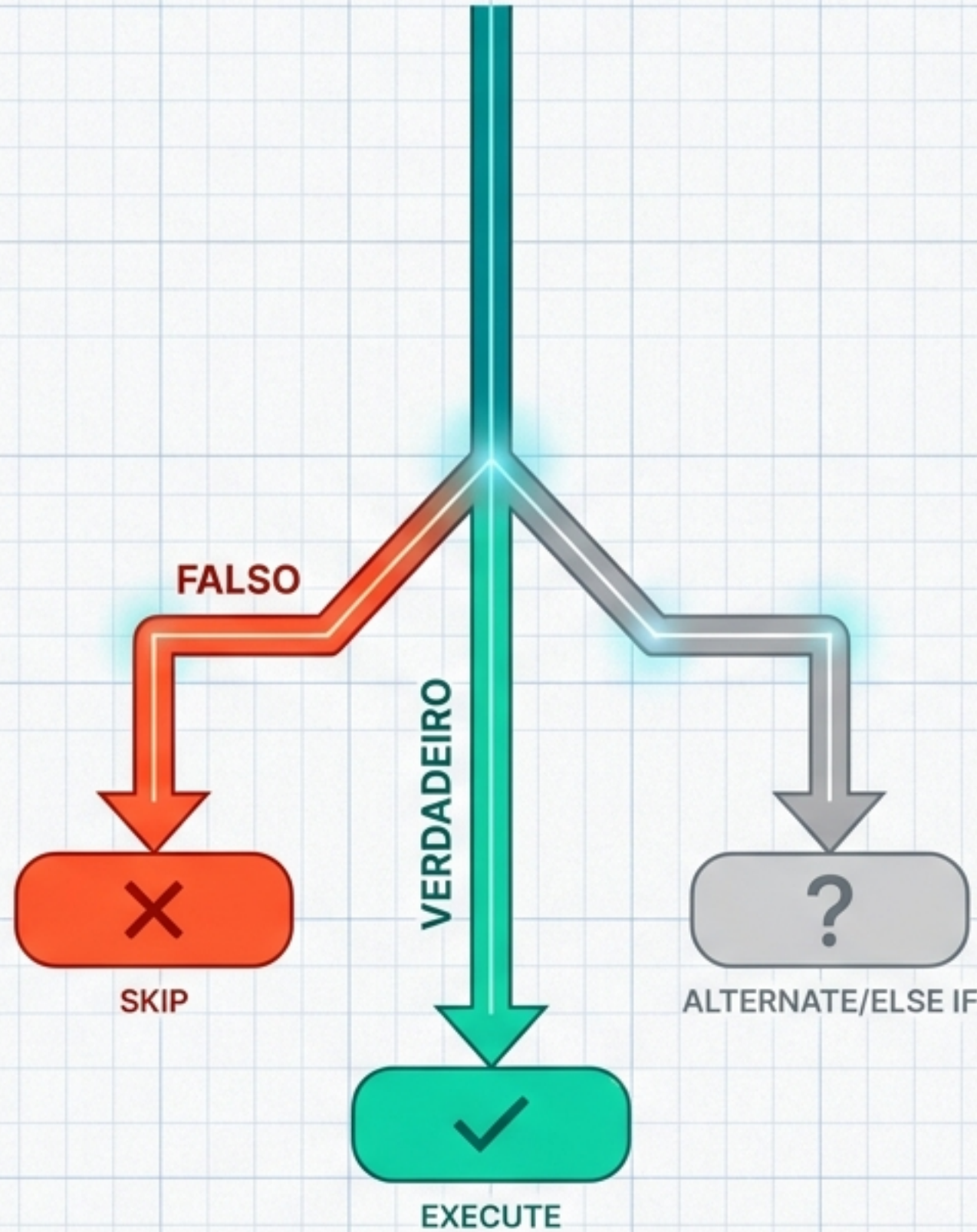


A Lógica das Decisões: Dominando IF, ELSE e ELSE IF



```
// Apostila interativa com  
conceitos, dicas de ouro,  
exercícios práticos e gabarito  
completo.
```

The Blueprint of Decisions

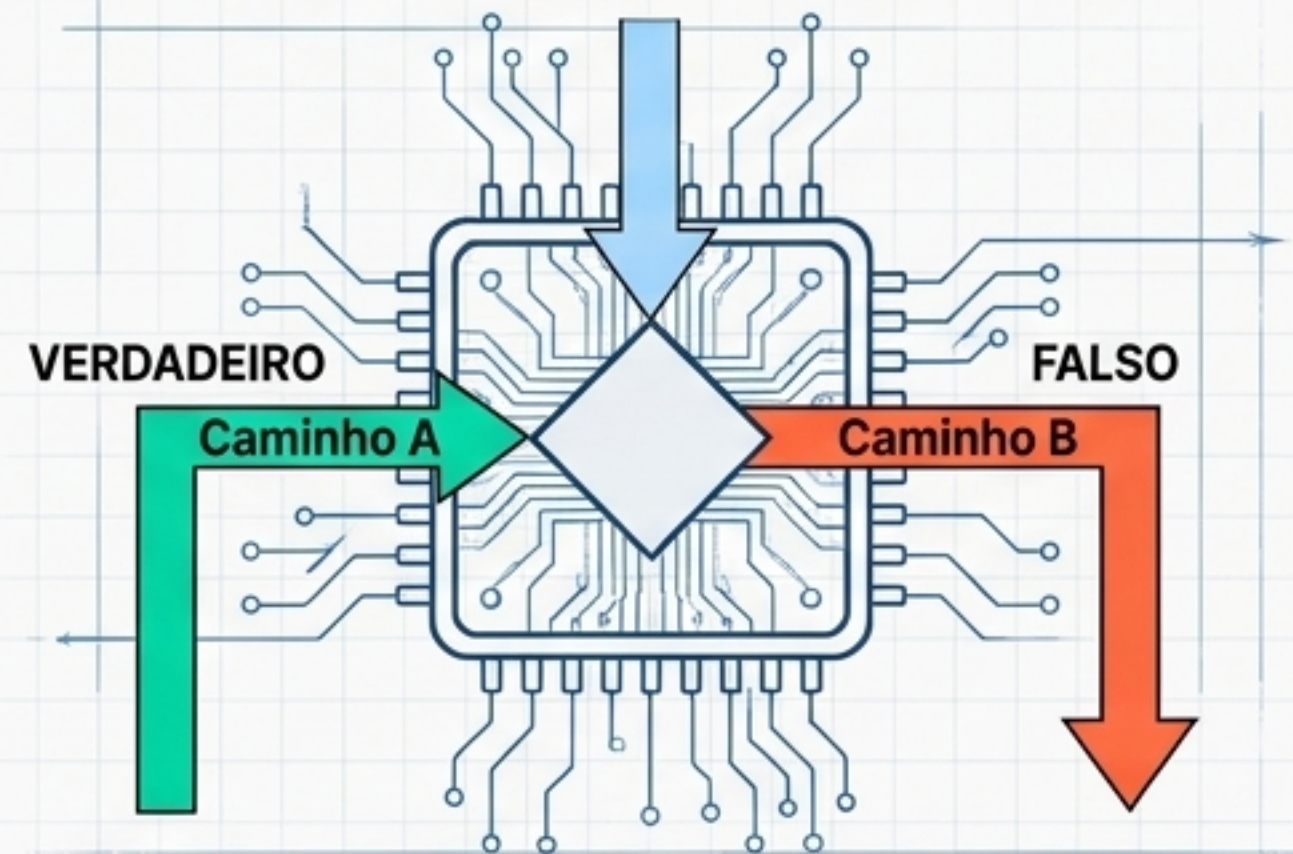
A Mente do Algoritmo

Lógica Humana



Na vida real, nossa mente funciona com lógicas condicionais: **"Se estiver chovendo, levo um guarda-chuva; senão, uso óculos de sol."**

Lógica do Algoritmo



Até agora, seus algoritmos executavam linha por linha, sem pensar. Com as **Estruturas Condicionais**, damos inteligência ao código. O algoritmo passa a decidir entre o **Caminho A** ou **Caminho B** com base no resultado de uma condição.

A Caixa de Ferramentas (Operadores Relacionais)

Toda condição em C é uma pergunta que só aceita duas respostas: **VERDADEIRO** ou **FALSO**. Para fazer essas perguntas, usamos os **Operadores Relacionais**.

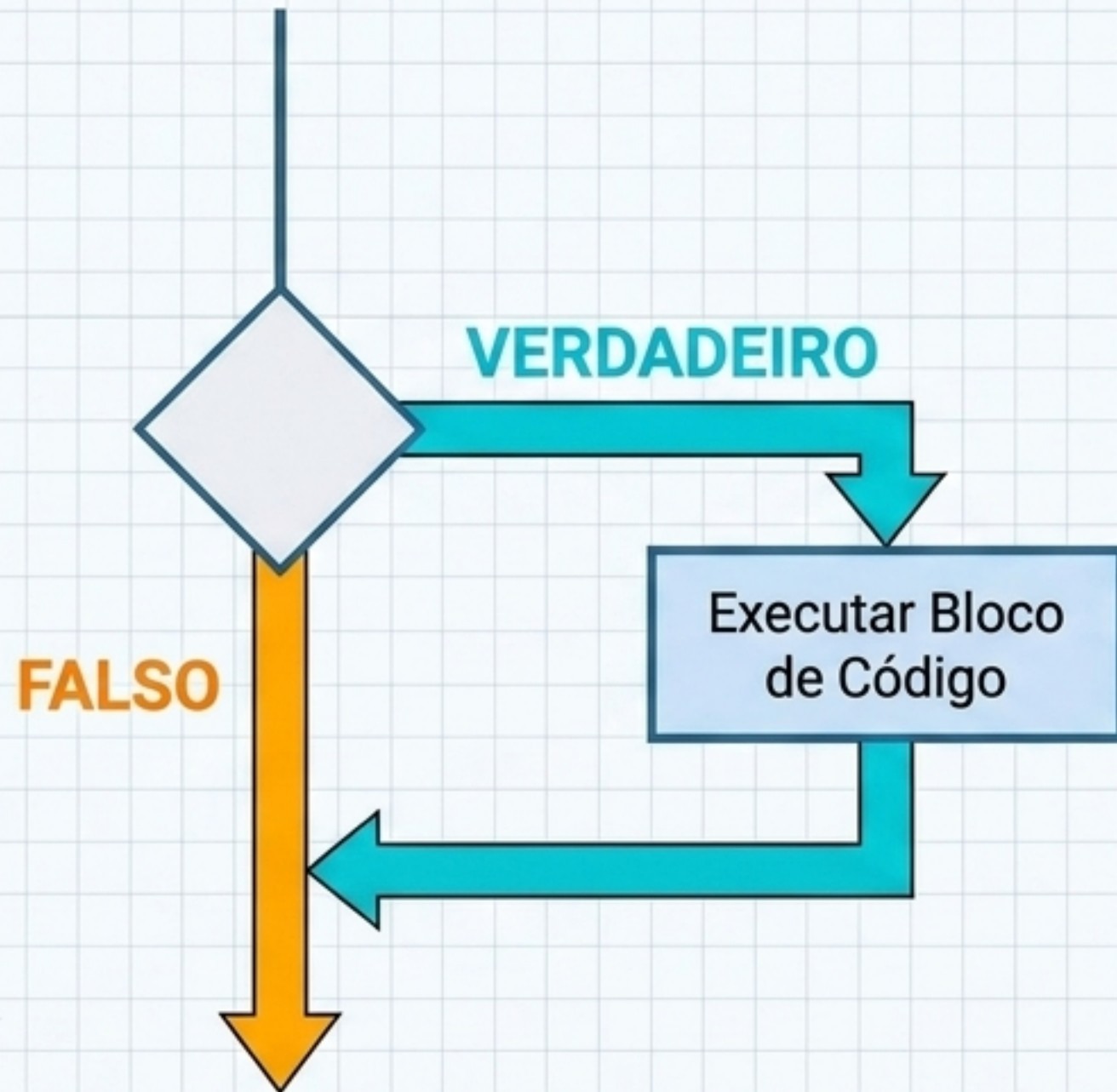
Símbolo	Significado	Exemplo de Pergunta
>	Maior que	<code>idade > 18</code>
<	Menor que	<code>peso < 70.5</code>
>=	Maior ou igual	<code>nota >= 6.0</code>
<=	Menor ou igual	<code>velocidade <= 120</code>
==	Igual a	<code>codigo == 10</code>
!=	Diferente de	<code>opcao != 0</code>



Atenção! Um sinal de igual (=) serve para atribuir um valor. Para comparar, use sempre dois sinais (==)!

Nível 1 – Nível 1 - A Decisão Simples (if)

O comando **if (SE)** avalia uma expressão. O bloco de código interno só será executado se a resposta for **VERDADEIRA**.



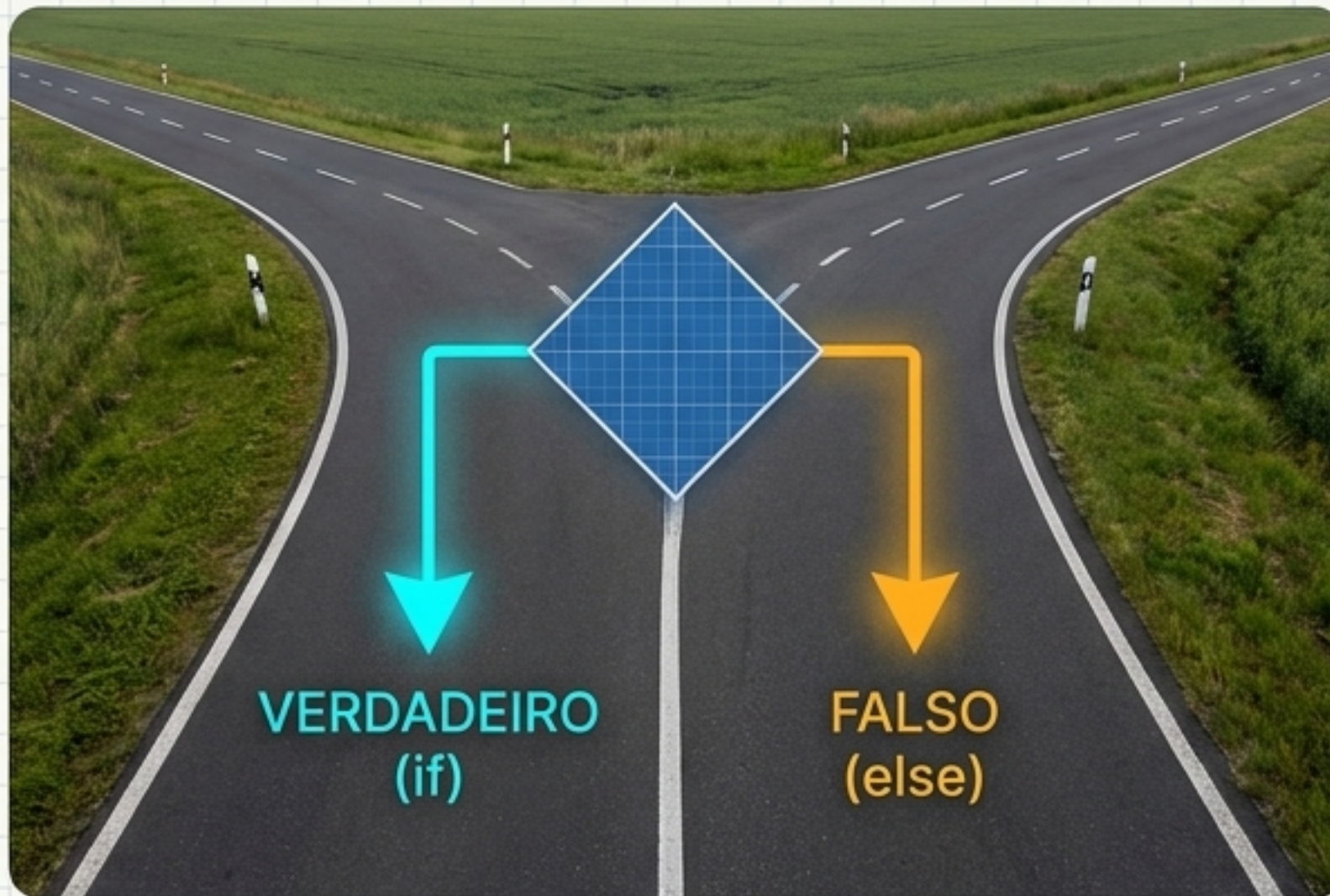
```
if (idade < 18) {  
    // Este código só roda se a  
    // verificação for VERDADEIRA  
    printf("Menor de idade");  
}
```

Se a idade for 15, a mensagem aparece. Se for 20, o programa simplesmente ignora o bloco e segue a vida.

The Blueprint of Decisions

Nível 2 - A Bifurcação (if / else)

E se precisarmos garantir uma ação alternativa quando a condição falhar?
Usamos o comando **else** (SENÃO).



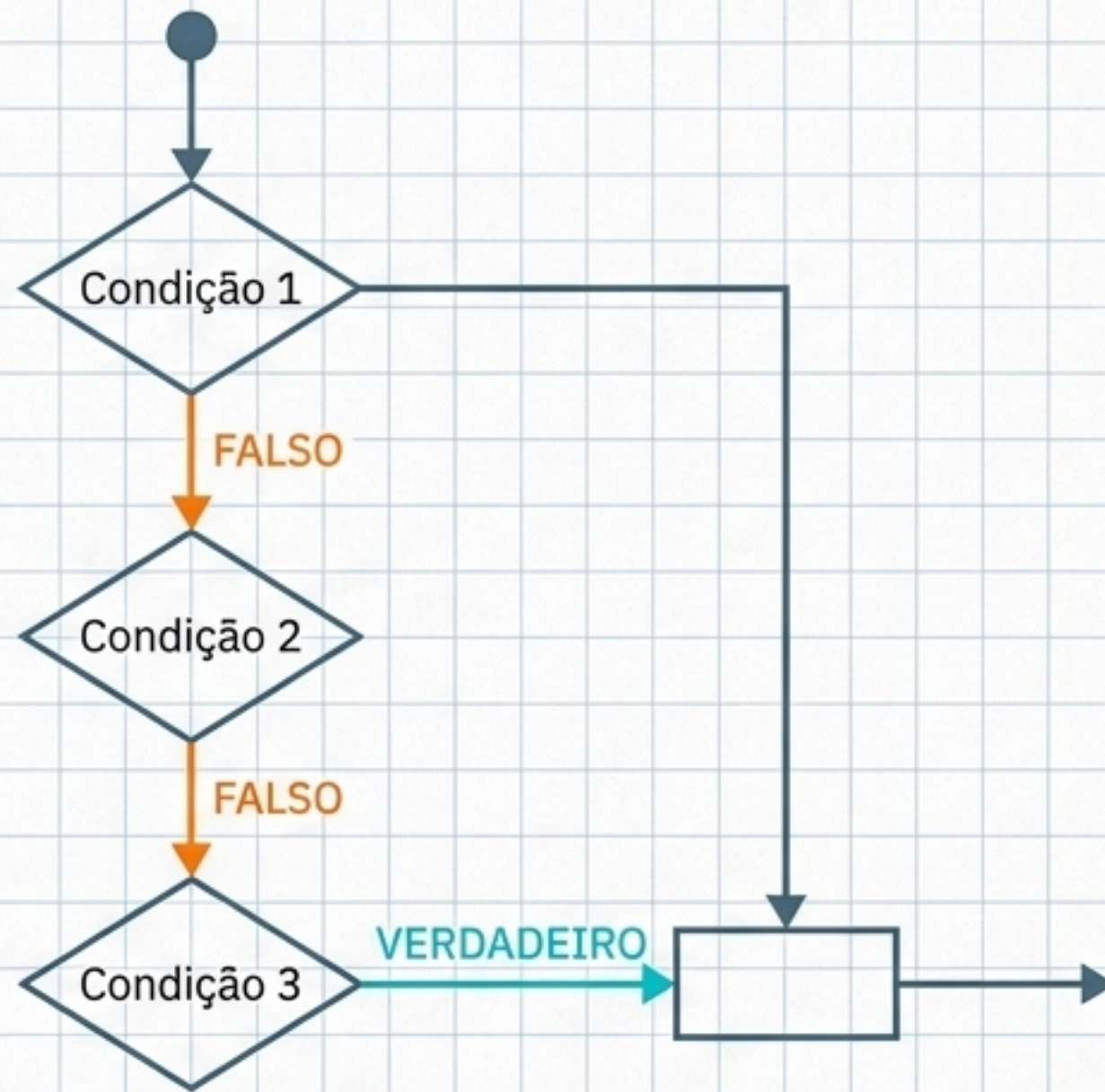
```
if (idade < 18) {  
    printf("Menor de idade");  
} else {  
    // Resultado FALSO cai aqui  
    printf("Maior de idade");  
}
```



O **else** nunca recebe uma condição própria. Ele é o **cesto de lixo** — abraça automaticamente tudo que foi rejeitado pelo **if**.

Nível 3 - Múltiplas Escolhas (else if)

Quando temos mais de dois cenários possíveis, encadeamos as decisões de forma inteligente.



```
if (expressao_1) {  
    // Executa se 1 for Verdadeira  
} else if (expressao_2) {  
    // Executa se 2 for Verdadeira  
} else if (expressao_N) {  
    // Pode ter quantos else if quiser  
} else {  
    // Se TUDO for falso, cai aqui (opcional)  
}
```

The Blueprint of Decisions

O Raio-X do Compilador (A Regra de Ouro)

O compilador testa a estrutura de cima para baixo. Apenas UMA saída é possível.

```
if (expressao_1) {  
    // Executa se 1 for Verdadeira  
} else if (expressao_2) {  
    // Executa se 2 for Verdadeira  
} else if (expressao_3) {  
    // Executa se 3 for Verdadeira  
} else {  
    // Se TUDO for falso, cai aqui  
}
```

Condição VERDADEIRA encontrada!

Assim que o programa encontra a primeira condição VERDADEIRA, ele executa aquele bloco e pula todo o resto da estrutura. Ele não procura outra verdade. Isso garante eficiência e evita bugs.

Por que usar `else if` e não vários `ifs` soltos?

Lento e Arriscado:
Vários IFs independentes

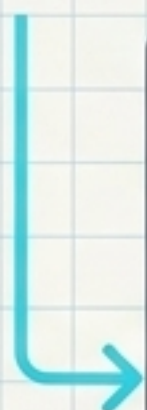


```
if (nota >= 9) {  
    printf("A");  
}  
if (nota >= 7) {  
    printf("B");  
}  
if (nota < 7) {  
    printf("C");  
}
```



Avaliação Forçada: O computador avalia TODAS as três condições obrigatoriamente, gastando processamento à toa. Pode gerar bugs se as lógicas se sobrepuserem (ex: uma nota 10 ativaria o primeiro e o segundo IF).

Otimizado e Seguro:
Encadeamento com ELSE IF



```
if (nota >= 9) {  
    printf("A");  
}  
else if (nota >= 7) {  
    printf("B");  
}  
else {  
    printf("C");  
}
```



Estrutura Inteligente: Achou a verdade na primeira linha? Ignora as demais instantaneamente. As condições são mutuamente exclusivas (só uma vence).

Dicas de Sobrevivência em C

Dica 1: O Ponto Cego (Valores Limite)

Se você testar `if (idade < 18)` e logo um `else`, o que acontece se a idade for exatamente 18? Ele vai pro `else`. Cuidado com os limites! Lembre-se de usar `<=` ou `>=` quando necessário.

Dica 2: As Chaves { }

Se o seu `if` tiver apenas uma linha, as chaves são opcionais, mas use-as sempre enquanto for iniciante para evitar bugs visuais de indentação.

```
if (x > 0) {  
    printf("Positivo");  
}
```

Dica 3: O Else é Opcional

Você não é obrigado a fechar um encadeamento com `else`, mas ele é excelente para capturar erros (ex: usuário digitou um número negativo inválido em um menu).

Caderno de Exercícios (Tente fazer antes de olhar o gabarito!)



Exemplo: Radar e Velocidade Permitida

Exercício 1 (Radar):

Monte um programa para um radar eletrônico. Leia a velocidade informada no teclado. Exiba "MULTADO" se a velocidade lida for **maior que 120km/h**. (Dica: Use if simples).

Notes:

Exercício 2 (Números): Leia um número e imprima se ele é POSITIVO (maior que 0) ou NEGATIVO (menor que 0). (Dica: Que tal usar o else para capturar o número zero como neutro?).

Notes:

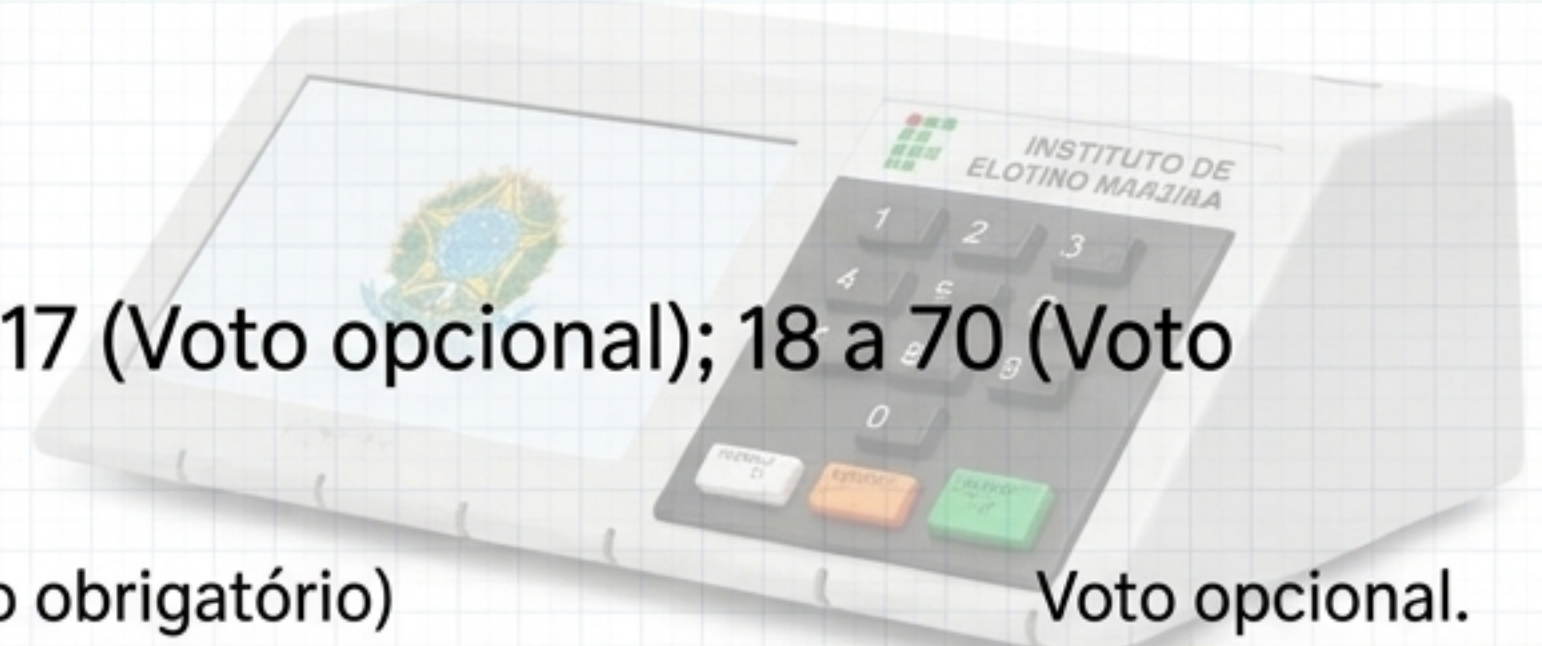


Reta Numérica: Negativos, Zero, Positivos

Exercício 3 (Sistema Eleitoral)

Leia uma idade.

Regras: Menor que 16 (Não pode votar); 16 e 17 (Voto opcional); 18 a 70 (Voto obrigatório); Maior que 70 (Voto opcional).



Exercício 4 (Palco do Show)

Solicite um código de apresentação e exiba a banda e o palco correspondente.

Códigos: 10 (Dilsinho/Mundo), 20 (Trio Dona Zefa/Forró), 30 (Dj Denis/Funk), 40 (Nós do Morro/Supernova).

(Dica: Encadeamento perfeito para else if).



Gabarito: Radar e Positivo/Negativo

Solução Ex. 1 - Radar

```
int vel;  
scanf("%d", &vel);  
if (vel > 120) {  
    printf("MULTADO\n");  
}
```

Solução Ex. 2 - Pos/Neg

```
int num;  
scanf("%d", &num);  
if (num > 0) {  
    printf("Positivo\n");  
} else if (num < 0) {  
    printf("Negativo\n");  
} else {  
    printf("Zero / Neutro\n");  
}
```

Repare como o else final captura inteligentemente o número 0 sem precisarmos testá-lo explicitamente com uma condição!

Gabarito: Sistema Eleitoral



```
int idade;  
scanf("%d", &idade);  
if (idade < 16) {  
    printf("Não pode votar\n");  
} else if (idade < 18 || idade > 70) {  
    // Usamos o operador lógico OU (||)  
    printf("Voto opcional\n");  
} else {  
    // Se não caiu nas de cima, com certeza está entre 18 e 70!  
    printf("Voto obrigatório\n");  
}
```



Note como a inteligência da estrutura em cascata nos poupou de testar ($\text{idade} \geq 18 \ \&\& \ \text{idade} \leq 70$). O compilador já filtrou essas possibilidades para nós nas linhas anteriores!

Gabarito: Agendamento de Shows

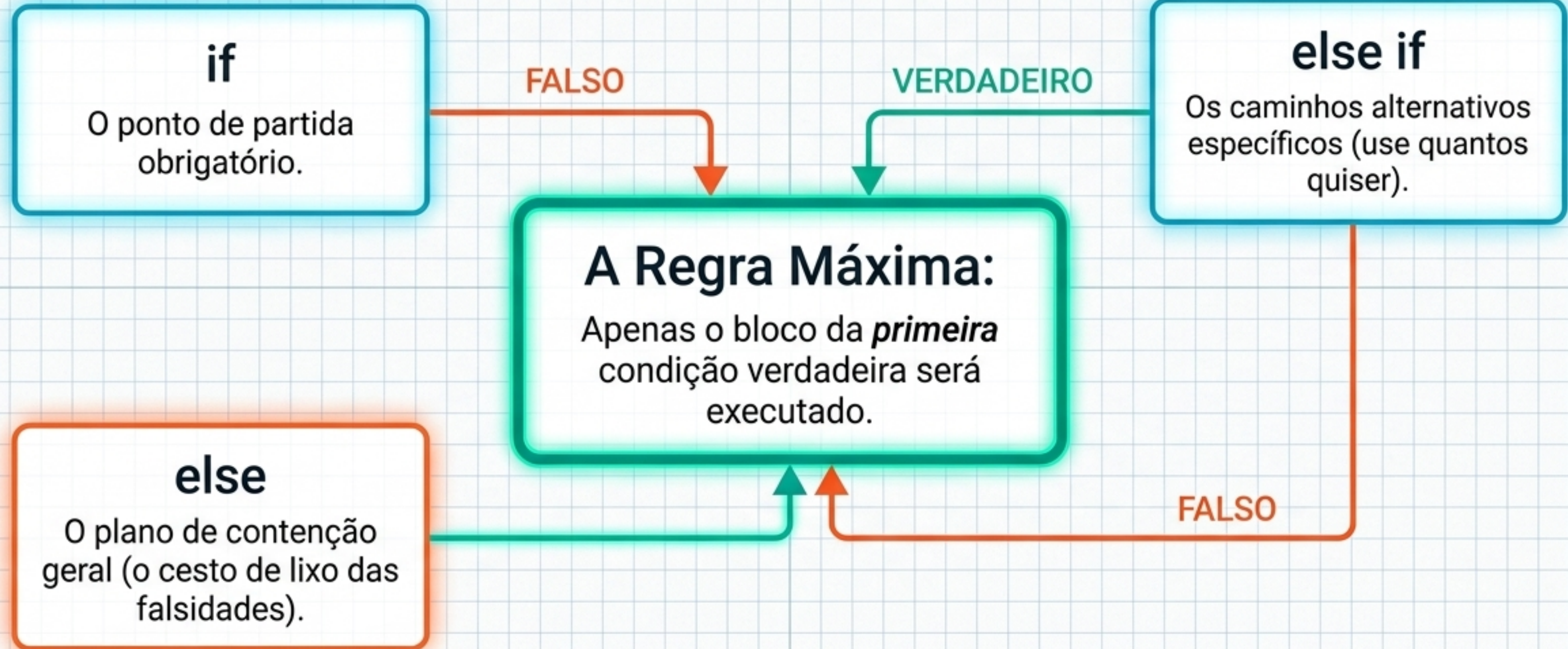


```
int cod;  
scanf("%d", &cod);  
  
if (cod == 10) {  
    printf("Dilsinho – Palco Mundo\n");  
} else if (cod == 20) {  
    printf("Trio Dona Zefa – Palco Forró\n");  
} else if (cod == 30) {  
    printf("Dj Denis – Palco Funk\n");  
} else if (cod == 40) {  
    printf("Nós do Morro – Palco Supernova\n");  
} else {  
    printf("Código Inválido!\n");  
}
```



O else final atua como um sistema de segurança contra digitações incorretas do usuário.

Revisão Express



Domine a lógica, compile sem medo e continue praticando. Seu código agora pensa!