

# Aula 12 - Plano de Gerenciamento de Tempo e Cronograma

**Disciplina:** Gestão de Projetos Tecnológicos

**Prof.** Me. João Paulo Biazotto

# Áreas de Conhecimento Projeto

- Escopo
- Tempo
- Qualidade
- Recursos Humanos
- ...

# Ciclo de Vida do Projeto

- Como monitorar e manter registros de todos esses aspectos?



# Plano de Gerenciamento do Projeto




Documento que descreve como o projeto será **executado, monitorado, controlado e encerrado.**

Este plano garante que os projetos sejam entregues dentro do prazo, orçamento e com a qualidade esperada.





# Plano de Gerenciamento do Projeto

Redução de riscos, melhor alocação de recursos e maior **previsibilidade** dos resultados.

# Plano de Gerenciamento do Projeto

- Escopo 
- Tempo
- Custos
- Qualidade
- Recursos Humanos
- Comunicação 
- Riscos
- ~~Aquisições~~
- Partes Interessadas 

# Plano de Gerenciamento de Tempo

- Escopo 
- Tempo 
- Custos
- Qualidade
- Recursos Humanos
- Comunicação 
- Riscos
- ~~Aquisições~~
- Partes Interessadas 

# Plano de Gerenciamento de Tempo

Documento que estabelece **como** as atividades do projeto serão **planejadas**, monitoradas e controladas ao longo do tempo.

Garante que o projeto seja concluído dentro do **prazo estabelecido**.

Gera **previsibilidade** ao gerenciamento do projeto.



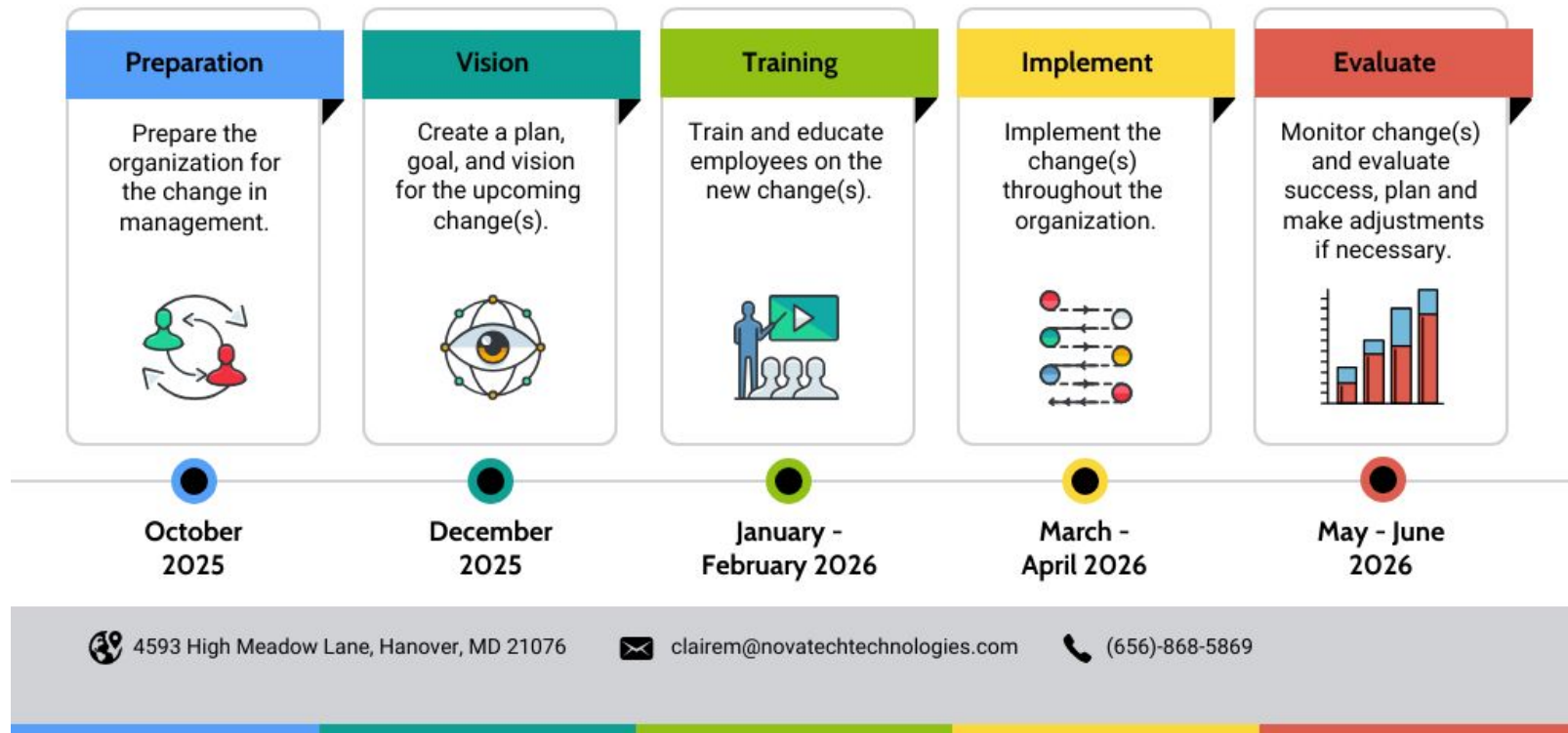
# Plano de Gerenciamento de Tempo

O plano de gerenciamento de tempo é composto por **dois documentos** principais:

- *Timeline*: Representação visual macro das principais fases e eventos do projeto.
- *Cronograma*: Documento detalhado com datas, dependências e duração de cada atividade.

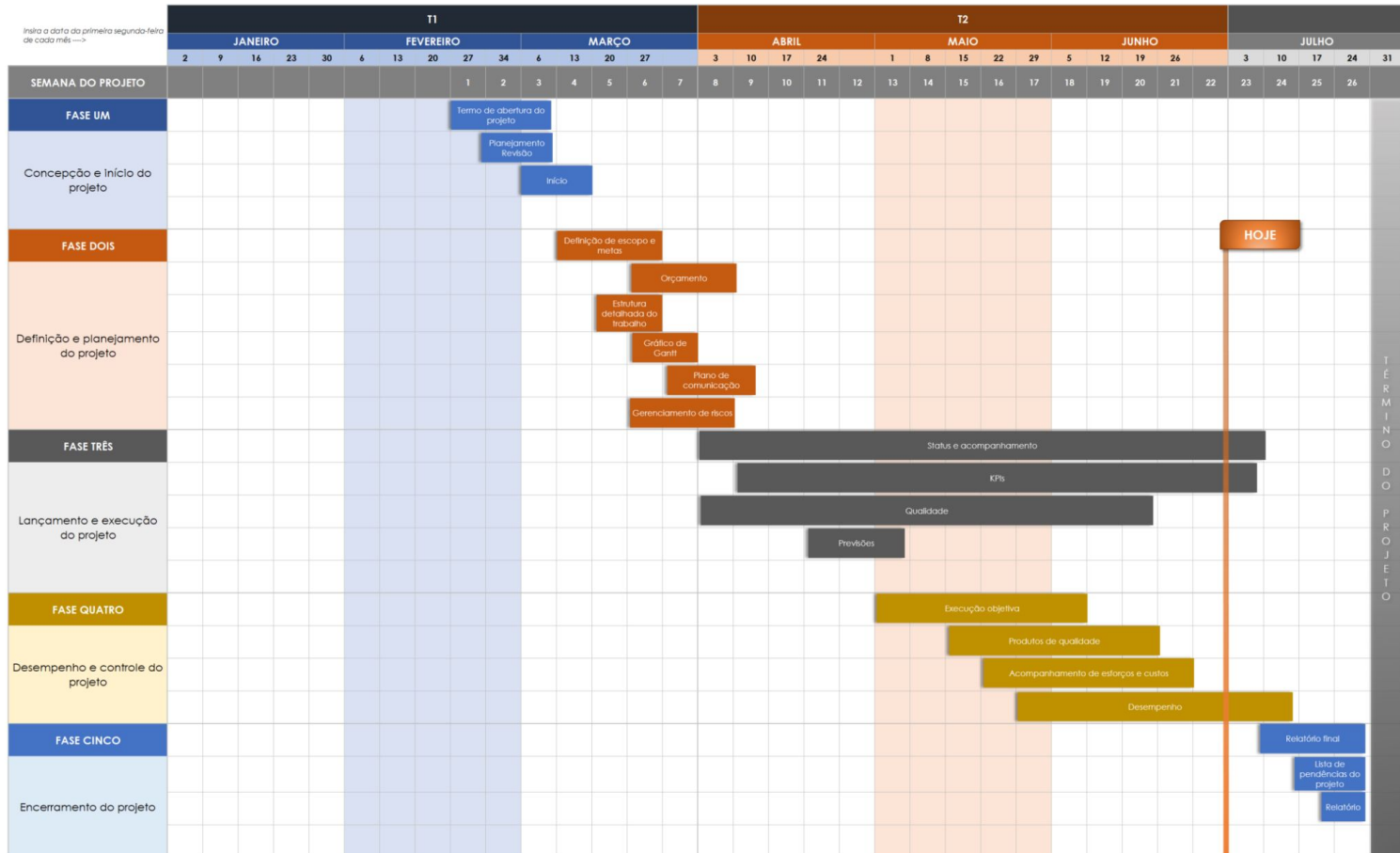
# Timeline x Cronograma

## Timeline for Change Management



# Timeline x Cronograma

## EXEMPLO DE MODELO DE CRONOGRAMA DE PROJETO



## Na aula de hoje...

Vamos focar no *Cronograma* na aula de hoje.

Aplicar técnicas para construção do Cronograma

# Marcos e Milestones

Projetos **sem prazos** definidos tendem ao fracasso.

O **tempo** é um dos três grandes pilares do gerenciamento de projetos (junto com custo e escopo).

Falhas no controle do cronograma geram **atrasos**, **retrabalho** e **custos** adicionais.

# O que é um cronograma?

"Modelo do plano do projeto que define as **datas** planejadas para a realização de atividades do projeto."

Auxilia a **monitorar** o progresso e garantir que as fases sejam **concluídas** corretamente.

# Diferenças importantes

<b>Termo</b>	<b>Definição rápida</b>
Cronograma	Planejamento com datas
Linha de base do cronograma	Versão aprovada usada para controle
Plano de gerenciamento do cronograma	Documento que define como o cronograma será planejado, desenvolvido e controlado

# Benefícios em utilizar Cronogramas

- Serve como guia visual e base para monitoramento.
- Ajuda na comunicação com stakeholders.
- Permite ajuste de recursos e prioridades.




# Benefícios em utilizar Cronogramas

- O cronograma depende da EAP (Estrutura Analítica do Projeto).
- Cada entrega precisa ser dividida em atividades com prazos.



# Como construir o cronograma?

- Planejar o gerenciamento do cronograma
- Definir as atividades
- Sequenciar as atividades
- Estimar os recursos
- Estimar a duração
- Desenvolver e controlar o cronograma




# Como construir o cronograma?

- Planejar o gerenciamento do cronograma 
- Definir as atividades
- Sequenciar as atividades
- Estimar os recursos
- Estimar a duração
- Desenvolver e controlar o cronograma

# Como construir o cronograma?

- Planejar o gerenciamento do cronograma 
- Definir as atividades 
- Sequenciar as atividades
- Estimar os recursos
- Estimar a duração
- Desenvolver e controlar o cronograma

# Como construir o cronograma?

- Planejar o gerenciamento do cronograma 
- Definir as atividades 
- Sequenciar as atividades 
- Estimar os recursos
- Estimar a duração
- Desenvolver e controlar o cronograma

# Sequenciar Atividades

- Estabelece dependências (predecessores)
- Determinar a melhor ordem de execução das tarefas.
- Garantir que dependências técnicas, lógicas e de recursos sejam respeitadas.
- Criar a base para o caminho crítico e cronograma do projeto.

# Tipos de Dependência

- **Obrigatórias** (Hard Logic): Derivadas da natureza do trabalho (ex: não dá pra testar sem antes codificar).
- **Discrecionárias** (Soft Logic): Baseadas na experiência ou boas práticas (ex: fazer primeiro a interface antes do backend).
- **Externas**: Fora do controle do projeto (ex: entrega de um fornecedor).
- **Internas**: Controladas pela equipe (ex: revisão antes da publicação).

# Tipos de Dependência

- Método do Diagrama de Precedência (PDM – Precedence Diagramming Method)
- Conecta atividades com base nos tipos de dependência:
  - Finish-to-Start (FS) – mais comum
  - Start-to-Start (SS)
  - Finish-to-Finish (FF)
  - Start-to-Finish (SF)



## Finish-to-Start (FS) – Finalizar para começar

- A tarefa sucessora só pode começar quando a predecessora terminar.
- Exemplo:
  - Desenvolver código do sistema (Tarefa A) → Testar funcionalidade (Tarefa B)
  - O teste só pode começar após o término da codificação.

## Start-to-Start (SS) – Começar junto

- A tarefa sucessora só pode começar quando a predecessora começar (ambas iniciam próximas uma da outra).
- Exemplo:
  - Criar layout visual (Tarefa A) → Implementar layout no front-end (Tarefa B)
  - Assim que o design começa a ser feito, a implementação já pode ir sendo iniciada.

## Finish-to-Finish (FF) – Finalizar junto

- A tarefa sucessora só pode terminar quando a predecessora terminar (ambas precisam ser concluídas simultaneamente).
- Exemplo:
  - Revisar documentação técnica (Tarefa A) → Finalizar entrega do projeto (Tarefa B)
  - O projeto só pode ser entregue quando a revisão estiver concluída.





## Start-to-Finish (SF) – Começar para finalizar

- A tarefa sucessora só pode terminar quando a predecessora começar (raro e geralmente em contextos operacionais).
- Exemplo:
  - Novo turno de suporte inicia (Tarefa A) → Encerrar turno anterior (Tarefa B)
  - O turno atual só encerra quando o próximo começar, garantindo continuidade.

# Organizar as dependências

ID	Atividade	Predecessores	Tipo
A	Criar especificação	-	-
B	Desenvolver sistema	A	FS
C	Testar sistema	B	FS
D	Criar manual	A	SS
E	Treinar usuários	C, D	FF / FS

# Como construir o cronograma?

- Planejar o gerenciamento do cronograma 
- Definir as atividades 
- Sequenciar as atividades 
- Estimar os recursos 
- Estimar a duração
- Desenvolver e controlar o cronograma

# Estimar Recursos

- Determinar quais recursos serão necessários (humanos, tecnológicos, materiais) e quanto tempo cada atividade levará, considerando esses recursos.

# Estimar Recursos






- Pessoas (desenvolvedores, analistas, designers, testers)
- Ferramentas (IDE, servidores, softwares pagos)
- Infraestrutura (máquinas, ambientes de teste)



# Estimar Recursos

Atividade	Recursos estimados
Criar protótipo de interface	1 UX designer + 1 ferramenta Figma
Desenvolver API REST	2 devs backend (nível pleno)
Escrever testes automatizados	1 QA engineer + framework Selenium
Implantar sistema em produção	1 devops + servidor em nuvem (AWS)

# Como construir o cronograma?

- Planejar o gerenciamento do cronograma 
- Definir as atividades 
- Sequenciar as atividades 
- Estimar os recursos 
- Estimar a duração 
- Desenvolver e controlar o cronograma

# Estimar Duração

A duração depende de:

- Quantidade de recursos
- Nível de habilidade
- Complexidade da tarefa
- Produtividade esperada

# Estimar Duração

Técnicas comuns:

- ***Analógica:*** baseada em projetos anteriores
- ***Estimativa por especialista:*** consulta a devs experientes.
- ***Análise paramétrica:*** baseia-se em métricas históricas (ex: “3 telas/dia por dev”).
- ***Três pontos (PERT):*** média ponderada entre cenários otimista, mais provável e pessimista.

# Estimar Duração

## Exemplo prático (PERT)

- Tarefa: Desenvolver módulo de login
- Otimista (O): 2 dias
- Provável (M): 4 dias
- Pessimista (P): 12 dias

## Fórmula:

- $Duração = (O + 4M + P) / 6 \rightarrow (2 + 4 \times 4 + 12) / 6 = 5$   
dias

# Caminhos Críticos e Folgas

- Caminho crítico: Sequência **mais longa** de atividades que determina a duração mínima do projeto.
- Se qualquer tarefa do caminho crítico atrasar, o projeto como um todo atrasará.

# Caminhos Críticos e Folgas

- Sem folga (float = 0)
- Define o prazo final do projeto
- Atividades críticas devem ser monitoradas de perto.

# Caminhos Críticos e Folgas

- Ajuda a priorizar esforços
- Suporte à tomada de decisão (ex: onde acelerar, onde aplicar recursos extras)



# Exemplo de Construção De Caminho Crítico

Tarefa	Descrição	Duração (dias)	Predecessora
A	Levantamento de requisitos	3	-
B	Design da interface	4	A
C	Desenvolvimento backend	5	A
D	Integração front-backend	2	B, C
E	Testes	3	D
F	Deploy	1	E

# Exemplo de Construção De Caminho Crítico

A (3)	/ \	<b>Tar efa</b>	<b>Descrição</b>	<b>Duração (dias)</b>	<b>Predeces sora</b>
B (4) C (5)	\ /	A	Levantamento de requisitos	3	-
D (2)		B	Design da interface	4	A
E (3)		C	Desenvolvimento backend	5	A
F (1)		D	Integração front-backend	2	B, C
		E	Testes	3	D
		F	Deploy	1	E

# Exemplo de Construção De Caminho Crítico

**A (3)**

<b>Tarefa</b>	<b>Descrição</b>	<b>Duração (dias)</b>	<b>Predecessora</b>
<b>A</b>	<b>Levantamento de requisitos</b>	<b>3</b>	-
B	Design da interface	4	A
C	Desenvolvimento backend	5	A
D	Integração front-backend	2	B, C
E	Testes	3	D
F	Deploy	1	E

# Exemplo de Construção De Caminho Crítico

<p>A (3)</p> <p>/</p> <p><b>B (4)</b></p>	Tarefa	Descrição	Duração (dias)	Predecessora
	A	Levantamento de requisitos	3	-
	<b>B</b>	<b>Design da interface</b>	<b>4</b>	<b>A</b>
	C	Desenvolvimento backend	5	A
	D	Integração front-backend	2	B, C
	E	Testes	3	D
	F	Deploy	1	E

# Exemplo de Construção De Caminho Crítico

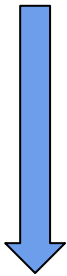
A (3) / \		Tarefa	Descrição	Duração (dias)	Predecessora
B (4) <b>C (5)</b>		A	Levantamento de requisitos	3	-
		B	<b>Design da interface</b>	4	A
		C	<b>Desenvolvimento backend</b>	5	A
		D	Integração front-backend	2	B, C
		E	Testes	3	D
		F	Deploy	1	E

# Exemplo de Construção De Caminho Crítico

A (3)

/ \

**B (4) C (5)**



PARALELISMO

Tarefa	Descrição	Duração (dias)	Predecessora
A	Levantamento de requisitos	3	-
<b>B</b>	<b>Design da interface</b>	<b>4</b>	<b>A</b>
<b>C</b>	<b>Desenvolvimento backend</b>	<b>5</b>	<b>A</b>
D	Integração front-backend	2	B, C
E	Testes	3	D
F	Deploy	1	E

# Exemplo de Construção De Caminho Crítico

A (3)		Tar efa	Descrição	Duração (dias)	Predeces sora
/	\				
B (4)	C (5)	A	Levantamento de requisitos	3	-
\	/	B	Design da interface	4	A
<b>D (2)</b>		C	Desenvolvimento backend	5	A
		<b>D</b>	<b>Integração front-backend</b>	<b>2</b>	<b>B, C</b>
		E	Testes	3	D
		F	Deploy	1	E

# Exemplo de Construção De Caminho Crítico

A (3)		Tar efa	Descrição	Duração (dias)	Predeces sora
/	\				
B (4)	C (5)	A	Levantamento de requisitos	3	-
\	/	B	Design da interface	4	A
D (2)		C	Desenvolvimento backend	5	A
		D	Integração front-backend	2	B, C
E (3)		E	<b>Testes</b>	<b>3</b>	<b>D</b>
		F	Deploy	1	E




# Exemplo de Construção De Caminho Crítico

A (3)	/ \	<b>Tar efa</b>	<b>Descrição</b>	<b>Duração (dias)</b>	<b>Predeces sora</b>
B (4) C (5)	\ /	A	Levantamento de requisitos	3	-
D (2)		B	Design da interface	4	A
E (3)		C	Desenvolvimento backend	5	A
<b>F (1)</b>		D	Integração front-backend	2	B, C
		E	Testes	3	D
		<b>F</b>	<b>Deploy</b>	<b>1</b>	<b>E</b>

## Exemplo de Construção De Caminho Crítico

- Caminho 1:  $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F = 3 + 4 + 2 + 3 + 1$   
13 dias
- Caminho 2:  $A \rightarrow C \rightarrow D \rightarrow E \rightarrow F = 3 + 5 + 2 + 3 + 1$   
14 dias

 Caminho Crítico =  $A \rightarrow C \rightarrow D \rightarrow E \rightarrow F = 14$  dias

# Folgas

- Folga (float): quanto uma tarefa pode atrasar sem impactar o fim do projeto.
- Folga Total (*total float*)
- Folga Livre (*free float*)

# Folga Total

Tempo que uma atividade pode ser atrasada sem afetar a **data de término do projeto**.

Útil para identificar flexibilidade no planejamento.

# Folgas

Caminho 1:  $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F = 3 + 4 + 2 + 3 + 1$

13 dias

Caminho 2:  $A \rightarrow C \rightarrow D \rightarrow E \rightarrow F = 3 + 5 + 2 + 3 + 1$

14 dias

→ B pode atrasar até 1 dia sem impactar a data final do projeto.

# Folga Livre

Tempo que uma atividade pode atrasar sem atrasar o início da **próxima atividade imediata**.

# Folgas

Caminho 1:  $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F = 3 + 4 + 2 + 3 + 1$

13 dias







Caminho 2:  $A \rightarrow C \rightarrow D \rightarrow E \rightarrow F = 3 + 5 + 2 + 3 + 1$

14 dias

Folga Livre da atividade B = 1 dia

→ B pode atrasar até 1 dia sem atrasar o início da tarefa D, porque **D só pode começar após C (que termina no dia 8).**

# Como construir o cronograma?

- Planejar o gerenciamento do cronograma 
- Definir as atividades 
- Sequenciar as atividades 
- Estimar os recursos 
- Estimar a duração 
- Desenvolver e controlar o cronograma 



# Controlar o Cronograma

- Processo de monitorar o andamento do projeto para garantir que ele esteja seguindo o cronograma planejado, e realizar ajustes quando necessário.

# Controlar o Cronograma

- Medir o progresso real vs. planejado
- Atualizar datas de início/fim
- Gerenciar mudanças aprovadas no cronograma
- Identificar desvios e atuar corretivamente

# Controlar o Cronograma

- Linha de base do cronograma
- Sistemas de controle de mudanças
- Relatórios de desempenho

Exemplo: Durante a Sprint 3, a equipe percebe que a integração com API externa atrasou. O gerente reprograma a atividade de testes e comunica o impacto no marco de homologação.

# DÚVIDAS?