

## **Disciplina: Processo de Desenvolvimento de Software**

### **Aula 6: A Implementação no Processo de Desenvolvimento de Software**

#### **Apresentação:**

Nesta aula, iremos definir o conceito de implementação para o processo de desenvolvimento de software.

A fase de implementação, ou codificação, tem como objetivo escrever o programa em uma linguagem de programação, seguindo normas e diretrizes da empresa à qual o desenvolvedor esteja ligado.

Na fase da implementação, o analista ou desenvolvedor detalha e implementa o que foi definido na etapa de desenho, através de componentes de código de programa e documentação detalhada.

---

#### **Objetivos:**

- Conhecer as atividades de implementação no processo de desenvolvimento de software;
- Entender as necessidades de definir uma tecnologia para a transformação do desenho para o projeto em um sistema binário;
- Analisar os diversos tipos de produto e utilizar a linguagem que atenda às necessidades;

# Definições

Nós vimos anteriormente, que **desenho**, é uma das etapas do processo de desenvolvimento de software, lembra?

A **implementação**, é o processo que realiza a transformação do desenho em diversos tipos de componentes de código de programação.



O código de programação pode ser dividido em 3 tipos:

```

CV_HOUGH_GRADIENT, 2, gray->width,100,200); if(circles->total > 0)
{
    float* p = (float*)cvGetSeqElem( circles, 1 );
    uchar* ptr = cvPtr2D(img, cvRound(p[1]), cvRound(p[0]), NULL);

    double region_size = 7;
    double red_avg = 0;
    double green_avg = 0;
    double blue_avg = 0;

    for(int y=-floor(region_size/2); y<ceil(region_size/2); y++)
    {
        uchar* ptr1 = (uchar*) (ptr + y * img->widthStep);
        for( int x=-floor(region_size/2); x<ceil(region_size/2); x++)
        {
            blue_avg += ptr1[3*x];
            green_avg += ptr1[3*x+1];
            red_avg += ptr1[3*x+2];
        }
    }
    red_avg = red_avg/(region_size*region_size);
    green_avg = green_avg/(region_size*region_size);
    blue_avg = blue_avg/(region_size*region_size);

    bool color = (green_avg-150)*(green_avg-150)<900 && (blue_avg-100)*
    if(color)
    {
        cvCircle( rgbimg, cvPoint(cvRound(p[0]),cvRound(p[1])),
            3, CV_RGB(0,255,0), -1, 8, 0 );
        cvCircle( rgbimg, cvPoint(cvRound(p[0]),cvRound(p[1])),
            cvRound(p[2]), CV_RGB(255,0,0), 3, 8, 0 );

        if(d = get_actual_depth(cvGet2D(depthimg, cvRound(p[1]), cvRound
        {
            templandmark->detected = true;
            X = 320.5 - cvRound(p[0]);
            mu = (240.5 - cvRound(p[1]))*d/FOCAL_LENGTH;
            w = X*d/FOCAL_LENGTH;

```

## Código Fonte

Conjunto de instruções geradas através de uma linguagem de programação, de maneira lógica e estruturada; após o processo de compilação ou interpretação, transformar-se-á em código objeto.

```

MOV id2 R1
MULT 2 R1
MOV id1 R2
ADD R1 R2
MOV R2 id1

```

## Código Objeto

Resultado da compilação do código fonte.

```
10100110011100000110101
00011100010101000111100
11000011111101010101011
0110100110101010101110
11000110101000011000111
10101010100001110011010
11010011101011001110110
10011010101101011101010
10101010001001100011110
11010101001110010110001
```

## **Código Máquina**

Sequência binária de ações diretamente direcionadas para o processador da máquina.

Vejamos mais algumas definições:

## **Compilador**

Programa que faz uma leitura do código fonte, desenvolvido em uma linguagem de alto nível, e transcreve para um novo tipo de linguagem chamada de baixo nível.

## **Interpretador**

Programa que, além de fazer a leitura do código fonte e transformá-lo em código objeto, efetua a execução do mesmo sequencialmente.

## Linguagem de baixo nível


Linguagem de programação que utiliza a arquitetura do processador para executar as ações. Esta linguagem é a que mais se aproxima dos códigos de execução direta do processador, ou seja, linguagem de máquina.

## Linguagem de alto nível

Comumente chamada de linguagem de programação, esta linguagem se aproxima mais da linguagem humana, ou seja, linguagem com um padrão de entendimento humano bem definido. Para essa linguagem não é levado em consideração a arquitetura do computador, nem as características do processador e seus registradores, visto que, na fase de interpretação ou compilação, esses programas transformarão em linguagem de baixo nível ou de máquina.

# Classificações das linguagens



 Linguagem de programação (Fonte: Shutterstock / Abscent)

## Linguagem de primeira geração

Desenvolvida no início da era dos computadores, esta linguagem é interpretada pelos microprocessadores. Cada microprocessador possui uma linguagem própria de entendimento, o que pode ocasionar erros de programação em processadores de uma mesma família de fabricantes.

Ex: Linguagem binária. 0 e 1.

## Linguagem de segunda geração

Surgida em meados dos anos 50, foi considerada a primeira linguagem de alto nível, visto que era de fácil entendimento e, portanto, considerada mais humana.

Ex: Assembly

## Linguagem de terceira geração

Em meados dos anos 80, surgiram com os conceitos de programação estruturada e programação orientada a objetos. Ex: Pascal, Cobol, C, C++

## Linguagem de quarta geração

É característica dessa linguagem dar suporte para execução de rotinas auxiliares a linguagens de terceira geração.

Ex: Linguagem de consulta, utilizada para conexão com banco de dados. (SQL)

Uma vez que o desenho será a base da implementação, o processo de documentação de uso do produto passa a ter importância nesta fase, onde a documentação e a programação devem andar lado a lado.





## Próximos Passos

---

Etapa de documentação e manutenção do produto no processo de desenvolvimento de *software*.

---

## Explore mais

---

Pesquise na internet sites, vídeos e artigos relacionados ao conteúdo visto.

Em caso de dúvidas, converse com seu professor online por meio dos recursos disponíveis no ambiente de aprendizagem.