A programação estruturada é independente da linguagem utilizada e se caracteriza por uma forma de programação que pode ser reduzida por apenas três estruturas: Sequencial, Decisão e Repetição; conceito criado por Michael A. Jackson no livro "Principles of Program Design" de 1975. Uma linguagem que representa esse conceito é a linguagem C.

. Estrutura Sequencial: o Algoritmo é executado linha por linha de forma ordenada sequencial:

Exemplo:

Comando 1

Comando 2

Comando 3

. Estrutura de Decisão: Possibilita a execução ou não execução de algoritmos através da análise de afirmações logicas (verdadeiro ou falso), essa estrutura pode ser representada pelo IF, ELSE e Switch na linguagem C.

Exemplo 1:

Se <condição for verdadeira> então

Comando 1

Comando 2

Comando 3

Exemplo 2 :

If(x>5){

X++;

}

. Estrutura de Repetição: Essa estrutura permite que uma ou mais comandos sejam executados enquanto a condição for verdadeira.

Exemplo 1:

Enquanto< condição verdadeira > então

Comando 1

Comando 2

Comando 3

Exemplo 2:

For(i=0; i<4; i++){

printf(“estrutura de repetição em execução \n”);

}

A Programação Estruturada por se caracterizar de estruturas simples e limitadas é necessário a decomposição do problema a ser resolvido de forma a se tornarem problemas menores que serão resolvidos de forma sequencial para a solução do problema maior anterior. Dessa forma o Programa costuma ter muitas linhas de códigos e ser muito complexo, por isso se aplica o conceito de modularização que se baseia em funções que são separadas da função principal, mas que dependem da função principal para retornar algum resultado. Dessa forma pode se dividir o programa e executar ações sem rescrever a função toda, colocando apenas a função separada da função principal.

A programação orientada a objeto se caracteriza por um padrão de desenvolvimento de algoritmos usados por diversas programações, que se baseia em uma representação de um objeto real. Existem 4 pilares básicos para uma programação orientada a objeto são eles:

**Abstração**: Consiste em dar três pontos essenciais para o objeto representado no algoritmo: identidade, propriedade e métodos; a identidade do objeto visa identificar o objeto no código; a propriedade visa dar características para esse objeto como tipo, tamanho velocidade entre outros; Os métodos visam definir as ações que serão realizadas pelo objeto.

**Encapsulamento:** É uma técnica que visa esconder as propriedades do objeto. Dessa forma adiciona segurança ao objeto e ao programa. 

**Herança:** Caracteriza-se pela hereditariedade de propriedades e características de outro objeto ou grupo de objetos. Dessa forma torna mais dinâmico o algoritmo e facilita a programação não sendo necessário reescrever todas as características do objeto apenas dizer sua hereditariedade.

**Polimorfismo**: Um dos pontos essenciais da programação orientada a objetos é o chamado polimorfismo, que se caracteriza por modificar características herdadas de outros objetos ou grupo de objetos.

A programação orientada a objeto traz conceitos de simulação de objetos reais a fim de tornar o programa mais voltado a realidade dessa forma ela representa os elementos do algoritmo em objetos ou classes. Utilizando-se de conceitos de objetos a programação orientada a objeto se torna extremamente mais versátil e dinâmica. 