológica
Engenharia de Software

Manutenção do Software

Manutenção Corretiva → tem como objetivo corrigir falhas de usuários e outros desenvolvedores;

Manutenção preventiva → constatado uma hipótese de falha, se antecipa a um possível mal funcionamento que o software pode ter. Exemplo sistemas de datas foram alterados na virada do século de DD/MM/AA para DD/MM/AAAA;

Manutenção Adaptativa → esse tipo de manutenção incluem adaptação a normas e regras, novas tecnologias ou mesmo uma customização requerida pelo cliente;

Manutenção evolutiva → alterar programas ou sistemas antigos, porém, ainda funcionais, para atualizações mais recentes



Graduação Tecnológica
Engenharia de Software

Gerência de Configuração

A ideia por trás de uma gerência de configuração é manter o controle de versionamento e revisões coerentes e atuais para:

- Códigos fonte;
- Documentação;
- Páginas web;
- Relatórios e outros



Graduação Tecnológica
Engenharia de Software

Gerência de Projetos

A gerência de projetos está relacionada na prática em toda cadeia do desenvolvimento de software, atuando desde gerência e planejamento com clientes, finanças, riscos, marketing, entretanto, a área que mais demanda tempo e análise sem dúvida nenhuma são os recursos humanos, analise a **“lei de brooks”** sobre o tema, na literatura sugerida.



Graduação Tecnológica
Engenharia de Software

Processos de Software

Historicamente, podemos dividir em 2 tipos de processos de desenvolvimento mais comumente usados:

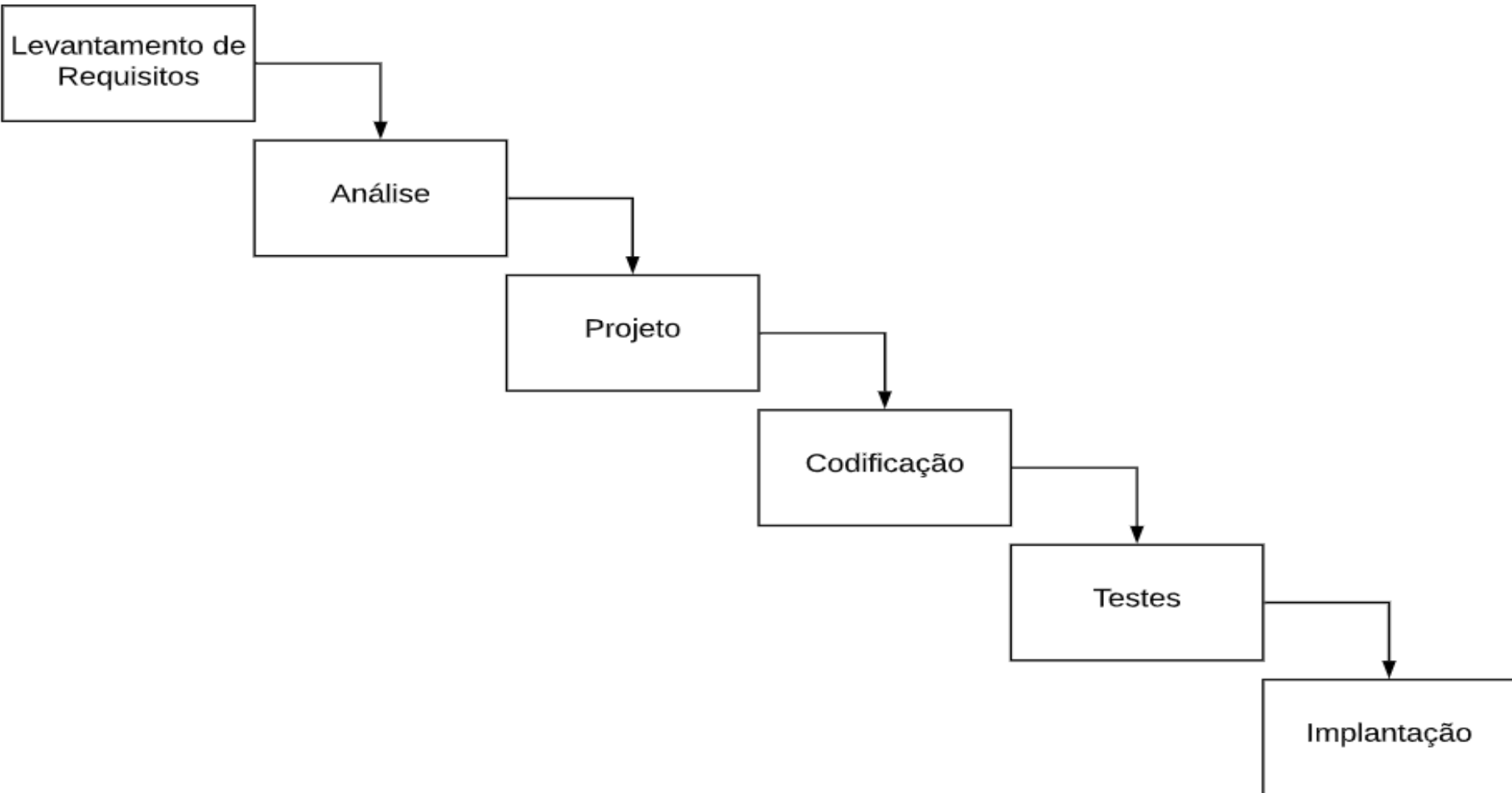
Processos Waterfall(Cascata) ;

Processos Ágeis ou incrementais ou interativos ;



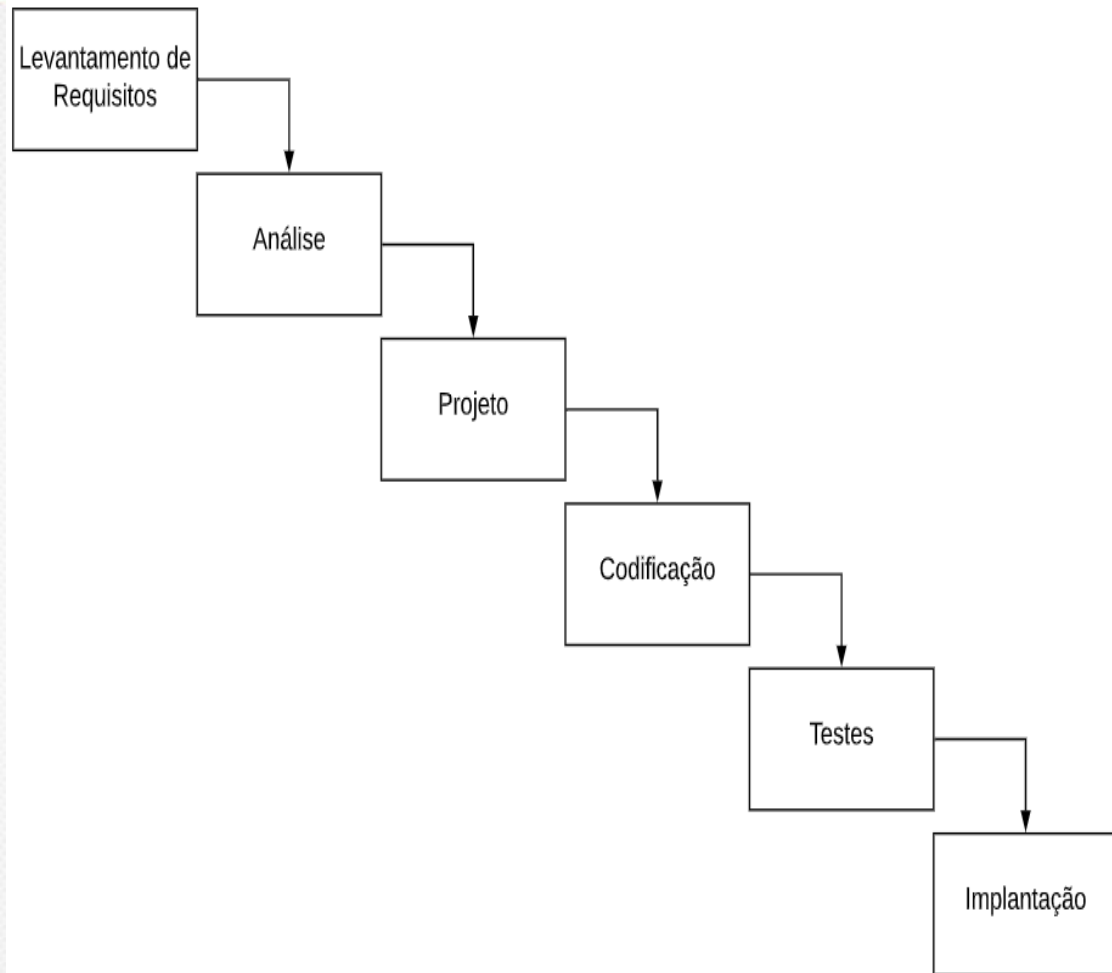
Graduação Tecnológica
Engenharia de Software

Processo Cascata



Processo Ágil

Nos métodos ágeis pressupõe verificações e validações constantes. Criação dos testes antes do código, integração diária. Os métodos mais usados atualmente: Scrum, XP, Kanban e Lean Development



Modelos de software

Um modelo oferece uma representação em mais alto nível de um sistema do que o seu código fonte.

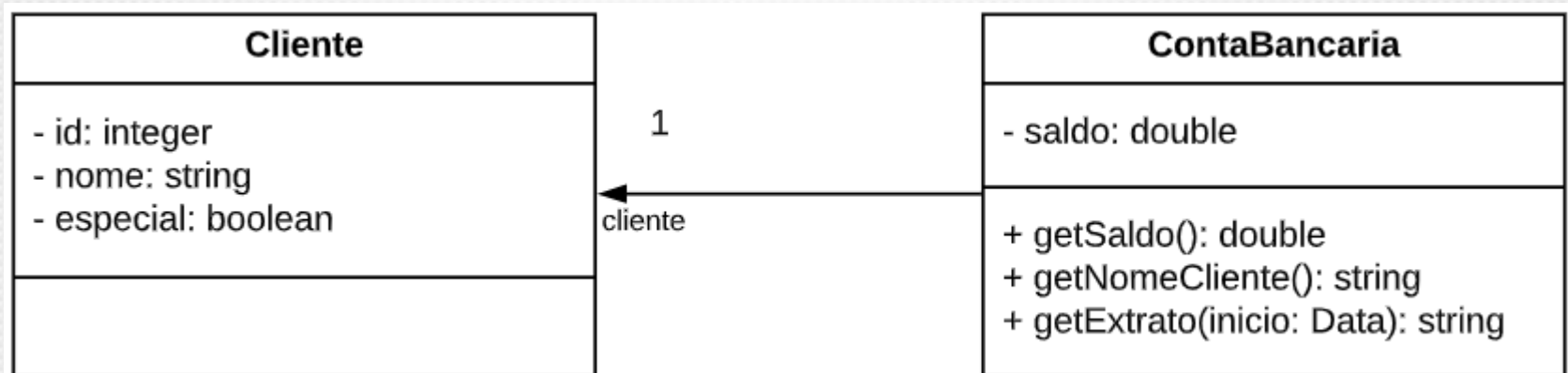
Engenharia avante ou comum → primeiro cria-se um modelo para ter um entendimento de mais alto nível de um sistema, antes de partir para a implementação do código;

Engenharia Reversa → para entender o funcionamento de um sistema complexo, decodifica-o em sistema de alto nível.



Diagramas UML

Frequentemente, modelos de software são baseados em notações gráficas. Diagramas UML são bastante comuns para essas etapas. Exemplo de um diagrama de classe UML



Qualidade de software

Podemos analisar qualidade de software em duas vertentes:

- Como qualidade no “**processo**”, o quanto meu processo é bom, geralmente são atreladas a “**qualidade interna**” e caracterizadas por modularização, legibilidade do código, manutenção simples e testabilidade;
- Como produto, o quanto o produto é bom e aí consideramos como “**qualidade externa**”, são caracterizadas por correção, robustez, eficiência, portabilidade, facilidade de uso, compatibilidade, dentre outros



Questionário

Responder o questionário da aula 01
(envio individual)

<https://forms.gle/FCijzPbRhm81CLz98>

