

---

## Resumo: Modularidade

---

O texto aborda a modularidade e, com isso, o conceito de m dulos: partes de um programa bem definidas e  nicas, arquitetadas de uma maneira tal que cada uma desempenha sua pr pria fun  o, sem alterar ou interferir nas demais. Para que tal parte do programa seja, de fato, um m dulo bem estruturado, o m dulo e o software precisam satisfazer cinco caracter sticas: decomponibilidade, componibilidade, compreensibilidade, continuidade e prote  o.

---

### **Decomponibilidade:**

A caracter stica de decomponibilidade indica que o software   arquitetado de tal forma que pode ser decomposto em partes menores que simultaneamente t m conex o umas com as outras e s o suficientemente diferentes de tal forma que possuem independ ncia uma das outras para realizar suas tarefas sem alterar o resto do software ou interferir nas tarefas de outras partes. Podemos visualizar essa caracter stica como um algoritmo de divis o e conquista que transforma um problema complexo em diversos problemas menores e menos complexo que s o resolvidos separadamente.

### **Componibilidade:**

A caracter stica de componibilidade indica que os m dulos s o arquitetados de tal forma que podem ser facilmente unidos para a produ  o de novos sistemas e softwares, principalmente em contextos n o originalmente planejados. Podemos visualizar essa caracter stica como uma fun  o de ordena  o compat vel com outras fun  es que lidam com inteiros, strings, listas, tuplas e demais elementos orden veis por algum par metro.

### **Compreensibilidade:**

A caracter stica de compreensibilidade indica que os m dulos do software s o arquitetados de tal maneira que um leitor humano consegue compreender com facilidade o que cada m dulo   e faz sem ter que ler os demais m dulos ou, no pior caso, que tenha que ler uma quantidade pequena de outros m dulos.

### **Continuidade:**

A caracter stica de continuidade indica que os m dulos do software s o arquitetados de tal maneira que, quando modificados, apenas alteram o comportamento do m dulo em quest o ou o comportamento de um n mero pequeno de m dulos al m de original, ou seja, o software continua como se o seu principal comportamento n o tivesse sido alterado, uma vez que uma mudan a em m dulo n o alterou significativamente o software.

---

## Resumo: Modularidade

---

### **Prote  o:**

A caracter stica de prote  o indica que os m dulos do software s o arquitetados de tal maneira que, quando um evento inesperado ocorre enquanto o software est  em execu  o, o problema causado por essa anormalidade afeta apenas o m dulo em que ocorre ou, no pior caso, afeta uma quantidade pequena de m dulos al m do m dulo original, sem mudar completamente o comportamento do software, mantendo assim sua funcionalidade.