

Gestão de projeto / produtos informáticos

Planeamento ágil de produtos de software

cedric.grueau@estsetubal.ips.pt

Planeamento ágil de produtos de software

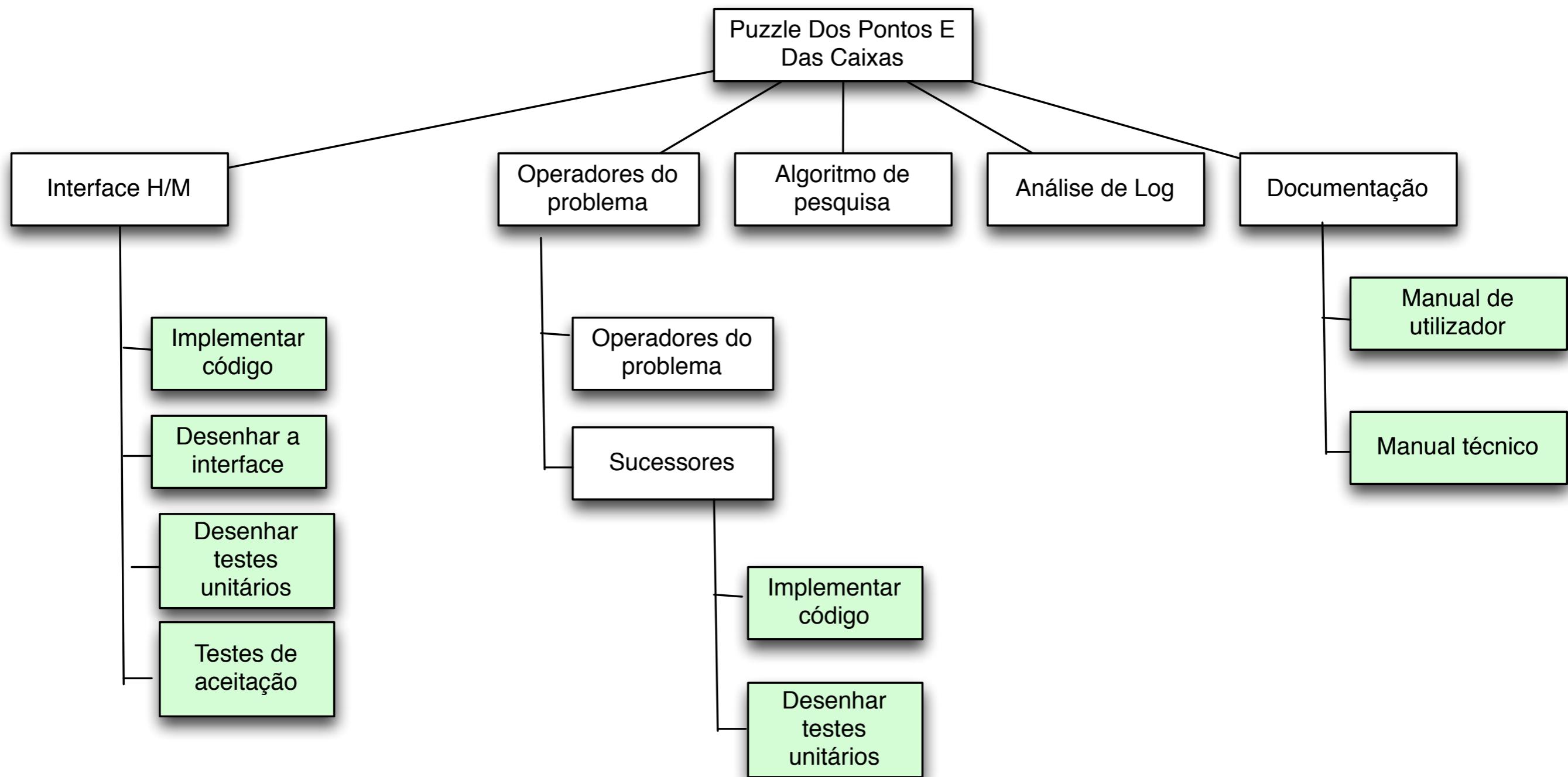
Estimativas

Objetivos de aprendizagem

- Definir e criar um ponto de história.
- Identificar diferentes maneiras de dimensionar uma tarefa.
- Explique o que é a velocidade, como calcular uma estimativa de velocidade, quais fatores influenciam a velocidade e o que torna a velocidade estável.
- Explicar o conceito de *feito*, e como uma história é contada para a velocidade, se for feita.
- Definir o conceito de time boxing e sua relação com o processo Scrum.
- Definir o conceito de lançamento (release).
- Explicar e criar um Gráfico de Gantt.
- Explicar e criar um plano de lançamento.

Exercício - estimativas

- De acordo com a lista de tarefas correspondentes ao desenvolvimento do projeto nº1 de IA, elaborar uma estimativa do tempo necessário para elaborar cada tarefa assinalada a verde.



Estimativas vs Pontos de História

Story points

- As estimativas são facilmente confundidas com compromissos porque ambas são medidas em unidades de tempo. É fácil concluir que o tempo sugerido numa estimativa pode ser simplesmente convertido em horas de trabalho. No entanto, isso não é muito preciso, e deve ser evitado.
- Os pontos de história foram desenvolvidos como um **sistema alternativo** às estimativas.
- Em vez de estimar quanto tempo falta para terminar o trabalho, os pontos de história são valores atribuídos às histórias para indicar quanto tempo será necessário para terminar uma parte de trabalho, **em relação a outras partes de trabalho**.
- Os pontos de história são, portanto, **medidas relativas**, e **sem unidades**, ao contrário de estimativas de tempo. Nas estimativas de tempo, horas ou dias são as unidades utilizadas, enquanto que em pontos de história, os pontos não são representativos de qualquer medida particular.

Como usar pontos de história

User stories:

- Descrições curtas, simples e formatadas de uma funcionalidade, descrita na perspectiva da pessoa que deseja a funcionalidade.
 - Formato:
 - “As a ‘__[1]__’, I want to ‘__[2]__,’ So that ‘__[3]__’”
 - “Como ‘__[1]__’, quero ‘__[2]__,’ de modo a ‘__[3]__’”
1. Persona
 2. Tarefa a desempenhar com a aplicação
 3. Problema / dificuldade a resolver

User Stories

SERVICE CARD IS A REMINDER OF A CONVERSATION THAT IS TO BE ABLE TO SUPPORT THE DRAFT UNTIL DRAFT IS APPROVED

* DIAGNOSTIC SCREEN:
- CONFIRMATION:
- ACCEPTANCE CRITERIA
TO SAY WE ARE DONE

User stories



Lista de histórias - backlog

Master Story List  

File Edit View Insert Format Data Tools Add-ons Help Last edit was 1 hour ago

Print  Refresh  Undo  Redo  \$ % .0 .00 123 Arial 10 **B** *I* A               

ID

	A	B	C	D	E	F	G	H	I
1	ID	Epic	Theme	As a...	I want...	So that...	90% CI lower	90% CI upper	Priority
2	1	Warp capability	Propulsion	Engineer	accelerate to warp speed	escape romulans	3	5	1
3	2	Warp capability	Propulsion	Navigator	know the velocity	predict arrival time	2	3	2
4	3	Warp capability	Nobody dies	Science officer	service the warp drive	ship can stay running	4	4	1
5	4	Warp capability	Nobody dies	Crewmember	protection from acceleration effect	not crushed to death	4	5	1
6	5	Warp capability	Fuel management	Engineer	store fuel onboard safely	refuel intervals reasonable	5	3	1
7	6	Warp capability	Fuel management	Captain	refuel easily	no need for extensive deviation	2	2	2

Como usar pontos de história

Backlog de produto - Scrum

- Conjunto ou **lista de funcionalidades** (features) de software que o gestor de produto de software e a equipa de desenvolvimento planeiam desenvolver para um produto, ao longo de um projeto.
- Escolher uma história e atribuir um valor de pontos
- Atribuir sucessivamente pontos as restantes histórias
 - exemplo: se uma tarefa for duas vezes maior do que outra, deve atribuir sensivelmente o dobro dos pontos.
 - O número total dos pontos permite ter uma ideia da extensão do projeto

Como usar pontos de história

- Não transformar os pontos em horas
- Não devem ser muito precisos mas sim relativos
- valores altos 10, 50, ou 100 não devem ser usados sob o risco de serem confundidos com percentagens.
- Usar valores fixos
- Conselho: usar a série de Fibonacci: 1, 1, 2, 3, 5, 8, 13, 21, etc.
 - Por exemplo, se tiver uma história que é aproximadamente o dobro de trabalho de outra história a qual foi atribuído o valor de 2, deverá atribuir-lhe o valor 5, e não 4.

Vantagens dos pontos de história

- Parecem levar a estimativas **mais precisas** (porque são relativas)
- **Menos estressante** para o programador: é difícil de associar uma “deadline” a um valor de pontos

Limitações dos pontos de história

- Não são aceites por todas as equipas:
 - Requerem a mudança da utilização de tempo para a utilização de pontos
 - Tentação de tratar os pontos como horas
 - Solução: aumentar a prática
- As estimativas são sempre difíceis de fazer de forma precisa. Mesmo com pontos é possível errar na estimativa
- Com equipas de maior dimensão, existe a tentação de inflacionar os pontos para dar sensação de produtividade. Se acontecer a meio de um projeto, os pontos perdem utilidade para estimar.
- **A consistência é essencial com os pontos de história.**

Outras unidades de estimativa

- jellybeans
- t-shirt sizing
- Casos de uso
- Pontos de função (Métricas de software deduzidas do código-fonte pela análise de algoritmo WMFP)
- COCOMO (modelo de regressão com base em projetos já desenvolvidos):
- parmetros de SLIM
- Cuidado: relativas vs absolutas!

https://en.wikipedia.org/wiki/Software_development_effort_estimation

Exercício 2 - estimativas com pontos de história

- Considere as 4 histórias seguintes e atribui pontos a cada uma delas, de acordo com o esforço envolvido
 - “Como utilizador, quero visualizar os puzzles de modo que possa escolher qual será resolvido.”
 - “Como avaliador, quero ter acesso as métricas de avaliação de cada procura de forma que possa avaliar se o algoritmo funciona de forma correta”
 - “Como utilizador, quero procurar a melhor solução possível para um determinado puzzle”
 - Como utilizador, quero adicionar ficheiros de forma a resolver novos puzzles”.

Estimativas de velocidade

- **Velocidade** : trabalho efetuado ao longo de um período de tempo
- Unidade: sprint
- Velocidade = número de pontos de história completados num período de tempo:
 - e.g. 15 pontos de história num sprint

Estimativas de velocidade

Útil para :

- **Estimar o esforço** da equipa de desenvolvimento ao longo do projeto
- Estimar o esforço que pode ser alcançado para trabalhos futuros
- Rastrear de forma indireta a produtividade da equipa

Estimativas de velocidade com pontos de história

- Como medir a velocidade em SCRUM e método ágeis?
 - Considerar que a história está completamente implementada:
 - **DONE** = deve passar os testes unitários **E** os testes de aceitação **E** documentada
- Não se deve contar os pontos de histórias que ainda não estão DONE mesmo se estiverem codificada (-> permite incentivar o fecho das histórias).

Desenvolvimento orientado pela velocidade

- Usar a velocidade de iterações anteriores para planear outras iteração de um projeto
 - Uma iteração de 15 pontos de história pode ser usada para planear outras iterações de dimensão similar
- É necessário haver estabilidade nas iterações anteriores
 - No arranque do projeto, é mais difícil determinar a velocidade

Desenvolvimento orientado pela velocidade

- 1^a situação:
 - No arranque do projeto, quando é mais difícil determinar a velocidade porque não existe trabalho prévio
- Solução:
 - Consultar peritos
 - Usar como base a experiência anterior da equipa para fazer estimativas

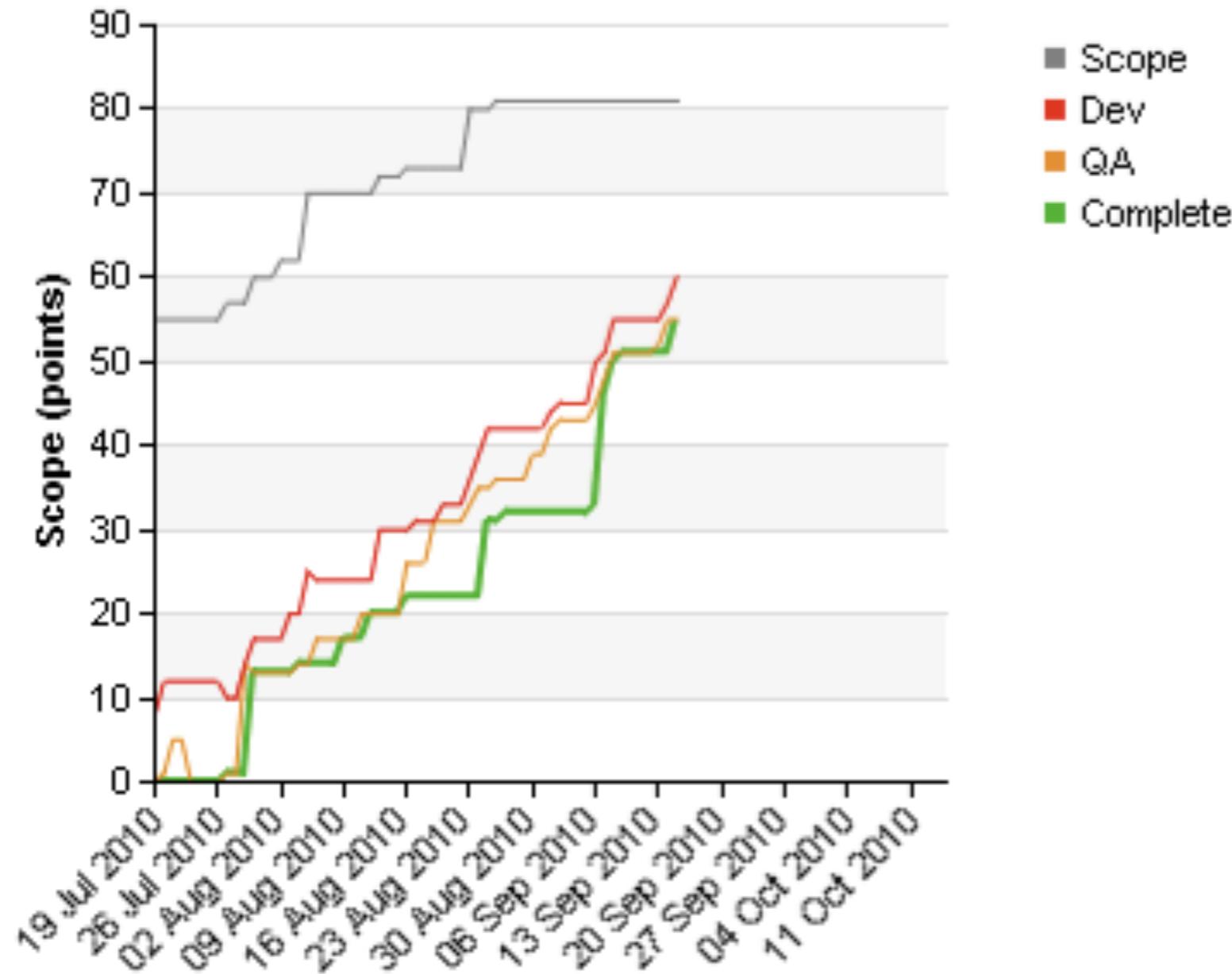
Desenvolvimento orientado pela velocidade

- 2^a situação:
 - A meio do projeto quando é + fácil estimar com base em trabalho anterior
- Solução : análise de dados prévios
 - **Extrapolação linear** (estendendo dados num gráfico para além dos dados conhecidos numa linha reta com base no comportamento dos dados anteriores),
 - A **média da velocidade** dos três sprints anteriores,
 - Usando a **velocidade mínima** dos três sprints anteriores.

Algumas considerações acerca da utilização dos pontos de história

- Limitações
 - Controversa
 - Não é precisa: a velocidade varia
 - Importância de só contabilizar as histórias DONE
 - Permanece uma estimativa
- Vantagens
 - Boa abordagem para o longo prazo (aumento da consistência)
 - **Boa para a tomada de decisão** (mesmo se não mede a produtividade individual)
- Alternativa
 - Desenvolvimento orientado pela aprendizagem

Rastrear o progresso



Cumulative Flow Diagram (CFD) / burn up chart / turndown chart / finger diagram

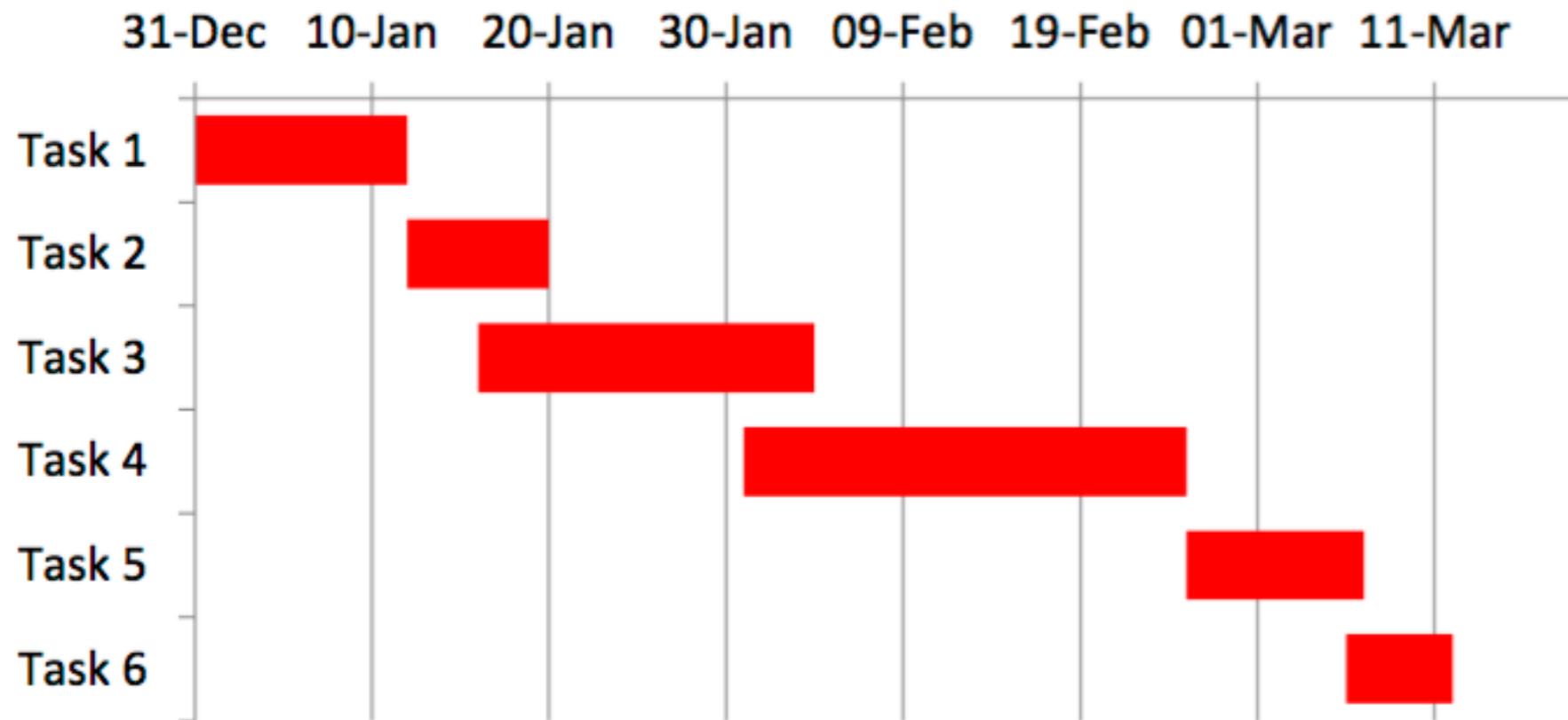
Time Boxing

- Período de **tempo restrito** em que algo deve ser construído, ou durante o certas tarefas devem ser concluídas
- O tempo é estritamente limitado - o evento deve permanecer dentro de um prazo pré-determinado.
- Particularmente importante para Scrum.
- Existe
 - um trabalho de **planeamento** no início do período de tempo e
 - um **espaço de reflexão** sobre o andamento do projeto no fim do período.

Time Boxing com SCRUM

- Duas a quatro semanas de duração.
- Quantidade relativamente pequena de trabalho que é restrito a um tempo de desenvolvimento relativamente curto.
- Períodos de planeamento e de revisão (retrospectiva) antes e depois do trabalho realizado e revisto.
- Ajuda a manter um projeto no caminho certo e num ritmo regular e sustentável, porque divide o projeto em pedaços mais fáceis de gerir.
- Os objetivos e metas gerados pelo período permitem manter o escopo e seguir o cronograma.
- Ajuda a determinar metas futuras.
- Cuidado em não incluir muitas tarefas numa iteração.
- Permite a equipa **aprender** com o trabalho dividido em intervalos regulares, por causa dos períodos de revisão.
- Um projeto pode ser lançado em etapas, no final de cada período: no final de cada período, uma versão funcional do software deve ser criada, que pode ser lançado como um produto.
- Permite manter o foco e o moral da equipa

Gráficos de Gantt



- tarefas no eixo esquerdo (y)
- tempo no eixo horizontal (x)

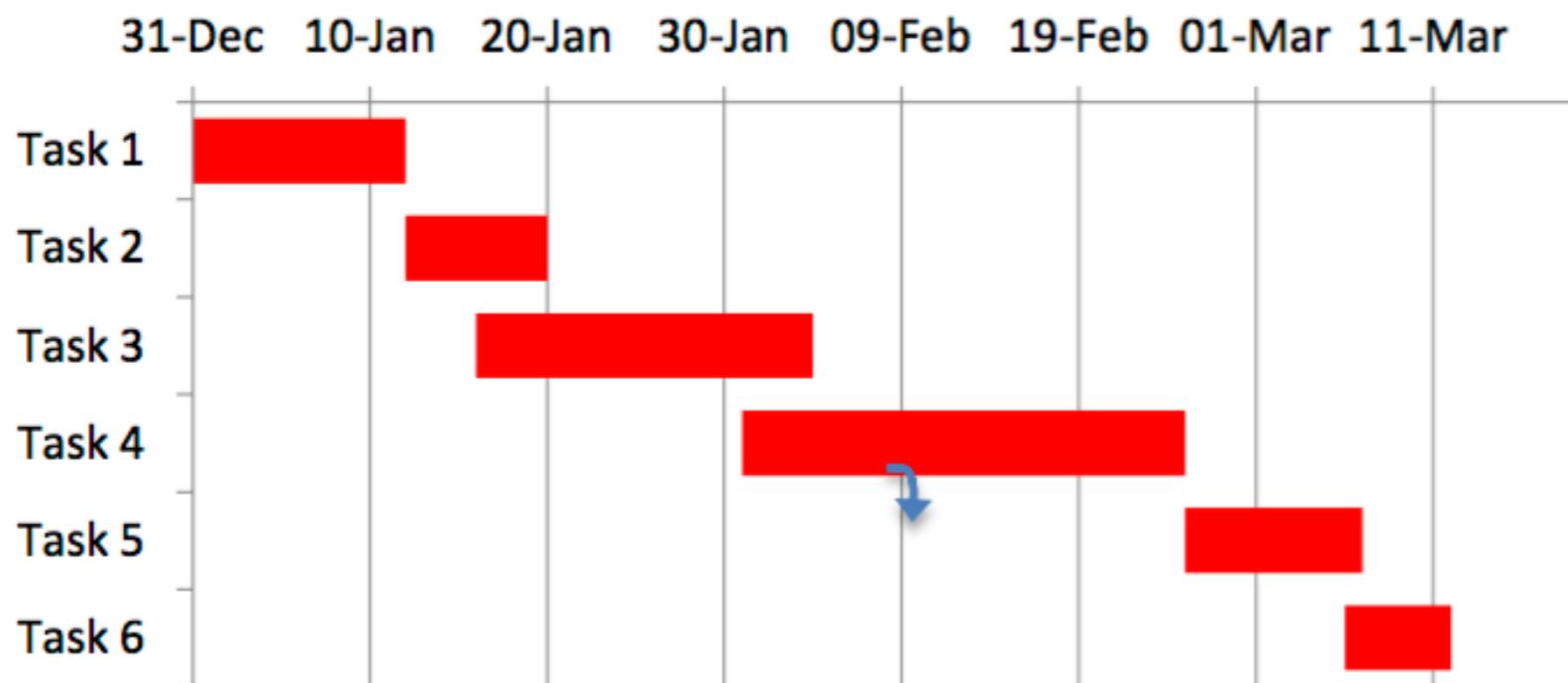
Gráficos de Gantt

Com métodos ágeis:

- Contexto adicional deve ser fornecido para tarefas (como identificar se há dependências de tarefas)
- Partes do gráfico podem ser seccionadas como sprints
- Os gráficos devem ser usados com uma compreensão de flexibilidade e adaptação ao projeto
- Mudança de Requisitos (em outras palavras, o gráfico pode mudar conforme o projeto se desenrola)

Dependências entre tarefas

- Para mostrar as **dependências** entre tarefas no gráfico de GANTT:
 - deve ser criado um backlog de tarefas do projeto com prioridades e identificação das dependências.



- Outras técnicas: Critical Path Method (CPM) charts and Program Evaluation Review Technique (PERT)

Ferramentas

- <http://www.ganttproject.biz/>
- Microsoft Excel
- Microsoft Powerpoint
- Microsoft project
- muitos outras ferramentas online de gestão de projeto
- ...

Planos de lançamento (release)

- O planeamento é um processo que pode ocorrer no nível de iteração e no nível de release:
planeamento a dois níveis.
- O planeamento de iteração: lida com tarefas a um nível detalhado
- O planeamento de release (lançamento) lida com histórias de utilizadores: nível mais abrangente.
- Uma unidade comum: o sprint.
- **Plano de iteração:**
 - gerado no início de um sprint nos métodos Scrum e Agile
 - o programador planeia as tarefas a serem concluídas durante esse sprint específico.
 - As tarefas são então auto-atribuídas pelos programadores na equipa antes do início do trabalho.
 - Output: **software** que pode ser mostrado ao cliente durante o evento de revisão.

Planos de lançamento (release)

- Usados para **planejar alguns sprints** de uma só vez, determinando quais histórias de utilizador do backlog do projeto devem ser concluídas e lançadas em qual sprint.
- Anterior aos planos de iteração
- As histórias de utilizadores são escolhidas **por prioridade**.
- O plano é **feito antes que qualquer sprint** seja iniciado - portanto, ocorre antes do planejamento da iteração.
- As histórias de utilizadores devem ser **distribuídas de forma lógica e uniforme** em sprints; Caso contrário, eles podem ser concluídos de forma desorganizada ou aleatória, o que pode causar problemas com dependências de tarefas.

Gestão dos requisitos - histórias de utilizadores

Remember MoSCoW ?

1. “Must do”(prioridade máxima)
2. “Should do”(prioridade média)
3. “Could do” (baixa prioridade)

“Won’t have”?

Critérios de atribuição das histórias de utilizadores

- A **experiência anterior** pode ajudar a informar quantas histórias de utilizadores podem ser concluídas num sprint
- Dados como **estimativas de velocidade e pontos de história** podem ser usados.
- A **velocidade** pode indicar quantos pontos de história podem ser concluído num sprint.
- As **Histórias** podem ser adicionadas até atingir o limite de pontos de história que o projeto permite.

Planeamento de releases

- Deve fornecer uma **projeção precisa do que será o produto** no final de uma próxima série de sprints.
- Um bom plano de lançamento **ajuda com as expectativas** do cliente e
- **Mantém o foco e o moral do programador** porque os resultados tangíveis são planeados e atingidos progressivamente.

Planeamento ágil de produtos de software

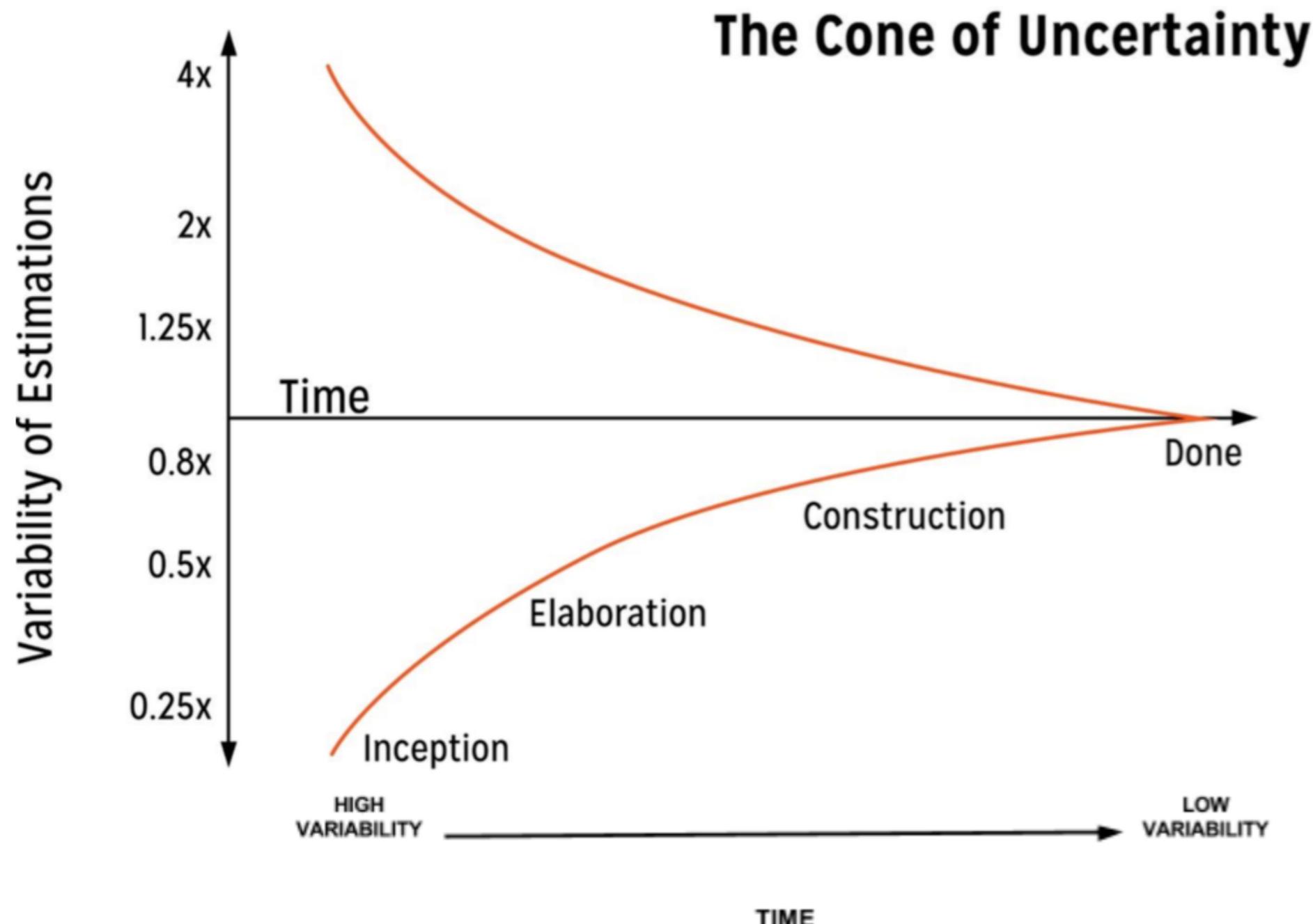
Planeamento

Objetivos de aprendizagem

- Descrever o que faz uma boa estimativa.
- Explicar questões que podem afetar a estimativa, incluindo o conceito de Cone of incerteza.
- Explicar diferentes abordagens para a criação de estimativas, tais como, de baixo para cima, analogia e Especialistas.
- Ser capaz de calcular o tempo de estimação da tarefa usando a fórmula de estimativa.
- Compreender dependências de tarefas e os diferentes tipos de dependências de tarefas: start-start, start-finish, finish-start,
- Explicar o método do gráfico de método de caminho crítico (CPM).
- Explicar o conceito de folga.
- Explicar e criar um gráfico PERT.
- Explicar as diferenças entre um gráfico CPM e um gráfico PERT.
- Explicar e criar um plano de iteração.
- Entender que o planeamento é um trabalho auto-designado feito pelos programadores e que não é atribuído pelos Clientes.

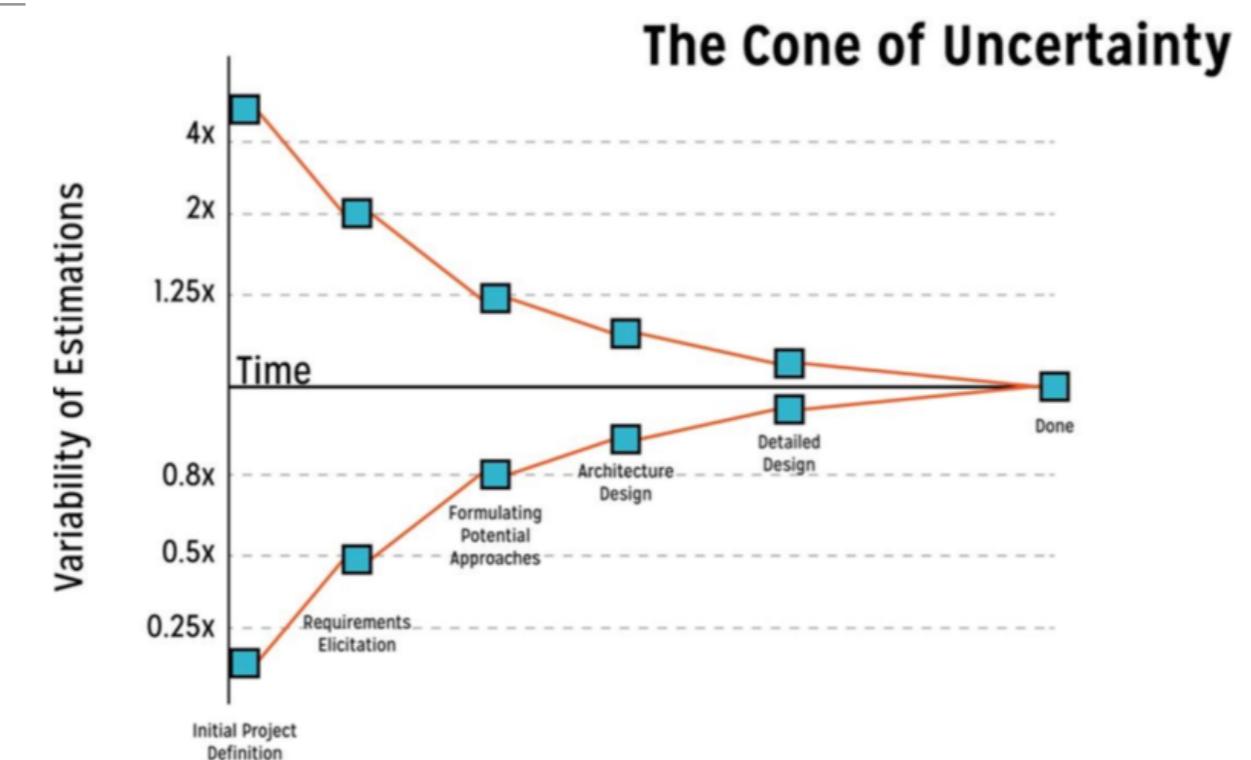
O cone da incerteza

A variabilidade refere-se aos fatores ou à extensão pela qual uma estimativa pode variar.



O cone da incerteza

- **Definição Inicial do Produto** - (formulação inicial da ideia do produto). Nesta fase, a variabilidade é elevada (entre o fator aproximado de 3 e 0,375), e as estimativas necessitarão de um grande intervalo.
- **Elicitação de Requisitos** - (quando as necessidades do produto e características foram identificadas). Nesta fase, a variação é menor (entre aproximadamente 2 e 0,5), e os intervalos de estimativa serão menores também.
- **Formulação das Abordagens Potenciais** - quando os métodos usados para fazer o projeto são escolhidos. Nesta fase, a variação (entre o fator aproximado de 1,25 e 0,8) e os intervalos de estimativa tornam-se ainda menores.
- **Desenho de Arquitetura e Desenho Detalhado**, quando a informação é conhecida sobre como e o que será criado no produto final. Nestas fases, a estimativa torna-se ainda mais fiável à medida que a variação diminui (a variação torna-se cada vez mais próxima de um factor de 1).



Intervalos de estimativas

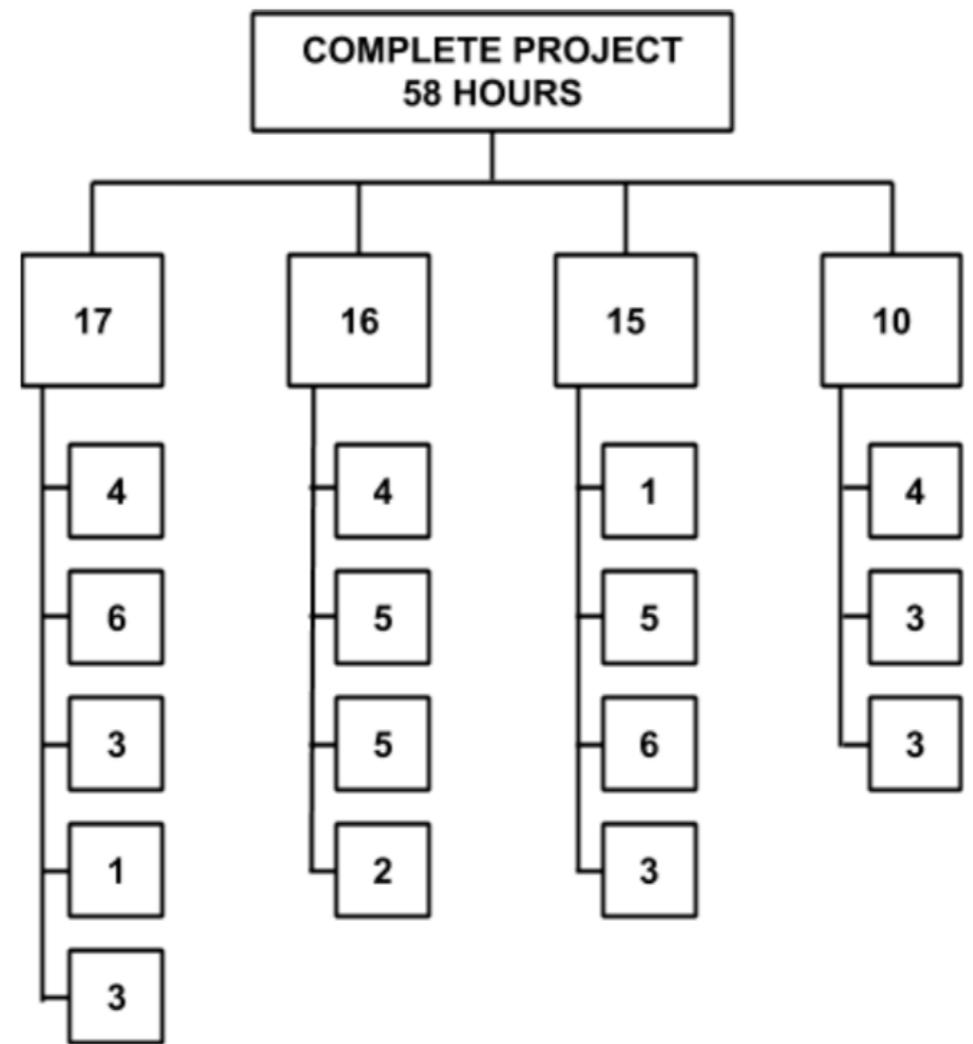
- Os intervalos de estimativa para o projeto podem ser avaliados usando a variabilidade ilustrada no gráfico:
 - A estimativa deve ser multiplicada pela sua variabilidade fatorial correspondente à fase em que se encontra o projeto. Isso permitirá determinar os valores superiores e inferiores do intervalo de estimativa.
 - Por exemplo, se uma tarefa é estimada a 6 horas na fase de elicitação de requisitos, isso significa que tem uma variabilidade de estimação de 2 a 0,5 no eixo y.
 - $2 \times 6 = 12$
 - $0,5 \times 6 = 3$
 - Isto dá um intervalo estimado de 3 a 12 horas.
 - Nesta fase, ainda há muita incerteza.
- Usar a variabilidade para criar intervalos de estimativa ajuda a garantir um intervalo mais preciso.
- As estimativas também devem ser continuamente revisadas à medida que o projeto se move através do Cone.

Criação de estimativas temporais

- 4 abordagens:
 - Bottom-up
 - Analogia
 - Peritos
 - Usar a formula de estimativa dos 3 pontos

Bottom up

- Baseada na estrutura dos produtos de trabalho
- Estimativa do tempo de cada tarefa individual
- Técnica **mais rápida** detidas se a informação está disponível
- pressupõe que nenhuma das tarefas na estrutura de divisão de trabalho (WBS) pode ser executada simultaneamente, mas sim que cada tarefa deve ser concluída antes que outra possa ser iniciada.



Analogia

- A estimativa de tempo é criada **comparando a tarefa ou projeto atual com uma experiência similar**. Se uma tarefa ou projeto semelhante foi realizado no passado com âmbito e custos semelhantes, então pode ser uma boa base para estimar o projeto atual.
- Esta técnica deve ser usada com alguma cautela:
 - Só pode funcionar se a **equipa for comum aos dois projetos**
 - O trabalho, o processo, as tecnologias e os membros da equipa devem ser iguais ou similares ao projeto atual para serem uma base significativa de dados.

Peritos

- Fusionar várias estimativas de peritos (antigos programadores que trabalharam em projetos semelhantes)
- Usada quando a equipa tem pouca experiência
- Tem um custo financeiro associada
- Pode ser bastante precisa

Formula da estimativa dos 3 pontos

- Estimativa do tempo de cada tarefa individual
- Técnica **mais rápida** se a informação está disponível
- Pressupõe que nenhuma das tarefas na estrutura de divisão de trabalho (WBS) pode ser executada simultaneamente, mas sim que cada tarefa deve ser concluída antes que outra possa ser iniciada.

Formula da estimativa dos 3 pontos

- Conhecida como a formula PERT de estimativa de 3 pontos
- Tempo mais provável (**T_m**): a estimativa amis provável do tempo que seria necessário para terminar a tarefa. Esta estimativa parte do pressuposto que o projeto ou a tarefa irão encontrar problemas normais. Qualquer um dos métodos de avaliação (baixo para cima, a analogia, ou especialistas) podem ser usados para gerar esta estimativa.
- Tempo otimista (**T_o**): É a melhor estimativa de cenário. Ele assume que tudo vai dar correr bem durante o projeto ou a tarefa e que a equipa de desenvolvimento irá trabalhar de forma eficiente. É, portanto, a menor quantidade de tempo na qual a tarefa ou o projeto poderia ser concluído.
- Tempo pessimista (**T_p**): É a pior das hipóteses. Assume-se que tudo ou quase tudo o que poderia correr mal ao longo do projeto ou da tarefa irá acontecer. É, portanto, a quantidade máxima de tempo na qual a tarefa ou o projeto poderia ser concluído.

Formula da estimativa dos 3 pontos

- Tempo expectável (**Te**):

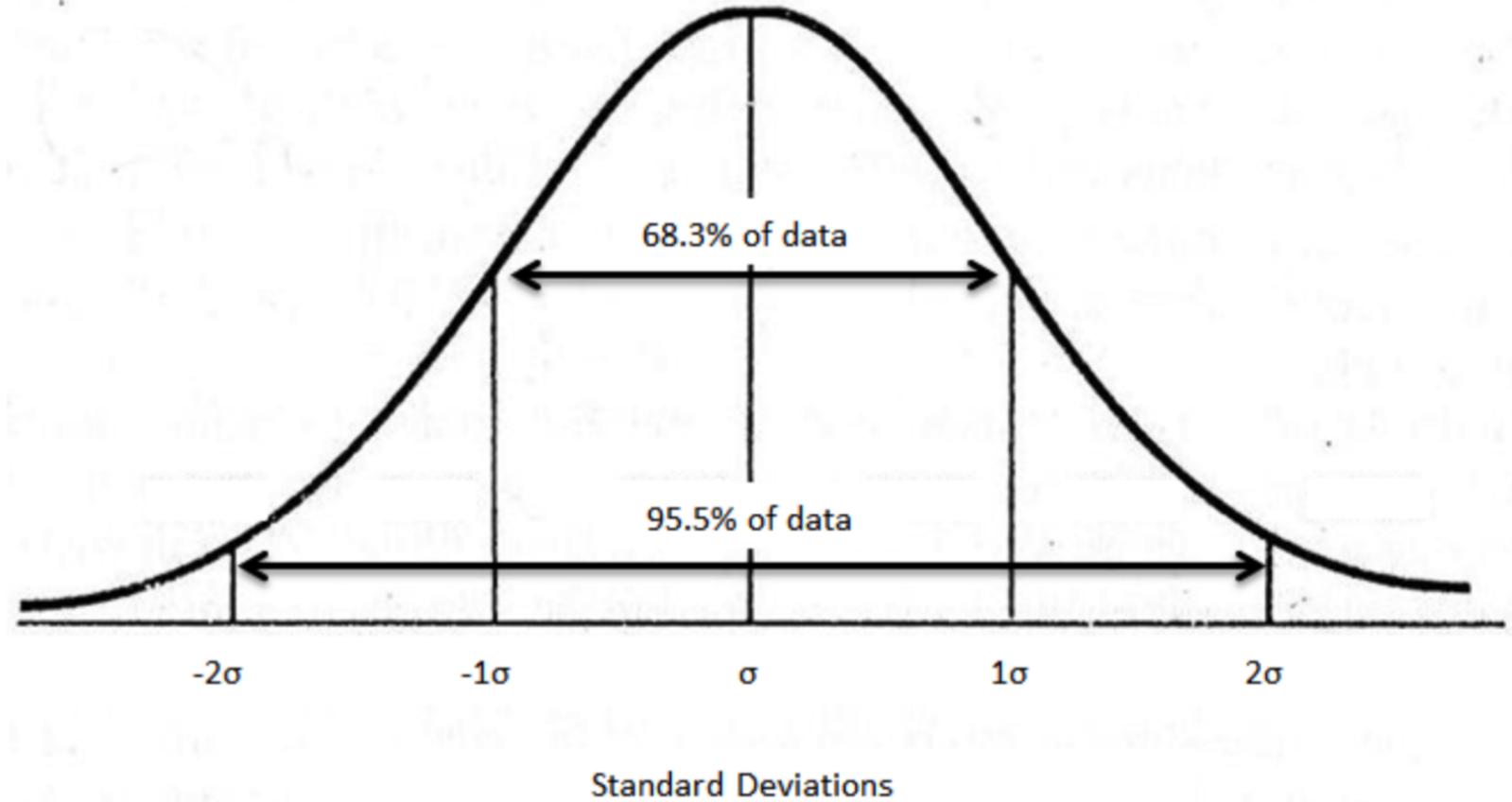
$$Te = \frac{(To+4Tm+Tp)}{6}$$

- Esta fórmula coloca ênfase no tempo mais provável que é multiplicado por 4, dando-lhe mais peso na fórmula. Depois, é calculada a média com o Tempo Optimista e o Tempo Pessimista.
- O tempo esperado é geralmente representado como um intervalo. Para criar um intervalo, precisamos de algumas informações sobre a probabilidade de que a estimativa esteja correta. Isto é feito calculando os desvios do Tempo Esperado.

Formula da estimativa dos 3 pontos

- Desvio padrão

$$\sigma = \frac{(Tp - To)}{6}$$



Os desvios são representados pela letra grega sigma (σ). O desvio padrão representa a forma como as medições podem ser distribuídas a partir do valor esperado. Estes são baseados em parcelas de distribuição normal com base em percentagens.

Uma distribuição normal pressupõe que os valores estão repartidas de acordo com o desvio padrão (neste caso, o Tempo Esperado) e que esses valores dentro de uma unidade de desvio (σ) do desvio padrão têm um grau de precisão provável de 68,3%. Se considerados duas unidades de desvio (2σ), há uma probabilidade de 95,5% do valor estar certo.

Exemplo

- $T_o = 5$ dias

$$T_e = \frac{(5 + (4 \times 8) + 17)}{6}$$

- $T_m = 8$ dias

$$T_e = \frac{(5 + 32 + 17)}{6}$$

- $T_p = 17$ dias

$$T_e = \frac{(54)}{6}$$

$$T_e = 9 \text{ dias}$$

$$\sigma = \frac{(17 - 5)}{6}$$

Desta forma, a estimativa do bloco de trabalho será de 7 a 11 dias, com um grau de certeza de 68,3%.

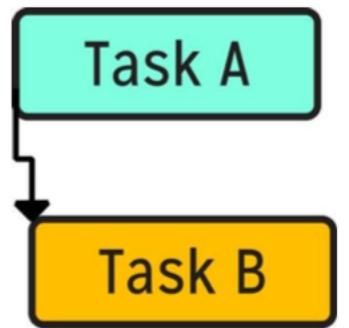
$$\sigma = \frac{(12)}{6}$$

$$\sigma = 2$$

Notas sobre o método dos 3 pontos

- Se o resultado da estimativa for um valor real, convém arredondar o valor calculado (ex: 6,1 para 6 dias)
- Para o cálculo da estimativa, é por vezes usada uma outra fórmula:

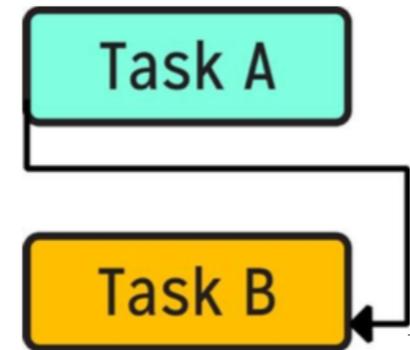
$$Te = \frac{(To+Tm+Tp)}{3}$$



Dependência entre tarefas - start-start

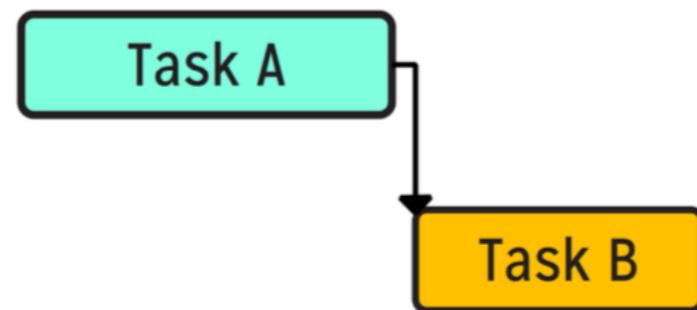
- Aplicam-se as dependências onde a primeira tarefa deve ser iniciada antes que a segunda tarefa possa ser iniciada. A altura em que estas tarefas terminam neste tipo de dependência não é importante.
- As dependências start-start são representadas visualmente da forma acima descrita, com uma seta que vai desde o início da Tarefa A até o início da Tarefa B. As duas tarefas são mostradas mais ou menos em paralelo. A tarefa B depende do início da Tarefa A.
- Este método de representação pode ser usado em qualquer tipo de representações visuais, incluindo gráficos de Gantt e PERT.
- Um exemplo de uma dependência start-start pode ser: estar a trabalhar num projeto e criar um glossário. O projeto deve ser iniciado antes do início do glossário. É possível trabalhar nestas duas em simultâneo, no entanto, não importa qual é delas termina primeiro.

start-finish



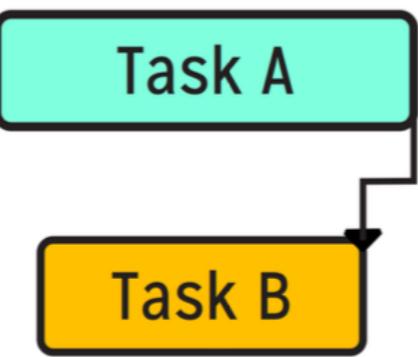
- Aplica-se às tarefas onde a primeira tarefa deve começar antes da segunda tarefa termine. Essas dependências geralmente envolvem algum tipo de data de entrega. Nas dependências de start-finish, o início da segunda tarefa não é importante. Em outras palavras, a primeira tarefa pode ser iniciada antes ou depois da segunda tarefa ser iniciada.
- Uma seta que vai desde o início da Tarefa A até o final da Tarefa B. Ambas as tarefas são mostradas mais ou menos em paralelo. A tarefa A deve ser iniciada antes da tarefa B terminar.
- Este método de representação pode ser usado em qualquer número de representações visuais, incluindo gráficos de Gantt e PERT.
- Um exemplo de uma dependência de início-fim poderia ser: estar a planear o próximo sprint durante o sprint atual enquanto este está em curso. Isso permite que a equipa de desenvolvimento comece a trabalhar imediatamente após iniciar o próximo sprint. Planeamento (Tarefa B) deve ser iniciado antes que o sprint atual (Tarefa A) termine, portanto, é dependente do sprint atual do tipo start-finish.

finish-start



- Aplica-se as tarefas onde a primeira tarefa deve terminar antes que a segunda comece. É o tipo mais comum de dependência.
- A tarefa B não se está sobreposto com a tarefa A. A deve terminar antes de B começar.
- Pode ser aplicada a qualquer tipo de representação visual de tarefas, incluindo os gráficos de Gantt e PERT.
- Um exemplo de uma dependência finish-start pode ser a necessidade de concluir uma funcionalidade e saber o que esta irá fazer antes de elaborar um manual de utilizador ou um tutorial para um produto.

finish-finish



- Acontece quando a primeira tarefa deve terminar antes da segunda ser concluída.
- As duas tarefas avançam em paralelo. As setas mostram que A deve terminar antes de B ser concluída.
- Pode ser usada em qualquer tipo de representação visual de tarefas, incluindo os gráficos de Gantt e PERT.
- Um exemplo de dependência finish-finish pode ser terminar faturar um cliente antes de entregar a totalidade do projeto para evitar falhas de pagamento.

Critical Path Method - CPM

Um gráfico CPM é uma representação visual das dependências

- A construção desta rede corresponde ao método das tarefas potenciais ou método do caminho critico. Pode também ser chamado PDM (Precedence Diagramming Method).
- Cada atividade é representada por uma caixa.
- As atividades são conectadas por ligações de dependência representadas por setas.
- Representação sintatica das relações lógicas entre atividades, construir da esquerda para a direita para representar a cronologia do projeto..
- Vantagem: rede de antecedentes que permite visualizar de forma clara a lógica das dependências. Dá a possibilidade de relações com prazos (intervalo / sobreposição)
- Representação existente em ferramenta de gestão de projetos: PMW (Project Management Workbench) e Microsoft Project.

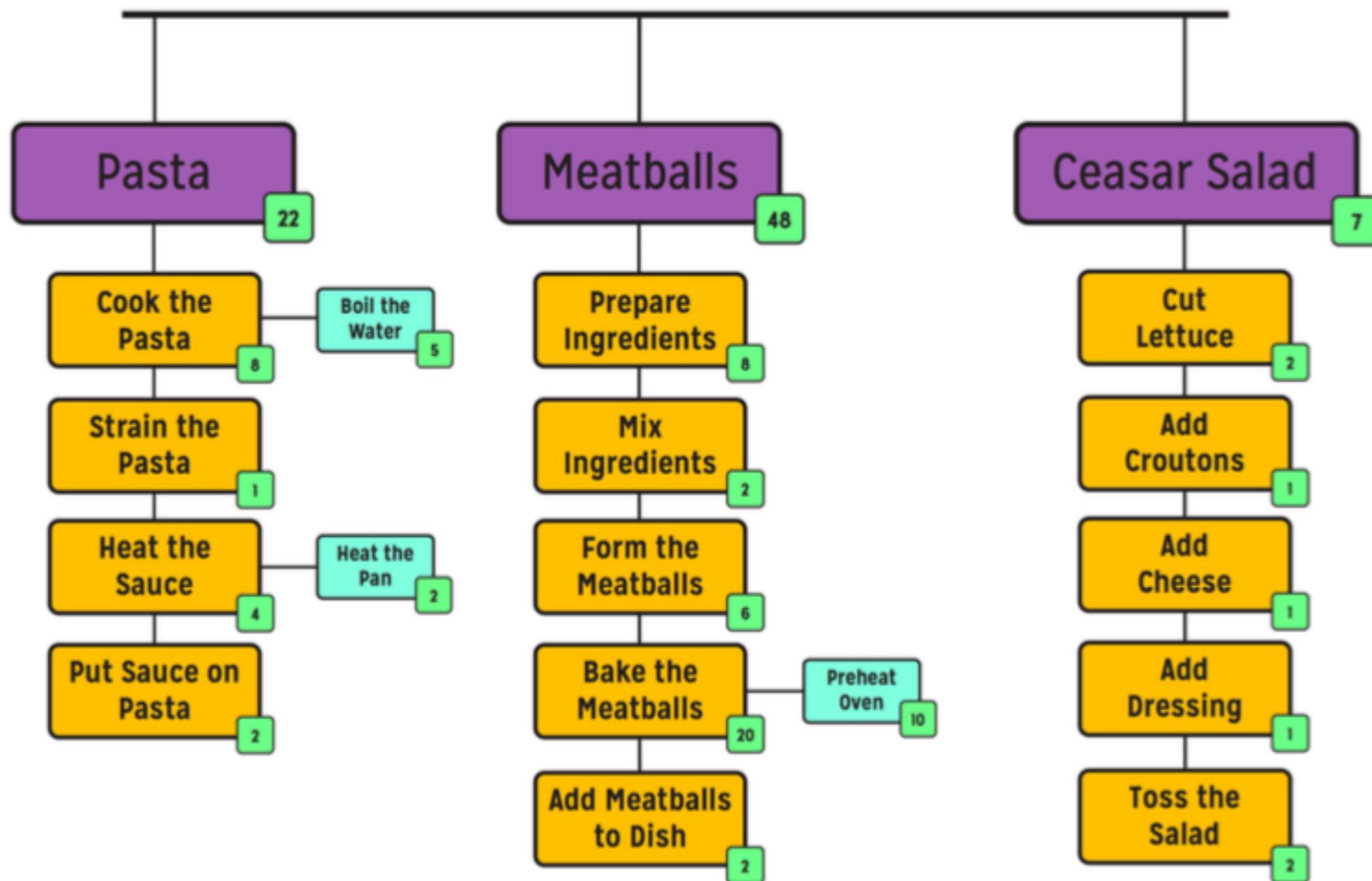
Critical Path Method - CPM

Método:

1. Fazer uma lista das tarefas necessárias para concluir o projeto / produto de trabalho e criar a seguir uma Estrutura de Composição do Trabalho (WBS) para essas tarefas.
2. Adicionar estimativas de tempo para cada tarefa.
3. Organizar as tarefas agrupando as dependências horizontalmente. Por exemplo, uma tarefa que é concluir-iniciar dependente de outra deve ser agrupada ao lado da tarefa que é dependente. As setas podem então ser usadas dependendo da tarefa para descrever a dependência.
4. A partir destes grupos de tarefas, é possível criar caminhos. Um caminho é uma seqüência coletadas por setas que pode levar de uma tarefa para outra (com as dependências ilustradas).

CPM - exemplo do jantar

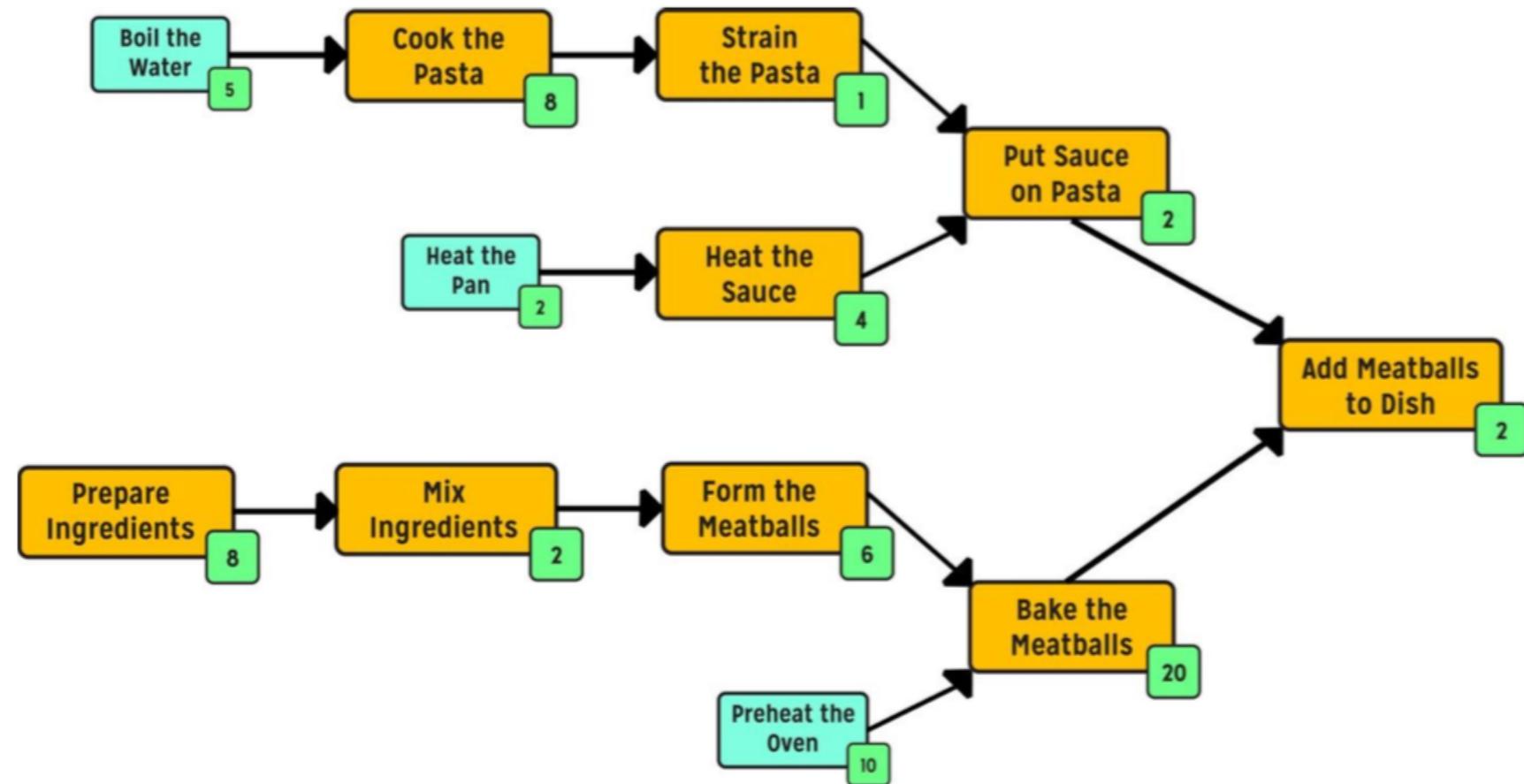
SPAGHETTI & MEATBALL DINNER



Critical Path Method - CPM

- Caminhos que vão desde o início até o final do gráfico são chamados **caminhos beginning-to-end** (início até o fim). Tarefas em diferentes caminhos de início até o fim podem ser concluídas de forma independente e, portanto, ao mesmo tempo.
- Se dois ou mais caminhos de início até o fim se encontram, este é um **ponto de coordenação** para dependências de tarefas - significa que as duas ou mais tarefas devem ser concluídas antes que a tarefa para a qual apontam possa começar. Em outras palavras, as tarefas devem ser coordenadas.
- Um gráfico de CPM pode ter tantos caminhos de início-até-fim e pontos de coordenação quanto necessário para concluir o projeto.

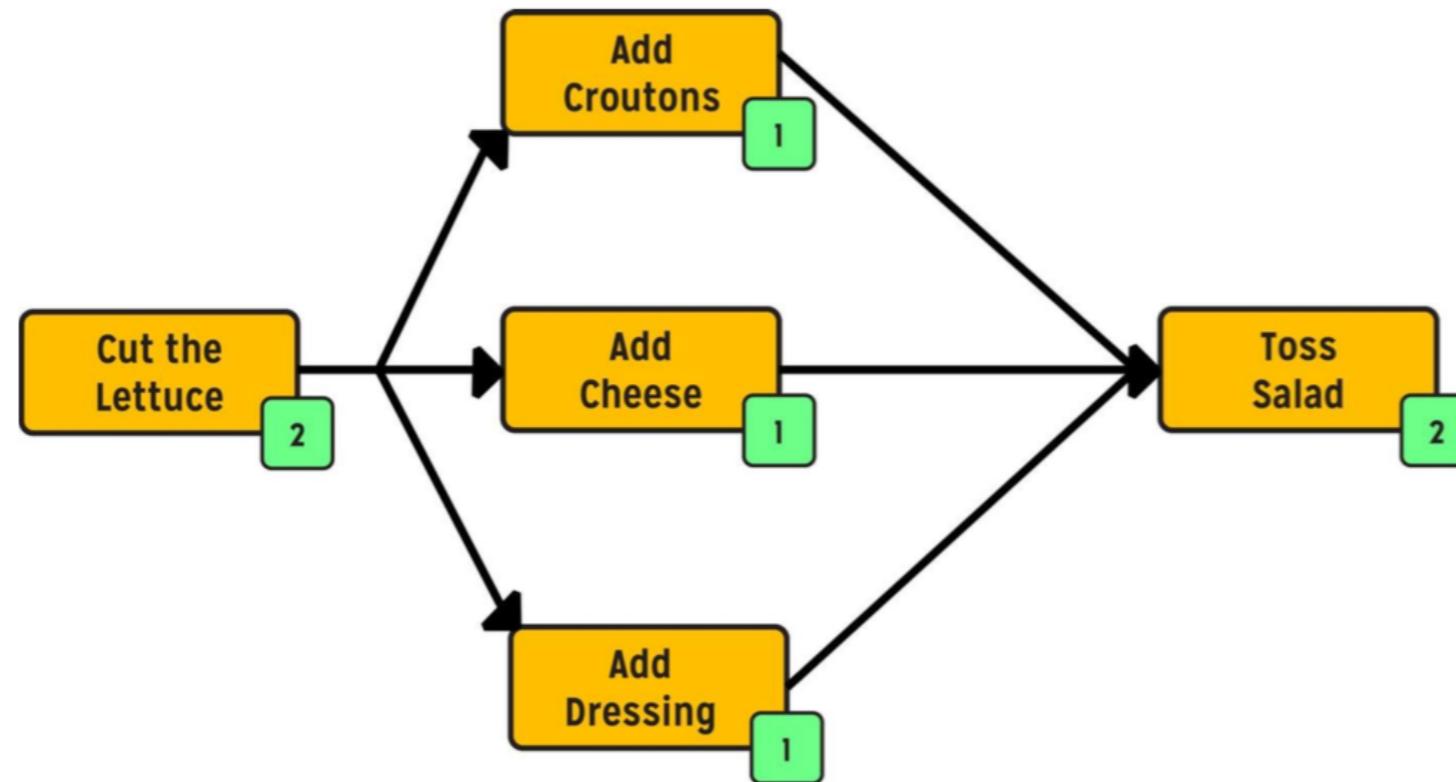
CPM - exemplo do jantar



- 4 caminhos
- “Boil the Water” -> “Add Meatballs to Dish”
- “Heat the Pan” -> “Add Meatballs to Dish”
- “Prepare Ingredients” -> “Add Meatballs to Dish”
- “Pre-heat Oven” -> “Add Meatballs to Dish”

3 pontos de coordenação

Tarefas paralelas num CPM

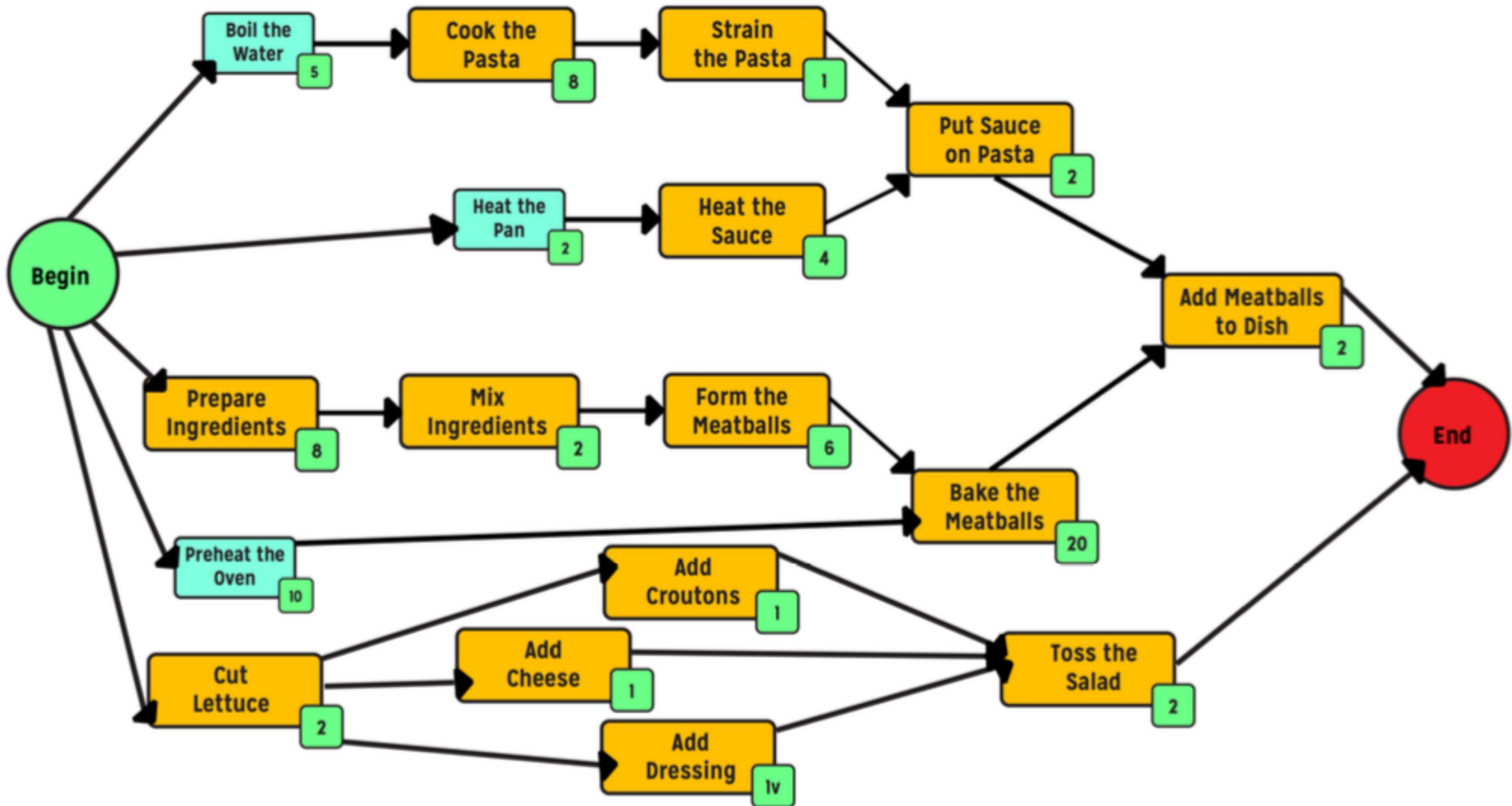


3 caminhos beginning-to-end:

- “Add Croutons”
- “Add Cheese”
- “Add Dressing”

CPM

Combinar todos os caminhos beginning-to-end para criar um CPM completo.



Caminhos críticos

- Os gráficos de CPM são uma ferramentas úteis para gerar um cronograma do projeto e determinar Estimativas de tempo mínimo. Isto é feito com base em caminhos críticos.
- Caminho crítico: é o caminho de maior duração de tarefas entre pontos lógicos. Em outras palavras, é o **caminho *beginning-to-end* com a maior estimativa de tempo**.
- Representa a quantidade de tempo necessária para completar nível de projecto, Assumindo que há recursos suficientes para que outros caminhos *beginning-to-end* possam ser executado em simultâneo.

Caminhos críticos

- Fornecem informações sobre a maior e a menor estimativas de tempo para as tarefas do projeto.
- Evidenciam as tarefas críticas para o projeto.
- Evidenciam também as tarefas que têm folga. São aquelas que não estão no caminho crítico.

As estimativas de tempo geradas a partir de caminhos críticos em gráficos CPM podem diferir das estimativas da Estrutura de Repartição do Trabalho (WBS). Num gráfico CPM, o caminho crítico determina a estimativa mínima e assume-se (onde os recursos estão disponíveis) que outros caminhos serão executados em simultâneo. Numa estimativa WBS, no entanto, cada tarefa é adicionada como se fosse feita uma após a outra.

Tarefas críticas

- As tarefas críticas no caminho crítico devem ser as **primeiras a serem agendadas** a partir de um gráfico CPM. Da facto, as tarefas de outros caminhos podem ter tempo extra para serem concluídas (dentro do limite de tempo do caminho crítico), sem adicionar ao projeto geral.
- A **folga** é útil para os projetos, pois fornece um certo grau de flexibilidade para certas tarefas, sem afetar necessariamente a data de conclusão desejada.

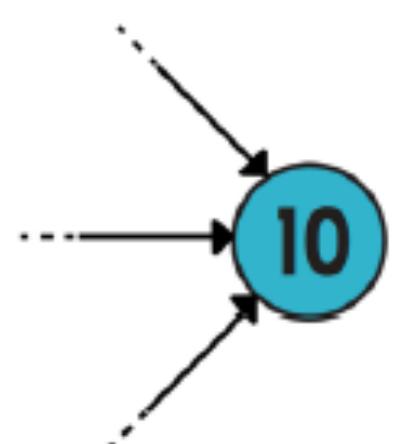
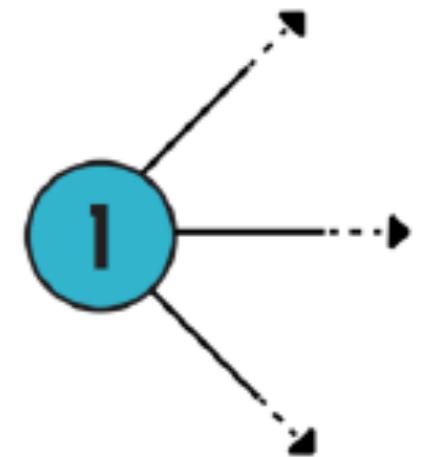
Gráfico PERT

Program Evaluation and Review Technique (PERT)

- Ferramenta de visualização de projeto (tal como CPM)
- Nós: milestones -> evento ou produção de um produto de trabalho
- Arcos: tarefas
 - com descrição e estimativa de tempo
 - com setas para mostrar as dependências
 - indicar a ordem da tarefa
 - Uma sequência de tarefas dependentes é representada como um caminho que segue a direção das setas.

Dependências entre tarefas

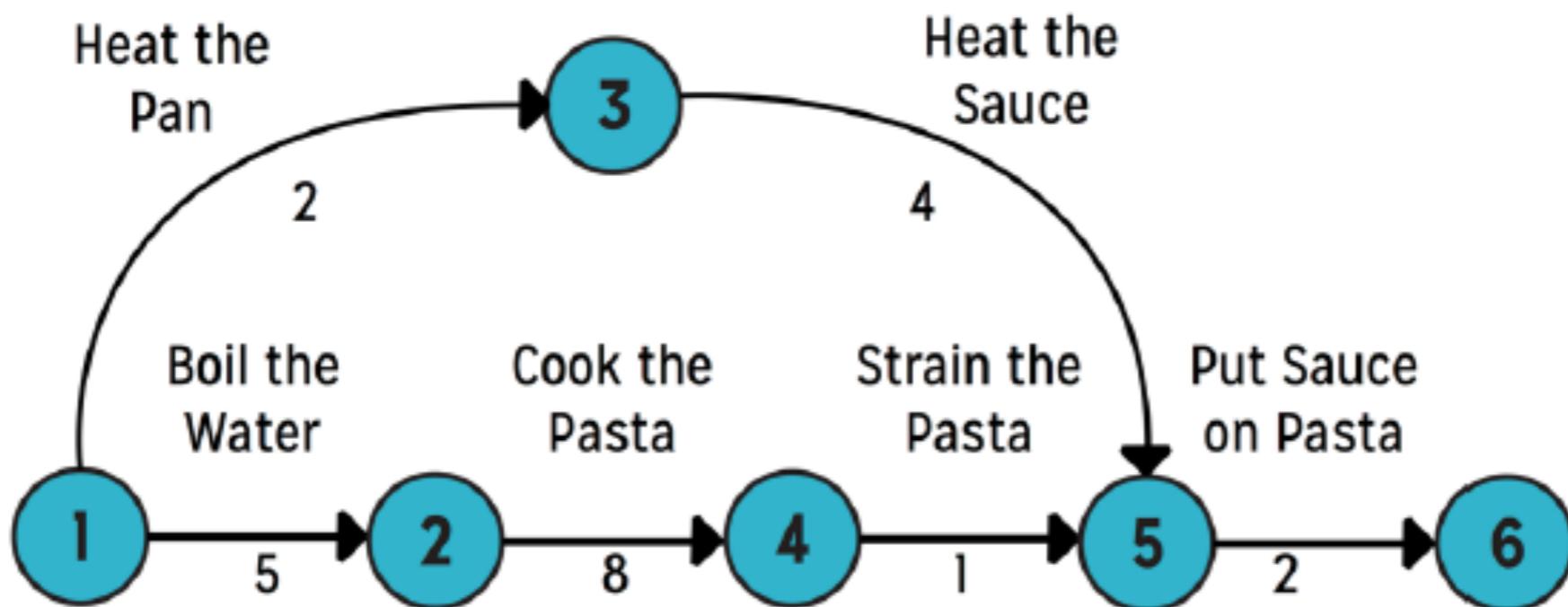
- Se vários caminhos independentes saírem de um nó (ou seja, se várias linhas de um nó), isso significa que essas tarefas podem ser concluídas ao mesmo tempo, assumindo que o tempo e os recursos estão disponíveis. **São executadas em paralelo.**
- Por outro lado, se vários caminhos se encontram num nó (se múltiplas setas apontam para um nó), isso significa que **esses caminhos devem ser sincronizados**. Em outras palavras, esses caminhos devem ser concluídos ao mesmo tempo, antes que o próximo caminho que sai do nó possa ser iniciado.



Exemplo de gráfico PERT

jantar de espaguete com almôndegas e salada

- Caminho para fazer a massa:



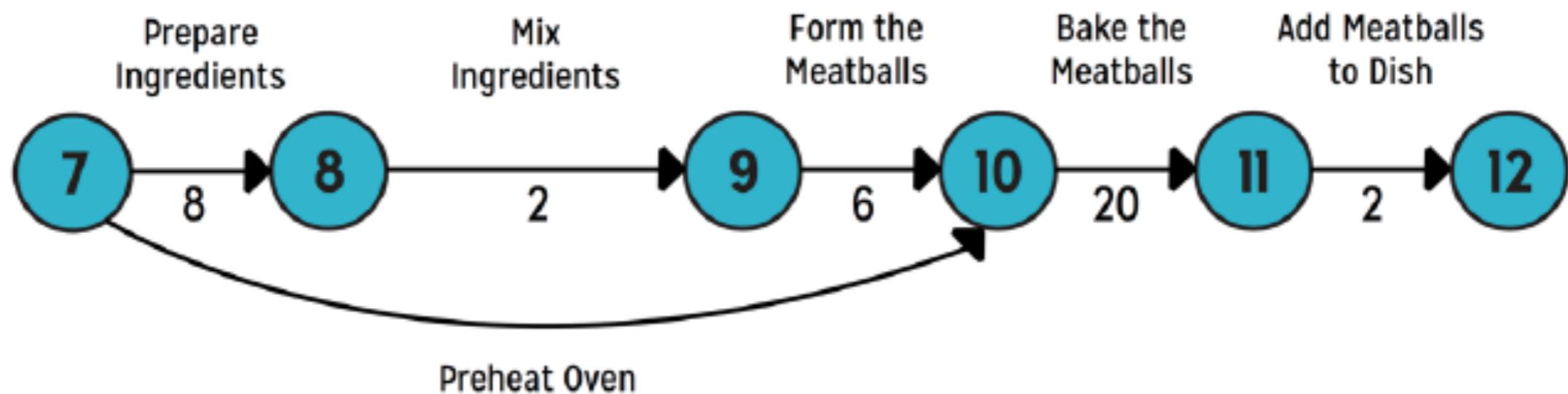
- 2 caminhos:

- 1-> 3 -> 5
- 1-> 2 -> 4 -> 5

Exemplo de gráfico PERT

jantar de espaguete com almôndegas e salada

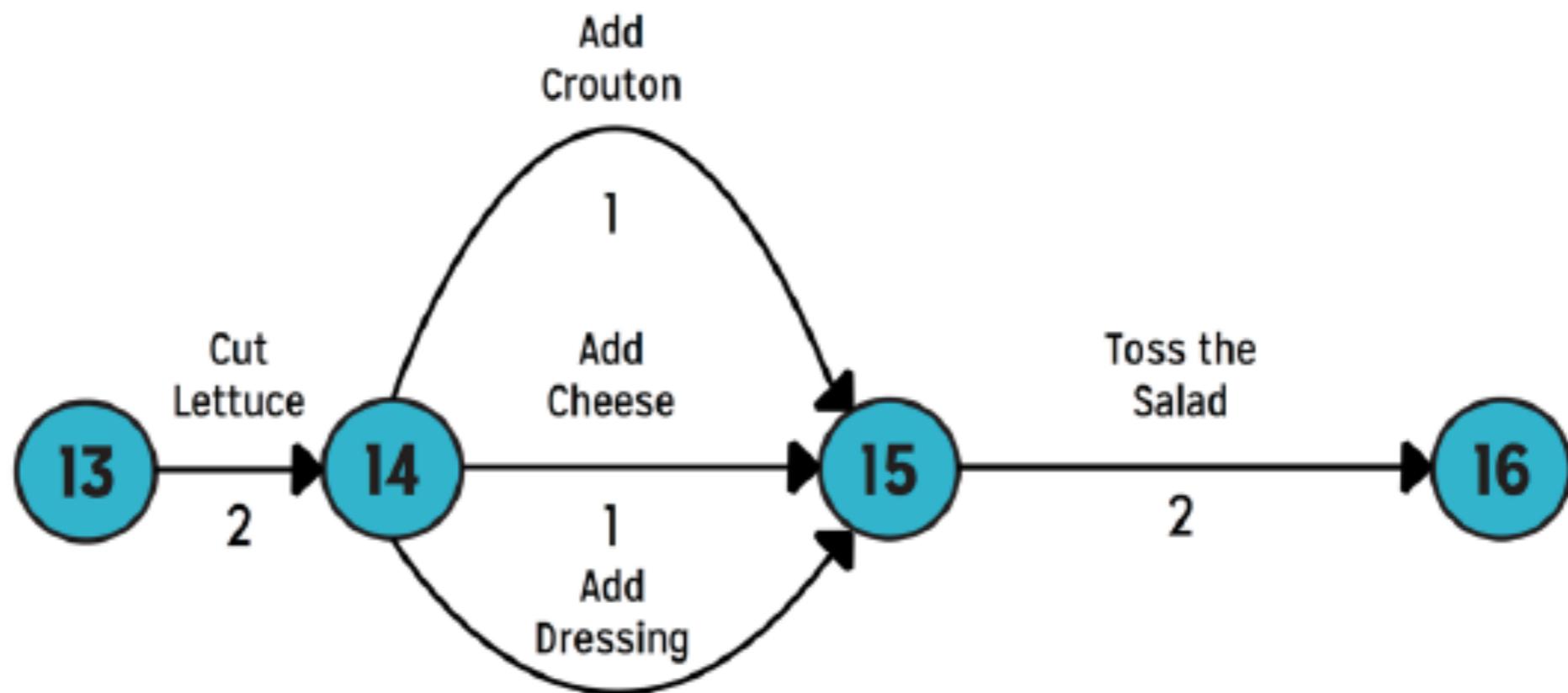
- Caminho para fazer as almôndegas:



Exemplo de gráfico PERT

jantar de espaguete com almôndegas e salada

- Caminho para fazer a salada:



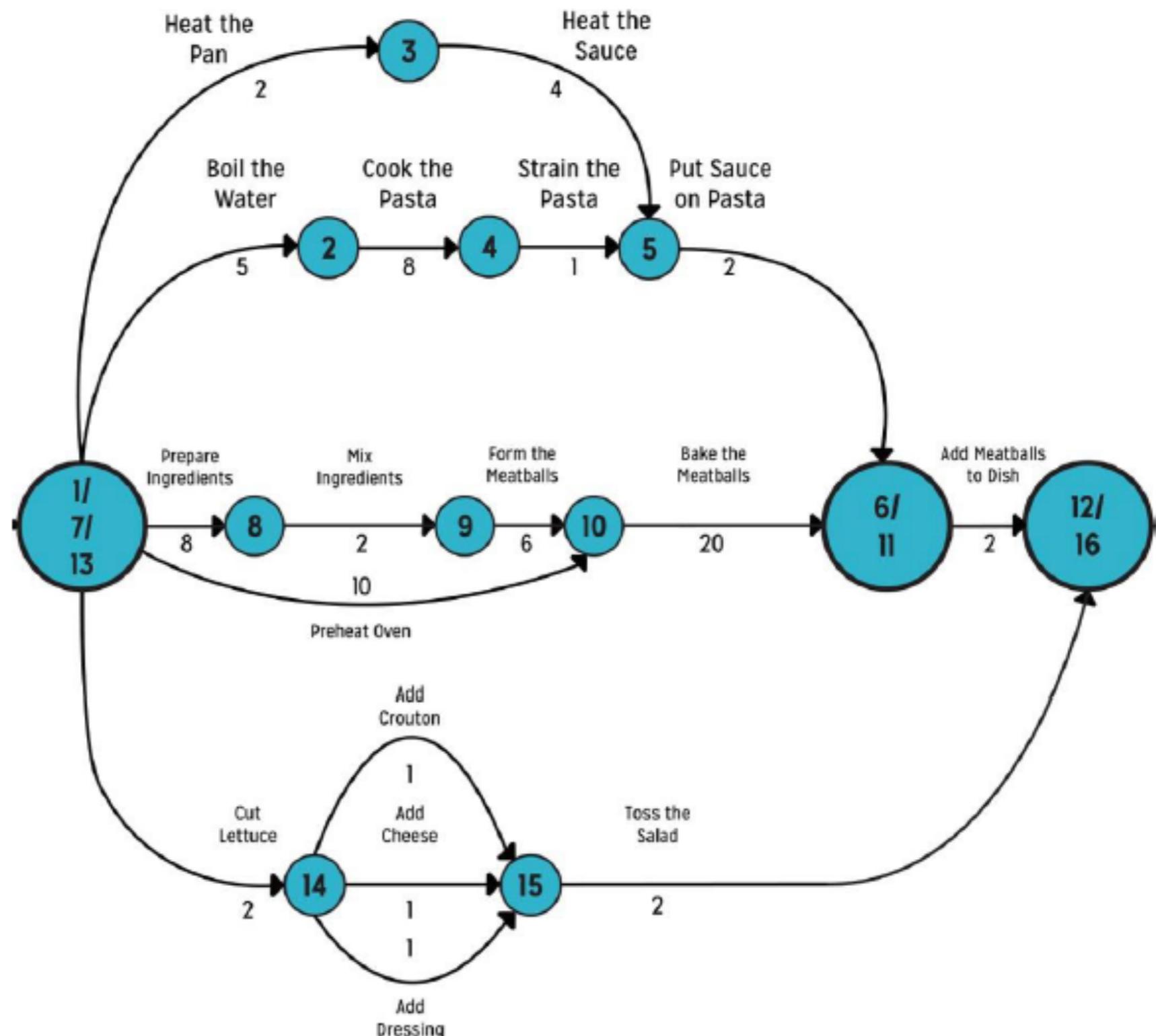
Note as tarefas paralelas “Add Croutons,” “Add Cheese,” “Add Dressing.” Estas tarefas podem ser feitas em qualquer ordem, mas todas devem ser concluídas antes que a tarefa final de "Mexer a Salada" possa começar.

Exemplo de gráfico PERT

jantar de espaguete com almôndegas e salada

- Gráfico completo para o jantar

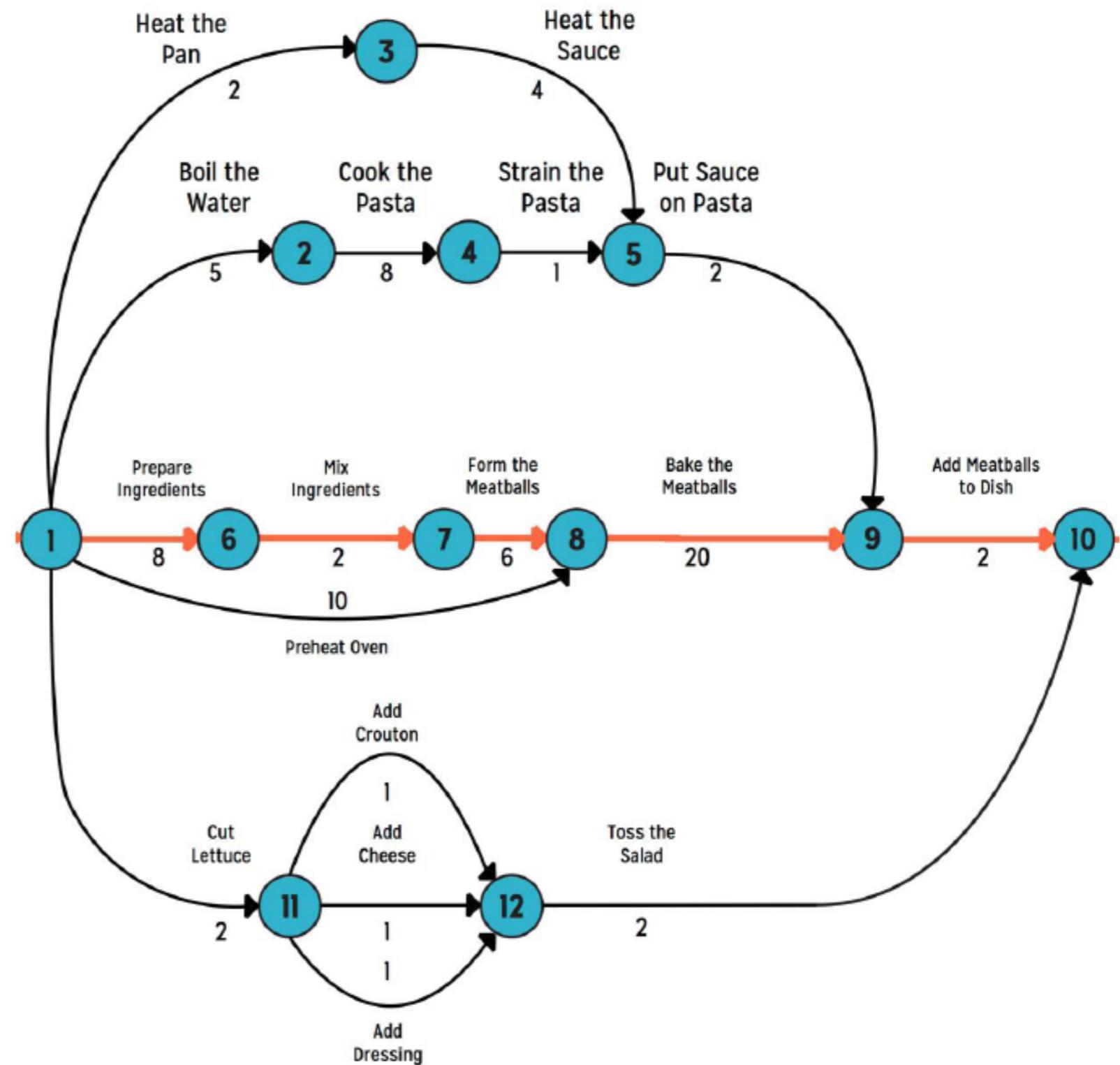
Como na construção de um gráfico CPM, num gráfico PERT, cada agrupamento de tarefas deve ser traçado primeiro e, depois, essas seções combinadas. Num gráfico PERT, isso pode exigir a limpeza de números que se sobreponem para a mesma tarefa em seções diferentes. Por exemplo, o nó 6 acima combinado com o nó 11 no gráfico final, e foi simplesmente rotulado 11



Exemplo de gráfico PERT

jantar de espaguete com almôndegas e salada

- Gráfico final com os números corrigidos após junção dos 2 gráficos (salada e massa)



Exercício

Gráfico PERT

- Desenhar um gráfico de PERT correspondente a lista de tarefas da tabela

Tarefas	Predecessores
A	-
B	-
C	A
D	A
E	D,B
F	C,E

PERT - Caminhos críticos

- Tal como nos gráficos CPM, o caminho crítico é o caminho com as maiores estimativas de tempo das tarefas.
- O caminho crítico na verdade representa a quantidade de tempo mais curta na qual se pode concluir o projeto, com todas as tarefas no caminho crítico concluídas.
- O caminho crítico é, portanto, a quantidade mínima de tempo que um projeto pode representar, assumindo que há recursos suficientes para que os outros caminhos podem ser concluídos em simultâneo.

PERT - Caminhos críticos

Exemplo do jantar

- O caminho crítico entre a preparação dos ingredientes até adicionar as almôndegas ao prato (1-> 2-> 6-> 8-> 1-> 12) tem um total de 38 minutos.
- É a mais longa estimativa de tarefas
- Outros caminhos vão terminar mais cedo:
 - 1-> 5-> 9-> 11-> 12 leva 10 minutos,
 - 1-> 4-> 7-> 9-> 11-> 12 leva 18 minutos,
 - 1-> 8-> 11-> 12 demora 32 minutos e
 - 1-> 3-> 10-> 12 leva 5 minutos. mesmo
- Embora o caminho crítico leva mais tempo, todas essas etapas devem ser terminadas, de modo que o total do projeto demore pelo menos 38 minutos.
- Além disso, como nos gráficos de CPM, os outros caminhos têm folgas, uma vez que têm algum espaço para que as estimativas possam mudar (contanto o caminho deve levar menos tempo do que o caminho crítico), sem afetar a estimativa global do projecto.

PERT - Caminhos críticos

Exemplo do jantar

- Mais uma vez, os caminhos $1 \rightarrow 5 \rightarrow 9 \rightarrow 11 \rightarrow 12$, $1 \rightarrow 4 \rightarrow 7 \rightarrow 9 \rightarrow 11 \rightarrow 12$, $1 \rightarrow 8 \rightarrow 11 \rightarrow 12$ e $1 \rightarrow 3 \rightarrow 10 \rightarrow 12$, são todos caminhos mais curtos do que o caminho crítico, e podem levar algum tempo extra para terminar, se necessário, contanto que eles não levam mais tempo do que um total de 38 minutos
- Qualquer tarefa no caminho crítico que se estende em termos de tempo irá resultar no aumento do tempo mínimo para terminar o projeto, porque não há folga no caminho crítico.
- No exemplo acima, o caminho crítico de $1 \rightarrow 2 \rightarrow 6 \rightarrow 8 \rightarrow 1 \rightarrow 12$ demora 38 minutos, de modo que o tempo mínimo do projeto será de 38 minutos.
- Entretanto, se a estimativa inicial de 6 minutos para terminar a Tarefa $6 \rightarrow 8$ estende-se para 10 minutos, então a estimativa do caminho crítico passa também de 38 minutos para 42 minutos. O tempo mínimo para o projeto foi alterado para um total de 42 minutos, na medida que o caminho crítico não tem nenhuma folga.
- Num gráfico PERT, caminhos críticos deve estar na horizontal. No exemplo acima para o projeto do jantar, o caminho crítico está assinalado a vermelho. Como os caminhos críticos estão sempre na horizontal nos gráficos PERT, eles são facilmente identificados, mesmo se as informações tais como as estimativas de tempo não forem disponíveis. Todos os outros caminhos no gráfico devem ser desenhados quer acima ou abaixo do caminho crítico.

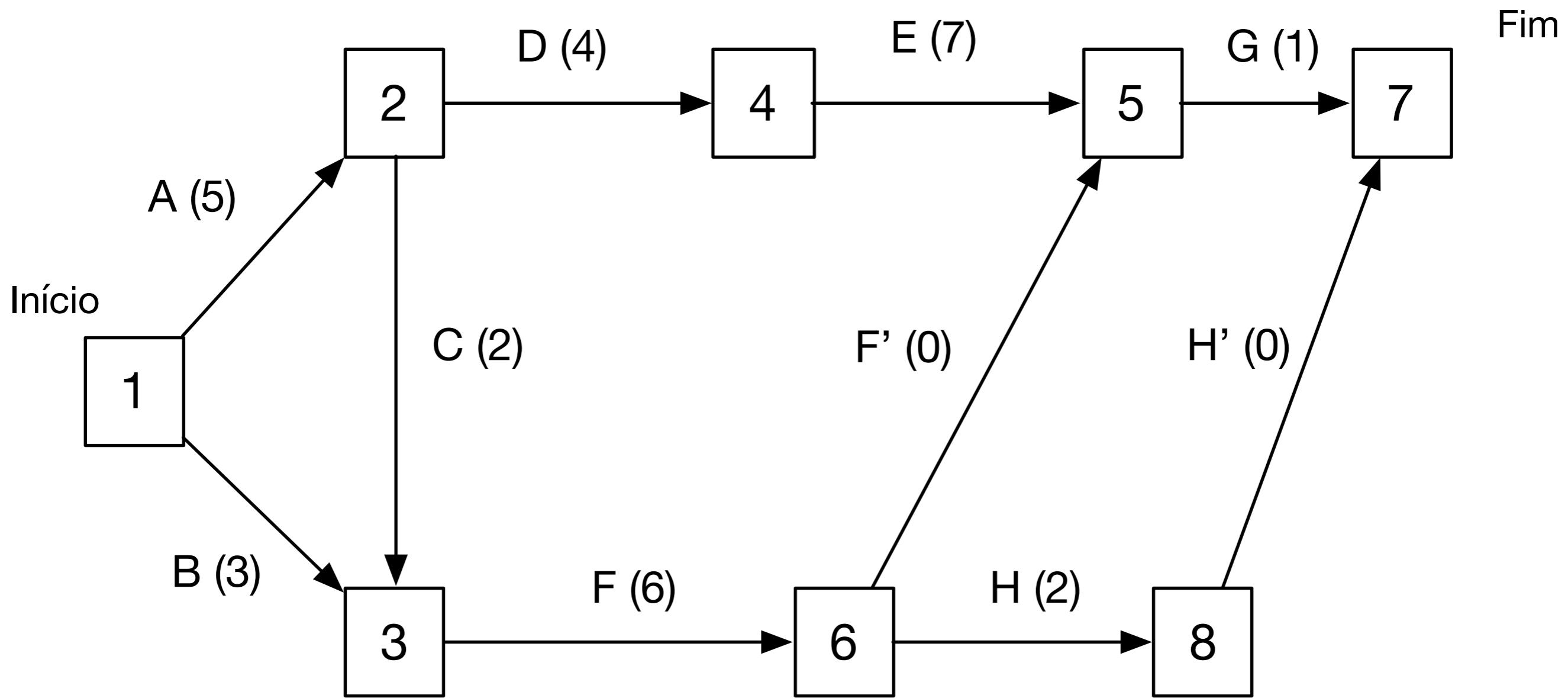
PERT vs CPM

- Em que circunstâncias é melhor usar um gráfico CPM ou um gráfico PERT?
 - Ambos podem ser usados para demonstrar dependências de projeto e gerar estimativas de tempo.
 - Ambos funcionam bem para planeamento de tarefas.
 - Quando qualquer gráfico pode ser usado com sucesso para um projeto, é então uma questão de preferência da equipa de desenvolvimento e do gestor de projeto.
 - Em circunstâncias em que o projeto está mais **focado nas tarefas**, o gráfico **CPM** é mais adequado. Nos gráficos CPM, é possível adicionar informações às tarefas, como custos.
 - Em circunstâncias em que o projeto está **mais focado nos eventos ou nas metas**, o gráfico **PERT** é preferível. Os gráficos PERT mostram quais tarefas precisam de ser concluídas antes de um evento ou de uma meta.

tipos de gráficos PERT

- PERT de metas potenciais
- PERT de tarefas potenciais

Exemplo



Duração das tarefas

O mais cedo

- Data “**o mais cedo**”: A data “o mais cedo” de qualquer passo é obtida calculando o tempo mais longo necessário para chegar a esta fase.
- Exemplo: para chegar a etapa 3
 - 2 caminhos: 1,3 e 1,2,3 - o mais cedo = 7

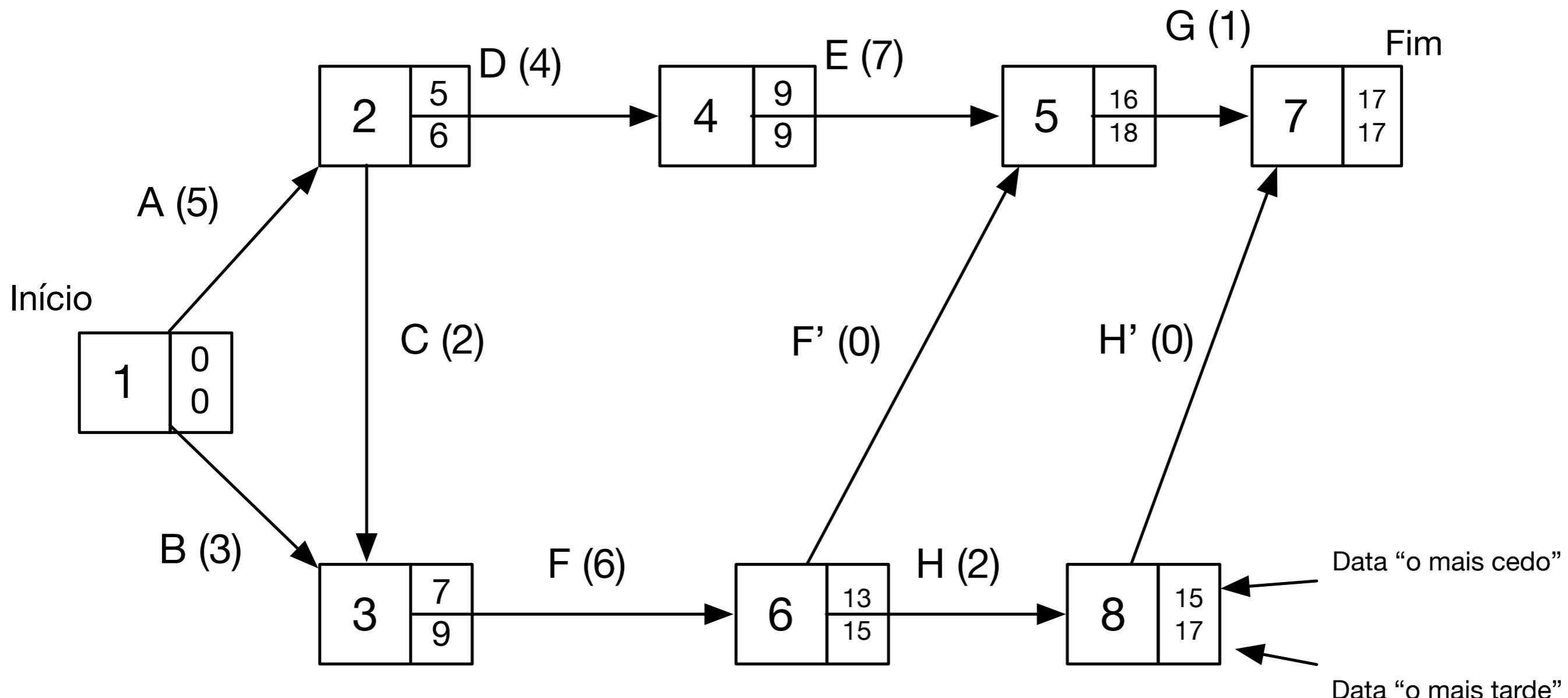
Duração das tarefas

O mais tarde

- Data “**o mais tarde**”: A data “o mais tarde” é obtida a partir da última etapa, fazendo a subtração do tempo que corresponde ao caminho mais longo para recuar a etapa considerada. (A data obtida deve ser a menor)
- Exemplo: cálculo de tempo mais tarde na etapa 6.
 - 3 caminhos:
 - $7,5,6 - 1+0=1$ dia \rightarrow o mais tarde : $17-1 = 16$ dias
 - $7,8,6 - 0+2=2$ dias \rightarrow o mais tarde : $17-2=15$ dias

o mais tarde ?

Exemplo

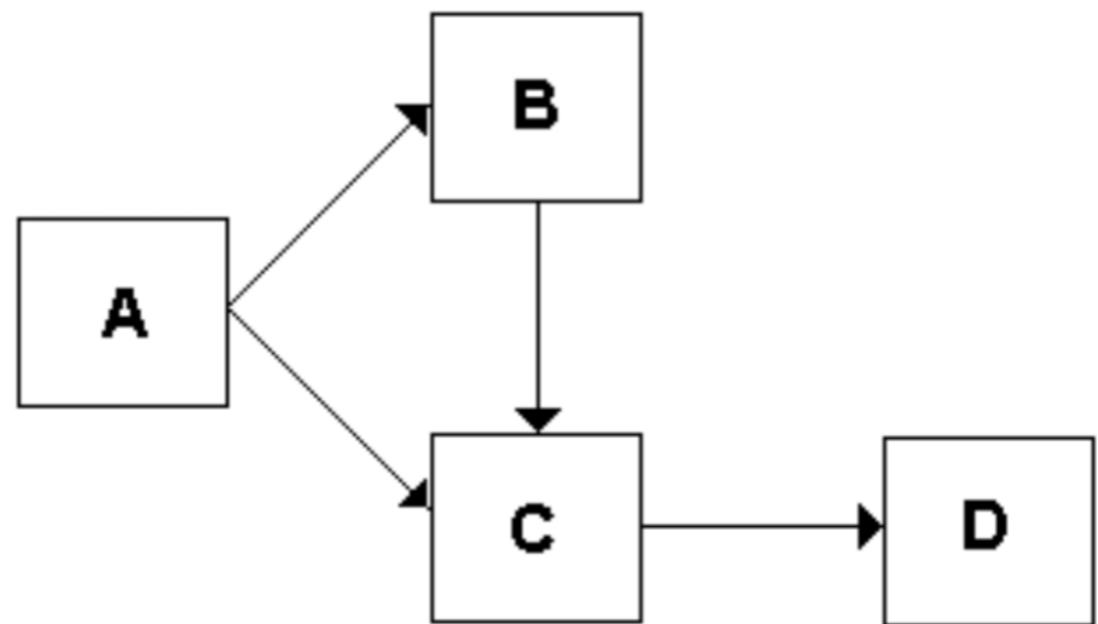


Convenção

MT - Margem Total	IMT - Início o mais tarde	ML - Margem Livre
IMC - Início o mais cedo	D - Duração	FMC - Fim o Mais Cedo
NOME - Nome da Tarefa	FMT - Fim o Mais Tarde	s - Desvio padrão

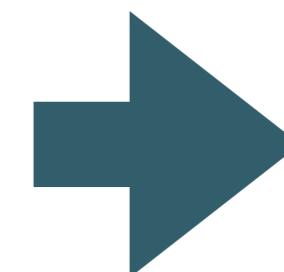
Exemplo 2

- A: 5
- B: 2
- C: 3
- D: 4



$$FMC = IMC + D$$

$$FMC(A) = IMC(A) + D(A) = 0 + 5$$



0	5	5
A		

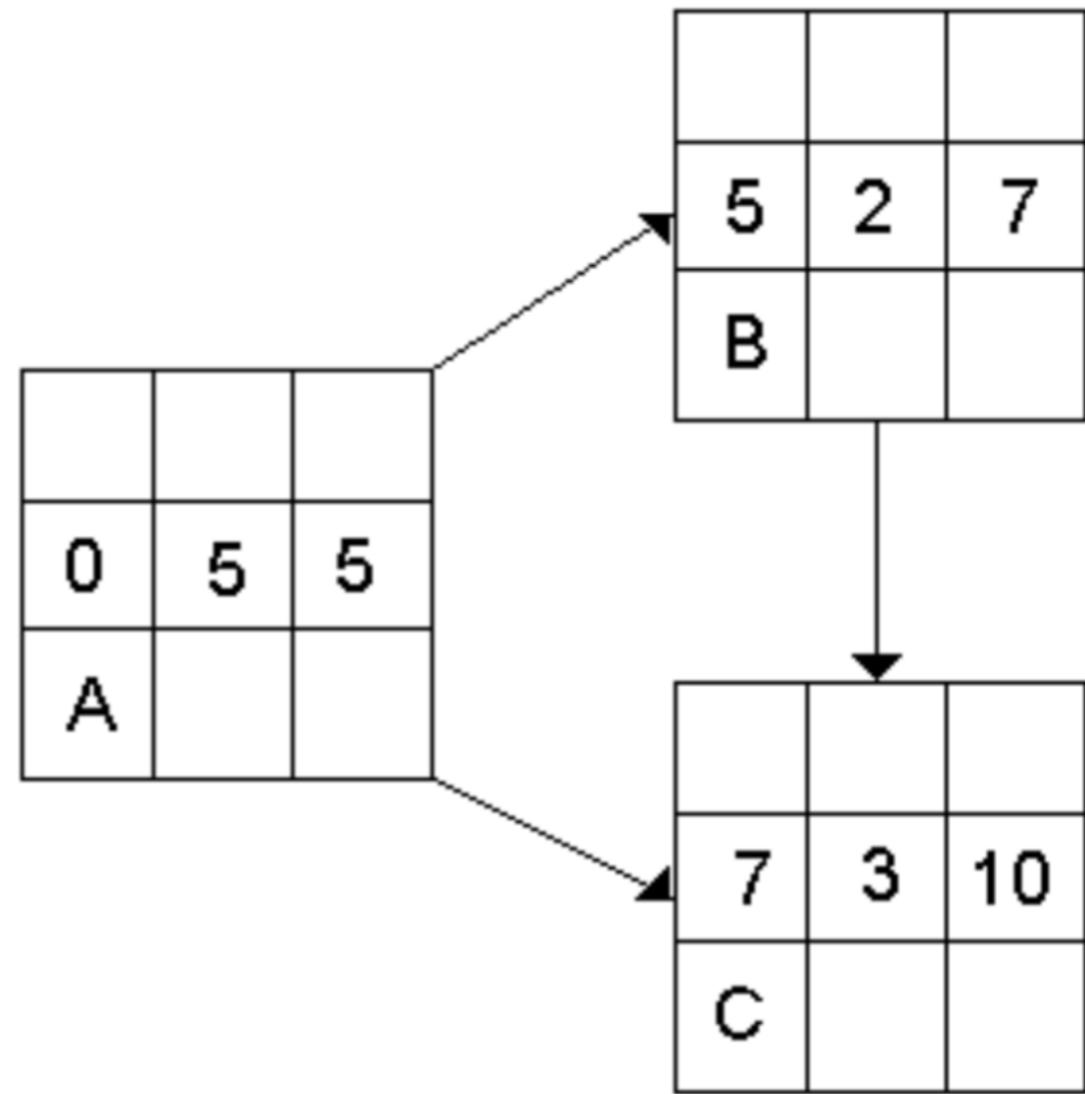
Exemplo 2 - tarefas B e C

$$\text{IMC (B)} = \text{FMC (A)}$$

$$\text{FMC (B)} = \text{IMC (B)}$$

$$+ D (B) = 5 + 2 =$$

$$7$$



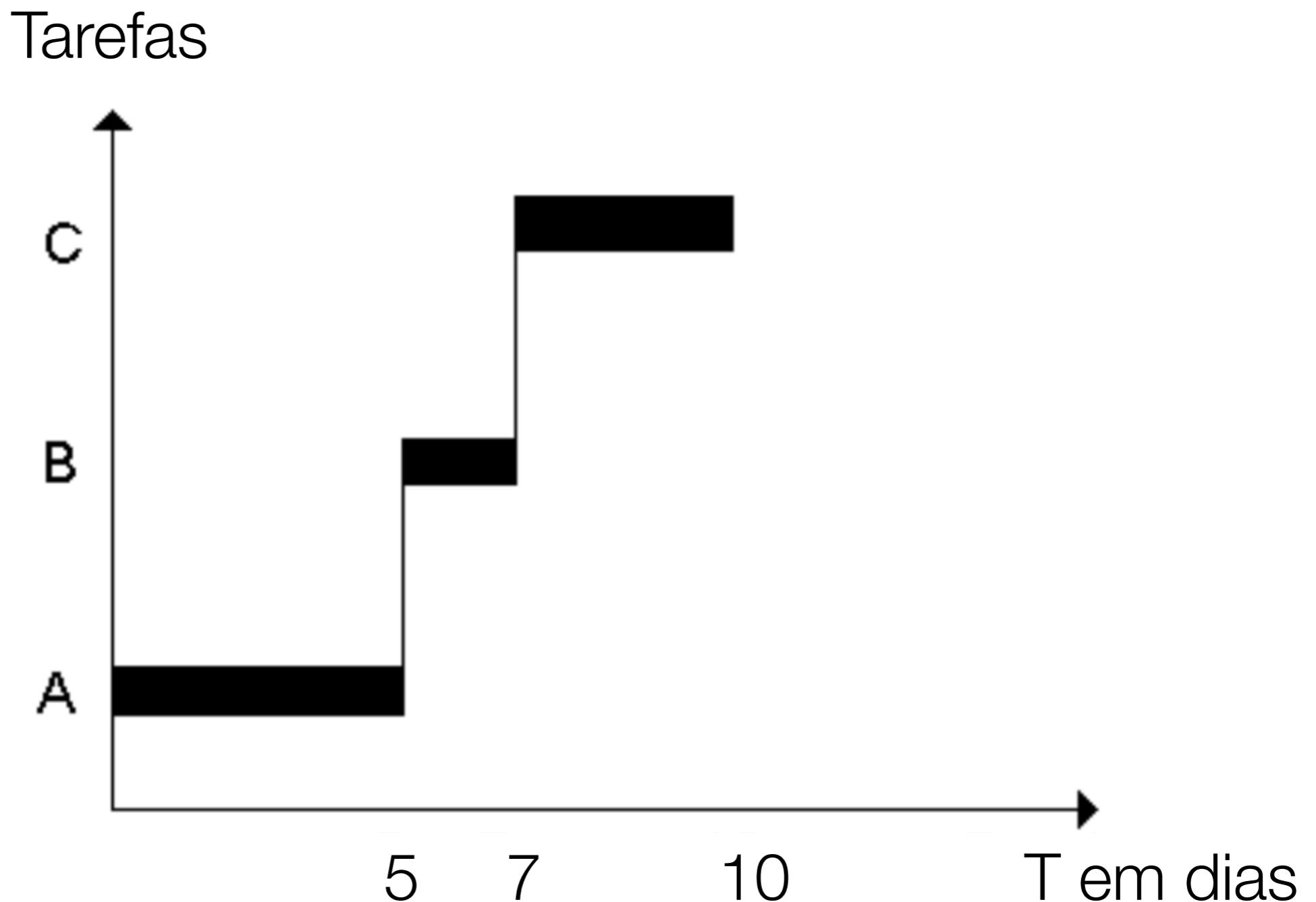
$$\text{IMC (C)} = \text{MAX} (\text{FMC (A)} ; \text{FMC (B)}) = 7$$

$$\text{FMC (C)} = \text{IMC (C)} + D (C) = 7 + 3 = 10$$

Exemplo 2

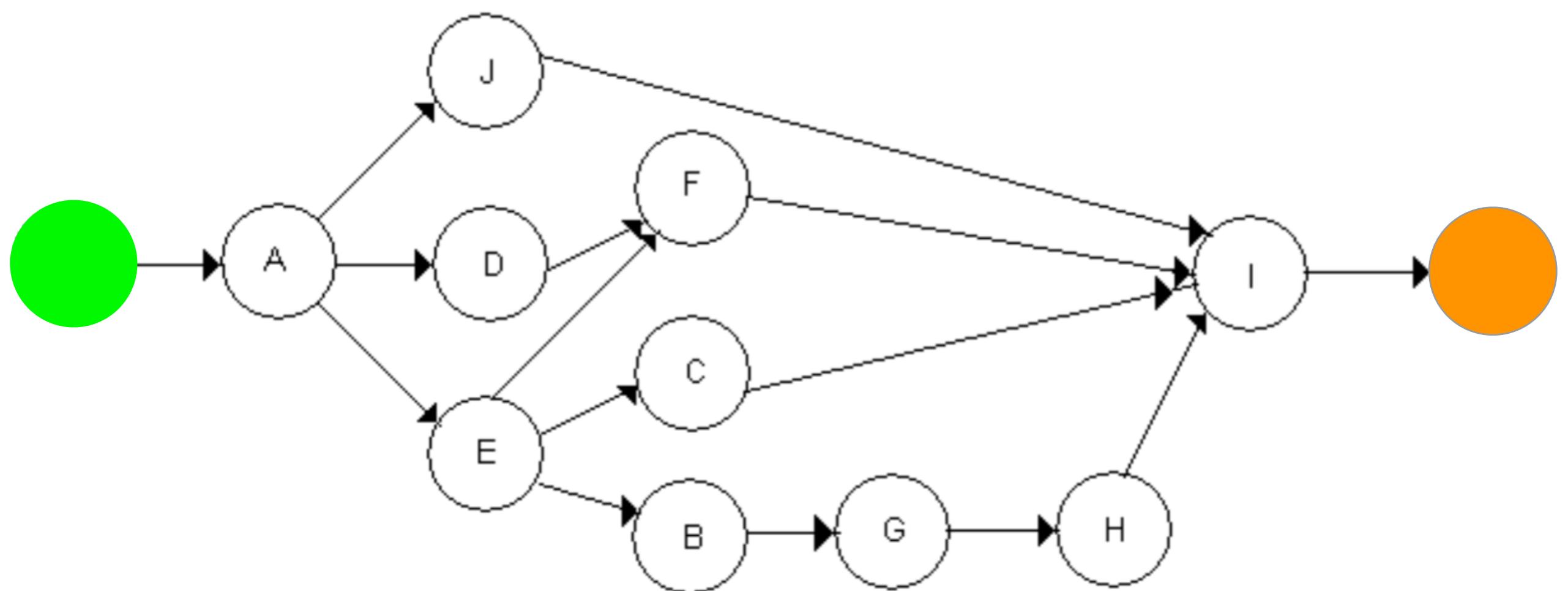
- A tarefa C não pode começar antes do fim de B (7 dias após o início) e que:

IMC (C) = FMC (B)



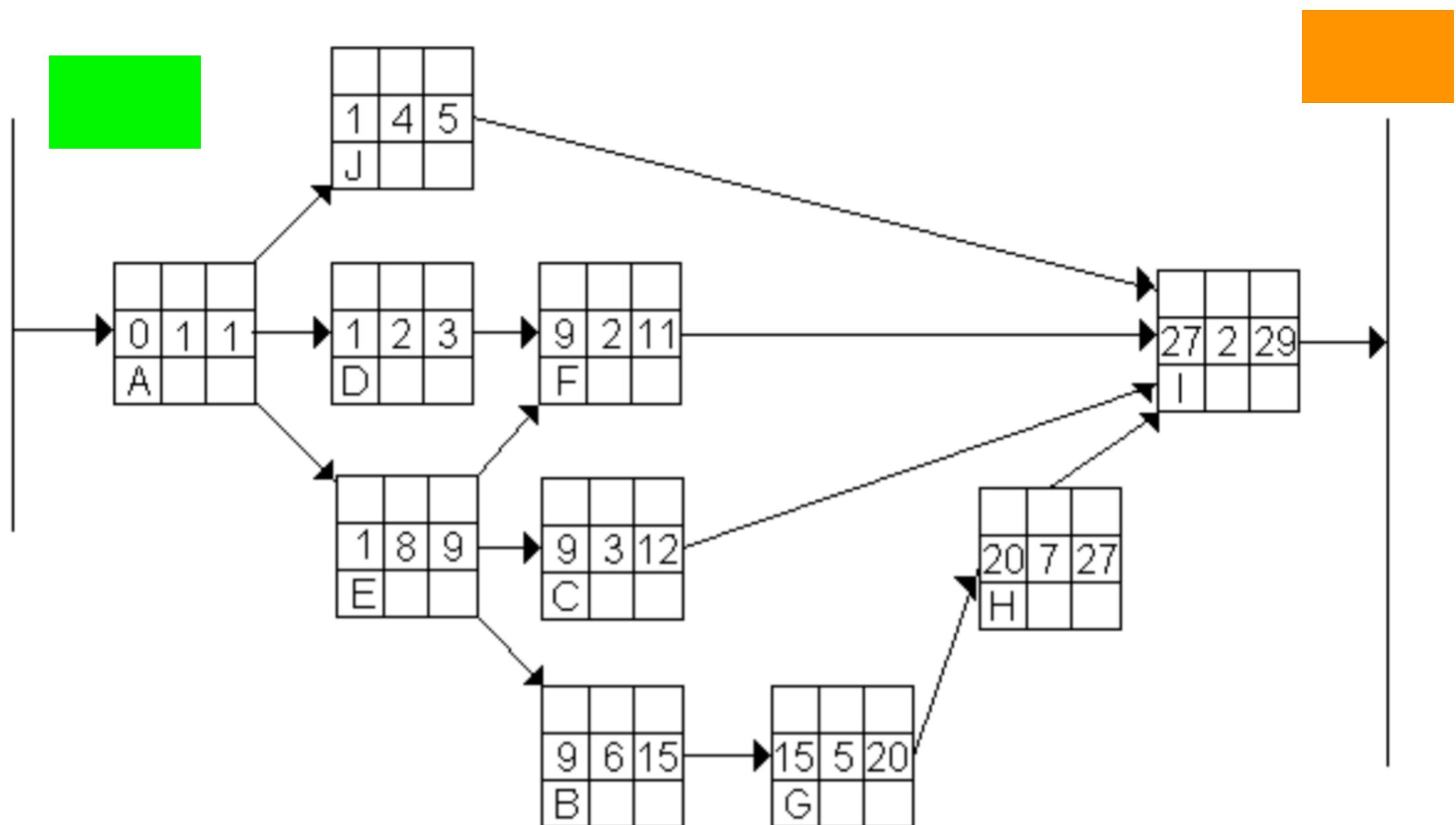
Exemplo 3

Para fazer	A	B	C	D	E	F	G	H	I	J
É preciso ter feito		E	E	A	A	D,E	B	G	J,C,H,F	A



Exemplo 3

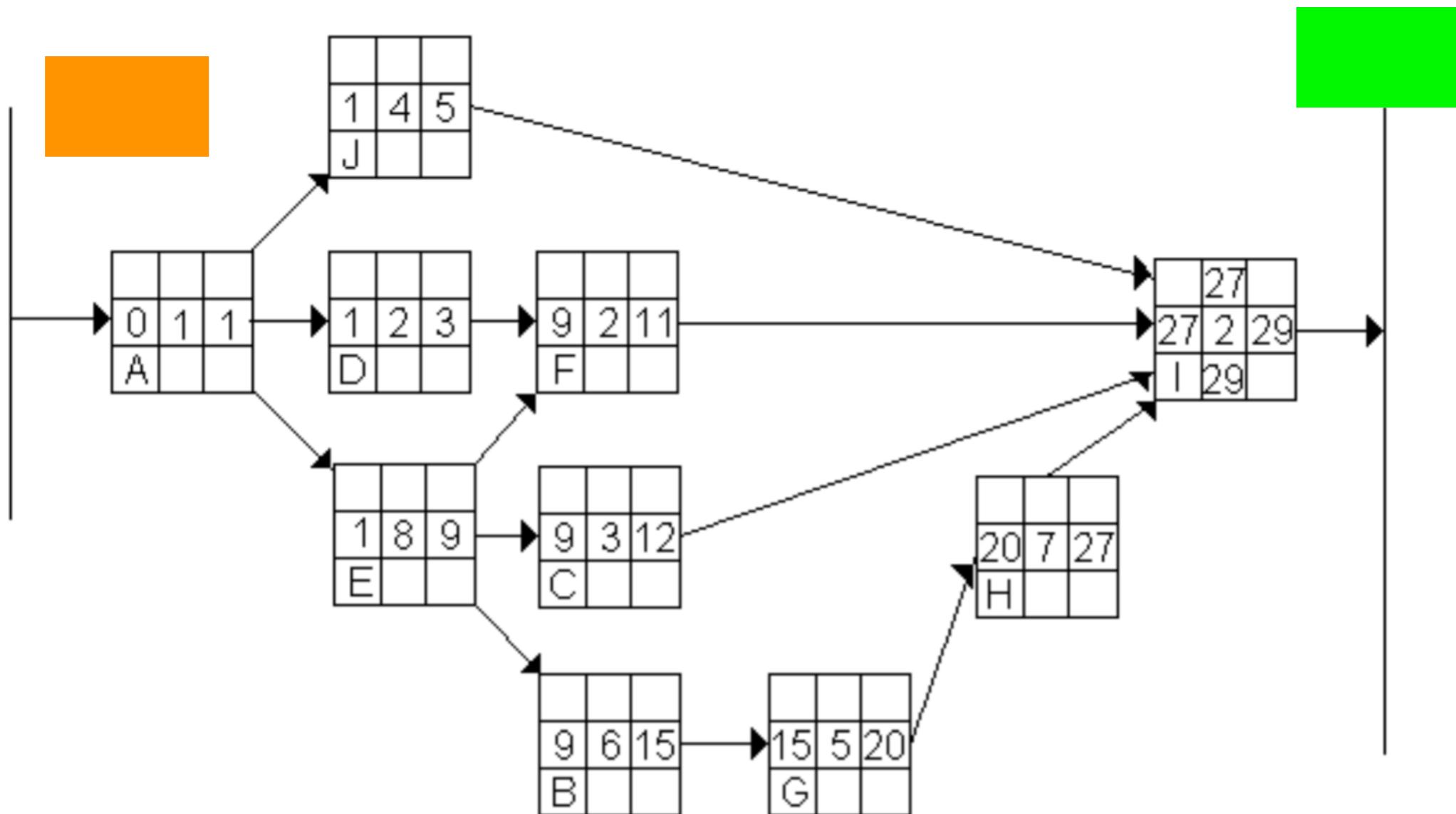
Tarefas	A	B	C	D	E	F	G	H	I	J
Duração	1	6	3	2	8	2	5	7	2	4



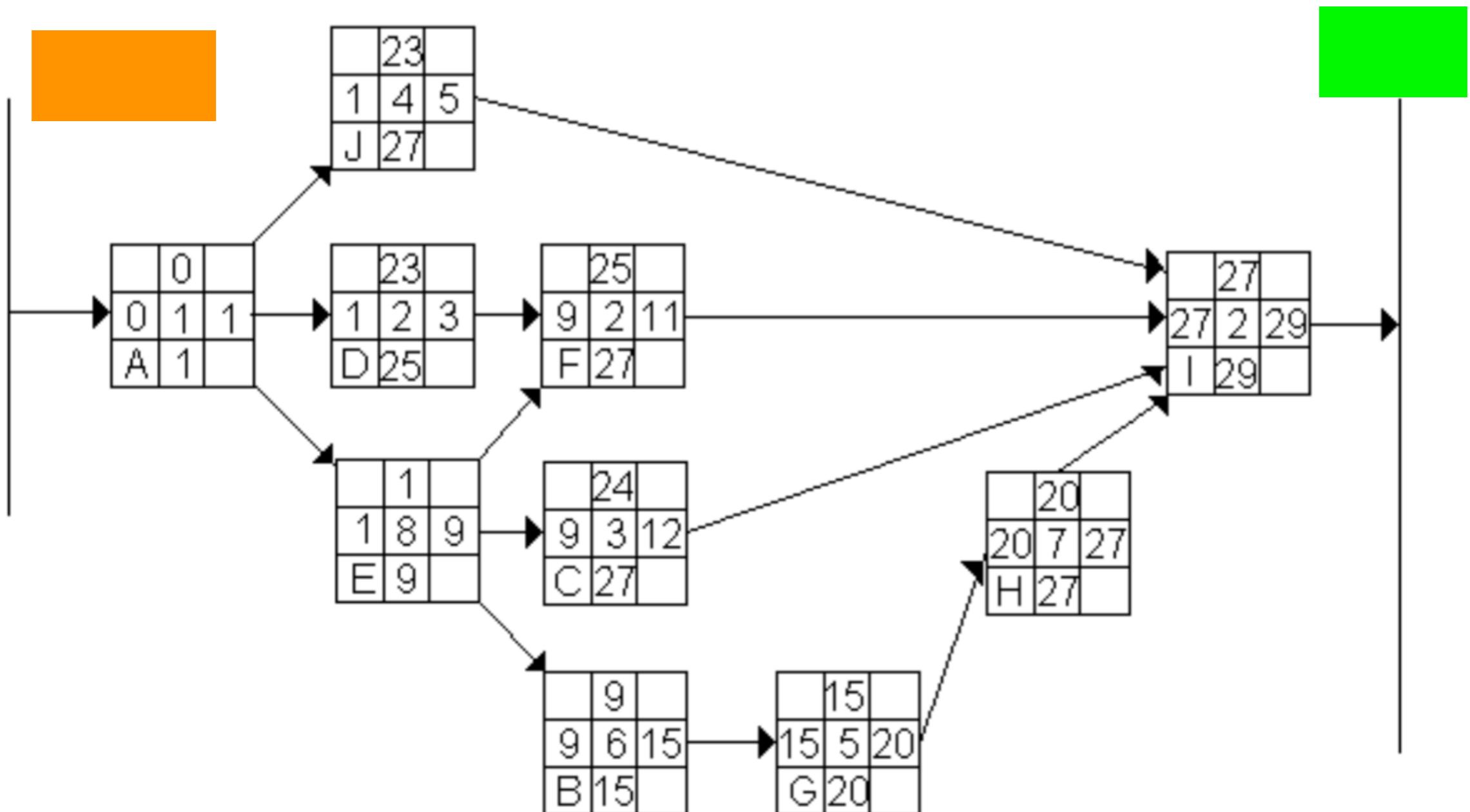
Cálculo das datas “O mais tarde”

- presuposto: compromisso com o cliente : 29 dias

$$\text{IMT}(I) = \text{FMT}(I) - D(I) = 29 - 2 = 27$$



Cálculo das datas “O mais tarde”

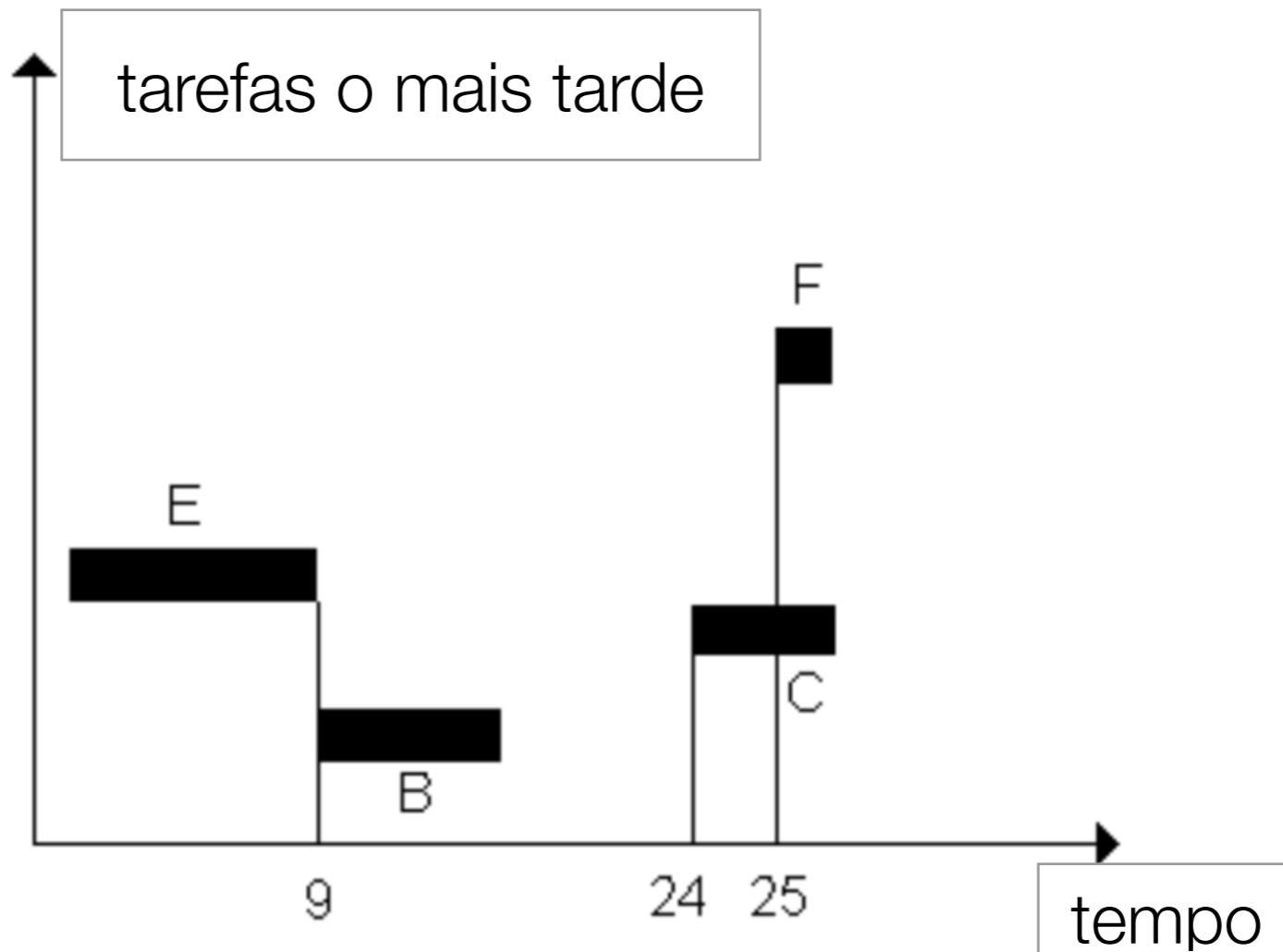


a data de fim o mais tarde das tarefas J, F, C e H é a data de início o mais tarde da tarefa I: 27 dias

Para calcular as datas de início o mais tarde de J, F, C e H, basta subtrair a duração às datas de fim o mais tarde e obtemos respetivamente 23, 25, 24 e 20

Exemplo 3

- As tarefas E e A têm mais de uma tarefa a seguir. A data de fim o mais tarde é a mais pequena das datas de início o mais tarde das tarefas seguintes.
- Por exemplo para a tarefa E que tem como tarefas seguintes F, C e B, a data de início o mais tarde será escolhida entre as datas 25, 24 e 9; a data escolhida é a mais pequena e portanto E vai terminar o mais tarde na data 9.

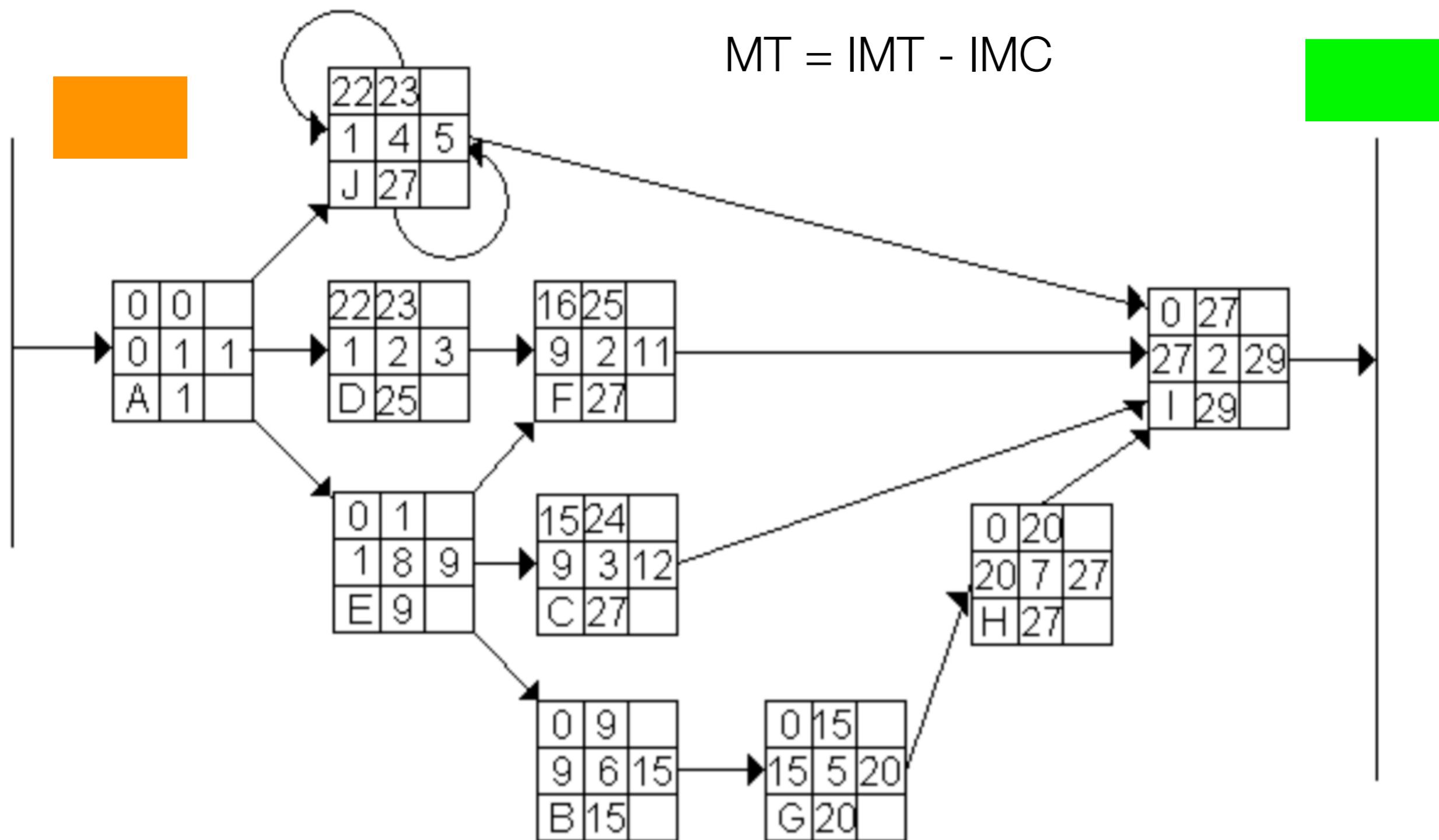


Exemplo 3 - Margem Total

- A margem total de uma tarefa é igual à diferença entre IMT e IMC de uma mesma tarefa.
- Ela indica o **atraso máximo** que poderia assumir a tarefa sem atrasar o término do projeto.

$$MT = IMT - IMC$$

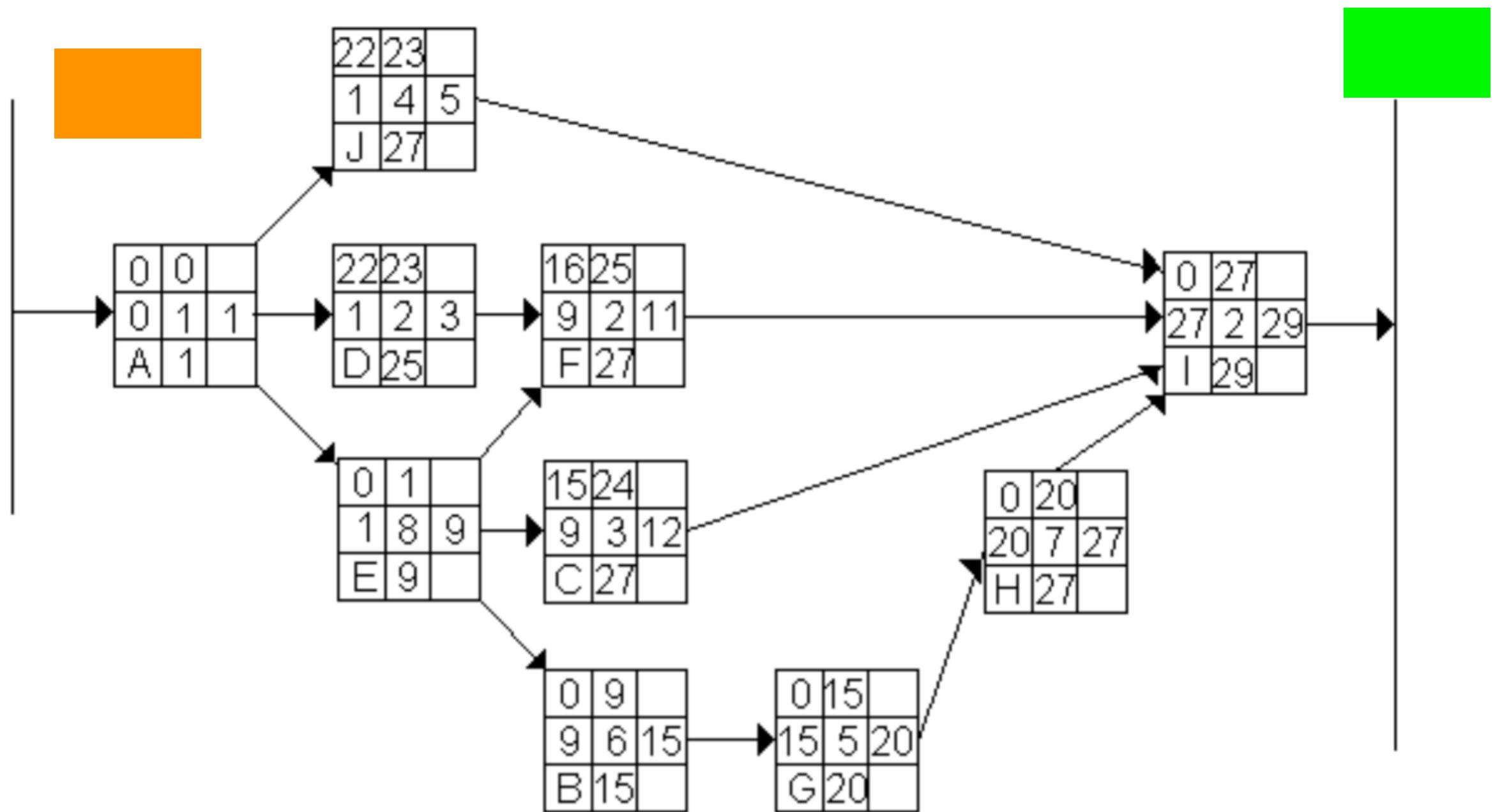
Exemplo 3 - Margem Total



Exemplo 3 - Caminho crítico

- Caminho (os) cuja duração é a mais longa entre o início ou o fim da rede. É constituída por tarefas, cuja a margem total é menor.
- Há sempre pelo menos um caminho crítico. Saber o caminho crítico é fundamental para a gestão do tempo.
- Se todas as margens das tarefas do caminho crítico têm zero, então qualquer atraso em qualquer uma dessas tarefas envolve um atraso na conclusão do projeto (se nenhuma ação corretiva é tomada). Como estas tarefas são, portanto, monitoramento prioritário.
- Além disso, se quisermos reduzir a duração total do projeto será nas tarefas do caminho crítico, que devermos agir primeiro.

Exemplo 3 - Caminho crítico



- Caminho crítico: A,E,B,G,H,I

Exemplo 3 - MARGEM LIVRE

Objetivos

- Cálculo das datas et cálculo das margens
- - Calcular as datas o mais cedo, o mais tarde
- - Calcular as margens livres, as margens totais
- Determinar o caminho crítico

Exercício

CPM

- A empresa Ordomeca estuda o lançamento de uma nova gama de produtos.
- Este lançamento exige a realização de tarefas identificadas pelas letras A a I e cujas características são:

Tarefas	Duração	Predecessores
A	5	D
B	2	G,H
C	5	B
D	4	-
E	2	G,H
F	4	E,I
G	3	-
H	2	D
I	6	A

1. Estabelecer a matriz dos predecessores e classificar as tarefas por níveis.
2. Realizar o gráfe das precedências.
3. Traçar o gráfico PERT das tarefas potenciais com datas o mais cedo, o mais tarde e as margens totais.

Criar planos de iteração

1. No início da reunião de Planeamento do Sprint: Criar um objetivo para a iteração.
2. Reportar a velocidade do projeto a partir do último sprint.
3. Determinar todas as histórias de utilizadores que potencialmente podem ser concluídas dentro do sprint e dividi-las em tarefas de programador.
4. Os programadores devem criar uma estimativa para cada tarefa
5. Reavaliar as histórias de utilizadores escolhidas para o sprint a seguir à realização das estimativas.
6. A equipe de desenvolvimento deve atribuir as tarefas.

Planeamento ágil de produtos de software

Planos de iteração

1. Criar um objetivo para a iteração

Só se o objetivo não estiver definido ainda para este sprint no plano de release.

O objetivo do Sprint é o foco geral para a iteração que está a ser planeada. Definir o objetivo ajuda a manter a equipa de desenvolvimento focada em tarefas relevantes ao longo do sprint.

2. Reportar a velocidade do projeto

Este é um passo importante, já que a velocidade do projeto anterior determinará os compromissos assumidos para o sprint que está a ser planeado.

Nos casos em que o último sprint foi anormal, a **velocidade mais baixa** de sprints anteriores deve ser usado.

3. Partir as histórias de utilizadores escolhidas em tarefas

- As histórias de utilizadores selecionadas para o sprint serão suportadas pelo plano de lançamento. No entanto, as histórias de utilizador do sprint anterior que não foram consideradas "concluídas" (por exemplo, se uma história não passou nos testes de aceitação) também devem ser incluídas no plano de iteração para este sprint.
- O número de histórias de usuários deve ser escolhido com base na velocidade do sprint anterior.
- Isso significa que o sprint só pode conter uma quantidade de histórias em acordo com a sua capacidade, de modo a que a soma das estimativas de ponto de história seja menor ou igual à velocidade anterior. Por exemplo, se o sprint anterior terminou 15 pontos da história, então as histórias de utilizadores devem ser selecionadas de forma a que a some dos pontos de história do sprint sejam igual a 15 pontos no máximo.

3. Partir as histórias de utilizadores escolhidas em tarefas

- As histórias de utilizadores são também escolhidas com base na prioridade - as mais importantes devem ser terminadas em primeiro lugar.
- Quaisquer histórias de utilizadores não escolhidas devem aguardar um sprint futuro.
- Uma vez que as histórias de utilizadores foram escolhidas para o sprint, elas devem ser divididas em tarefas de programador pela equipa de desenvolvimento. Os clientes não estão envolvidos com esta divisão de tarefas. **Uma tarefa de programador deve ser concluída em apenas um ou dois dias.**
- Depois das tarefas terem sido discriminadas, elas devem ser expressas usando técnicas como uma estrutura de divisão de trabalho (WBS), um gráfico de método de caminho crítico (CPM), um gráfico de técnica de avaliação de programa e revisão (PERT), um quadro Kanban, etc.

Um quadro Kanban é um sistema de agendamento desenvolvido pela Toyota. Este sistema organiza a produção numa cadeia lógica.

4. Criar estimativas de tarefas

- Todas as estimativas de tarefas devem ser **acordadas** por toda a equipe de programadores.
- Em Scrum, as tarefas são consideradas inter-funcionais. Isso significa que qualquer programador deve ser capaz de assumir qualquer tarefa. No entanto, cada programador tem um ritmo único, portanto, as estimativas podem diferir para a mesma tarefa entre programadores.
- As estimativas das tarefas são, portanto, uma **média acordada em toda a equipa**. Mesmo as tarefas que exigem competências especializadas e, portanto, devem ser feitas por uma pessoa específica da equipe de desenvolvimento, devem ter uma estimativa com a qual toda a equipa concorda.
- As estimativas são dadas em **horas, meio-dias ou dias**. Qualquer tarefa com uma estimativa menor do que essas medidas deve ser agrupada com outras tarefas. Do mesmo modo, qualquer tarefa que se estime ter mais de três dias deve ser dividida em tarefas menores.

5. Reavaliar as histórias de utilizadores

Quando as tarefas têm estimativas, é mais fácil garantir que podem ser realisticamente feitas dentro do sprint.

A reavaliação das histórias de utilizadores, se se verificar que não podem ser concluídas no sprint, é preciso reajustar o compromisso para o sprint.

As histórias de utilizadores e suas tarefas associadas devem ser removidas do plano de iteração até que o tempo total estimado corresponda ao tempo disponível dos programadores.

As estimativas feitas ao nível de uma iteração substituem as estimativas feitas no nível de release. É mais provável que as estimativas do nível de iteração estejam precisas, porque os são mais específicas. Se as estimativas do nível de iteração alterarem as estimativas do nível da release, o plano de release deve ser alterado em conformidade. Normalmente, os planos de release só enquadram alguns sprints e são ajustados regularmente.

6. Atribuição para as tarefas

- Em planos de iteração, as tarefas são auto-atribuídas. Isso significa que os gestores não atribuem tarefas aos membros da equipa de desenvolvimento, mas sim, os programadores optam por trabalhar no que eles acham interessante.
- Isso resulta geralmente no aumento do nível de satisfação dos colaboradores, e da qualidade do software.
- Os programadores devem escolher tarefas com base na disponibilidade face a outros compromissos ou projetos nos quais estão envolvidos na altura.

Exemplo de plano de iteração

- **OBJETIVO DO SPRINT:** Neste sprint, a equipe está a criar os elementos do perfil do utilizador. Na conclusão deste sprint, os utilizadores devem ser capazes de criar contas, fazer login e editar os seus perfis.
- **ÚLTIMO SPRINT:** No último sprint, completamos 33 pontos de história. Com base em três programadores que trabalham aproximadamente 40 horas por semana para o sprint de uma duração de duas semanas.

Nome / Descrição	Prioridade	Tamanho estimado (pontos)	Atribuido a	Estimativa de esforço (dias)
Criar base de dados para as contas de utilizador	1	2	João	3
Desenhar a página de Log in	1	2	João	1
Desenhar a página de criação de conta	1	1	Ana	1
Desenhar a página de perfil de utilizador	1	1	Paulo	1
Programar a funcionalidade da base de dados das páginas de login	3	1	João	2
Programar a funcionalidade da base de dados para a página da criação de conta	3	1	Ana	2
Programar página de login	2	1	João	2
Programar página de criação de conta	2	1	Ana	2
Programar página de perfil	2	2	Paulo	2
Escrever testes para o login na conta	2	1	João	2
Escrever testes para a criação da conta	2	1	Ana	2

Nome / Descrição	Prioridade	Tamanho estimado (pontos)	Atribuído a	Estimativa de esforço (dias)
Escrever testes para a mudança de perfil de utilizador	2	1	Paulo	2
Escrever testes para a mudança da informação do utilizador	2	1	Paulo	2
Executar teste de aceitação da página de login	4	2	João	2
Executar teste de aceitação da página de criação de conta	4	2	Ana	2
Executar teste de aceitação da página de perfil	4	2	Paulo	2
Escrever descrição do manual de utilizador para a página de login	4	1	João	1
Escrever descrição do manual de utilizador para a página de login	4	1	Ana	1
Escrever descrição do manual de utilizador para a mudança de perfil	4	1	Paulo	1
TOTAL		25		33

Exemplo de cronograma parcial sob a forma de tabela

Software Selection Project Schedule								
ID	Task Name	Work	Duration	Start	Finish	Predecessors	Resource Names	
0	Software Selection Project	1,112 hrs	46.5 days	Mon 1/12/04	Tue 3/16/04			
1	1 Planning	32 hrs	2 days	Mon 1/12/04	Tue 1/13/04			
2	1.1 Determine selection strategies	24 hrs	8 hrs	Mon 1/12/04	Mon 1/12/04		Technology Advisor,Business Analyst,Project Leader	
3	1.2 Determine final schedule	8 hrs	8 hrs	Tue 1/13/04	Tue 1/13/04	2	Project Leader	
4	2 Define Requirements	144 hrs	11 days	Wed 1/14/04	Wed 1/28/04	1		
5	2.1 Develop Technical Requirements	24 hrs	24 hrs	Wed 1/14/04	Fri 1/16/04		Technology Advisor	
6	2.2 Develop Vendor Requirements	16 hrs	8 hrs	Mon 1/19/04	Mon 1/19/04	5	Technology Advisor,Business Analyst	
7	2.3 Develop Future State Business Requirements	16 hrs	16 hrs	Tue 1/20/04	Wed 1/21/04	6	Business Analyst	
8	2.4 Prioritize Requirements	48 hrs	16 hrs	Thu 1/22/04	Fri 1/23/04	5,6,7	Technology Advisor,Business Analyst,Project Leader	
9	2.5 Identify Vendor Knockout Criteria	12 hrs	4 hrs	Mon 1/26/04	Mon 1/26/04	8	Technology Advisor,Business Analyst,Project Leader	
10	2.6 Determine Vendor Long List	12 hrs	4 hrs	Mon 1/26/04	Mon 1/26/04	9	Technology Advisor,Business Analyst,Project Leader	
11	2.7 Finalize Requirements Document	16 hrs	16 hrs	Tue 1/27/04	Wed 1/28/04	10	Business Analyst	
12	3 Develop Vendor Short List	88 hrs	7 days	Tue 1/27/04	Wed 2/4/04			
13	3.1 Research Vendors on Long List	40 hrs	5 days	Tue 1/27/04	Mon 2/2/04	10	Technology Advisor	
14	3.2 Determine Vendor Short List	8 hrs	1 day	Tue 2/3/04	Tue 2/3/04	13	Technology Advisor	
15	3.3 Review Vendor Short List	8 hrs	1 day	Wed 2/4/04	Wed 2/4/04	14	Technology Advisor	
16	4 Develop Vendor Finalist List	120 hrs	18.5 days	Thu 1/29/04	Tue 2/24/04			
17	4.1 Develop Request for Proposal (RFP)	64 hrs	8 days	Thu 1/29/04	Mon 2/9/04	4		
18	4.1.1 Create Request for Proposal	32 hrs	32 hrs	Thu 1/29/04	Tue 2/3/04		Business Analyst	
19	4.1.2 Approve Request for Proposal	24 hrs	24 hrs	Wed 2/4/04	Fri 2/6/04	18	Business Analyst	
20	4.1.3 Distribute Request for Proposal	8 hrs	8 hrs	Mon 2/9/04	Mon 2/9/04	19	Business Analyst	
21	4.2 Process RFP Responses	24 hrs	1.5 days	Mon 2/16/04	Wed 2/18/04	17FS+5 days		
22	4.2.1 Vendor RFP Responses Due	0 hrs	0 days	Mon 2/16/04	Mon 2/16/04			
23	4.2.2 Review and Summarize Vendor Responses	16 hrs	8 hrs	Tue 2/17/04	Tue 2/17/04	22	Technology Advisor,Business Analyst	
24	4.2.3 Research Vendor References	8 hrs	4 hrs	Wed 2/18/04	Wed 2/18/04	23	Technology Advisor,Business Analyst	
25	4.3 Develop Preliminary Fit Analysis	16 hrs	16 hrs	Wed 2/18/04	Fri 2/20/04	21	Technology Advisor	
26	4.4 Determine Vendor Demonstration List	8 hrs	8 hrs	Fri 2/20/04	Mon 2/23/04	25	Technology Advisor	
27	4.5 Schedule Demonstrations	8 hrs	8 hrs	Mon 2/23/04	Tue 2/24/04	26	Project Leader	
28	5 Evaluate Finalists	400 hrs	25.5 days	Tue 2/10/04	Tue 3/16/04			
29	5.1 Develop Demo Test Scripts	48 hrs	24 hrs	Tue 2/10/04	Thu 2/12/04	17	Technology Advisor,Business Analyst	
30	5.2 Assess Performance	224 hrs	8 days	Tue 2/24/04	Fri 3/5/04	16		
31	5.2.1 Assess Package Performance	128 hrs	6 days	Tue 2/24/04	Wed 3/3/04			
32	5.2.1.1 Working Session - Vendor A	12 hrs	4 hrs	Tue 2/24/04	Tue 2/24/04	16	Technology Advisor,Business Analyst,Project Leader	
33	5.2.1.2 Working Session - Vendor B	12 hrs	4 hrs	Wed 2/25/04	Wed 2/25/04	32	Technology Advisor,Business Analyst,Project Leader	
34	5.2.1.3 Vendor "A" Demonstration	48 hrs	16 hrs	Wed 2/25/04	Fri 2/27/04	32,29,27,33	Technology Advisor,Business Analyst,Project Leader	
35	5.2.1.4 Vendor "B" Demonstration	48 hrs	16 hrs	Fri 2/27/04	Tue 3/2/04	34	Technology Advisor,Business Analyst,Project Leader	
36	5.2.1.5 Record Demo Scores	8 hrs	8 hrs	Tue 3/2/04	Wed 3/3/04	35	Business Analyst	
37	5.2.2 Assess Vendor Performance	96 hrs	8 days	Tue 2/24/04	Fri 3/5/04			
38	5.2.2.1 Research vendor references	48 hrs	16 hrs	Tue 2/24/04	Thu 2/26/04	16	Technology Advisor,Business Analyst,Project Leader	
39	5.2.2.2 Possible Site visits	48 hrs	16 hrs	Wed 3/3/04	Fri 3/5/04	31	Technology Advisor,Business Analyst,Project Leader	
40	5.3 Develop Final Fit Analysis	80 hrs	5 days	Fri 3/5/04	Fri 3/12/04	30		
41	5.3.1 Perform Fit-Gap analysis for each option	16 hrs	8 hrs	Fri 3/5/04	Mon 3/8/04		Technology Advisor,Business Analyst	
42	5.3.2 Assess risk for each option	16 hrs	8 hrs	Mon 3/8/04	Tue 3/9/04	41	Technology Advisor,Business Analyst	
43	5.3.3 Develop high-level schedule and budget forecast	16 hrs	8 hrs	Tue 3/9/04	Wed 3/10/04	42	Technology Advisor,Business Analyst	

Exemplo de cronograma parcial sob a forma de diagrama de Gantt.

