

# FIAAP

Compliance & Quality Assurance

2025

# Cálculo de Esforço, Prazo e Custo de Projeto

# Cálculo de Prazo e Custos de Projeto

O cálculo de prazo e custo de um projeto envolve uma combinação de análise, experiência e consideração cuidadosa das variáveis envolvidas



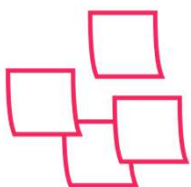
Com as identificações preliminares de Escopo e Estimativa de Esforço realizados previamente o prazo e custo se utiliza de parâmetros pré-estabelecidos.



# Cálculo de Prazo e Custos de Projeto



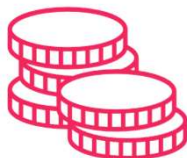
**Definição do Escopo:** Tenha uma compreensão clara e detalhada do escopo do projeto, incluindo todos os requisitos, entregáveis e objetivos.



**Identificação de Tarefas:** Divida o projeto em tarefas individuais ou subtarefas. Liste todas as atividades que precisam ser realizadas para concluir o projeto.



**Estimativa de Duração:** Para cada tarefa, estime o tempo necessário para concluí-la. Isso pode ser feito em dias, semanas ou meses, dependendo da escala do projeto.



**Estimativa de Custo:** Determine os recursos necessários para cada tarefa, incluindo pessoal, materiais e outros custos associados.



# Cálculo de Prazo e Custos de Projeto



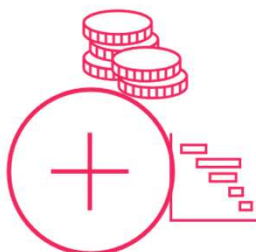
**Alocação de Recursos:** Determine quantas pessoas ou membros da equipe serão atribuídos a cada tarefa e quanto tempo eles gastarão nela.



**Cálculo de Prazo:** Some as durações estimadas de todas as tarefas para obter a duração total do projeto. Isso fornecerá uma estimativa inicial do prazo.



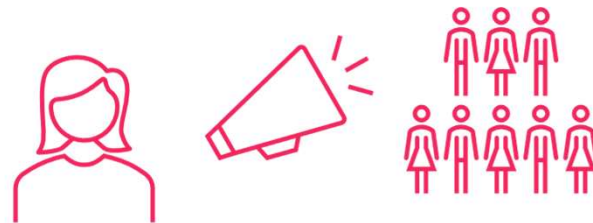
**Cálculo de Custo:** Some os custos estimados de todas as tarefas para obter o custo total do projeto. Isso inclui custos de mão de obra, materiais, equipamentos e outros.



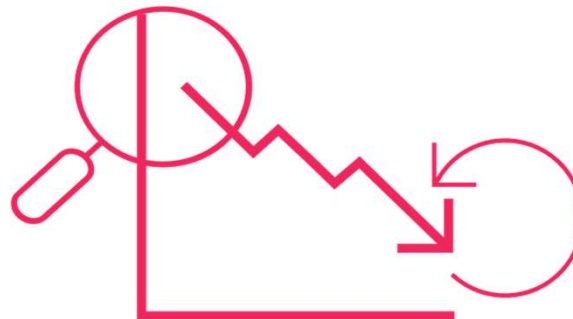
**Buffer de Tempo e Custo:** Adicione um buffer de tempo e custo para lidar com imprevistos e riscos. Isso ajuda a acomodar possíveis atrasos e custos adicionais.

# Cálculo de Prazo e Custos de Projeto

**Comunicação e Aprovação:** Comunique as estimativas de prazo e custo à equipe e partes interessadas. Certifique-se de obter aprovação e feedback.



**Monitoramento e Atualização:** À medida que o projeto avança, monitore o progresso real em relação às estimativas. Atualize as estimativas conforme necessário com base no aprendizado e nas mudanças no escopo.



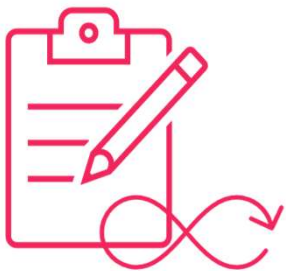
# Cálculo de Prazo e Custos de Projeto



**Ferramentas de Gestão:** Utilize ferramentas de gestão de projetos, como software de gerenciamento de projetos, para facilitar o rastreamento do progresso, prazos e custos.



**Relatórios e Avaliações:** Crie relatórios regulares para acompanhar o progresso do projeto em relação às estimativas iniciais. Avalie desvios e tome medidas corretivas, se necessário.



**Aprendizado Contínuo:** Use os resultados e insights do projeto para melhorar a precisão das estimativas em projetos futuros.



# Cálculo de Prazo e Custos de Projeto

## Exemplo Prático

**Projeto:** Desenvolvimento de um Site de E-commerce

**Passos:**

**Definição do Escopo:** O projeto envolve criar um site de e-commerce para uma loja de roupas, incluindo catálogo de produtos, carrinho de compras, checkout e integração com pagamento.





# Cálculo de Prazo e Custos de Projeto

## Exemplo Prático

### Identificação de Tarefas:

Design de Interface  
Desenvolvimento Front-end  
Desenvolvimento Back-end  
Integração de Pagamento  
Testes e Depuração  
Lançamento



# Cálculo de Prazo e Custos de Projeto

## Exemplo Prático

**Estimativa de Duração e Custo:** Suponhamos que as estimativas sejam as seguintes:

Design de Interface: 2 semanas

Desenvolvimento Front-end: 4 semanas

Desenvolvimento Back-end: 6 semanas

Integração de Pagamento: 2 semanas

Testes e Depuração: 3 semanas

Lançamento: 1 semana

Custo estimado (incluindo mão de obra, ferramentas e outros):  
\$50.000



# Cálculo de Prazo e Custos de Projeto

## Exemplo Prático

**Alocação de Recursos:** 2 designers, 2 desenvolvedores front-end, 2 desenvolvedores back-end e 1 gerente de projeto.

**Cálculo de Prazo:** Soma das durações estimadas das tarefas:  $2 + 4 + 6 + 2 + 3 + 1 = 18$  semanas.

**Cálculo de Custo:** Custo estimado total: \$50,000.



# Cálculo de Prazo e Custos de Projeto

## Exemplo Prático

**Buffer de Tempo e Custo:** Adicionamos um buffer (tempo extra) de 2 semanas para possíveis atrasos e custos adicionais:  $\text{Prazo} = 18 + 2 = 20$  semanas,  $\text{Custo} = \$50,000 + \$5,000 = \$55,000$ .  
 $((50k/18)*2)$

**Comunicação e Aprovação:** As estimativas de prazo e custo são comunicadas à equipe e ao cliente para aprovação.

**Monitoramento e Atualização:** À medida que o projeto avança, as estimativas são comparadas ao progresso real. Se houver desvios significativos, os ajustes são feitos.



# Cálculo de Prazo e Custos de Projeto

## Exemplo Prático

**Relatórios e Avaliações:** Relatórios semanais mostram o progresso do projeto em relação às estimativas, identificando áreas que estão no caminho certo e aquelas que precisam de atenção.

**Aprendizado Contínuo:** Depois do projeto, a equipe analisa o que funcionou bem e o que pode ser melhorado para refinar as estimativas em projetos futuros.

Esse é um exemplo simplificado. Em projetos reais, muitos outros fatores podem influenciar as estimativas de prazo e custo, como mudanças de escopo, imprevistos e recursos humanos adicionais. A precisão das estimativas é crucial para um planejamento e execução bem-sucedidos.

# Cerimônias de acompanhamento e técnicas de revisão de projeto

As cerimônias de acompanhamento e as técnicas de revisão de projeto são práticas fundamentais para manter um projeto no caminho certo, identificar problemas e fazer ajustes conforme necessário. São algumas cerimônias e técnicas comuns:

**Reuniões de Status:** Reuniões regulares em que a equipe relata o progresso, problemas, obstáculos e próximos passos. Isso ajuda a manter todos informados e identificar problemas em estágios iniciais.



# Cerimônias de acompanhamento e técnicas de revisão de projeto

**Sprints (Scrum):** Cerimônias do método Scrum, como o Sprint Planning, Daily Standup, Sprint Review e Sprint Retrospective, que permitem um acompanhamento constante do progresso e ajustes a cada iteração.

**Revisões de Marcos:** Realizadas após atingir marcos específicos do projeto. São oportunidades para avaliar o progresso, revisar os resultados e planejar as próximas etapas.

**Reuniões de Gate:** Realizadas em pontos de decisão-chave do projeto, como antes de avançar para uma nova fase. As partes interessadas revisam o progresso e decidem se o projeto deve continuar.



# Cerimônias de acompanhamento e técnicas de revisão de projeto

## Técnicas de Revisão de Projeto:

**Análise de Valor Agregado (EVA):** Uma técnica que mede o desempenho do projeto em relação ao cronograma e orçamento. Ajuda a identificar desvios e permite ajustes para manter o projeto sob controle.

**Avaliação de Riscos:** Identificação, análise e monitoramento contínuo dos riscos do projeto. Isso permite tomar medidas preventivas ou corretivas para lidar com potenciais problemas.





# Cerimônias de acompanhamento e técnicas de revisão de projeto

**Revisões de Código:** Para projetos de desenvolvimento de software, as revisões de código garantem a qualidade do código, a adesão a padrões e a detecção de possíveis problemas.

**Revisões de Design:** Avaliação do design do projeto em busca de adequação aos requisitos, eficiência e viabilidade técnica.

**Avaliações Técnicas:** Exame detalhado da arquitetura e das soluções técnicas do projeto para garantir a integridade, segurança e desempenho.



# Cerimônias de acompanhamento e técnicas de revisão de projeto

**Auditorias de Qualidade:** Avaliações independentes das práticas de qualidade do projeto para garantir que os processos estejam alinhados com as normas e melhores práticas.

**Feedback do Cliente:** Manter um diálogo constante com o cliente para garantir que suas necessidades e expectativas sejam atendidas. Isso ajuda a ajustar o projeto conforme necessário.

**Retrospectivas de Projeto:** Realizadas após a conclusão do projeto, permitem que a equipe reflita sobre o que funcionou bem, o que poderia ter sido melhor e quais lições foram aprendidas.

Cada projeto pode exigir diferentes cerimônias e técnicas de revisão, dependendo de sua natureza, tamanho e complexidade. O importante é manter um acompanhamento constante e uma abordagem pró-ativa para identificar problemas e fazer ajustes ao longo do tempo.



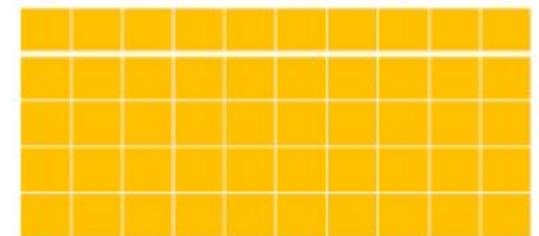
# Diferenças entre estimativas diretas como o Poker Scrum e estimativas paramétricas

As estimativas diretas, como o Poker Scrum, e as estimativas paramétricas são duas abordagens diferentes para estimar o esforço, o prazo e/ou o custo de um projeto. Aqui estão as principais diferenças entre elas:



**Paramétrica**

■ X 100 =



# Diferenças entre estimativas diretas como o Poker Scrum e estimativas paramétricas

## Estimativas Diretas (Poker Scrum):

**Baseadas em Percepção:** As estimativas diretas são baseadas na percepção dos membros da equipe sobre o esforço necessário para realizar uma tarefa ou completar um entregável.

**Colaborativas:** O processo de estimativas diretas envolve a participação ativa de todos os membros da equipe. Isso promove a colaboração e o compartilhamento de conhecimento.



# Diferenças entre estimativas diretas como o Poker Scrum e estimativas paramétricas

## Estimativas Diretas (Poker Scrum):

**Subjetivas:** As estimativas diretas podem ser subjetivas, uma vez que diferentes membros da equipe podem ter opiniões diferentes sobre o esforço envolvido em uma tarefa.

**Métodos como o Poker Scrum:** Métodos como o Planning Poker, onde a equipe escolhe cartas com valores de esforço, são exemplos de estimativas diretas.

**São Relativas:** As estimativas diretas muitas vezes fornecem valores de esforço relativos em vez de estimativas absolutas, o que ajuda a comparar a complexidade relativa das tarefas.

# Diferenças entre estimativas diretas como o Poker Scrum e estimativas paramétricas

## Estimativas Paramétricas:

Baseadas em Dados Históricos: As estimativas paramétricas são baseadas em dados históricos e em relações matemáticas entre variáveis do projeto, como tamanho, produtividade e esforço.

Modelos Matemáticos: As estimativas paramétricas usam modelos matemáticos para prever o esforço, o prazo ou o custo com base em parâmetros específicos do projeto.



# Diferenças entre estimativas diretas como o Poker Scrum e estimativas paramétricas

## Estimativas Paramétricas:

**Menos Subjetivas:** As estimativas paramétricas tendem a ser menos subjetivas, pois se baseiam em dados e relações matemáticas objetivas.

**Levam em Consideração Variáveis:** As estimativas paramétricas consideram fatores como tamanho do projeto, produtividade da equipe, complexidade técnica e outros.

**Exemplos:** Modelos COCOMO (Constructive Cost Model) e PERT (Program Evaluation and Review Technique) são exemplos de abordagens paramétricas.

# Diferenças entre estimativas diretas como o Poker Scrum e estimativas paramétricas

## Diferenças Gerais:

**Natureza:** Estimativas diretas são baseadas em opiniões e percepções da equipe, enquanto as estimativas paramétricas são baseadas em análises matemáticas e dados históricos.

**Precisão:** As estimativas paramétricas tendem a ser mais precisas, uma vez que se baseiam em modelos matemáticos e dados reais.

**Flexibilidade:** As estimativas diretas são mais flexíveis e adequadas para projetos com muitas incertezas ou mudanças, enquanto as estimativas paramétricas são mais apropriadas para projetos com padrões repetitivos e bem definidos.

Em última análise, a escolha entre estimativas diretas e paramétricas dependerá da natureza do projeto, das necessidades da equipe e da disponibilidade de dados históricos confiáveis. Muitas vezes, uma combinação de abordagens pode ser usada para obter estimativas mais abrangentes.





# Modelos paramétricos de estimativa

Existem vários modelos paramétricos de estimativa amplamente utilizados na indústria para prever esforço, prazo e custo de projetos. Aqui estão alguns dos principais modelos paramétricos:

**COCOMO (Constructive Cost Model):** Um dos modelos paramétricos mais conhecidos, desenvolvido por Barry Boehm. O COCOMO possui várias versões, incluindo COCOMO I, COCOMO II e COCOMO III. Ele estima o esforço, o custo e o prazo do projeto com base em características do projeto, como tamanho do código, complexidade e fatores de produtividade.

**Function Point Analysis:** Um método que avalia o tamanho funcional de um sistema de software, medindo as funções que ele executa para os usuários. As funções são atribuídas a diferentes categorias (entrada, saída, consulta, arquivo interno e interface externa) e convertidas em pontos de função, que são usados para estimar o esforço e o prazo.

**Estimativa de Linhas de Código:** Embora não seja um modelo paramétrico estrito, a estimativa de linhas de código ainda é usada para prever esforço, prazo e custo. Ela se baseia no número de linhas de código que serão escritas e considera a produtividade da equipe.

# Modelos paramétricos de estimativa

**PERT (Program Evaluation and Review Technique):** Um método que estima o tempo necessário para concluir uma tarefa, levando em consideração três cenários possíveis: otimista, pessimista e mais provável. Ele calcula uma estimativa ponderada que leva em consideração a incerteza.

**Delphi Method:** Embora não seja estritamente um modelo paramétrico, o método Delphi é uma técnica de estimativa que envolve especialistas fornecendo opiniões anônimas sobre o esforço, o prazo e o custo do projeto. Essas opiniões são consolidadas para chegar a uma estimativa.

**Function Point Productivity Rates:** Este modelo utiliza os pontos de função como métrica e estima a produtividade da equipe com base em projetos anteriores. A produtividade é então usada para calcular o esforço necessário para o projeto atual.

**SLIM (Software Life Cycle Management):** Um modelo paramétrico que utiliza uma abordagem estatística para prever o esforço, o prazo e o custo do projeto. Ele leva em consideração variáveis como tamanho, produtividade da equipe e complexidade técnica.

## CoCoMo Intermediário

$$E = a.(KLoC)^b . (EAF)$$

- **E**: estimativa de esforço, dado em pessoa.mês (PM).
- **KLoC**: estimativa de linhas de código, em milhares (*Thousands Lines of Code*).
- **EAF**: fator de ajuste de esforço (*Effort Adjustment Factor*).
- **'a'** e **'b'**: constantes baseadas no tipo de projeto e ambiente de desenvolvimento.

$$D = 2,5 . E^c$$

- **D**: estimativa de tempo de desenvolvimento (*Duration*).
- **'c'**: constante baseada no tipo de projeto e ambiente de desenvolvimento.



# CoCoMo Intermediário

$$E = a \cdot (KLoC)^b \cdot (EAF)$$

|               | $a_i$ | $b_i$ | $c_i$ |
|---------------|-------|-------|-------|
| Organic       | 3.2   | 1.05  | 0.38  |
| Semi-detached | 3.0   | 1.12  | 0.35  |
| Embedded      | 2.8   | 1.20  | 0.32  |

- **Organic:** times de desenvolvimento relativamente pequenos, trabalhando em projetos simples, bem compreendidos ou similares a algo do passado, em um ambiente familiar de desenvolvimento (ferramentas, linguagem, etc.)
- **Embedded:** desenvolvimento com restrições severas em termos de requisitos, complexidade, regulação, etc. Exige um time de desenvolvimento grande e com muita experiência.
- **Semi-detached:** projeto que apresente características de complexidade, experiência e tamanho da equipe em um nível intermediário entre o *Organic* e o *Embedded*.

# CoCoMo Intermediário

$$E = a.(KLoC)^b \cdot (EAF)$$

| Direcionadores de Custo   | Muito Baixo | Baixo | Normal | Elevado | Muito Elevado | Extremamente Elevado |
|---|-------------|-------|--------|---------|---------------|----------------------|
| ATRIBUTOS DO PRODUTO  |             |       |        |         |               |                      |
| Confiabilidade exigida do software                                | 0,75        | 0,88  | 1,00   | 1,15    | 1,40          | -                    |
| Tamanho do banco de dados   | -           | 0,94  | 1,00   | 1,08    | 1,16          | -                    |
| Complexidade do produto   | 0,70        | 0,85  | 1,00   | 1,15    | 1,30          | 1,65                 |
| ATRIBUTOS DO HARDWARE   |             |       |        |         |               |                      |
| Restrições ao tempo de execução                                   | -           | -     | 1,00   | 1,11    | 1,30          | 1,66                 |
| Restrições de memória   | -           | -     | 1,00   | 1,06    | 1,21          | 1,56                 |
| Volatilidade do ambiente de máquina virtual                       | -           | 0,87  | 1,00   | 1,15    | 1,30          | -                    |
| Tempo de <i>turnaround</i> (tempo para completar o ciclo) exigido | -           | 0,87  | 1,00   | 1,07    | 1,15          | -                    |
| ATRIBUTOS DE PESSOAL  |             |       |        |         |               |                      |
| Capacidade do analista  | 1,46        | 1,19  | 1,00   | 0,86    | 0,71          | -                    |
| Experiência em aplicações   | 1,29        | 1,13  | 1,00   | 0,91    | 0,82          | -                    |
| Capacidade do programador   | 1,42        | 1,17  | 1,00   | 0,86    | 0,70          | -                    |
| Experiência em máquina virtual                                    | 1,21        | 1,10  | 1,00   | 0,90    | -             | -                    |
| Experiência com a linguagem de programação                        | 1,14        | 1,07  | 1,00   | 0,95    | -             | -                    |
| ATRIBUTOS DE PROJETO  |             |       |        |         |               |                      |
| Uso de práticas modernas de programação                           | 1,24        | 1,10  | 1,00   | 0,91    | 0,82          | -                    |
| Uso de ferramentas de software                                    | 1,24        | 1,10  | 1,00   | 0,91    | 0,83          | -                    |
| Cronograma exigido de desenvolvimento                             | 1,23        | 1,08  | 1,00   | 1,04    | 1,10          | -                    |

Fonte: Boehm (1981)

# CoCoMo Intermediário

## Exemplo

Projeto: desenvolvimento de um site simples de e-commerce, por uma software-house com experiência nesse tipo de produto.

$$E = a.(KLoC)^b \cdot (EAF)$$

- **KLoC**: 40 KLoC (40.000 linhas de código)
- **'a'** = 3,20
- **'b'** = 1,05

|               | <b>a<sub>i</sub></b> | <b>b<sub>i</sub></b> | <b>c<sub>i</sub></b> |
|---------------|----------------------|----------------------|----------------------|
| Organic       | 3.2                  | 1.05                 | 0.38                 |
| Semi-detached | 3.0                  | 1.12                 | 0.35                 |
| Embedded      | 2.8                  | 1.20                 | 0.32                 |

# CoCoMo Intermediário

$$E = 3,2.(40)^{1,05} \cdot (0,598)$$

| Direcionadores de Custo   | Muito Baixo | Baixo | Normal | Elevado | Muito Elevado | Extremamente Elevado |
|---|-------------|-------|--------|---------|---------------|----------------------|
| ATRIBUTOS DO PRODUTO  |             |       |        |         |               |                      |
| Confiabilidade exigida do software                                | 0,75        | 0,88  | 1,00   | 1,15    | 1,40          | -                    |
| Tamanho do banco de dados   | -           | 0,94  | 1,00   | 1,08    | 1,16          | -                    |
| Complexidade do produto   | 0,70        | 0,85  | 1,00   | 1,15    | 1,30          | 1,65                 |
| ATRIBUTOS DO HARDWARE   |             |       |        |         |               |                      |
| Restrições ao tempo de execução                                   | -           | -     | 1,00   | 1,11    | 1,30          | 1,66                 |
| Restrições de memória   | -           | -     | 1,00   | 1,06    | 1,21          | 1,56                 |
| Volatilidade do ambiente de máquina virtual                       | -           | 0,87  | 1,00   | 1,15    | 1,30          | -                    |
| Tempo de <i>turnaround</i> (tempo para completar o ciclo) exigido | -           | 0,87  | 1,00   | 1,07    | 1,15          | -                    |
| ATRIBUTOS DE PESSOAL  |             |       |        |         |               |                      |
| Capacidade do analista  | 1,46        | 1,19  | 1,00   | 0,86    | 0,71          | -                    |
| Experiência em aplicações   | 1,29        | 1,13  | 1,00   | 0,91    | 0,82          | -                    |
| Capacidade do programador   | 1,42        | 1,17  | 1,00   | 0,86    | 0,70          | -                    |
| Experiência em máquina virtual                                    | 1,21        | 1,10  | 1,00   | 0,90    | -             | -                    |
| Experiência com a linguagem de programação                        | 1,14        | 1,07  | 1,00   | 0,95    | -             | -                    |
| ATRIBUTOS DE PROJETO  |             |       |        |         |               |                      |
| Uso de práticas modernas de programação                           | 1,24        | 1,10  | 1,00   | 0,91    | 0,82          | -                    |
| Uso de ferramentas de software                                    | 1,24        | 1,10  | 1,00   | 0,91    | 0,83          | -                    |
| Cronograma exigido de desenvolvimento                             | 1,23        | 1,08  | 1,00   | 1,04    | 1,10          | -                    |

Fonte: Boehm (1981)

# CoCoMo Intermediário

## Exemplo

Projeto: desenvolvimento de um site simples de e-commerce, por uma software-house com experiência nesse tipo de produto.

$$E = 3,2 \cdot (40)^{1,05} \cdot (0,598)$$

$$E = 92,05 \text{ PM}$$

$$D = 2,5 \cdot E^c$$

$$D = 2,5 \cdot 92,05^{0,38}$$

$$D = 13,94$$

$$P = E / D$$

$$P = 92,05 / 13,94$$

$$P = 6,6$$





FIAP

THE WAY WE ARE