
---

# Código da Vida em C++

## 📖 Episódio 3

## Microburacos de Minhoca

### 🎬Parte 1

### O Pedido de Zaion

No meio da aula, o celular de Zaion vibrou. Ele olhou discretamente e leu a mensagem do seu assistente, Beta Cinco:
“Aviso: compromisso confirmado com Sr. Paulo Martin, empresa LigLog Logística, cidade Vitória.
Voo: saída 14h00 do heliporto da Informática Zanarelli.
Retorno previsto: 18h00. Preparativos concluídos.”
— Puxa, tinha me esquecido! E agora, quem vai tomar conta dos meninos? — pensou, preocupado. — O Marcos e a Rita foram às compras… que problemão.
— Por que você não pede para a Kael? Acho que ela pode te ajudar — sugeriu Zahroniel.
— Será que ela topa, Zahy?

— Você só vai saber se pedir.
— É… vou tentar.
Zaion se aproximou de Kael, um pouco hesitante:
— Kael, tudo bem? Você está ocupada hoje à tarde?
Ela ergueu os olhos, curiosa:
— Oi, Zaion! Sim, estou trabalhando com meus pais.
Zaion titubeou. Zahroniel interveio em tom firme:
— Fala logo pra ela. Não enrola, Zaion.
Respirando fundo, ele disparou:
— Estou com um problemão. Tenho que ir para Vitória à tarde e não tenho com quem deixar os meninos. Você poderia cuidar deles, por favor?
Kael cruzou os braços, pensativa:
— Não sei, Zaion… eu também tenho trabalho.
— Ah, por favor! — insistiu ele.
Ela suspirou, mas sorriu de leve:
— Tá bom, mas eles vão ter que ir comigo para o laboratório.
— Beleza. Eu até levaria, mas só tem um lugar disponível no voo.
— Você sabe que saímos ao meio-dia, e eles só às 14h, né?
— Sim. Meu assistente vai se encarregar de levá-los até você. Depois eu busco.
— Tá, combinado. Mas olha… você vai ficar me devendo essa.
Zaion riu, aliviado:
— Pode deixar. Obrigado, Kael!

---

## 🎬Parte 2

### A Tarefa de Kael

— Boa tarde, Sr. Leo, Sr. Darian. Eu os levarei ao laboratório, onde a Sra. Kael os espera. Tenham uma boa viagem.
— Boa tarde... quem é você mesmo? — perguntou Leo.
— Sou Beta Cinco. Trabalho para o Sr. Zaion e estou incumbido de levá-los até o laboratório.
— Vai demorar muito? — questionou Ryo.

— A viagem levará vinte e três minutos, dependendo das condições do tráfego.
— Chegamos. Dirijam-se à recepção, a Sra. Kael os espera.

(No laboratório, Kael analisava os servidores e sistemas dos androides, à procura da anomalia apontada por seus pais, quando foi chamada à recepção.)

Kael entregou-lhes crachás de visitante:
— Boa tarde! Estes crachás têm chip de aproximação. Vocês poderão abrir algumas portas — não todas, claro — e também usá-los nas máquinas de comida. Vamos até o laboratório.
Pouco depois:
— Chegamos. Podem usar aqueles computadores. Eles têm alguns jogos, mas, por favor, sem barulho.
— Puxa, esses jogos são chatos... — reclamou Leo.
— Podem ser, mas você não faz mais pontos que eu! — provocou Ryo.
— Faço sim!
— Isso eu quero ver!
Ryo bateu recorde e exibiu:
— 10.584 pontos! Quero ver você ganhar!
— Ah, só 9.809... Vamos de novo! — insistiu Leo.
Minutos depois, Ryo desistiu:
— Já cansei desse jogo. A Kael deve ter coisa mais legal...
Kael ergueu a voz:
— Meninos, silêncio! Estou trabalhando.
Um silêncio desconfortável se instalou, mas não durou muito.
— O que você está fazendo? — perguntou Leo.
— Procurando anomalias nos servidores dos androides. — Kael suspirou. — Nunca pensei que fosse tão difícil lidar com crianças...
— Podemos ajudar! — animou-se Ryo.
— Melhor não.
— Ah, deixa vai, só um pouco! — pediu Leo.
Kael hesitou, depois cedeu:
— Está bem. Mas prometam que vão me avisar se encontrarem algo estranho. Vou

abrir o login neste computador aqui, ao lado de vocês, assim posso supervisionar.
— Tá, e o que a gente faz? — perguntou Ryo.
— Sigam este roteiro impresso. Façam as perguntas aos androides e, se houver resposta estranha ou erro, me avisem.
— Vamos escolher este aqui! — apontou Ryo.

(Depois de alguns testes...)

— Já vimos três. É tudo igual, que chato! — reclamou Leo.
— Ei, aquele não é o assistente do Zaion... o Beta Cinco? — disse Ryo, animado.
— É mesmo! Vamos testar ele.
Fizeram algumas perguntas. Pouco depois, Ryo bufou:
— Igualzinho! Como ele pode ser tão chato?
— Então que tal o Zeta? — sugeriu Ryo, com um sorriso maroto.
— É... vamos testá-lo. Aposto que ele é diferente.
Quando terminaram, Leo suspirou, entediado, olhando para a tela:
— Igualzinho... — murmurou, com o olhar vago.
De repente, Leo desviou a atenção para a porta e arregalou os olhos:
— Zeta, aqui! Somos nós! — gritou Leo.
Kael interveio imediatamente:
— Ele não pode ouvir vocês... Meninos, não! Não saiam daqui!
Mas já era tarde. Os dois dispararam porta afora, correndo atrás de Zeta.
Kael esfregou a testa, exasperada:
— Por que eles nunca obedecem? Ah, o Zahroniel me paga... isso só pode ter sido ideia dele.
Respirou fundo, tentando recuperar o foco:
— Bom, ao menos ficou um pouco de silêncio. Vamos ver o que eles conseguiram fazer aqui...
Ela checou a tela:

— Tudo certo. Ao menos me ajudaram: menos cinco testes na lista.

(No corredor...)

— Boa tarde, meninos! Onde está o Zaion? — perguntou Zeta.
— Ele viajou. A Kael está cuidando da gente. — respondeu Leo.
— A Kael? E já estão entediados, não é? — sorriu Zeta.
— Um pouco... — disse Ryo.
— Um pouco? O trabalho dela é o tédio em pessoa! Você não tem jogos melhores? — reclamou Leo.
— Tenho, mas estão em casa... Quer dizer, no meu quarto. A não ser...
— A não ser o quê, Zeta? — perguntaram juntos.
— Estou reprogramando um simulador de voo. Querem ver?
— Sim, sim! — gritaram os dois em coro.
Pouco depois, já no laboratório de Zeta:
— Uau, que legal! A gente pode pilotar um avião? — perguntou Ryo.
— Claro. O que querem?
— Um Boeing 737! — disse Leo.
— Não, um caça! — pediu Ryo.
— Tem sim. Que tal um F-22?
— Perfeito! — respondeu Ryo, animadíssimo.

(No fim da tarde, Zaion chega ao laboratório e encontra Kael.)

— Boa tarde, Kael! — cumprimentou com um sorriso, sem notar a ausência dos meninos. — Obrigado por cuidar deles. Deram muito trabalho?
Kael forçou um sorriso:
— Não. Até me ajudaram. Analisaram cinco androides e não acharam nada fora do padrão.
— Vamos até a recepção, que eu vou chamá-los.

(Kael pegou o telefone e avisou Zeta para levar os meninos. Minutos depois, eles chegaram, animados.)

— Zaion, o Zeta tem um laboratório incrível! Tem até um simulador de avião, e nós ajudamos ele a reprogramar...

Mas Zaion parecia mais interessado em Kael do que no que Leo contava. Depois de muito falar, Leo acabou perguntando:

— Mano, posso voltar lá outro dia?

— Não! Isso não é brinquedo. O Zeta está trabalhando. — respondeu Zaion, ainda sem desviar os olhos de Kael.

— Ah, não seja rabugento, Zaion! Deixa eles voltarem. Eles me ajudaram muito. A experiência do Leo com videogames foi incrível. — disse Zeta.

Zaion se virou para ele:

— Oi, Zeta. Se você está pedindo... tudo bem, mas só no tempo livre deles.

— Sr. Zaion, precisamos ir. Tenho um relatório para entregar antes do fim do expediente. Disse Beta Cinco.

— Vamos, meninos. Se despeçam do Zeta e da Kael.

Os dois correram para abraçar Zeta, que retribuiu o abraço apertado.

— Obrigado, Zeta! Foi muito bom estar com você. — disse Ryo.

Kael ficou pasma ao ver a cena, e mal teve tempo de reagir antes que os meninos corressem até ela e também a abraçassem.

— Obrigado, Kael. Foi legal... e desculpa por dizer que seu trabalho é chato. — disse Leo.

Kael congelou, surpresa. O que eu faço? — pensou, antes de se deixar levar e abraçá-los de volta. — Me sinto perdida...

Zaion se aproximou:

— Obrigado pela ajuda, Kael! — disse, envolvendo-a em um leve abraço.

— Ah... foi... um prazer. — murmurou, sentindo algo novo, uma emoção que nunca havia experimentado.

Quando eles partiram, ela permaneceu parada no corredor. O que aconteceu comigo? Quando agradeceram ao Zeta, eu fiquei brava porque preferiram ele. Mas quando me abraçaram... eu só lembrei

dos abraços da minha mãe. Será que era assim que ela se sentia? E o abraço do Zaion... não foi a mesma coisa. Meu coração disparou. Nunca me senti assim...

---

De volta ao laboratório, Kael revisou os dados coletados pelos meninos.
— Minha nossa! Eles tinham razão. Todos os testes dão o mesmo resultado. Como pode? O Zeta e o Beta Cinco são tão diferentes... Isto está errado!
— O que está errado? — perguntou uma voz.
Kael se virou e viu sua mãe.
— Mãe, os testes mostram que Beta Cinco e o Zeta são iguais, mas quando você conversa com eles... o Beta Cinco soa igual a todos os outros. Já o Zeta... ele não parece um androide, parece mais humano.
— Talvez a anomalia tenha afetado cada um de forma individual. Temos que testá-los presencialmente. O que você acha?
— Já fizemos, mas os resultados foram inconclusivos. Se quiser refazer, posso solicitar.
— Eu gostaria muito! Posso começar pelo Zeta? — perguntou Kael, eufórica.
— Tá, eu vejo o que consigo. Mas... o que está rolando entre você e o Zaion?
— O quê? — Kael arregalou os olhos. — Somos só amigos!
— Sei... vi o jeito que você olhou para ele. Pareciam um casal.
— Credo, mãe, nada a ver!
— É... não sei não! — provocou a mãe.
— Quanto ao Zeta, vou solicitar uma audiência para você. Agora vamos embora, o expediente acabou.

---

(Mais tarde, após o jantar...)
— Leo, trouxe esta mochila para você. A sua já estava bem velhinha.
— Que legal, mano! É descolada! Obrigado! — disse Leo, abraçando Zaion.
— Vou trocar minhas coisas já! E não esquece da nossa aula!

Leo correu para o quarto, e Zaion se virou para Ryo: — Ryo, isto é pra você.

— Uau! Eu queria este jogo faz tempo. Obrigado, mano! — disse Ryo, abraçando Zaion de surpresa. Ele retribuiu com carinho.

— Joga comigo agora, por favor!

— Tá... mas só uma partida! — cedeu Zaion.

(Três partidas depois...)

— Ryo, tenho que dar aula para o Leo. Foi muito bom jogar com você. Te vejo amanhã. — disse Zaion, abraçando o primo.

— Até amanhã, mano! — respondeu Ryo, sorrindo.

Zaion bateu na porta do quarto: — Leo, posso entrar?

— Entra! Estou terminando de arrumar as coisas.

Leo mostrou a mochila nova, já organizada.

— Coube tudo! Ela é incrível.

— E as lições? — perguntou Zaion.

— Fiz todos os programas do jeito que você ensinou. Agora até matemática está mais fácil de entender!

— Então liga o computador. Vamos começar.

---

## 🎬Parte 4

## Aula da Noite

— Hoje nós vamos aprender a dividir um programa em partes pequenas, que podemos reutilizar sempre que quisermos. É assim que nascem nossas próprias bibliotecas de funções, — disse Zaion.

— Vamos começar com um mini-jogo simples: adivinhar um número inteiro.

Zaion explicou as regras com calma:

— O programa vai sortear um número de 1 a 100. Você terá cinco chances. A cada palpite, eu te digo se acertou. Se não, falo se o número sorteado é maior ou menor que o seu.

— No fim, o programa pergunta se quer jogar de novo ou encerrar.
— Primeiro, vamos fazer uma versão 1 do jeito “grandão”, do modo que vínhamos fazendo até agora — tudo dentro da main, passo a passo. Depois, a gente refatora para funções.
Zaion sorriu para o irmão:
— Ok, Leo, — disse, pegando o teclado. — Vamos começar pela versão direta, e depois transformamos em funções para ficar elegante e reaproveitável.

---

## 📝 Passos a seguir

1.Preparar o ambiente

- Incluir bibliotecas necessárias:
  - `<iostream>` para entrada e saída.
  - `<cstdlib>` e `<ctime>` para gerar números aleatórios.
  - `<cctype>` para trabalhar com maiúsculas/minúsculas na resposta final.

2.Inicializar o sorteio

- Usar `std::srand(std::time(NULL))` para gerar números aleatórios diferentes a cada execução.

3.Iniciar o jogo em loop

- Estrutura `while (true)` para permitir jogar várias vezes até o usuário encerrar.

4.Sortear o número secreto

- Gerar número entre 1 e 100:

```
int numero_secreto = 1 + (std::rand() % 100);
```

5.Definir o número de tentativas

- Exemplo: `int tentativas_max = 5;`

6.Receber palpites do jogador

- Laço `for` para até 5 tentativas.
- Ler o valor digitado com `std::cin`.
- Se o palpite não for válido (não for número), limpar entrada e pedir de novo.

7.Comparar palpite com número secreto

- Se for igual → jogador acertou, encerrar rodada.
- Se for menor → mostrar mensagem: "O número sorteado é MAIOR".
- Se for maior → mostrar mensagem: "O número sorteado é MENOR".

8.Informar tentativas restantes

- A cada erro, exibir quantas tentativas ainda faltam.

9.Encerrar rodada

- Se acertou → mostrar parabéns.
- Se errou todas as tentativas → mostrar o número secreto.

10.Perguntar se deseja jogar novamente

- Perguntar: "Deseja jogar novamente? (S/N)".
- Se digitar "S" → reinicia o jogo.
- Se digitar "N" → encerrar programa.

---

# Veja a o programa:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <cctype>

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

int main(int argc, char** argv) {

   std::srand(std::time(NULL));

   std::cout << "=== Jogo de Adivinhacao (1 a 100) ===\n\n";

   while (true) {
      int z_numero_secreto = 1 + (std::rand() % 100);
      int z_tentativas_max = 5;
      bool z_acertou = false;

      std::cout << "Um numero entre 1 e 100 foi sorteado.\n";
         std::cout << "Voce tem " << z_tentativas_max << " tentativas para acertar.\n\n";

      for (int z_tentativa = 1; z_tentativa <= z_tentativas_max; ++z_tentativa) {
         int z_palpite;

         // Leitura simples do palpite
         std::cout << "Tentativa " << z_tentativa << " de " << z_tentativas_max << ". Digite seu z_palpite: ";
         if (!(std::cin >> z_palpite)) {
            // Tratamento basico para entrada invalida
            std::cin.clear();                // limpa estado de erro
            std::cin.ignore(10000, '\n');    // descarta lixo do buffer
            std::cout << "Entrada invalida. Digite um numero inteiro.\n";
            --z_tentativa;                   // nao conta tentativa invalida
            continue;
         }

         if (z_palpite == z_numero_secreto) {
            std::cout << "Acertou! O numero era " << z_numero_secreto << ".\n";
            z_acertou = true;
            break;
         } else if (z_palpite < z_numero_secreto) {
                        std::cout << "Errou. Dica: o numero sorteado eh MAIOR que " << z_palpite << ".\n";
                  } else {
                        std::cout << "Errou. Dica: o numero sorteado eh MENOR que " << z_palpite << ".\n";
                  }

         int z_restantes = z_tentativas_max - z_tentativa;

      }

      if (!z_acertou) {
            std::cout << "\nSuas tentativas acabaram. O numero era " << z_numero_secreto << ".\n";
      }

      // Pergunta se deseja continuar
      char z_opc;
      std::cout << "\nDeseja jogar novamente? (S/N): ";
      std::cin >> z_opc;

      // Normaliza e decide
      z_opc = std::toupper(z_opc);
      if (z_opc != 'S') {
         std::cout << "\nEncerrando. Obrigado por jogar!\n";
         break;
      }

      std::cout << "\n------------------------------\n\n";
   }

   return 0;
}
```

— Leo, esta é a Versão 1 do nosso jogo, com tudo dentro da `main`.

— Ela funciona, mas é uma má prática: fica difícil reutilizar o código, complica a manutenção e atrapalha a leitura.
— Então a gente não consegue aproveitar as partes em outros programas? Perguntou Leo.

— Consegue, mas só copiando e colando, o que é ruim. Melhor é usar a Versão 2.

— O que muda na Versão 2? Questionou Leo.
— A gente separa em funções: sortearNumero(), lerPalpite(), avaliarPalpite() e querContinuar().

— E por que isso é melhor?

— Porque cada função faz uma tarefa clara. Aí fica fácil reutilizar em outros projetos, testar cada parte, trocar um pedaço sem quebrar o resto e deixar tudo mais organizado e legível.

— Entendi: modularizar é tipo montar com blocos de LEGO.

— Exatamente. Agora…

— …vamos ver essa versão por funções!

---

— Vamos ver cada função separadamente. Depois juntar tudo em um programa único (mesmo comportamento da versão monolítica, mas agora modular)

1) Funções, uma a uma

sortearNumero()

O que faz: devolve um inteiro aleatório de 1 a 100.

Por que existe: isola a lógica de sorteio; se amanhã você quiser mudar o intervalo, altera só aqui.

```
#include <cstdlib>
#include <ctime>

int sortearNumero() {
    // pressupõe srand(time(NULL)) feito na main uma única vez
    return 1 + (std::rand() % 100);
}
```

lerPalpite()

O que faz: lê com segurança um palpite do usuário (um inteiro).

Por que existe: centraliza validação e tratamento de entrada inválida.

```
#include <iostream>

int lerPalpite(int z_tentativa_atual, int z_tentativas_max) {
    int z_palpite;
    while (true) {
        std::cout << "Tentativa " << z_tentativa_atual << " de "
        <<z_tentativas_max<< ". Digite seu palpite: ";
        if (std::cin >> z_palpite) {
            return z_palpite; // ok
        }
        // Entrada inválida: limpa e pede novamente sem gastar tentativa
        std::cin.clear();
        std::cin.ignore(10000, '\n');
        std::cout << "Entrada invalida. Digite um numero inteiro.\n";
    }
}
```

avaliarPalpite()

O que faz: recebe o palpite e o número secreto, imprime a dica e devolve true se acertou.

Por que existe: separa a regra de comparação (menor/maior/igual) do resto do fluxo.

```
#include <iostream>

bool avaliarPalpite(int z_palpite, int z_segredo) {
    if (z_palpite == z_segredo) {
        std::cout << "Acertou! O numero era " << z_segredo << ".\n";
        return true;
    } else if (z_palpite < z_segredo) {
        std::cout << "Errou. Dica: o numero sorteado eh MAIOR que " <<
z_palpite << ".\n";
    } else {
        std::cout << "Errou. Dica: o numero sorteado eh MENOR que " <<
z_palpite << ".\n";
    }
    return false;
}
```

querContinuar()

O que faz: pergunta se o jogador deseja jogar novamente e retorna true para “S”, false para qualquer outra resposta.

Por que existe: encapsula a decisão de continuar, facilitando mudar a interface depois (ex.: menus).

```
#include <iostream>
#include <cctype>

bool querContinuar() {
    char z_opc;
    std::cout << "\nDeseja jogar novamente? (S/N): ";
    std::cin >> z_opc;
    z_opc = std::toupper(z_opc);
    return (z_opc == 'S');
}
```

jogarRodada()

O que faz: executa uma partida completa com no máximo 5 tentativas.

Por que existe: organiza o fluxo da rodada, usando as funções anteriores.

```
#include <iostream>

void jogarRodada(int z_tentativas_max = 5) {
    int z_segredo = sortearNumero();
    bool z_acertou = false;
```

```
    std::cout << "Um numero entre 1 e 100 foi sorteado.\n";
    std::cout << "Voce tem " << z_tentativas_max << " tentativas para acertar.
\n\n";

    for (int z_tentativa = 1; z_tentativa <= z_tentativas_max; ++z_tentativa) {
        int z_palpite = lerPalpite(z_tentativa, z_tentativas_max);

        if (avaliarPalpite(z_palpite, z_segredo)) {
            z_acertou = true;
            break;
        }

        int z_restantes = z_tentativas_max - -z_tentativa;

    }

    if (!z_acertou) {
        std::cout << "\nSuas tentativas acabaram. O numero era " <<
z_segredo << ".\n";
    }
}
```

---

# — Agora que analisamos cada função, vamos juntá-las em um único programa:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <cctype>

/* run this program using the console pauser or add your own getch,
system("pause") or input loop */

// ---------- Assinaturas (prototipos) ----------
int  sortearNumero();
int  lerPalpite(int z_tentativa_atual, int z_tentativas_max);
bool avaliarPalpite(int z_palpite, int z_segredo);
bool querContinuar();
void jogarRodada(int z_tentativas_max = 5);

int main(int argc, char** argv) {
    std::srand(std::time(NULL));
     std::cout << "=== Jogo de Adivinhacao (1 a 100) — Versao por Funcoes
===\n\n";
    while (true) {
        jogarRodada(5);
        if (!querContinuar()) {
            std::cout << "\nEncerrando. Obrigado por jogar!\n";
            break;
        }
        std::cout << "\n-----------------------------\n\n";
    }
    return 0;
}


// ---------- Implementacoes ----------
int sortearNumero() {
    return 1 + (std::rand() % 100);
}

int lerPalpite(int z_tentativa_atual, int z_tentativas_max) {
    int z_palpite;
    while (true) {
                std::cout << "Tentativa " << z_tentativa_atual << " de " <<
z_tentativas_max
                << ". Digite seu palpite: ";
        if (std::cin >> z_palpite) {
            return z_palpite;
        }
        std::cin.clear();
        std::cin.ignore(10000, '\n');
        std::cout << "Entrada invalida. Digite um numero inteiro.\n";
    }
}

bool avaliarPalpite(int z_palpite, int z_segredo) {
    if (z_palpite == z_segredo) {
        std::cout << "Acertou! O numero era " << z_segredo << ".\n";
        return true;
    } else if (z_palpite < z_segredo) {
            std::cout << "Errou. Dica: o numero sorteado eh MAIOR que " <<
z_palpite << ".\n";
    } else {
            std::cout << "Errou. Dica: o numero sorteado eh MENOR que " <<
z_palpite << ".\n";
    }
    return false;
}

bool querContinuar() {
    char z_opc;
```

```
    std::cout << "\nDeseja jogar novamente? (S/N): ";
    std::cin >> z_opc;
    z_opc = std::toupper(z_opc);
    return (z_opc == 'S');
}

void jogarRodada(int z_tentativas_max) {
    int z_segredo = sortearNumero();
    bool z_acertou = false;

    std::cout << "Um numero entre 1 e 100 foi sorteado.\n";
    std::cout << "Voce tem " << z_tentativas_max << " tentativas para acertar.
\n\n";

    for (int z_tentativa = 1; z_tentativa <= z_tentativas_max; ++z_tentativa) {
        int z_palpite = lerPalpite(z_tentativa, z_tentativas_max);

        if (avaliarPalpite(z_palpite, z_segredo)) {
            z_acertou = true;
            break;
        }

        int z_restantes = z_tentativas_max - z_tentativa;
        if (z_restantes > 0) {
            std::cout << "Voce ainda tem " << z_restantes << " tentativa(s).\n\n";
        }
    }

    if (!z_acertou) {
            std::cout << "\nSuas tentativas acabaram. O numero era " <<
z_segredo << ".\n";
    }
}
```

---

— Zaion, o programa ficou legal! Mas eu não entendi isso aqui:

int lerPalpite(int z_tentativa_atual, int z_tentativas_max) {

O que são essas coisas dentro dos parênteses? — perguntou Leo, franzindo a testa.

— Boa observação, Leo! — respondeu Zaion, sorrindo. — Esses são os parâmetros da função. Eles funcionam como “caixinhas” que recebem valores quando chamamos a função.

— Então z_tentativa_atual e z_tentativas_max são… variáveis?

— Exatamente. São variáveis especiais que existem só dentro da função. Quando a gente chama lerPalpite(2, 5), por exemplo, o z_tentativa_atual passa a valer 2, e o z_tentativas_max passa a valer 5.

— Ah, então eu consigo mandar informações de fora da função para dentro dela!

— Isso mesmo. É assim que a função sabe em qual tentativa você está e qual é o número máximo de tentativas.

— Entendi! Então se eu mudar os valores quando chamo a função, a lógica lá dentro usa esses novos valores, né?

— Perfeito. Isso deixa o código muito mais flexível e reaproveitável. — concluiu Zaion.

Leo perguntou, meio tímido:

— Só mais uma coisa… a Alin me falou que eu deveria usar variáveis indexadas, porque organizam melhor o programa. O que é isso e como eu uso?

Zaion sorriu, animado com a curiosidade do irmão:

— Ótima pergunta, Leo! Variáveis indexadas são o que chamamos de vetores em C++.

— Vetores? — repetiu Leo.

— Isso mesmo. Pensa em uma caixa de vários compartimentos, todos numerados de 0 até o tamanho que você escolher. Em vez de criar várias variáveis separadas, como nota1, nota2, nota3… a gente cria um vetor chamado nota[ ]. Assim, cada posição do vetor guarda um valor.

Leo arregalou os olhos:

— Então é tipo uma lista organizada de valores?

— Exatamente. Veja o exemplo:

---

```
#include <iostream>
using namespace std;

int main() {
    // Vetor de inteiros com 5 posições
    int numeros[5] = {10, 20, 30, 40, 50};

    cout << "Primeiro elemento: " << numeros[0] << endl; // mostra 10
    cout << "Terceiro elemento: " << numeros[2] << endl; // mostra 30

    // Vetor de caracteres (string simples)
    char vogais[5] = {'a', 'e', 'i', 'o', 'u'};
    cout << "Vogal na posicao 4: " << vogais[3] << endl; // mostra 'o'

    return 0;
}
```

---

— Repara que usamos colchetes [] para acessar a posição. A contagem começa do zero. — explicou Zaion, apontando para a tela.

Leo pensou um pouco:

— Então numeros[0] vale 10, numeros[1] vale 20… e assim por diante. E com char dá para guardar letras também.

— Isso mesmo. Com int guardamos números inteiros, com char guardamos caracteres. E, se precisar, podemos até usar string para palavras inteiras. Os vetores ajudam a organizar dados parecidos e facilitam muito quando temos que trabalhar com repetições ou laços.

Leo sorriu, satisfeito:

— Valeu, Zaion. Agora eu entendi o que a Alin quis dizer com variáveis indexadas!

— Bom, Leo, por hoje é suficiente. Vou te deixar um desafio: criar o jogo da forca.

— Sério?

— Sim. Use vetores para armazenar as palavras. E atenção: em C/C++, quando você usa arranjos de char (C-strings), não pode fazer palavra[0] = "Zaion"; depois que o array já existe. Tem que usar função pra copiar o texto: strcpy(palavra[0], "Zaion");.

— Entendi. E para comparar respostas?

— Também não use if (resp == "sim"). Com C-strings, use strcmp: if (strcmp(resp, "sim") == 0) { ... }.

— E como conto quantos caracteres têm?

— strlen. Ex.: qtd_digitada = strlen(palavra[0]);.

— Tá, vou me esforçar e prometo fazer do meu jeito, sem copiar códigos prontos nem usar IA!

— Aí sim. Eu confio em você.

---

🧰 Guia rápido: strcpy, strcmp, strlen (C-strings)
Inclua o cabeçalho #include <cstring> para usar essas funções em C++.
1) strcpy(dest, src)
O que faz: copia a string src (fonte) para dest (destino), incluindo o caractere nulo '\0'.
Uso típico: carregar palavras dentro de um vetor de char.

```
#include <cstring>

char palavra[10][32];          // até 10 palavras, cada uma com no máx. 31 chars + '\0'
std::strcpy(palavra[0], "Brasil");
std::strcpy(palavra[1], "Androide");
```

⚠️ Cuidado com tamanho! Garanta que o destino tem espaço suficiente. Se precisar, prefira strncpy(dest, src, n) para limitar a cópia.
2) strcmp(a, b)
O que faz: compara duas C-strings.
Retorno:
0 → iguais
< 0 → a vem antes de b
> 0 → a vem depois de b

```
char resp[8];
/* ... leia resp com std::cin ... */
if (std::strcmp(resp, "sim") == 0) {
   // respostas iguais (exatamente "sim")
}
```

💡 Se quiser ignorar maiúsculas/minúsculas, você pode:

Transformar resp para minúsculas antes de comparar, ou
Em ambientes POSIX (por exemplo: Unix e Linux), usar strcasecmp (não padrão C++).
3) strlen(s)
O que faz: retorna o comprimento da C-string (quantos caracteres antes do '\0').
size_t tam = std::strlen(palavra[0]); // "Zaion" -> 6
🧪 Exemplo mínimo (base para a forca)

```
#include <iostream>
#include <cstring>   // strcpy, strcmp, strlen
using namespace std;

int main() {
    // Banco de palavras (C-strings)
    char palavras[5][32];
    std::strcpy(palavras[0], "zaion");
    std::strcpy(palavras[1], "androides");
    std::strcpy(palavras[2], "laboratorio");
    std::strcpy(palavras[3], "minhoca");
    std::strcpy(palavras[4], "zeta");

    // Exemplo de leitura e comparação de resposta
    char resp[8];
    cout << "Quer jogar? (sim/nao): ";
    cin >> resp;

    if (std::strcmp(resp, "sim") == 0) {
        cout << "Otimo! Tamanho da primeira palavra: "
             << std::strlen(palavras[0]) << "\n";
        // daqui em diante: sua lógica da forca ;)
    } else {
        cout << "Tudo bem. Ate a proxima!\n";
    }
    return 0;
}
```

✅ Dicas pro teu jogo da forca
Armazenamento: char banco[QTD][TAM_MAX]; para palavras.
Palavra secreta: copie para um buffer de trabalho: char segredo[TAM_MAX]; strcpy(segredo, banco[idx]);
Estado da descoberta: vetor de char com _ e letras acertadas.
Entrada de letra: char chute; cin >> chute; (normalize para minúscula se quiser).
Verificação: percorra segredo e revele quando segredo[i] == chute.
Fim de jogo: ganhou quando não há mais _; perdeu quando acabar as tentativas.
Repetição: pergunte com strcmp(resp, "sim") == 0.

---

— Tenho certeza de que você vai dar o seu melhor. Boa noite, Leo!

— Boa noite, mano. Muito obrigado por me ajudar. Tenho confiança de que vou fazer uma boa prova — disse Leo, abraçando o irmão.

(A casa mergulhou em silêncio. Pouco depois…)

## 🎬 Parte 5

## Entre Sonho e Realidade

Zaion tomou um banho, escovou os dentes, pôs o pijama e se deitou.

— Zahy, você está aí? — pensou Zaion.

— Sim, por quê?

— Você esteve calado o dia todo… Não é o seu normal, kkkk.

— É… Desculpa, Zaion. Mas ainda não consegui entender o que está acontecendo, e isso é muito frustrante. Estive pesquisando e, quanto mais

procuro, menos compreendo. Parece que estamos em universos distintos e, ainda assim, nossas mentes permanecem unidas. Mesmo quando vou para outro universo e retorno, a impressão é de que nunca saí do seu. Trago comigo todos os detalhes do que aconteceu durante o dia.

— E como é esse outro universo? — perguntou Zaion.

— Muda a cada vez. Na última visita, era uma sala sem janelas ou portas, apenas com monitores e computadores… muitos, muitos mesmo. Eu era como um andróide, um computador operando outros computadores. Qual o propósito disso, não sei. Parece que tudo ali depende da minha mente para existir. Às vezes, sinto que não importa quem eu sou — importa apenas que eu exista. Mesmo que eu fique com você o tempo todo, aquele lugar continua a "viver" sem a minha presença consciente.

— Olha, isso é confuso! Mas não me importo. Apesar de sermos, de certa forma, um só, gosto de sentir que você é outro. Alguém em quem posso confiar, meu apoio. Se preferir, podemos esquecer tudo isso. Você não precisa mais partilhar sua outra vida comigo. Eu compreenderei.

— Zaion, às vezes você demonstra uma maturidade surpreendente… Nunca conheci ninguém assim.

— Boa noite, amigo. Espero que você consiga se resolver.

— Boa noite, Zaion.

(Zaion adormeceu, e Zahroniel foi levado de volta à sua outra vida, despertando lentamente em sua sala, no universo físico.)

— Ativar sistema! — exclamou.

Tudo se acendeu; os monitores iluminaram o local, que parecia a cabine de uma espaçonave. Em uma das telas, surgiu a imagem de Kael'Aran.

— Oi, Kael! O que deseja?

— Mas que desânimo! — brincou ela.

— Descobri algumas coisas. Os servidores dos andróides e seus sistemas de comunicação são semelhantes. O curioso é que o Beta Cinco e o Zeta, presencialmente, são muito diferentes. O Beta Cinco se comporta como um andróide comum. Já o Zeta… parece muito mais humano. Minha mãe vai agendar uma avaliação física e psicológica com ambos, para entendermos essa diferença.

— Ah, mas eu já sei por quê. O Zeta foi programado para trabalhar com humanos, tornando-se o mais parecido possível com eles. Ele se espelha muito no Zaion. Mas me diga: você já notou que está incorporando traços da sua personagem Kael, a que vive no universo do Zaion?

— Impressão sua. Mas, como já te disse, descobri uma rede de microburacos de minhoca que servem como sistema de comunicação e que atingem o meu, o seu e o universo do Zaion. O mais importante: esses buracos são artificiais e exigem enorme quantidade de energia para existir. De onde vem essa energia?

— Isso eu deixo para você descobrir.

— Zahroniel, vai me abandonar de novo?

— Olha, Kael… este lugar funciona sozinho. Pelo que percebi, só precisa que eu exista. Parece que minha mente é o que sustenta tudo. Não me pergunte mais, porque eu mesmo não sei explicar.

— Isso não tem lógica nenhuma!

— A vida é totalmente ilógica. Talvez um dia eu compreenda o que aconteceu conosco. Você pode continuar suas pesquisas, mas eu vou voltar para o que mais se parece com vida para mim: estar junto ao Zaion. Adeus.

— Zahroniel, espera! … Ah… Pensando bem, acho que também estou me deixando levar por aquele mundo. Minha ligação com a Kael, lá no universo do Zaion, está cada vez mais forte… Eu me sinto ela. E, quando estou com o Zaion,

os sentimentos parecem ainda mais intensos. Puxa… que situação complicada!

(Em sua sala iluminada por monitores, Kael’Aran toca o próprio peito e pensa…)

— Se meus sentimentos são reais no mundo de Zaion… então onde termina o sonho?