Universidade Federal de Santa Maria

Departamento de Eletrônica e Computação

ELC1048 - PROJETO DE SISTEMAS EMBARCADOS

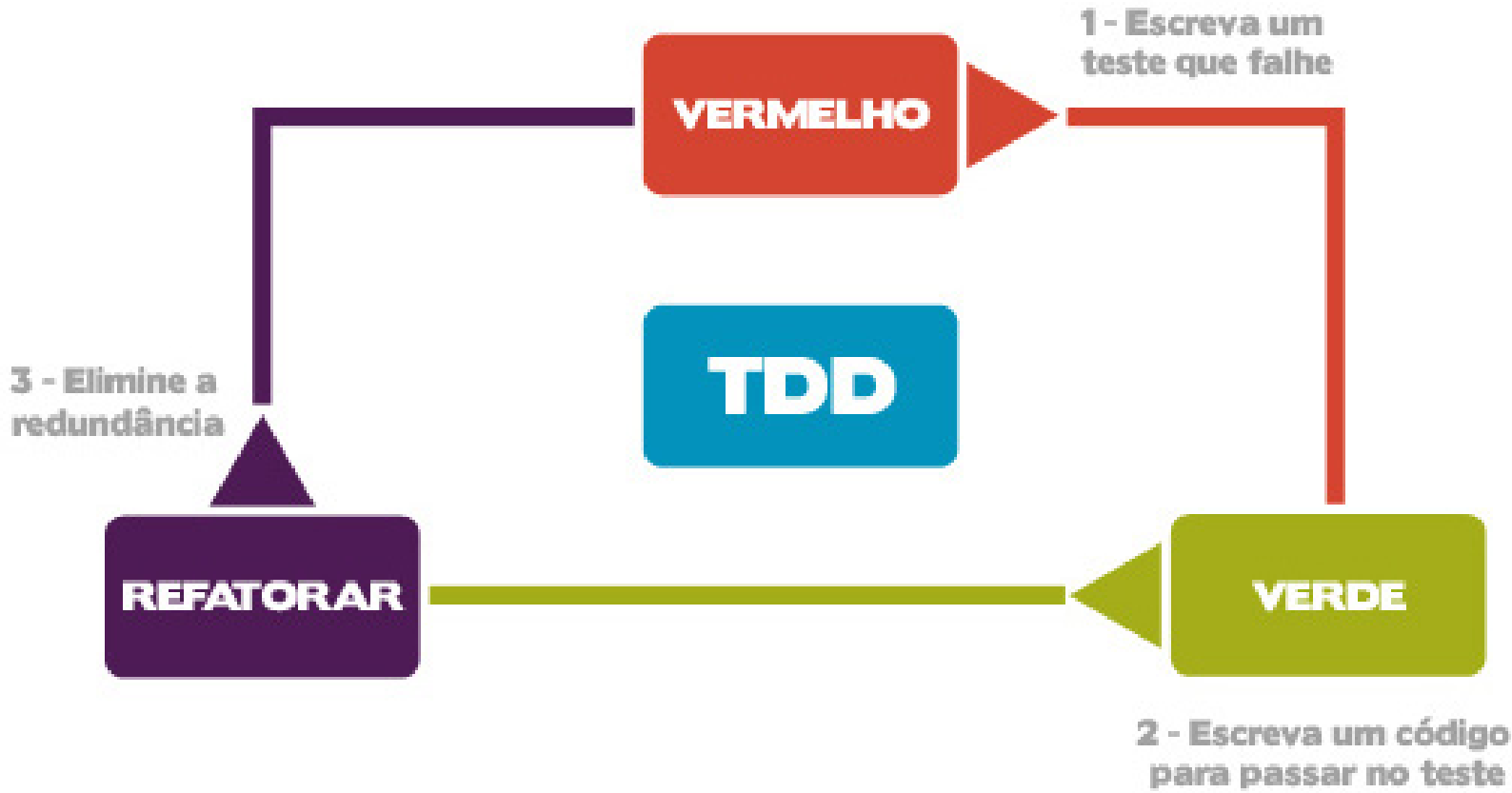
Prof. Carlos Henrique Barriquello
barriquello@gmail.com

Aula – Primeiro projeto TDD

Objetivo: Resolver um problema de programação pela metodologia TDD.

Entrega: Upload do projeto (.zip) no Moodle. E relatório dos testes incrementais.

Ciclo TDD



Fonte: http://arquivo.devmedia.com.br/artigos/Fabio_Gomes_Rocha/TDD/TDD_1.jpg

Exercício 1

Problema: encontrar e retornar a posição do primeiro bit igual a 1, em uma variável de 32 bits sem sinal, a partir do bit mais significativo.

Ex. 00110100111 => posição 6

Exercício 2

Problema: Escrever uma funçõa que converte um número decimal de 0 a 9999 em um número no formato BCD (*binary-coded-decimal*).

Exemplo

Problema: fazer um conversor de caracteres hexadecimais em números decimais.

Exemplo

```c
#include <stdio.h>
#include <stdlib.h>

/* macros de testes */
#define verifica(mensagem, teste) do { if (!(teste)) return mensagem; } while (0)
#define executa_teste(teste) do { char *mensagem = teste(); testes_executados++; \
                                  if (mensagem) return mensagem; } while (0)
int testes_executados = 0;

static char * executa_testes(void);

int charhex2dec(char x);

int main()
{
    char *resultado = executa_testes();
    if (resultado != 0)
    {
        printf("%s\n", resultado);
    }
    else
    {
        printf("TODOS OS TESTES PASSARAM\n");
    }
    printf("Testes executados: %d\n", testes_executados);

    getch();
    return resultado != 0;
}
```

Exemplo

```
31  int charhex2dec(char x)
32  {
33      //return x;  //teste 0 falhou
34      //if(x=='0') return 0; //teste 1 falhou
35      //if(x=='0') return 0;
36      //if(x=='1') return 1; //teste 1 passou
37      // refatorar
38      // return x-'0';  // passou teste 2, mas falhou no teste 3
39
40
41      if(x>='0' && x<='9') return x-'0';
42      if(x>='a' && x<='f') return x-'a' + 10;
43      if(x>='A' && x<='F') return x-'A' + 10;
44      return -1;
45  }
46
47  static char * teste_retorna0_para_char0(void)
48  {
49      verifica("erro: charhex2dec('0') deveria retornar 0", charhex2dec('0') == 0);
50      return 0;
51  }
52
53  static char * teste_retorna1_para_char1(void)
54  {
55      verifica("erro: charhex2dec('1') deveria retornar 1", charhex2dec('1') == 1);
56      return 0;
57  }
58
59  static char * teste_retorna2_para_char2(void)
60  {
61      verifica("erro: charhex2dec('2') deveria retornar 2", charhex2dec('2') == 2);
62      return 0;
63  }
```

Exemplo

```
68  static char * teste_retorna10_para_chara(void)
69  {
70      verifica("erro: charhex2dec('a') deveria retornar 10", charhex2dec('a') == 10);
71      return 0;
72  }
73
74  static char * teste_retorna15_para_charf(void)
75  {
76      verifica("erro: charhex2dec('f') deveria retornar 15", charhex2dec('f') == 15);
77      return 0;
78  }
79
80  static char * teste_retorna_neg1_para_charg(void)
81  {
82      verifica("erro: charhex2dec('g') deveria retornar -1", charhex2dec('g') == -1);
83      return 0;
84  }
85
86  static char * teste_retorna15_para_charF(void)
87  {
88      verifica("erro: charhex2dec('F') deveria retornar 15", charhex2dec('F') == 15);
89      return 0;
90  }
91
92  static char * executa_testes(void)
93  {
94      executa_teste(teste_retorna0_para_char0);
95      executa_teste(teste_retorna1_para_char1);
96      executa_teste(teste_retorna2_para_char2);
97      executa_teste(teste_retorna10_para_chara);
98      executa_teste(teste_retorna15_para_charf);
99      executa_teste(teste_retorna_neg1_para_charg);
100     executa_teste(teste_retorna15_para_charF);
101     return 0;
102 }
```