



5954014 - Engenharia de Software

Aula 07 - Planejamento e Estimativas

Prof. Dr. Denis Martins

DCM | FFCLRP | USP

Objetivos de Aprendizagem

Conceitos Fundamentais

- Diferenciar **tamanho, esforço, duração e custo**
- Explicar por que estimativas possuem **incerteza** de forma inerente

Práticas de Estimativa

- Elaborar estimativas por **analogia e decomposição**
- Utilizar **dados históricos** para apoiar o processo de estimativa

Técnicas e Abordagens

- Reconhecer diferentes **técnicas de estimativa** no ciclo de software
- Identificar as **premissas essenciais** de cada estimativa

Comunicação e Resultados

- Apresentar estimativas como **faixas de valores**, e não apenas como números absolutos

Antes de responder...

Que informações você gostaria de conhecer? Pense em pelo menos **3 informações essenciais**.

Uma Pergunta Comum

Um cliente procura sua equipe de desenvolvimento e diz:

"Precisamos de um sistema web para gerenciamento de eventos."

Uma Pergunta Comum

Um cliente procura sua equipe de desenvolvimento e diz:

"Precisamos de um sistema web para gerenciamento de eventos."

Antes de responder...

Que informações você gostaria de conhecer? Pense em pelo menos **3 informações essenciais**.

01 Escopo e Complexidade dos Requisitos

Quais são as funcionalidades indispensáveis? É necessário suporte a **pagamentos online**, controle de lotes, credenciamento via QR Code ou múltiplos perfis de acesso?

02 Integrações e Infraestrutura

O sistema precisará se integrar a **CRMs existentes**, ERPs ou plataformas de e-mail marketing? Onde a aplicação será hospedada e qual a expectativa de acessos simultâneos?

03 Restrições de Prazo e Orçamento

Existe alguma data de evento já agendada que sirva como **prazo limite inegociável**? Há uma expectativa orçamentária prévia para o desenvolvimento inicial (MVP)?

Processo de Estimativa

Estimar é construir uma base **quantitativa racional** para **guiar decisões** de engenharia de software.

Sem alguma estimativa, decisões de projeto passam a depender apenas de intuição.

O que é e por que estimar?

O que é uma estimativa?

Uma estimativa é uma **previsão quantitativa** construída de forma racional a partir das informações que estão disponíveis no momento.

Pode envolver:

- Tamanho e Esforço
- Duração e Custo

Atenção: Estimativa não é sinônimo de compromisso, promessa ou meta inegociável.

Por que estimar software?

Estimativas sólidas ajudam a responder perguntas críticas de viabilidade e planejamento:

- Quanto trabalho será necessário?
- Quantas pessoas podem ser necessárias?
- Quanto o projeto poderá custar?
- A solução pretendida é realmente viável?
- Qual alternativa de design ou arquitetura demanda mais esforço?

Alta Incerteza

Muitos projetos exigem estimativas iniciais complexas mesmo em cenários de alta incerteza estrutural e técnica.

As decisões iniciais de planejamento precisam enfrentar a falta de clareza antes do desenvolvimento começar.

Mas estimar software é difícil...

Fontes de Incerteza

É desafiador realizar estimativas antes de conhecer completamente:

- Todos os **requisitos** do sistema
- Todas as **dificuldades técnicas**
- Todas as **integrações** necessárias
- Todos os **detalhes** de implementação

A Visão de Sommerville

"As estimativas devem ser **revistas à medida que mais informações** sobre o sistema se tornam disponíveis."

O processo de estimativa é contínuo e iterativo, exigindo calibração constante conforme o escopo se torna claro.

Quatro Conceitos Diferentes

É crucial entender que estes conceitos são relacionados, mas totalmente distintos:

1. Tamanho

Escopo físico (ex: linhas, pontos de função)

2. Esforço

Trabalho necessário (ex: pessoas-hora)

3. Duração

Tempo de calendário (ex: semanas, meses)

4. Custo

Valor financeiro do projeto (R\$ ou \$)

Dimensões Críticas

Para planejar e estimar com sucesso, é preciso diferenciar claramente estas três dimensões.

Cada conceito exige técnicas próprias de medição e interfere diretamente na viabilidade do projeto.

Dimensões da Estimativa de Software

1. Tamanho

Representa **quanto software** deverá ser produzido. Expresso por:

- funcionalidades
- componentes
- pontos de função
- histórias ou pontos
- linhas de código

Exemplo: Um sistema possui > 12 funcionalidades principais (tamanho funcional).

2. Esforço

Representa a quantidade de **trabalho humano** necessário para a execução.

Exemplos:

- 80 pessoa-horas
- 30 pessoa-dias
- 12
pessoa-semanas

Exemplo: Desenvolver determinado módulo exige aproximadamente **120 pessoa-horas**.

3. Duração

Representa o **tempo de calendário** necessário para o projeto.

Exemplos:

- 10 dias
- 6 semanas
- 4 meses

Atenção: Esforço e duração **não são a mesma coisa**.

Esforço ≠ Duração

Entender que a alocação de mais pessoas não reduz proporcionalmente o prazo de entrega é um dos maiores desafios na gestão de projetos de software.

A relação entre esforço e tempo de calendário não é linear devido a fatores humanos e organizacionais.

Por que mais pessoas não equivalem a menos tempo?

Cenário Teórico

Considere uma atividade de software estimada em:

100 pessoa-horas

10 pessoas concluem em 10 horas?

Não necessariamente.

*Em teoria, a divisão parece perfeita, mas a realidade do desenvolvimento de software impõe **custos invisíveis** que impedem essa paralelização.*

Esforço ≠ Duração

Entender que a alocação de mais pessoas não reduz proporcionalmente o prazo de entrega é um dos maiores desafios na gestão de projetos de software.

A relação entre esforço e tempo de calendário não é linear devido a fatores humanos e organizacionais.

Por que mais pessoas não equivalem a menos tempo?

Cenário Teórico

Considere uma atividade de software estimada em:

100 pessoa-horas

10 pessoas concluem em 10 horas?

Não necessariamente.

*Em teoria, a divisão parece perfeita, mas a realidade do desenvolvimento de software impõe **custos invisíveis** que impedem essa paralelização.*

Fatores de Complexidade

Projetos de software envolvem:

- **Dependências** entre tarefas
- Necessidade de **comunicação** contínua
- Esforço de **coordenação** e gestão
- **Revisões** e controle de qualidade
- Processos complexos de **integração**
- **Conhecimento** muito especializado
- Trabalho que **não pode** ser paralelizado

Portanto:

Duração ≠ Esforço / Pessoas

4. Custo

Para planejar e estimar com sucesso, é preciso diferenciar claramente as dimensões do desenvolvimento de software.

O custo financeiro consolida e reflete as decisões tomadas em relação a tamanho, esforço e duração.

Composição do Custo

Representa o **impacto financeiro** associado ao desenvolvimento do projeto.

Pode envolver:

- **Trabalho da equipe** (desenvolvedores, gestores)
- **Infraestrutura** (servidores, nuvem)
- **Serviços e APIs** de terceiros
- **Licenças** de software
- **Ferramentas** de desenvolvimento
- **Contratação externa** e consultorias

4. Custo

Para planejar e estimar com sucesso, é preciso diferenciar claramente as dimensões do desenvolvimento de software.

O custo financeiro consolida e reflete as decisões tomadas em relação a tamanho, esforço e duração.

Composição do Custo

Representa o **impacto financeiro** associado ao desenvolvimento do projeto.

Pode envolver:

- **Trabalho da equipe** (desenvolvedores, gestores)
- **Infraestrutura** (servidores, nuvem)
- **Serviços e APIs** de terceiros
- **Licenças** de software
- **Ferramentas** de desenvolvimento
- **Contratação externa** e consultorias

Exemplo Simplificado

Cálculo básico considerando esforço e custo médio de hora trabalhada:

Esforço estimado:

200 horas

Custo médio da hora:

R\$ 100 / h

Impacto Financeiro Direto:

$200 \times 100 = \text{R\$ } 20.000$

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

*A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.*

Como podemos estimar?

01

Julgamento de especialistas

Baseia-se na experiência e intuição de profissionais seniores que já realizaram projetos semelhantes.

02

Estimativa por analogia

Utiliza dados históricos de projetos passados similares como base de comparação para o novo sistema.

03

Decomposição

Divide o software em partes menores (WBS / APF) para estimar cada componente individualmente de forma mais precisa.

04

Modelos algorítmicos

Utiliza fórmulas matemáticas e métricas (como COCOMO ou Pontos de Função) baseadas em fatores de complexidade do software.

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.

Como podemos estimar?

01

Julgamento de especialistas

Profissionais experientes analisam o problema detalhadamente e elaboram uma estimativa de esforço.

Exemplo Prático

"Já desenvolvemos três integrações semelhantes. Normalmente precisamos de aproximadamente 40 a 60 horas."

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.

Como podemos estimar?

01

Julgamento de especialistas

Profissionais experientes analisam o problema detalhadamente e elaboram uma estimativa de esforço.

Exemplo Prático

"Já desenvolvemos três integrações semelhantes. Normalmente precisamos de aproximadamente 40 a 60 horas."

Vantagens

♦ Rápido e Eficiente

Permite obter previsões ágeis sem a necessidade de métricas extremamente complexas.

♦ Experiência Acumulada

Valoriza o histórico profissional e o conhecimento prático da equipe de desenvolvimento.

Viés Individual

A estimativa pode ser influenciada por excesso de otimismo, pressões externas ou percepções puramente subjetivas do especialista.

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

*A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.*

Como podemos estimar?

02

Estimativa por analogia

Referência: Projeto anterior

- ◆ 12 funcionalidades
- ◆ Complexidade média
- ◆ **Histórico: 300 horas**

Novo Projeto

Análise comparativa:

- ◆ 14 funcionalidades (+2)
- ◆ Tecnologia semelhante
- ◆ Complexidade semelhante

O projeto anterior pode servir como referência?

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

*A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.*

Como podemos estimar?

02

Estimativa por analogia

Referência: Projeto anterior

- ◆ 12 funcionalidades
- ◆ Complexidade média
- ◆ **Histórico: 300 horas**

Novo Projeto

Análise comparativa:

- ◆ 14 funcionalidades (+2)
- ◆ Tecnologia semelhante
- ◆ Complexidade semelhante

O projeto anterior pode servir como referência?

Analogia não significa copiar: Analisar Diferenças

O novo projeto pode possuir **variáveis distintas** que impactam o esforço:

- ◆ Mais funcionalidades ou maior **complexidade técnica**
- ◆ **Novas tecnologias** envolvidas ou mais integrações
- ◆ Curva de aprendizado e **maturidade da equipe**

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.

Como podemos estimar?

03

Decomposição

Dividir para estimar

Em vez de perguntar o sistema inteiro, dividimos o problema:

- ◆ Quanto exige o cadastro?
- ◆ Quanto exige a autenticação?
- ◆ Quanto exige o pagamento e relatórios?

Funcionalidade	Esforço
Cadastro de usuários	20 h
Cadastro de eventos	32 h
Inscrição	28 h
Notificações / Relatórios	60 h

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.

Como podemos estimar?

03

Decomposição

Dividir para estimar

Em vez de perguntar o sistema inteiro, dividimos o problema:

- ◆ Quanto exige o cadastro?
- ◆ Quanto exige a autenticação?
- ◆ Quanto exige o pagamento e relatórios?

Funcionalidade	Esforço
Cadastro de usuários	20 h
Cadastro de eventos	32 h
Inscrição	28 h
Notificações / Relatórios	60 h

Por que decompor? Partes menores são:

- ◆ **Mais fáceis de compreender** e visualizar o escopo real.
- ◆ **Mais fáceis de comparar** com históricos de outros projetos.
- ◆ **Mais fáceis de discutir e estimar** de forma colaborativa.

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.

Como podemos estimar?

04

Modelos algorítmicos

Fatores quantitativos

Modelos relacionam matematicamente variáveis críticas:

- ◆ Tamanho do software
- ◆ Produtividade histórica
- ◆ Complexidade técnica e esforço

Exemplo clássico

Modelo COCOMO:

- ◆ COnstructive COst MOdel
- ◆ Foco no entendimento geral
- ◆ Sem necessidade de cálculos nesta aula

Foco conceitual no uso de equações.

Estimativas

Para planejar e mitigar riscos com sucesso, a Engenharia de Software utiliza **técnicas estruturadas** para prever esforço, prazo e custo.

A **escolha do método** correto impacta diretamente a precisão do cronograma e a viabilidade econômica do projeto.

Como podemos estimar?

04

Modelos algorítmicos

Fatores quantitativos

Modelos relacionam matematicamente variáveis críticas:

- ◆ Tamanho do software
- ◆ Produtividade histórica
- ◆ Complexidade técnica e esforço

Exemplo clássico

Modelo COCOMO:

- ◆ COConstructive COSt MOdel
- ◆ Foco no entendimento geral
- ◆ Sem necessidade de cálculos nesta aula

Foco conceitual no uso de equações.

Estimativas baseadas em dados

O importante é compreender a fundamentação dos modelos:

- ◆ Também podemos construir estimativas a partir de **modelos baseados em dados**.
- ◆ Equações matemáticas convertem métricas em esforço real.

Referências: Pressman & Maxim; Sommerville.

Importância de Dados Históricos

o analisar métricas de projetos passados, as equipes substituem a intuição e o otimismo por evidências concretas, tornando o planejamento futuro mais preciso, confiável e profissional.

A

Empresa A (Intuição)

- ◆ Estimativa baseada em opinião ou sentimento.
- ◆ Sem dados de suporte para justificar o esforço.

"Achamos que essa funcionalidade levará umas 30 horas."

B

Empresa B (Evidência)

- ◆ Baseada em métricas de projetos concluídos.
- ◆ Mediana estatística reduz o impacto de ruídos.

"Nas últimas 18 funcionalidades desse tipo, a mediana foi de 27 horas."

Qual possui uma base melhor para estimar?

Empresa B possui uma fundamentação científica e empiricamente validada:

- **Dados históricos** eliminam o "achômetro" e o viés otimista da equipe.
- **Amostras passadas (18 funcionalidades)** conferem previsibilidade matemática.

Exercício

Qual projeto serve como referência?

Analise os dados históricos ao lado para estimar o novo escopo com seu parceiro.

Proj	Funcs	Comp	Esforço
A	12	Baixa	240 h
B	15	Média	390 h
C	20	Alta	700 h

Novo Projeto a Estimar

- **16 funcionalidades**
- Complexidade **Média**
- Tecnologias semelhantes às anteriores

Questões para Discussão

1. **Qual projeto** deveria ter **maior peso** em uma estimativa por analogia? Por quê?
2. Qual seria uma **estimativa inicial** razoável de esforço para as 16 funcionalidades?
3. Vocês apresentariam ao cliente um **valor único** ou uma **faixa de esforço**? Justifiquem.
4. Quais **informações adicionais** poderiam alterar significativamente essa estimativa?

Exercício

Discussão

Uma análise aprofundada sobre os limites e calibrações da estimativa por analogia.

💬 **Debate aberto**

A Analogia mais Próxima

O **Projeto B** se destaca como referência:

- **15 funcionalidades**
- **Complexidade Média**
- **Histórico: 390 horas**

Limite da Simplificação

Não podemos simplesmente afirmar:

Novo Projeto = 390h

Variáveis Críticas de Calibração

Para uma estimativa realista por analogia, precisamos analisar sistematicamente:

- **Diferenças entre funcionalidades:** Complexidades específicas que divergem do modelo padrão.
- **Equipe:** Senioridade, velocidade e nível de entrosamento dos membros atuais.
- **Tecnologia:** Mudanças de frameworks, versões de linguagem ou arquitetura técnica.
- **Integrações:** Dependência de APIs externas, microsserviços e complexidade de dados.
- **Conhecimento disponível:** Curva de aprendizado, documentação e suporte técnico existente.

Conceito

Incerteza

Uma estimativa realista não esconde a incerteza inerente ao desenvolvimento.

Faixas vs. Valor Único

Opção A (Valor Único)

O projeto exige 417 horas.

Transmite uma falsa sensação de precisão absoluta para o cliente.

Opção B (Faixa de Incerteza)

Estimamos entre 380 e 470 horas, sendo ~420h o cenário provável.

Comunica a variação real e os riscos de forma transparente.

Por que usar uma faixa de estimativa?

Uma faixa deixa explícito que a estimativa depende diretamente de cenários e hipóteses:

Otimista

24 h

Poucos problemas ou imprevistos acontecem durante o desenvolvimento.

Mais Provável

30 h

O cenário padrão considerado realista e balanceado pela equipe.

Pessimista

42 h

Dificuldades técnicas, retrabalho ou dependências complexas aparecem.

Apresentar os 3 cenários comunica **muito mais informação** sobre o nível de risco do que simplesmente cravar um único valor arbitrário de ~~"30 horas"~~.

Conceito

Premissas

Toda estimativa de esforço ou prazo está diretamente amarrada a um conjunto de condições iniciais.

Exemplo de Cenário Estimado

Estimamos entre 300 e 360 horas para a entrega do projeto.

Esta faixa de valores assume como verdade quatro premissas fundamentais de viabilidade.

As 4 premissas que sustentam a estimativa:

O cálculo de horas depende da manutenção destas condições estáveis durante a execução:

1. Requisitos Definidos

Os requisitos principais do sistema estão acordados e não sofrerão alterações.

2. Domínio Tecnológico

A equipe de desenvolvimento conhece e domina a tecnologia que será utilizada.

3. APIs Externas

A API de terceiros permanecerá continuamente disponível e documentada.

4. Sem Migração de Dados

Não haverá necessidade de realizar migrações complexas de bases legadas.

Se uma única premissa mudar... a estimativa também precisará ser revisada. Alterações nas premissas quebram o acordo inicial de esforço e exigem recalibração imediata do cronograma.

Conceito

Diretrizes

Mais importante do que acertar um número de forma cega é construir uma base sólida e justificável.

O que vale mais que "acertar o número"?

Uma estimativa profissional e madura deve seguir cinco diretrizes fundamentais:

1. Coerente

Alinhada à complexidade real, histórico de velocidade e capacidade da equipe.

2. Justificável

Baseada em argumentos lógicos claros, premissas de negócio e engenharia.

3. Baseada em Evidências

Sustentada por métricas passadas, dados de projetos similares e referências.

4. Transparente (Premissas)

Deixa explícitas as condições iniciais e os cenários de viabilidade adotados.

5. Transparente (Incertezas)

Comunica claramente as variações possíveis, faixas de risco e limites da estimativa.

Na prática

Uma estimativa sem justificativa pode parecer precisa e ainda assim estar completamente errada.

Conexão

Para estimar com precisão, precisamos olhar para trás. Na aula anterior, consolidamos os fundamentos que tornam as previsões possíveis.

Na aula anterior discutimos como capturar a realidade para projetar o futuro.

Métricas e Indicadores

Registros de projetos anteriores:

- ◆ Esforço e tamanho real
- ◆ Produtividade histórica
- ◆ Taxa de defeitos e retrabalho

Estimativas Futuras

Destino dos dados:

Esses dados históricos alimentam diretamente os modelos matemáticos e algorítmicos.

Garantindo previsibilidade e precisão.

Como essa conexão gera valor?

O registro contínuo de métricas cria a fundação empírica necessária:

- ◆ **Dados históricos** eliminam o viés otimista e o "achômetro".
- ◆ **Amostras passadas** fornecem a calibração que os algoritmos precisam.

Referências: Pressman & Maxim; Sommerville.

Conclusão

Estimativas maduras combinam métricas, técnicas estruturadas e premissas para apoiar decisões de negócio mais embasadas e seguras.

Dimensões Principais

1. Tamanho

Escopo físico (ex: linhas, pontos de função)

2. Esforço

Trabalho necessário (ex: pessoas-hora)

3. Duração

Tempo de calendário (ex: semanas, meses)

4. Custo

Valor financeiro do projeto (R\$ ou \$)

Técnicas de Estimativa

- **Julgamento:** Baseado em experiência profissional prévia.
- **Analogia:** Comparação direta com projetos similares.
- **Decomposição:** Divisão do problema em partes menores.
- **Modelos:** Equações e relações quantitativas matemáticas.

Mensagem Principal

Estimativa não é promessa.

Uma boa estimativa deve sempre combinar:

**Evidências + Premissas +
Técnica + Incerteza**

Dúvidas e Discussão