

# AULA 1 - Introdução

Iniciais dos alunos: *AVC, JPP, PC*

---

Disciplina: *Programação Modular (INF1301)* – Professor: *Flavio Bevilacqua*  
:

As vantagens do modelo modular incluem: particionar problemas (subdividi-los, divide and conquer) e permitir que diversos indivíduos contribuam para a solução de um mesmo problema. Tudo que é computável pode ser desenvolvido.

Módulo genérico -> void\*

Vantagens da modularidade:

- Permite vencer barreiras de complexidade.
- Facilita o trabalho em grupo através da paralelização de tarefas.
- Permite reuso do código.
- Permite a criação de acervos de módulos que fazem parte de determinados assuntos.
- Facilita a administração de baselines com módulos já testados.
- Facilita o desenvolvimento incremental: agrega, testa, agrega, testa, agrega, testa, etc.
- Facilita aprimoramento individual, no sentido de que um aspecto pode ser modificado sem que o resto do projeto seja afetado ou tenha que ser modificado.
- Reduz o tempo de compilação. Isso porque somente os módulos que foram modificados devem ser recompilados.

Teste unitário é o teste individual de cada módulo feito pelo desenvolvedor. Teste de integração é o teste do executável, o conjunto dos módulos.

Baseline é o estado testado e estável do executável, depois que todos os testes unitários eo teste de integração obtiveram sucesso.

Se algo é atualizado/modificado novos testes devem ser executados. Ao final dos testes, se tudo estiver correto, uma nova baseline é estabelecida.

## **Princípios da Modularidade**

### **1. Módulo**

Definição física: unidade independente de compilação.

Definição lógica: um único conceito.

### **2. Elementos de uma aplicação**

Hierarquia: Sistemas -> Programas -> Módulos -> Classes (smente se for orientado a objeto) -> Funções -> Blocos de código -> Linha de comando

Artefato é qualquer elemento versionado criado ao longo de um processo de desenvolvimento. Por exemplo: programa, documento, ata, relatório.

Construto (build) é uma versão, mesmo que incompleta, da aplicação. Algo que é funcional e apresentável. Todo construto é um artefato, mas nem todo artefato é um construto.

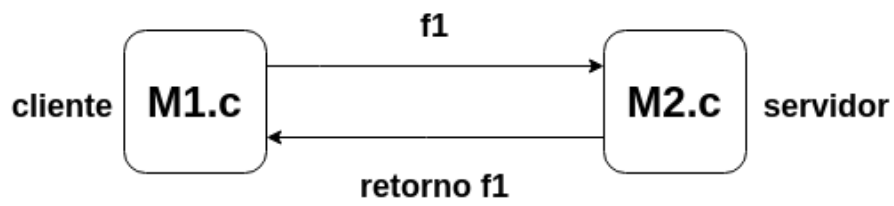
### **3. Interface**

Definição: mecanismo de troca de dados, estados e comandos entre elementos de uma aplicação.

OBS: a interface somente ocorre no mesmo nível da hierarquia dos elementos de uma aplicação. Portanto, interfaces ocorrem entre sistemas, entre programas, entre módulos entre funções, etc.

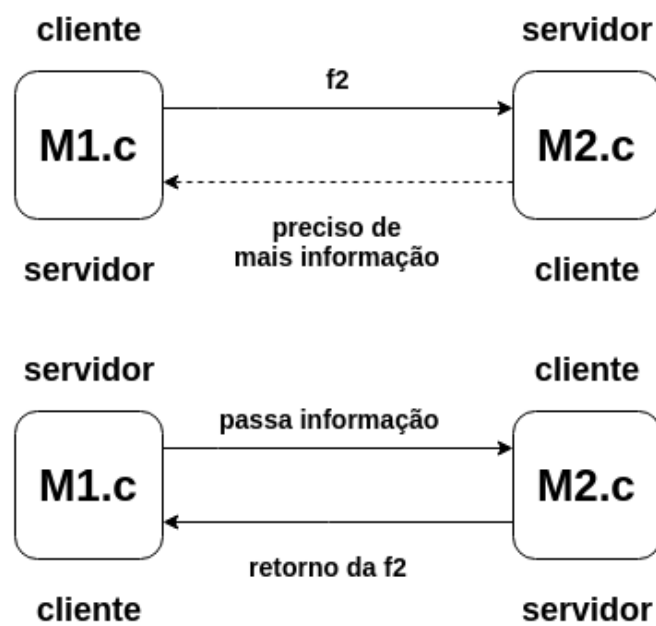
Formas de interface: Sistemas se comunicam por APIs e arquivos. Módulos se comunicam por funções de acesso. Blocos de código se comunicam por variáveis globais.

## Relacionamento Cliente-Servidor

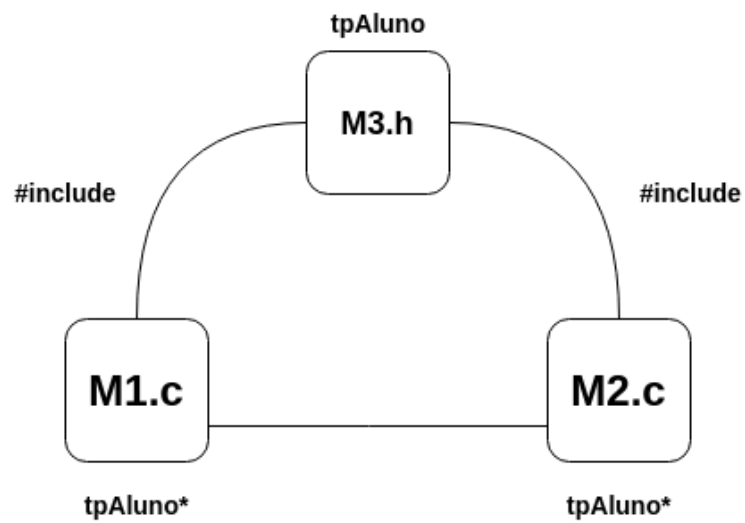


M2 pode ser cliente do servidor M1.

## Caso especial: callback



Interface fornecida por terceiros:



Caso em que dois módulos usam a mesma struct. Em vez de duplicar o código, devemos criar um módulo de definição (.h) e dar include deste módulo em ambos os módulos de implementação (.c).