



# Ponteiros em C – Parte 2

Prof. Sérgio Carlos Portari Júnior

2010

# Da história pro mundo real em C

- Em nossa história criamos um mundo fictício, baseado no mundo real que ainda representa toda a teoria de Ponteiros em C.
- Para facilitar todo o entendimento, vamos agora introduzir no contexto a comparação da história com a linguagem C junto com o mundo real.

# Da história pro mundo real em C

- Definição: Ponteiros em C são um tipo especial de variáveis (que contém um endereço de memória próprio como qualquer outra) que não guardam um conteúdo particular (como um número em inteiro) e mas sim um endereço de memória de outra variável por exemplo.
- Os ints guardam inteiros. Os floats guardam números de ponto flutuante. Os chars guardam caracteres. Ponteiros guardam endereços de memória.

# Da história pro mundo real em C

- *“Como nossas residências, o endereço de cada casa era único, isto é, não existia dois endereços iguais em todo o país.*
- *Sendo assim, cada casa pode ser identificada através de seu endereço completo sem que seja confundida com outra.”*
- No exemplo da história, mostra-se que cada variável possui um endereço próprio, de forma a não ser possível confundirmos umas com as outras mesmo que tenham a mesma rua ou nome de rua na vida real.

# Da história pro mundo real em C

- *No entanto, os habitantes de qualquer dos locais referenciam, sempre que possível, as casas uns dos outros pelo nome da pessoa que lá mora, tal qual fazemos, por exemplo:*
  - *Bom dia, você foi à casa do João hoje?”*
- Em C isso também ocorre. Podemos acessar uma variável pelo seu nome ou pelo seu endereço de memória. O ponteiro é um tipo de variável que possibilita acessar uma variável pelo endereço e não pelo seu nome.

# Da história pro mundo real em C

- Resumindo: quando usamos um ponteiro em C, criamos uma variável normal (que possui nome, que possui um espaço de memória reservada para ela) cujo conteúdo armazenado é o endereço de memória, geralmente de outra variável no programa ou de uma região de memória conhecida.
- Através desse endereço poderemos manipular seu conteúdo (como veremos mais adiante).

# Da história pro mundo real em C

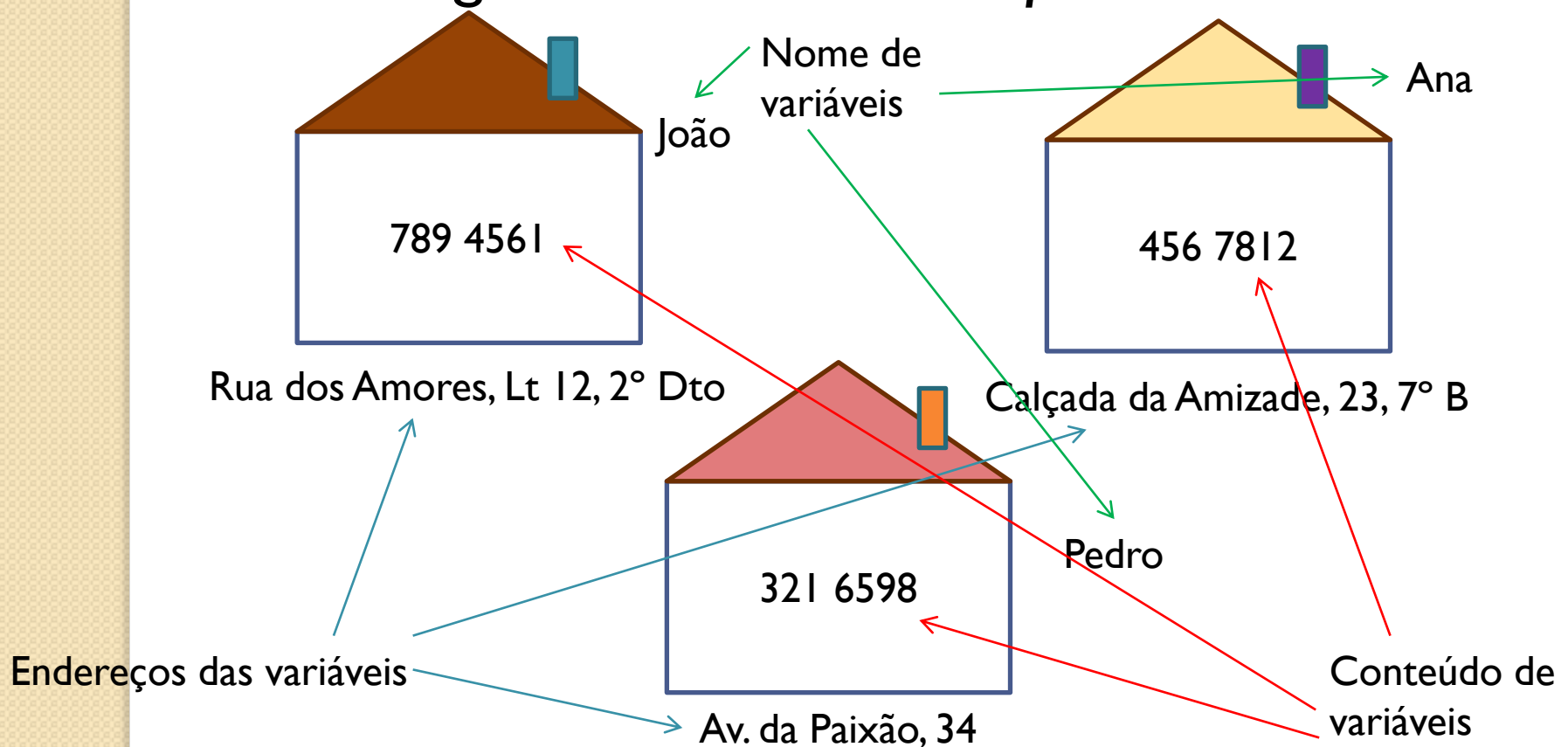
- *Suponhamos que existam 3 amigos que trocavam telefonemas entre si (João, Ana e Pedro).*

Nome	Telefone	Endereço
João	789 4561	R. dos Amores, Lote 12, 2º Dto.
Ana	456 7812	Calçada da Amizade, 23, 7º B
Pedro	321 6598	Av. da Paixão, 34

Aqui não temos nenhum ponteiro ainda

# Da história pro mundo real em C

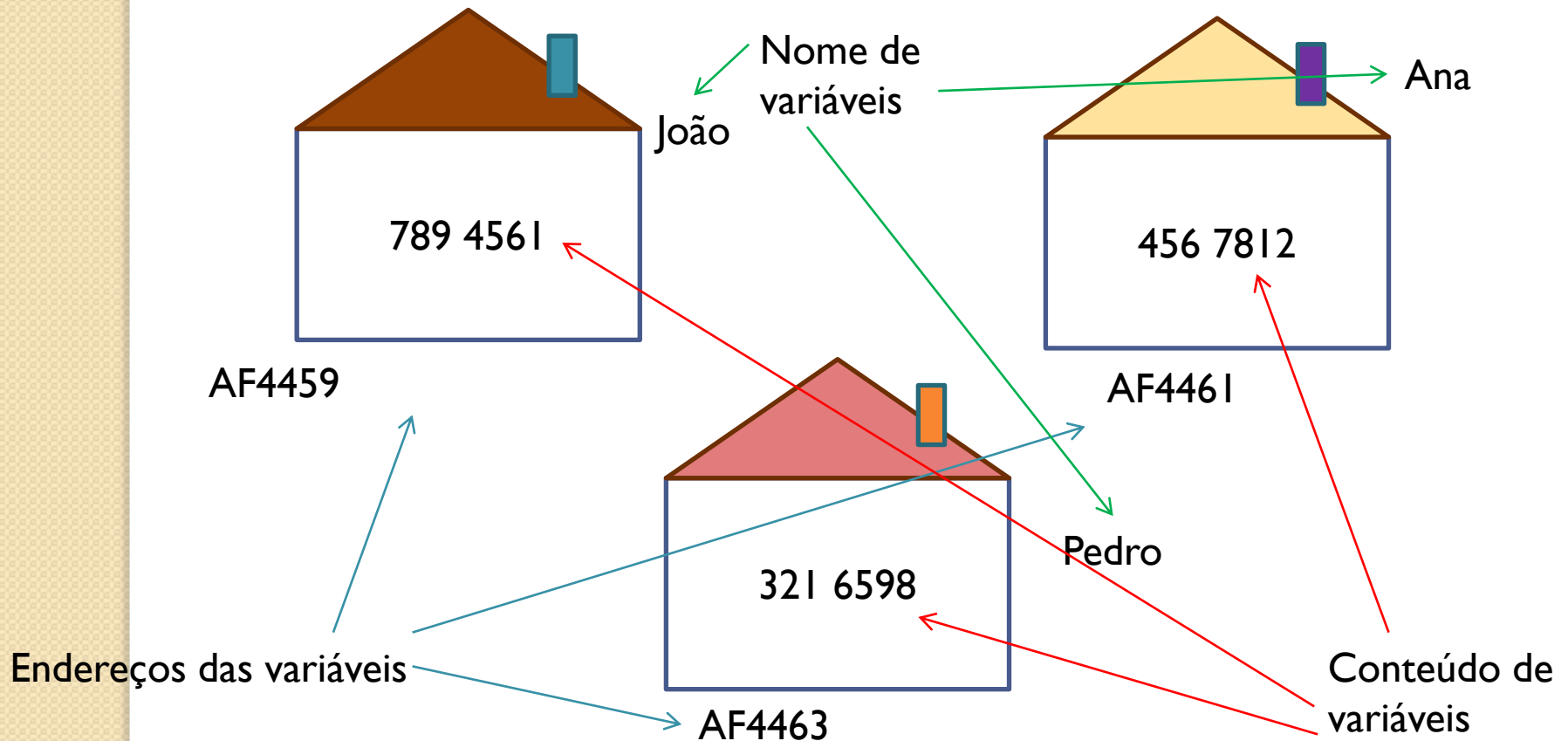
- Vamos mostrar graficamente o conjunto dos 3 amigos montados até aqui:*





# Da história pro mundo real em C

- Vamos converter para C:



# Da história pro mundo real em C

- Apesar de trabalharmos apenas com os nomes das pessoas (João, Ana e Pedro), existe ainda uma outra informação que não estamos utilizando (os endereços) onde estão situadas as respectivas casas que correspondem à localização destas pessoas.
- Por exemplo, o endereço de João é Rua dos Amores, Lt 12, 2º Dto. O de Ana é Calçada da Amizade, 23, 7º B, etc.
- Isso acontece também em C. Quando fazemos `int pedro=10`, chamamos apenas o nome pedro quando queremos utilizar o valor 10 e até esquecemos que pedro tem um endereço de memória reservado.

# Da história pro mundo real em C

- *Ora, em C existe um operador (&) que permite saber o endereço de uma variável (assim como já usamos em scanf).*
- *Dessa forma, se escrevêssemos em C novamente os printf's, teríamos:*
  - *printf("%s",&João); // => sairia Rua dos Amores, Lt 12, 2º Dto.*
- *De fato utilizamos sempre esse artifício. Num scanf("%d",&a); passamos o endereço de memória de a para a função scanf guardar o que o usuário digitou.*

# Da história pro mundo real em C

- *Em resumo, o nome de cada indivíduo é uma forma bem mais simples de representar qualquer posição no país dos telefones.*
- *$X \Rightarrow$  representa aquilo que está dentro da casa chamada  $X$ ;*
- *$\&X \Rightarrow$  representa o local onde a casa  $X$  se situa.*
  - *`printf("%d",João); // => 789 4561`*
  - *`printf("%s",&João); // => Rua dos Amores, Lt 12, 2º Dto.`*
- **Com uma variável inteira, por exemplo, acontece exatamente isso.**

# Da história pro mundo real em C

- *“Como já dissemos, nem todas as pessoas no país dos telefones podem ter telefone em casa.*
- *É o caso de SAMAD, ...”*
- Na história SAMAD será o primeiro ponteiro a ser utilizado.
- Por que ele não pode ter telefone? Telefone é o conteúdo de uma variável. Ponteiros “não tem conteúdos concretos”. Tem apenas endereços de memórias de outras variáveis.

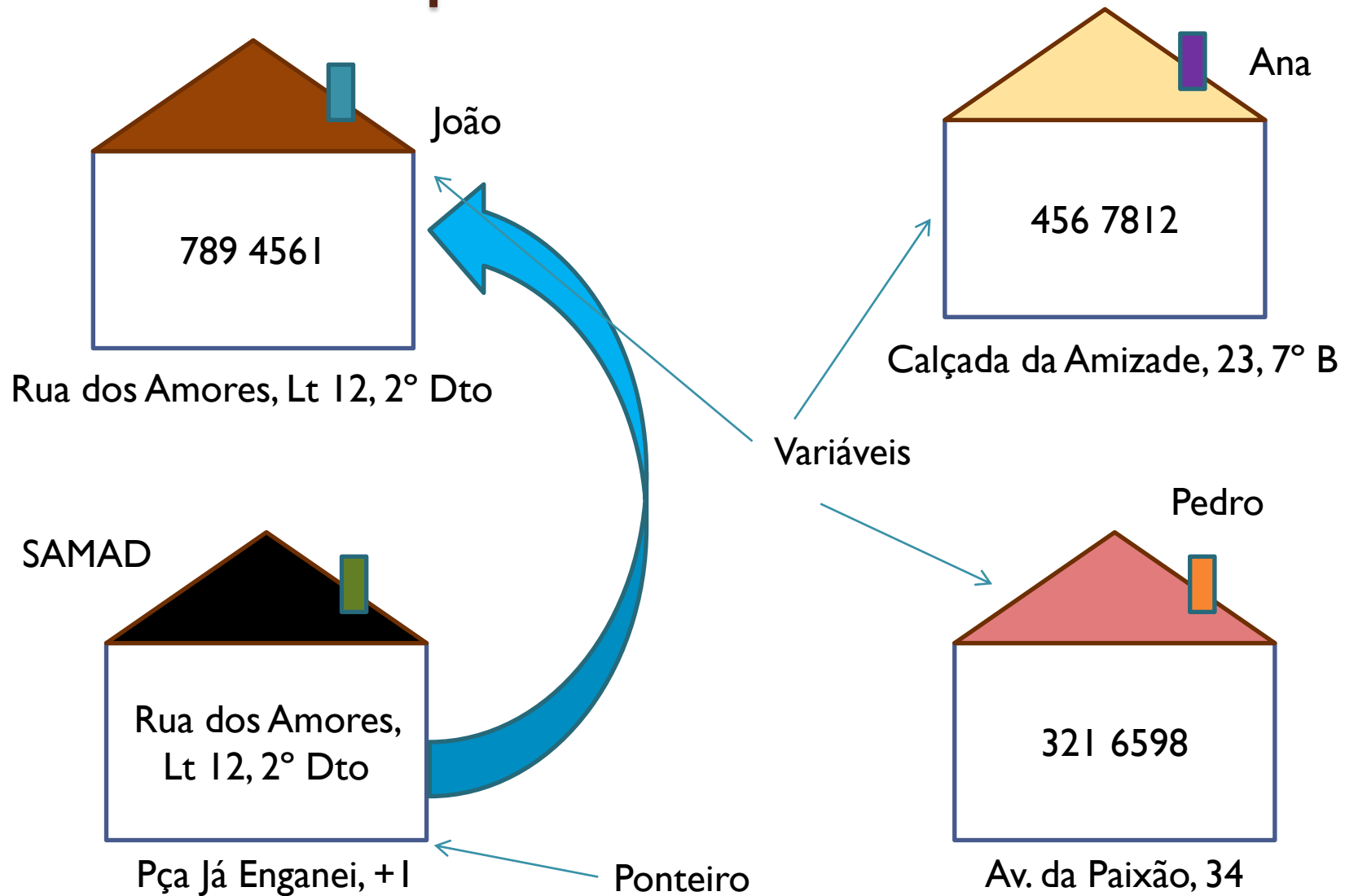
# Da história pro mundo real em C

- *Por falta de provas foi-lhe permitido viver em liberdade em sua casa, mas no entanto ele não poderia ter um telefone de modo a dificultar as suas atividades contra a sociedade, embora seu endereço seja bem conhecido por todos: Pça Já Enganei, +1.*
- SAMAD possui, como uma variável qualquer, um endereço para armazenar o seu conteúdo (endereço de outra variável) e possui um nome (SAMAD) para podermos acessar esse conteúdo.

# Da história pro mundo real em C

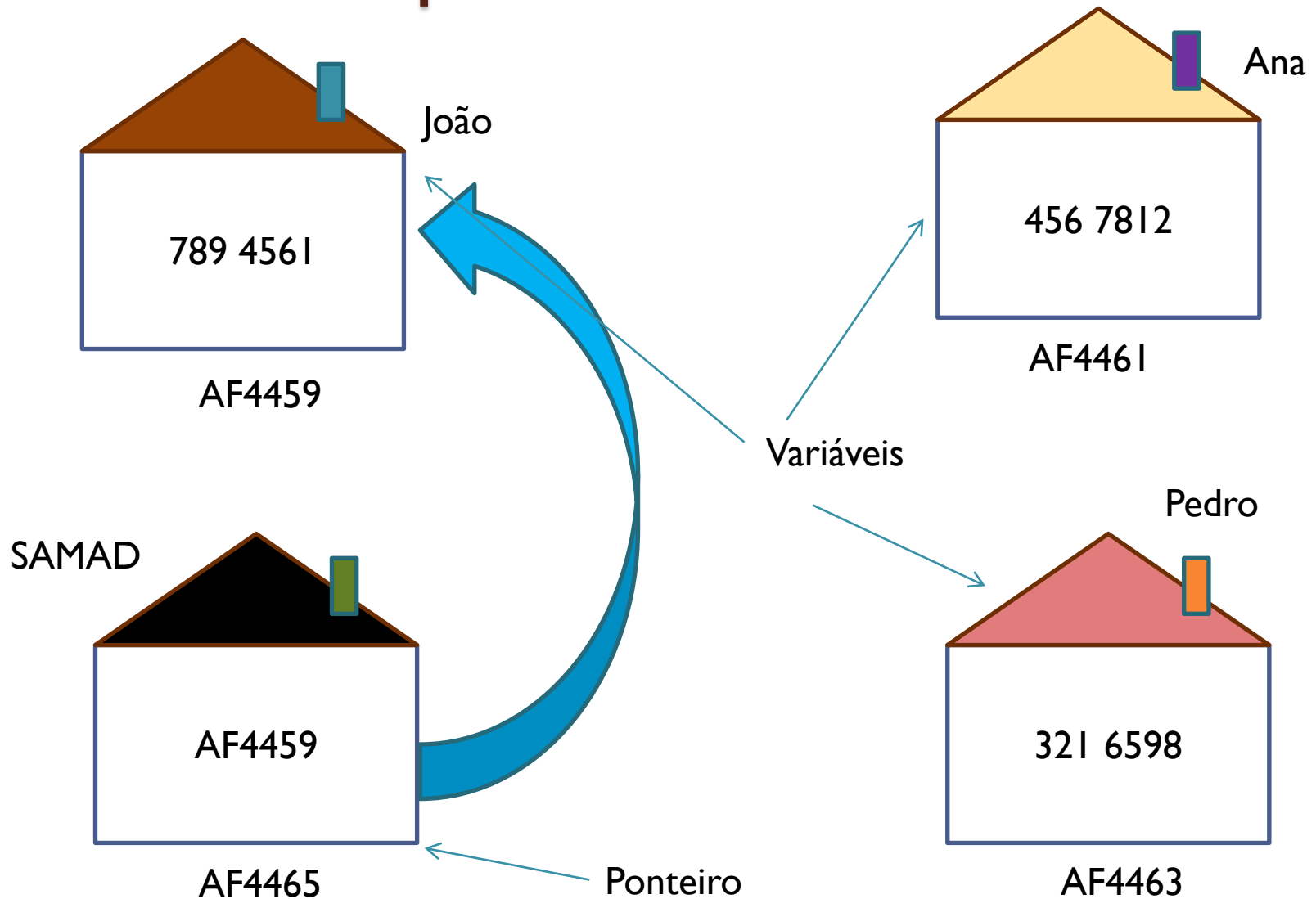
- *“Então, em sua casa, não existirá um número telefônico, mas terá um local onde poderá realizar a referida chamada.*
- *E se falamos em local, falamos em endereço.”*
- Nesta situação, como dito antes, SAMAD não possui conteúdo. Possui apenas um endereço de memória para acessar outra variável.
- Casa é a região da memória que está reservada para a variável.
- Número telefônico é o conteúdo desta variável.

# Da história pro mundo real em C





# Da história pro mundo real em C



# Da história pro mundo real em C

- *Como podemos ver, a casa de SAMAD é um pouco diferente das outras. Se as outras podiam ser delcaradas como*
  - *Telefone João, Ana, Pedro;*
- *SAMAD terá que ser declarado como uma casa que tem o endereço de outra casa, isto é, que aponta para uma casa com telefone. Para diferenciar, faremos assim:*
  - *Telefone \*SAMAD;*
  - *Não tem telefone, tem o endereço de onde tem um telefone*
- **Aqui está claro a diferença da declaração de uma variável para um ponteiro. Em C:**
  - **int a; (para variável) / int \*a; (para ponteiro)**

# Da história pro mundo real em C

- Se *SAMAD* contém o endereço de João é porque foi iniciado do seguinte modo:
  - *SAMAD = &João; //Lembrem do anúncio?*
- Em C nunca poderemos utilizar um ponteiro sem antes “apontá-lo” para um endereço de memória conhecido.
- Ex:  

```
int a; int *b; //declara a variável a e ponteiro b  
b = &a; // coloca em b o endereço de a.
```

# Da história pro mundo real em C

- Veja a tabela como ficaria:

Expressão	Valor
João	789 456 I
&João	R. dos Amores, Lote 12, 2º Dto.
SAMAD	R. dos Amores, Lote 12, 2º Dto.
&SAMAD	Pça Já Enganei, + I

Expressão	Valor
a	789456 I
&a	AF4459
b	AF4459
&b	AF4465

# Da história pro mundo real em C

- *Note que, falar de SAMAD é o mesmo que falar de &João, pois SAMAD contém o valor do endereço de João*
- *O endereço de SAMAD é o local onde fica sua casa.*
- Em C, vale o mesmo conceito, a variável `b` é o mesmo que o endereço de `a` (`b = &a`).

Expressão	Valor	Descrição
<code>a</code>	789 4561	Valor de <code>a</code>
<code>&amp;a</code>	AF4459	Endereço de <code>a</code>
<code>b</code>	AF4459	Valor de <code>b</code>
<code>&amp;b</code>	AF4465	Endereço de <code>b</code>
<code>*b</code>	789 4561	Valor <b>apontado</b> por <code>b</code>

# Da história pro mundo real em C

- *Depois da última ida ao hospital, João proibiu que SAMAD fosse a sua casa telefonar por ter dado falta de duas garrafas de ar.*
- *Apesar de contrariada, Ana deixou o famoso bandido utilizar a sua casa (depois de colocar todo seu patrimônio no seguro).*
- Em C um mesmo ponteiro pode mudar o endereço para onde ele aponta no decorrer o programa, ou seja, ele não é constante.
- *Dados de Ana:*

**ANA**

**456 7812**

**Calçada da Amizade, 23, 7º B**

# Da história pro mundo real em C

- *Pergunta:*
  - *Qual o conjunto de passos que precisamos executar?*
- *Resposta*
  - *SAMAD = &Ana;*
- *Essa pequena mudança faz com que:*

Expressão	Valor	Descrição
SAMAD	Calçada da Amizade, 23, 7º B	Valor de SAMAD
&SAMAD	Pça Já Enganei, + I	Endereço de SAMAD
*SAMAD	456 8712	Valor <b>apontado</b> por SAMAD

- Supondo uma variável chamada c (de mesmo tipo de a e b), se mudarmos como no exemplo teremos:

# Da história pro mundo real em C

- `int a=3, *b, c=5;`
- `b=&a;`

Expressão	Valor	Descrição
<code>b</code>	AF4459	Valor de <code>b</code>
<code>&amp;b</code>	AF4467	Endereço de <code>b</code>
<code>*b</code>	3	Valor <b>apontado</b> por <code>b</code>

- `b=&c;`

Expressão	Valor	Descrição
<code>b</code>	AF4465	Valor de <code>b</code>
<code>&amp;b</code>	AF4467	Endereço de <code>b</code>
<code>*b</code>	5	Valor <b>apontado</b> por <code>b</code>