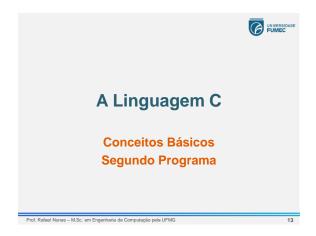


Programa 01 – alomundo.c Abram a ferramenta e após criar um novo projeto digitem o seguinte programa abaixo: 01 /* Meu Primeiro Programa */ 02 #include <stdio.h> 03 int main (void) 04 { 05 //Imprime a seguinte mensagem na tela 06 printf ("Ola! Eu nasci!\n"); 07 08 return 0; 09 10 }//Fim</stdio.h>	
01 /* Meu Primeiro Programa */ 02 #include <stdio.h> 03 int main (void) 04 { 05 //Imprime a seguinte mensagem na tela 06 printf ("Ola! Eu nasci!\n"); 07 08 return 0; 09 10)//Fim</stdio.h>	
Tipos e Modificadores	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 5	
UNIVESIDADE	
Tipos da Linguagem C	
□ O C possui 5 tipos básicos:	
□ char	
□ int	
□ float	
□ void □ double ————————————————————————————————————	
- double	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 6	

	UNIVERSIDADE	
Modifica	idores de Tipo	
☐ Para os	s tipos de variáveis também podemos	
	os modificadores de tipo.	
☐ Os mod	dificadores de tipo do C são quatro:	
□ signe	d	
unsig	ned,	
long		
■ short.		
Prof. Rafael Nunes – M.Sc.	em Engenharia de Computação pela UFMG 7	
	_	
	FUMEC STATE	
	Onevedence	
	Operadores	-
Prof. Rafael Nunes – M.Sc.	em Engenharia de Computação pela UFMG 8	
Operado	pres Aritméticos	
Os opera	adores aritméticos são usados para desenvolver es matemáticas.	
	apresentamos a lista dos operadores aritméticos do C:	
Operador	Ação Soma (inteira e ponto flutuante)	
+	Subtração ou Troca de sinal (inteira e ponto flutuante)	
*	Multiplicação (inteira e ponto flutuante)	
1	Divisão (inteira e ponto flutuante)	
%	Resto de divisão (de inteiros)	
++		
	Incremento (inteiro e ponto flutuante)	
	Incremento (inteiro e ponto flutuante) Decremento (inteiro e ponto flutuante)	

UNIVERSIDADE FUMEC	
Operadores Relacionais	
☐ Os operadores relacionais do C <i>realizam comparações</i>	
entre variáveis.	
☐ São eles:	
== (<i>igual</i>),	
!= (diferente de),	
> (maior que),	
< (menor que),	
>= (maior ou igual),	
<= (menor ou igual).	
Os operadores relacionais retornam verdadeiro (1) ou falso (0)	
laiso (o)	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 10	
UNIVERSIDADE	
Operadores Lógicos	
☐ Para fazer operações com valores lógicos	
(verdadeiro e falso) temos os operadores lógicos:	
(voludadilo o laloo) tollioo oo opolaacioo logicoo.	
0.0 /	
&& (and (e)),	
II (or (ou)),	
! (not (não)).	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 11	
Objetivos de Aprendizagem	
□ Controle de <i>Fluxo</i>	
☐ Estrutura de decisão	
☐ O comando switch	
□ Estrutura de repetição	
□ Comandos break e continue	
□ Funções <u> </u>	
UNIVERSIDADI	
FUMEC	



Abra a ferramenta *Eclipse*e após criar um *novo projeto*digite o programa a seguir!

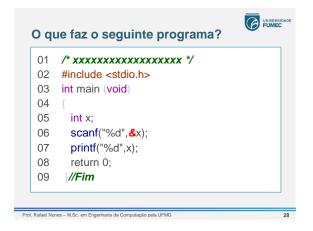
UNIVERSID FUMEC Programa 02 - convDiaAno.c /* Meu Segundo Programa */ #include <stdio.h> 3 #include <stdlib.h> 3 int main (void) 4 5 int dias; 6 float anos; 7 setbuf(stdout, NULL); /*ATENCAO!!!*/ 8 printf ("Digite o número de dias: "); 9 fflush(stdin); scanf ("%d",&dias); 10 11 anos=dias/365.25; 12 printf ("\n%d dias equivalem a %f anos.\n",dias,anos); 13 return 0; 14 }//Fim

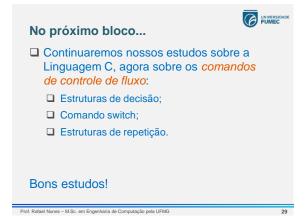
UNIVERSIDADE	
Analisando meu Segundo Programa em C	
Segundo i Tograma em C	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 16	
Analisando o Segundo Programa	
☐ São declaradas duas <u>variáveis</u> chamadas dias e anos.	
 □ A primeira é um <i>int</i> (<u>inteiro</u>) e a segunda um float (<u>ponto flutuante</u>). □ int = apenas valores inteiros □ float = apenas valores do conjunto real (ponto 	
flutuante)	
☐ É feita então uma chamada à função printf() , que coloca uma mensagem na tela.	
Prof. Rafael Nunes – M.Sc., em Engenharia de Computação pela UFMG 17	
Analisando o Segundo Programa	
Queremos agora ler um dado que será fornecido pelo usuário e colocá-lo na variável dias.	
☐ Devemos utilizar a função scanf()	
<pre>08 printf ("Digite o número de dias: "); 09 fflush (stdin); 10 scanf ("%d",&dias);</pre>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 18	

UNIVERSIDACE	
A função scanf()	
Linha 10 do programa "convDiaAno.c"	
Out Data Name Mig. or Francis of Commission NEW	
Prof. Rafael Nunes – M.Sc. em Engerharia de Computação pela UFMG 19	
A Função scanf()	
☐ Leitura de dados da entrada padrão ☐A tag "%d" diz à função que iremos ler um inteiro.	
O segundo parâmetro passado à função diz que o <u>dado capturado</u> deverá ser armazenado na variável dias.	
08 printf ("Digite o número de dias: "); 09 fflush (stdin);	
10 scanf ("%d",&dias);	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 20	
UNIVERSIDADE	
A função scanf()	
& → Atenção	
Prof. Rafael Nunes – M.Sc. em Engerharia de Computação pela UFMG 21	

UNIVERSIDADE FUNEC	
A Função scanf()	
☐ É importante ressaltar a necessidade de se colocar um & antes do nome da variável a ser	
<u>lida</u> quando se usa a função scanf().	
☐ O motivo disto só ficará claro mais tarde.	
☐ Observe que, no C, quando temos <i>mais de um</i>	
parâmetro para uma função, eles serão	
separados por vírgula.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 22	
UNIVERSIDADE FUMEC	
FUMEC	
Continuando a Analise do	
Segundo Programa	
Coganao i rograma m	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 23	_
_	
Analisando o Segundo Programa	
☐ Temos então uma expressão matemática	
simples que atribui a anos o valor de dias	
dividido por 365.25.	
anos=dias/365.25;	
□ Como anos é uma variável float o	
compilador fará uma conversão	
compilador fará uma conversão	
compilador fará uma conversão	

Analisando o Segundo Programa	
☐ A segunda chamada à função printf() tem três argumentos:	
printf ("\n%d dias equivalem a %f anos.\n", dias, anos);	
 A string "\n%d dias equivalem a %f anos.\n" diz à função para saltar para a próxima linha 	
 colocar um <i>inteiro</i> na tela; colocar a <u>mensagem</u> " dias equivalem a "; 	
colocar um valor <i>float</i> na tela;	
colocar a mensagem " anos.";	
realizar mais um retorno de carro.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 25	
Analisando o Segundo Programa	
☐ Os parâmetros restantes são as variáveis	
das quais deverão ser lidos os valores do	
inteiro e do float, respectivamente.	
printf (" $\n^{m}d$ dias equivalem a %f anos. \n^{m} , dias, anos);	
	-
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 26	
UNIVERSIDADE PUMEC	
Exercício de <i>Fixação</i>	
	-
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 27	



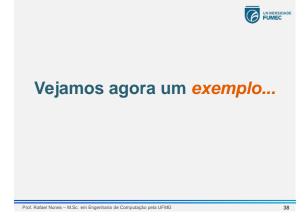




UNIVERSIDADE	
	<u></u>
A Linguagem C	
Controle de Fluxo	
	-
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 31	
UNIVERSIDADE	
Estruturas de <i>Decisão</i>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 32	
UNIVERSIDACE	
O comon do "f	
O comando <i>if</i>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 33	

O comando if	UNIVERSIDADE FUMEC			
☐ O comando if representa uma tom decisão do tipo "SE isto ENTÃO a	nada de aquilo".	_		
☐ A sua forma geral é: if (<i>condição</i>) declaração;		-		
		-		
Prof. Rafael Nunes – M.Sc. em Engenharía de Computação pela UFMO	34	- -		
O comando if	UNIVERSIDADE FUMEC			
☐ A condição do comando if é uma expressão que será avaliada .		_		
□Se o resultado for zero a declaraçã será executada .	o <u>não</u>	-		
□Se o resultado for <i>qualquer coisa</i> diferente de zero a declaração sero executada.	á	-		
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	35	_		
	UNIVERSIDADE FUMEC	_		
		-		
Vamos entender		-		
		_		
		_		
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	36	_		

O comando if	SIDADE C
teste = 1;	
1. if (teste == 0)	
2. printf("ALO!") // <u>NÃO</u> IRÁ EXECUTAR	
//	
1. if (teste != 0)	
2. printf("ALO!") //IRÁ EXECUTAR	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	37



```
#include <stdio.h>
int main ()

{

setbuf(stdout,NULL);
int num;
printf ("Digite um numero: ");
scanf ("%d",&num);
if (num>10)
printf ("\n\nO numero e maior que 10");
if (num==10)
{

printf ("\n\nVoce acertou!\n");
printf ("O numero e igual a 10.");
}
if (num<10)
printf ("\n\nO numero e menor que 10");
return 0;
}

Prot. Rafael Nunes — M.Sc. em Engenharia de Computação pela UFMG 39
```

	UNIVERSIDADE	
O comando if		
□ No programa acima a expressão num> avaliada e retorna um valor diferente de :		
se verdadeira, e zero, se falsa.	,	
☐ Repare que quando queremos testar		
igualdades usamos o operador = e não		
□Isto é porque o operador = representa a uma <i>atribuição</i> .	penas	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	40	
	UNIVERSIDADE FUMEC	
O comando <i>if-else</i>		
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	41	
	_	
O comando if-else	UNIVERSIDADE FUMEC	
☐ Podemos pensar no comando else	como	
sendo um complemento do comar		
□O comando if completo tem a seguir forma geral:	nte	
if (condição) declaração_1;		
<i>else</i> declaração_2;		
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	42	

	UNIVERSIDADE
O comando if-else	Pomeo
A expressão da condição será avaliada	l
☐ Se ela for <i>diferente de zero</i> (verdadeir	·0)
a declaração 1 será executada.	0)
☐ if (<i>condição</i> !=0) executa <i>declaração_1</i>	
☐ Se for zero (falso) a declaração 2 será	à
executada, ou seja, o bloco do <u>else</u>	
☐ if (condição==0) executa declaração_2	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMS	43
	UNIVERSIDADE FUMEC
O comando if-else	
□ É importante nunca esquecer que, qua	ndo
usamos a estrutura <i>if-else</i> , estamos garantindo que uma das duas declaraç	ões
sempre será executada	
 Nunca serão executadas as duas ou nenhuma delas 	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	44
	UNIVERSIDADE FUMEC
Vejamos um exemplo	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	45

O comando if-else	SIDADE C
#include <stdio.h></stdio.h>	
int main ()	
{	
int num;	
printf ("Digite um numero: ");	
scanf ("%d",#);	
if (num==10)	
(10111-10)	
printf ("\n\nVoce acertou!\n"); Bloco Verdadeiro	
printf ("O numero e igual a 10.\n");	
printi (O numero e iguar a 10. vi),	
else	
erse	
{	
printf ("\n\nVoce errou!\n"); Bloco falso	
printf ("O numero e diferente de 10.\n"); = else	
}	
return 0;	
}	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	46



O comando if-else-if A estrutura if-else-if é apenas uma extensão da estrutura if-else. Sua forma geral pode ser escrita como sendo: if (condição_1) declaração_1; else if (condição_2) declaração_2; else if (condição_3) declaração_3; ... else if (condição_n) declaração_n; else declaração_default; //Opcional

O comando if-else-if	
□ <u>Atenção!</u>	
A última declaração (default) é a que será	
executada no caso de todas as condições falharem, ou seja, serem falsas (zero)	
Ela <u>é opcional !</u>	
Prof. Rafael Nunes – M.Sc., em Engerharia de Computação pela UFMG 49	
UNIVERSIDADE	
FUMEC	
Vejamos um exemplo	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 50	- <u> </u>
_	
O comando if-else-if #include <stdio.h></stdio.h>	
int main () {	
int num; printf ("Digite um numero: "); scanf ("%d",#);	
<pre>if (num>10) printf ("\n\nO numero e maior que 10");</pre>	
else if (num==10)	

printf ("\n\n\oce acertou!\n"); printf ("O numero e igual a 10.");

else if (num<10)
printf ("\n\nO numero e menor que 10");
return(0);

Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG

UNIVERSIDADE	
Vamos refletir um pouco sobre	
a expressão condicional	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 52	
UNIVERSIDADE FUNCE.	
A expressão condicional ☐ Quando o compilador avalia uma condição, ele espera um valor de retorno para poder	
tomar sua decisão.	
Mas esta expressão não precisa ser uma expressão no sentido convencional.	
Uma variável sozinha pode avaliada como "expressão"	
☐ Ela apenas retorna seu próprio valor.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 53	
UNIVERSIDADE FUMEC	
A expressão condicional ☐ Isto quer dizer que teremos as seguintes expressões: int num;	
if (num=0) if (num==0)	
for (i = 0; string[i] != '\0'; i++) == equivalem a int num;	
if (num) if (!num) for (i = 0; string[i]; i++)	
☐ e que podemos simplificar essas expressões.	
Deck Defeat Morary M.Co. on Expendencia de Computancia anto UEMC	

if's aninhados	UNIVERSIDADE FUMEC
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	55

if's aninhados O if aninhado é simplesmente um comando if declarado dentro de um outro comando if. Devemos ter cuidado na organização dos diversos comandos e saber exatamente qual else pertence a qual if

```
if's aninhados

if (num==10)

{
    printf ("\n\n\ocorrected acertou!\n");
    printf ("O numero e igual a 10.\n");
}
else
{
    if (num>10)
    {
        printf ("O numero e maior que 10.");
    }
    else
    {
        printf ("O numero e menor que 10.");
    }
}
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMS

57
```

UNIVERSIDADE FUNEC	
O comando switch	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 58	
WINDSCIDAGE PUMES	
O comando switch ☐ O comando if-else e o comando switch são os dois comandos de tomada de decisão.	
Sem dúvida alguma o mais importante dos dois é o if, mas o comando switch tem aplicações valiosas.	
O comando switch é próprio para se testar uma variável em relação a diversos valores pré-estabelecidos.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 59	
UNIVERSIDADE	
O comando switch ☐ Forma geral:	
switch(<expressão>){ case <<i>const1</i>>: < <i>instruções1</i>></expressão>	
case < <i>const2</i> >: < <i>instruções2></i> case < <i>constN</i> >: < <i>instruçõesN</i> >	
default: <instruções></instruções>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMS 60	<u> </u>

ONLY CONTROL OF THE C	RSIDADE EC
O comando switch - Exemplo	
switch(dia){	
case 0: printf("Domingo\n"); break;	
case 1: printf("Segunda\n"); break;	
case 2: printf("Terça\n"); break; case 3: printf("Quarta\n"); break;	
case 4: printf("Quinta\n"); break;	
case 5: printf("Sexta\n"); break;	
case 6: printf("Sabado\n"); break;	
default: printf("Opção Inválida\n"); break;	
}	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	61
Fiol. Raidel Nulles - M.Sc. elli Lingellialia de Collipulação pela Ol MS	01
_	
WIND AND AND AND AND AND AND AND AND AND A	RSIOADE EC
O comando switch	
☐ Podemos fazer uma analogia entre o switch e a	A
estrutura if-else-if apresentada anteriormente:	
aceita expressões	
□ Somente aceita <i>constantes</i> .	
O switch testa a variável e executa a	
declaração cujo case corresponda ao valor atual da variável.	
atuai da variavei.	
☐ A declaração default é opcional	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	62
_	
O comando switch	RSIDADE EC
□ O comando break , faz com que o switch	
seja interrompido assim que uma das declarações seja executada.	
☐ Mas ele <u>não</u> é essencial ao comando switch.	
☐ Se após a execução da declaração não	
houver um break, o programa continuará	
executando normalmente. (<i>Cuidado!</i>)	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	63



```
#include <stdio.h>
int main ()

{
    int num;
    printf ("Digite um numero: ");
    scanf ("%d",&num);
    switch (num) {
        case 9:
            printf ("\n\nO numero eh igual a 9.\n");
            break;
        case 10:
            printf ("\n\nO numero eh igual a 10.\n");
            break;
        case 11:
            printf ("\n\nO numero eh igual a 11.\n");
            break;
        default:
            printf ("\n\nO numero não é nem 9 nem 10 nem 11.\n");
        }
    return 0;
}
```



UNIVERSIDADE FUMEC	
O comando for	
	-
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMS 67	
O comando for	
□ O loop (laço) <i>for</i> é utilizado para <i>repetir um comando</i> , ou bloco de comandos, <i>diversas vezes</i> ;	
☐ Sua forma geral é: for (inicialização;condição;incremento) declaração;	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 68	
O comando for	
☐ A declaração no comando for também pode ser um bloco ({ }) e neste caso o ; é omitido.	
☐ O melhor modo de se entender o loop for é observando a <i>maneira como ele funciona</i> "internamente".	
	-
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMS 69	

UNIVERSIDADE FUMEC	
O comando for	
☐ O <i>loop for</i> é equivalente a seguinte estrutura:	
ininializaçõe	
inicialização; if (condição)	
{	
declaração;	
incremento;	
"Volte para o comando <u>if</u> "	
}	
	-
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 70	
UNIVERSIDADE FUMEC	
O comando for	
☐ Podemos ver então que o <i>for</i> executa a	
inicialização incondicionalmente e testa a	
condição.	
□Se a <i>condição</i> for falsa ele não faz mais nada. Sai do loop.	
□Se a <i>condição</i> for verdadeira ele executa a	
declaração (apenas uma vez), o <i>incremento</i> e	-
volta a testar a condição novamente.	
☐ Ele repete estas operações até que a	
condição se torne falsa .	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 71	
UNIVERSIDADE FUMEC	
Vejamos um exemplo	
vejanios uni exemplo	
	-
Ded Oder Marry M.Co. or Countries of Co.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 72	

O comando for	
#include <stdio.h></stdio.h>	
int main () {	
int count;	
<pre>for (count=1;count<=100;count=count+1) printf ("%d ",count); return 0;</pre>	
}	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMS 73	
UNIVERSIDADE	
O comando <i>for</i>	
Loop infinito	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 74	
UNIVERSIDADE	
O comando for – Loop Infinito O loop infinito tem a forma	
for (inicialização;;incremento) declaração;	
 Este loop chama-se loop infinito porque será executado para sempre, a não ser que ele seja 	
interrompido □ Para interromper um loop como este usamos o comando	
break. ☐ O comando break vai quebrar o loop infinito e o programa continuará sua execução normalmente.	
отнината заа влеоцкао поппаннене.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 75	

O comando for – Loop Infinito	4DE
 O programa que faz a leitura de uma tecla e sua impressão na tela, até que o usuário aperte uma tecla sinalizadora de final (um FLAG). O nosso FLAG será a letra 'X'. 	
<pre>#include <stdiio.h> #include <stdiib.h> int main () { setbuf(stdout, NULL); int Count; char ch; for (Count=1; ;Count++) { ch = getchar(); if (ch == 'X') break, printf("\nLetra: %c",ch); } return 0;</stdiib.h></stdiio.h></pre>	
}	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	76



O comando while



☐ O comando while tem a seguinte forma geral:

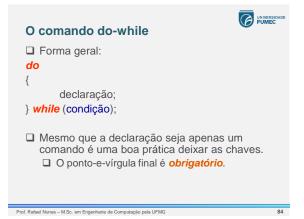
while (condição) declaração;

Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMC

_	
O comando while	
 Assim como fizemos para o comando for, vamos tentar mostrar como o comando 	
while funciona fazendo uma analogia.	
D. Falia a subtle and a substante	
☐ Então o while seria equivalente a:	
if (condição)	
{	
declaração;	
"Volte para o comando if"	
,	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 79	
- UNIVERSITATION OF	
O comando while	
☐ Podemos ver que o comando while testa uma	
condição.	
 Se esta condição for verdadeira a declaração é executada e faz-se o teste novamente, e 	
assim por diante.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 80	
UNIVERSIDADE FUMEC	
O comando while	
☐ Assim como no caso do for, podemos fazer um	
loop infinito. □ Para tanto basta colocar uma expressão eternamente	
verdadeira na condição.	
D. Dada sa também amitin a dadamañ a tana	
Pode-se também omitir a declaração e fazer um loop sem conteúdo	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 81	

O comando while	UNIVERSIDADE FUMEC
O programa abaixo espera o usuário digitar a tecla 'q' e só d finaliza:	lepois
<pre>#include <stdio.h> int main () { setbuf(stdout, NULL); char Ch; Ch=\0'; while (Ch!='q') { Ch = getchar(); } return 0; }</stdio.h></pre>	
rof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	82





O comando do-while



- Vemos pela análise do bloco anterior que a estrutura do-while executa a declaração, testa a condição e, se esta for verdadeira, volta para a declaração.
- A grande novidade no comando do-while é que ele, ao contrário do for e do while, garante que a declaração será executada pelo menos uma vez.

Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG

85

```
int main () {
                                                                                               UNIVERSIDA FUMEC
      int i;
      do {
           printf ("\n\nEscolha a fruta pelo numero:\n\n"):
           printi ( "Ntlesconia a iruta
printf ("\t(1)...Mamão\n");
printf ("\t(2)...Abacaxi\n");
printf ("\t(3)...Laranja\n\n");
scanf("%d", &i);
      } while ((i<1)||(i>3));
      switch (i) {
            case 1:
                        printf ("\t\tVoce escolheu Mamão.\n");
                       break;
            case 2:
                       printf ("\t\tVoce escolheu Abacaxi.\n");
                       break;
            case 3:
                        printf ("\t\tVoce escolheu Laranja.\n");
                       break:
            return 0;
```

Exercício de Fixação



- Escreva um programa que peça dois inteiros, correspondentes a horas e minutos (Ex: 23:59).
- Peça os números até conseguir valores que estejam na faixa correta (horas entre 0 e 23 e minutos entre 0 e 59).
- 3. Imprima a hora digitada pelo usuário no formato "xx:xx"
- 4. Comente seu programa

rof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFM

No próximo bloco	
☐ Continuaremos nossos estudos sobre a Linguagem C, agora sobre:	
☐ Comando break e continue	
☐ Funções	
Bons estudos!	
DONS estudos:	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 88	
UNIVERSIDADE FUMEC	
Algoritmos e Estrutura de Dados	
Videoaula 01 (Parte 3/3)	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	
5	
UNIVERSIDADE	
O comando <i>break</i>	
O Comando break	

O compando breste	UNIVERSIDADE
O comando break Nós já vimos dois usos para o	comando brask
☐ interrompendo os comandos switch	
■ Na verdade, estes são os dois	usos do comando
break:	
□ele pode quebrar a execução de ur caso do switch) ou	n comando (como no
☐interromper a execução de qualque caso do for, do while ou do do while	er loop (como no
	•
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	91
	UNIVERSIDADE
O comando break	Formed
 O break faz com que a execuç continue na primeira linha segu 	
bloco que está sendo interrom	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	92
	UNIVERSIDADE FUMEC
0.0000001	-4:
O comando col	ntinue
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	93

O comando continue	SIDADE
 O comando continue pode ser visto como sendo o oposto do break. Ele só funciona dentro de um loop. 	
 Quando o comando continue é encontrado, o loop pula para a próxima iteração, sem o abandono do loop 	
Ao contrário do que acontecia no comando break.	
Prof. Rafael Nunes – M.Sc., em Engenharia de Computação pela UFMG	94

```
int main() {
    setbuf(stdout, NULL);
    int opcao;
    while (opcao!=5) {
        printf("\n\n Escolha uma opcao entre 1 e 5: ");
        fflush(stdin);
        scanf("%d", &opcao);
        /* Opcao invalida: volta ao inicio do loop */
        if ((opcao > 5)||(opcao < 1)) continue;
        switch (opcao) {
            case 1: printf("\n --> Primeira opcao.."); break;
            case 2: printf("\n --> Segunda opcao.."); break;
            case 3: printf("\n --> Terceira opcao.."); break;
            case 4: printf("\n --> Quarta opcao.."); break;
            case 5: printf("\n --> Abandonando.."); break;
        }
    }
    return 0;
}
```



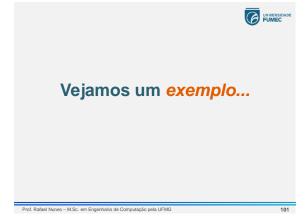
A Linguagem C

Funções

Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMC

UNIVERSIDADE FUMEC	
O que é uma <i>função?</i>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 97	-
Universidade Fumec	
O que é uma função? ☐ Uma função é um bloco de código de	
programá que <u>pode ser usado diversas vezes</u> em sua execução.	
O uso de funções permite que o programa fique <i>mais legível</i> , mais bem <u>estruturado</u> .	-
Um programa em C consiste, no fundo, de várias funções colocadas juntas.	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 98	
UNIVERSIDADE FUNEC	
Estrutura de uma <i>função</i>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 99	

Estrutura Geral de uma Função	ERSIDADE MEC
tipo_de_retorno nome_da_função (lista_de_argumento { código_da_função }	os)
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	100



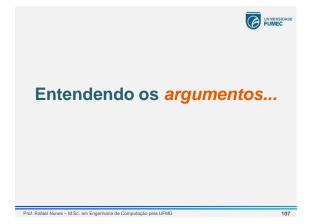
```
Exemplo de Função

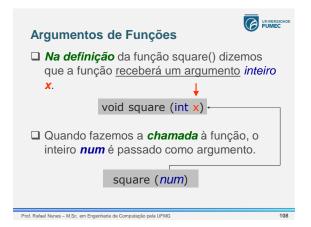
#include <stdio.h>
/* Funcao simples: Só imprime Ola! */
void mensagem ()
{
    printf ("Ola! ");
}
/* Funcao principal */
int main ()
{
    mensagem();
    printf ("Tudo Bem?\n");
    return 0;
}

Prol. Rafael Nunes — M.Sc. em Engerharia de Computação pela UFMG 102
```

UNIVERSIDADE FUMEC	
Funções	
3000	
Argumentos	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 103	
Argumentos de Funções	
Argumentos são as entradas que a função recebe.	
☐ É através dos argumentos que passamos parâmetros para a função.	
☐ Ex: As funções printf() e scanf() são	
funções que <i>recebem argumentos</i> .	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMS 104	
UNIVERSIDADE	
Majamaa uus avananta	
Vejamos um exemplo	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 105	

Argumentos de Funções #include <stdio.h> #include <stdlib.h></stdlib.h></stdio.h>	UNIVERSIDADE FUMEC
/* Calcula o quadrado de x */ void square (int x) { printf ("O quadrado e %d",(x*x)); }	
<pre>int main () { setbuf(stdout, NULL); int num; printf ("Entre com um numero: "); fflush(stdin); scanf ("%d",#); printf ("\n\n"); square(num); return 0; }</pre>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	106





Argumentos de Funções	WERSIDADE IMEC
□ Em primeiro lugar temos de satisfazer aos requisitos da função quanto ao tipo e à quantidade de argumentos quando a chamamos.	
//void square (int x) /*Funcao Square*/ // int num; square(num);	
Apesar de existirem algumas conversões de tipo, que o C faz automaticamente, é importar ficar atento.	nte
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	109



UNIVERSIDADE	
E quando precisamos passar mais do que um argumento para a função?	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 112	
Múltiplos Argumentos	
Repare que, neste caso, os argumentos são separados por vírgula e que deve-se explicitar o tipo de cada um dos argumentos, um a um.	
void mult (<i>float</i> a, <i>float</i> b, <i>float</i> c) Note também que os argumentos passados para a função <i>não necessitam</i> ser todos <i>variáveis</i> porque mesmo sendo <u>constantes</u> serão copiados para a variável de entrada da função	
mult (x,y, 3.87);	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 113	
UNIVERSIDADE	
Retornando Valores	
Para que eu preciso retornar valores após realizar uma função?	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG 114	

Retornando Valores	ersidade AEC
 Muitas vezes é necessário fazer com que uma função retorne um valor. As funções que vimos até aqui não retornam nada pois especificamos um retorno void. 	,
 Podemos especificar um tipo de retorno indicando-o antes do nome da função. Mas para dizer ao C o que vamos retornar precisam da palavra reservada return. 	ios
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	115



Sabendo disso fica fácil fazer uma função para multiplicar dois inteiros e que retorne o resultado da multiplicação

Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG

116

Retornando Valores #include <stdio.h> int prod (int x,int y) { return (x*y); } int main () { int saida; saida = prod(12,7); printf ("A saida e: %d\n",saida); return 0; }

O UN	VERSIDADE
Retornando Valores	MEC
☐ Veja que como <i>prod</i> retorna o valor de 12 multiplicado por 7, este valor pode ser usado em uma expressão qualquer.	
□ No programa fizemos a atribuição deste resultado à variável saida, que posteriormente)
foi impressa usando o printf().	
<pre>saida = prod (12,7); printf ("A saida e: %d\n",saida);</pre>	
Prot. Rafael Nunes – M.Sc. em Engenharia de Computação pela UPMG	118
	VERSIDADE
Retornando Valores	MEC
Uma observação adicional:	
Se não especificarmos o tipo de retorno de uma função, o compilador C automaticamente suporá que este tipo é inteiro.	
☐ Porém, <u>não</u> é uma boa prática <i>deixar de</i> especificar o val de retorno	or
□ Nesta disciplina, este valor será sempre deverá ser especificado .	
<pre>int prod (int x, int y)</pre>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	119
Retornando Valores – Outro exemplo	VERSIDADE IMEC
□ A função agora recebe dois floats e també retorna um float.	m
□ Repare que no exemplo a seguir especificamos um valor de retorno para a função main (int) e retornamos zero.	
☐ Normalmente é isto que fazemos com a	
função main:	
□retorna zero quando é executada <u>sem</u> qualquer tipo de erro)
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	120

<pre>Retornando Valores #include <stdio.h> float prod (float x, float y) { return (x*y); }</stdio.h></pre>	UNIVERSIDADE PUMEC
<pre>int main () { float saida; saida=prod (45.2,0.0067); printf ("A saida e: %f\n",saida); return 0; }</pre>	
Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMG	121

Exercício de Fixação



- 1. Implemente uma mini calculadora com as quatros funções básicas:
 - Soma
 - Subtração
 - Divisão
 - Multiplicação (Você irá encontrar nos exercícios apresentados!)
- Obs.: Você deverá escolher os tipos de dados (int ou float) para as operações solicitadas

Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFMC

122

Para finalizar...



Faça os exercícios propostos e não deixe de acessar o ambiente virtual

Bons estudos e até a nossa próxima videoaula!

Prof. Rafael Nunes – M.Sc. em Engenharia de Computação pela UFM

Referências	
Waldemar Celes, Renato Cerqueira, José Lucas Rangel, Introdução a Estruturas de Dados, Editora Campus. (2004)	