

Compiladores

Fabio Lubacheski

`fabio.lubacheski@mackenzie.br`

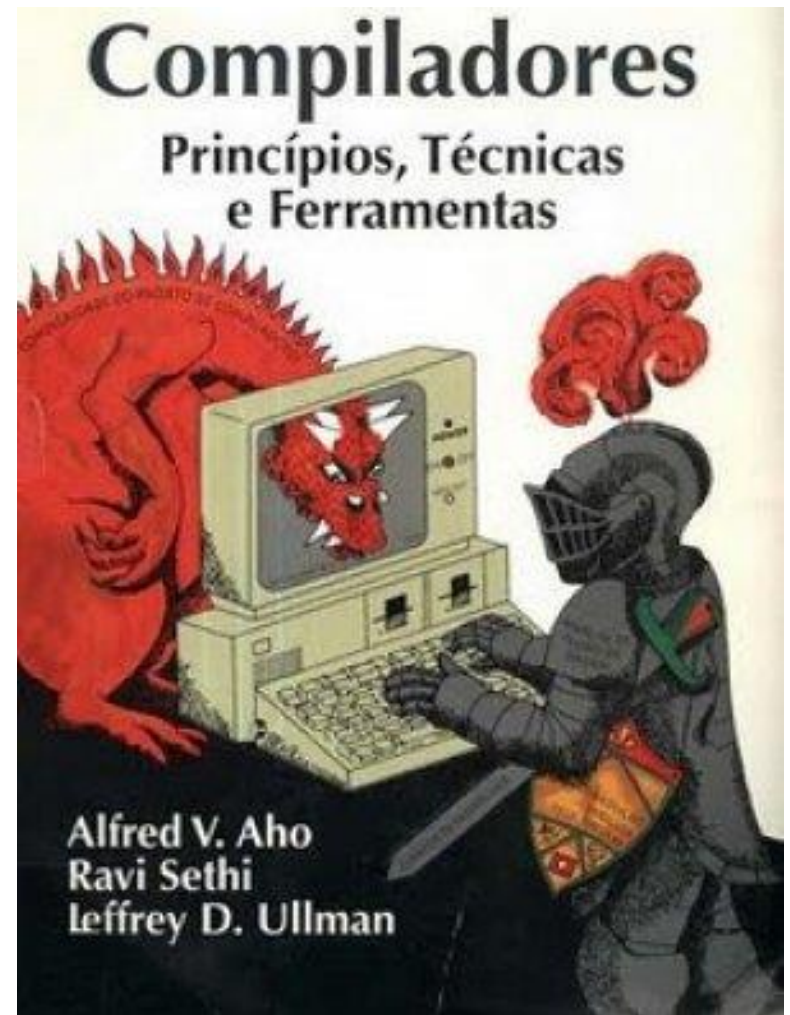
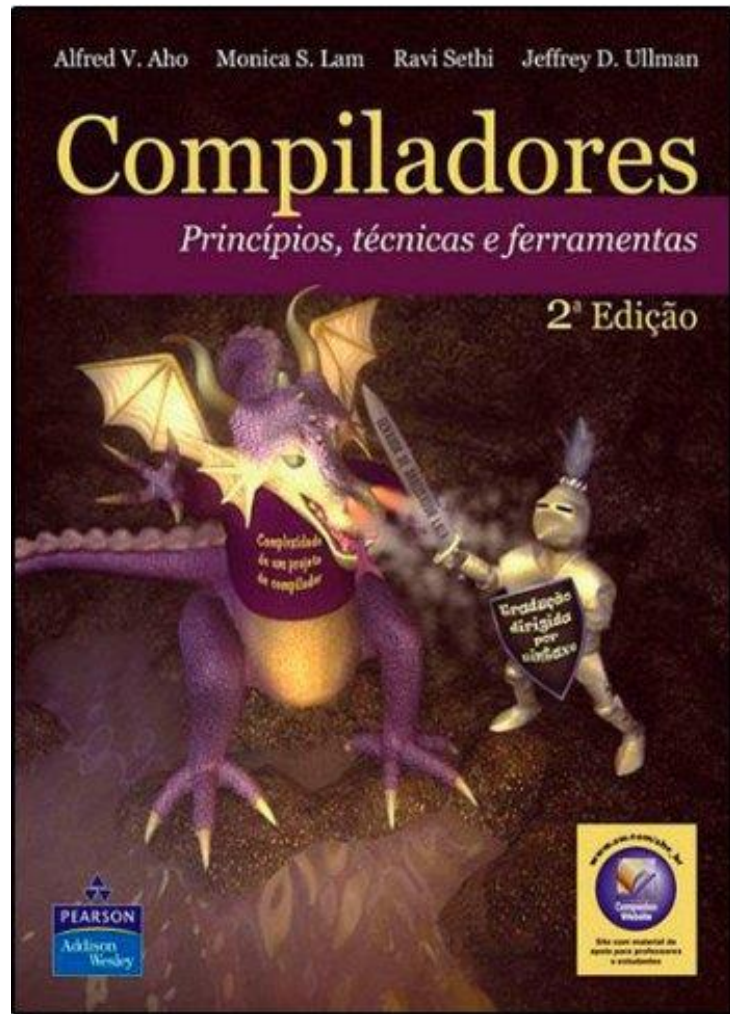
`http://lattes.cnpq.br/9894811024725114`

Plano de Ensino

- Conteúdo Programático
- Metodologia
- Critérios de Avaliação
- Bibliografia básica e complementar

Livros e referências

- Dragão



Livros e referências

- Aulas do **Prof. Tomasz Kowaltowski** em:

<https://sites.google.com/unicamp.br/ic-mo403-mc900/main>

E também o livro disponível em:

<https://drive.google.com/file/d/1DyqeBUgayQpjh1yaHziQgm8WCE7u7ZaN/view>



Compiladores

- Motivação

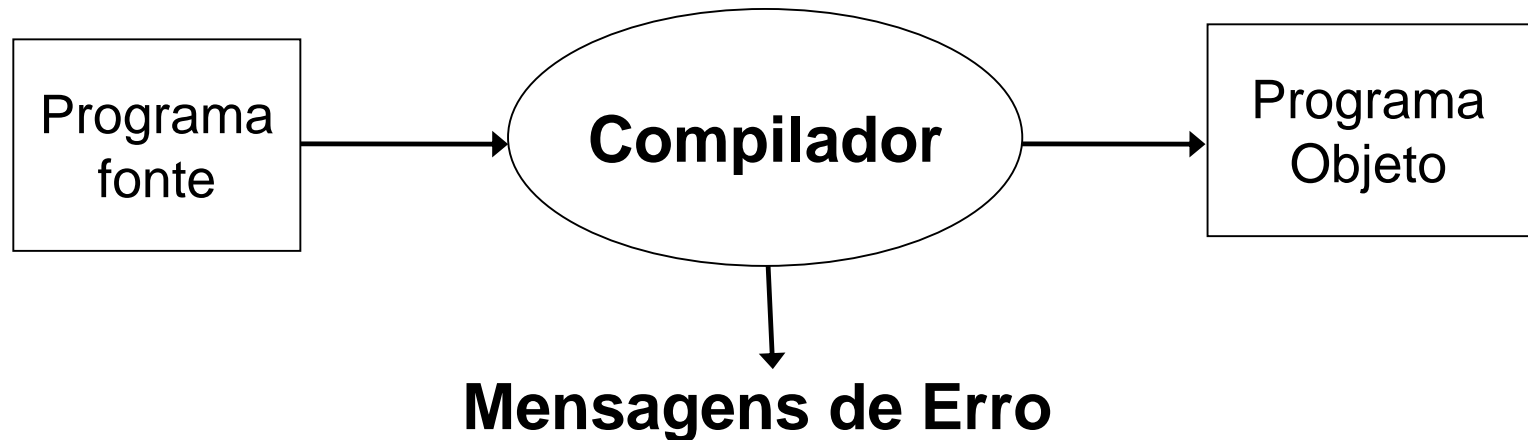
Os princípios e técnicas da construção de compiladores são usadas muitas vezes na carreira de um cientista da computação.

- Pré-requisitos

- Linguagens de Programação;
- Linguagens Formais e Autômatos;
- Sistemas Operacionais;
- Arquitetura de Computadores;
- Algoritmos e Estrutura de Dados;e
- Engenharia de Software.

Definições iniciais

Um compilador é um **processador** de linguagem, cuja função é receber como **entrada** um programa escrito em uma linguagem de programação (linguagem fonte) e converte para um programa equivalente em outra linguagem (linguagem objeto).

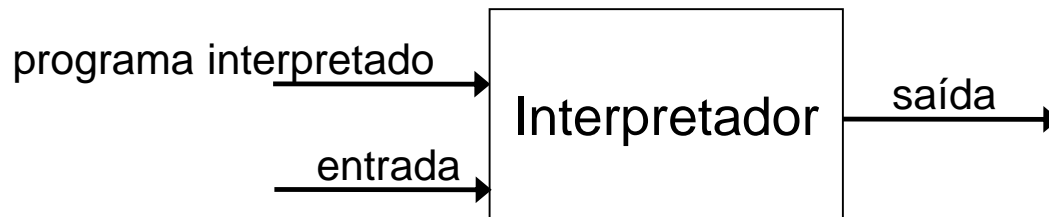


Definições iniciais

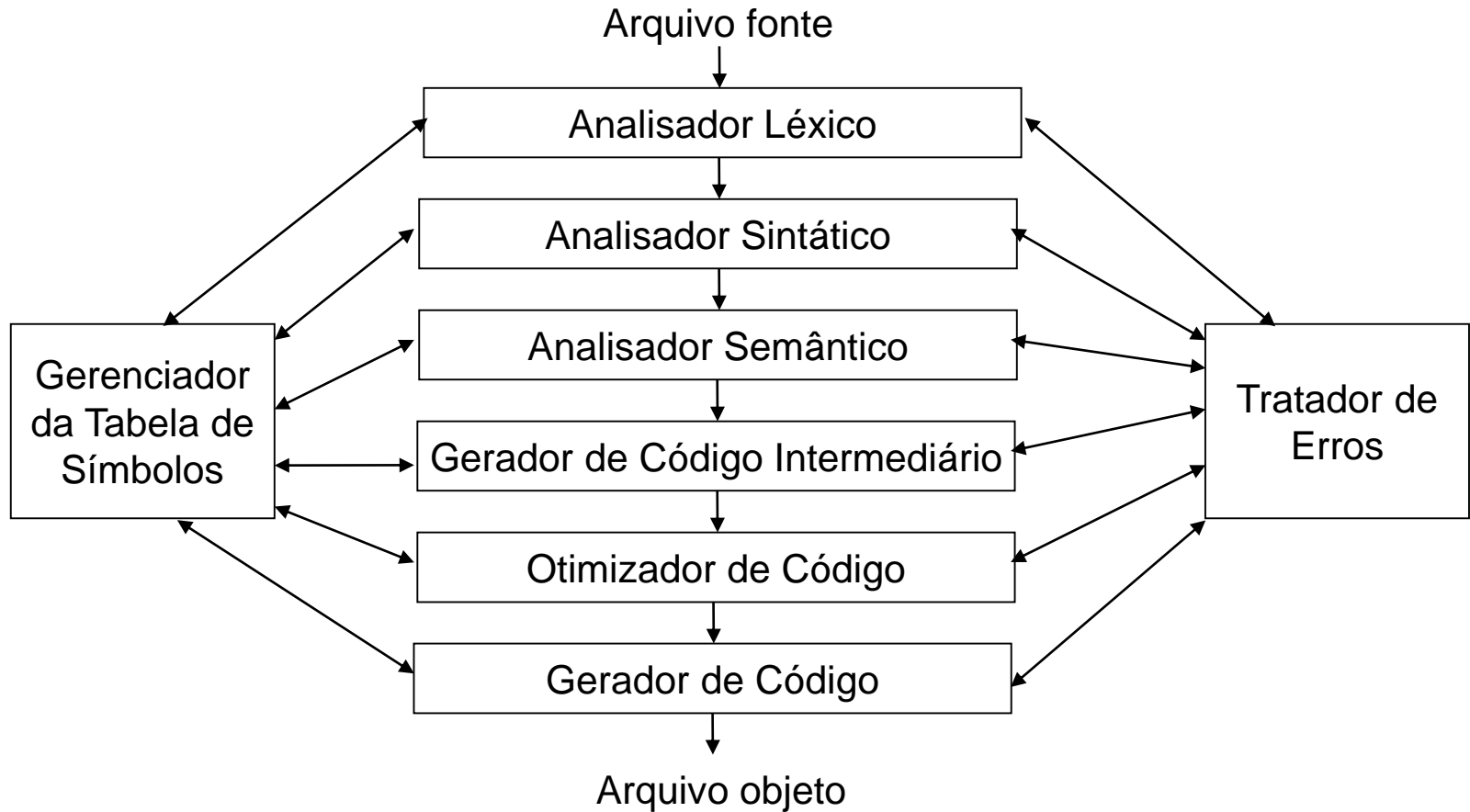
Se o programa objeto for uma programa em uma linguagem de máquina executável, poderá ser chamado pelo usuário para processar **entradas** e produzir **saída**.



Um **interpretador** é outro tipo de **processador** de linguagem, mas não produz um **programa objeto** como saída, um interpretador executa diretamente as operações especificadas no programa fonte sobre as entradas fornecidas pelo usuário.



Fases do Compilador



Fases do Compilador

- Gerenciador da tabela de símbolos

Sua função é registrar os identificadores usados no programa fonte e coletar informações sobre os seus atributos. A tabela de símbolos é um estrutura de dados com um registro para cada identificador, com campos contendo os seus atributos, além disso essa estrutura deve permitir uma recuperação bastante eficiente da informação armazenada.

- Tratador de erros

Cada fase do compilador pode conter erros, que precisam ser tratados de tal forma que a compilação possa continuar

Analizador léxico

lê o arquivo fonte, agrupando os caracteres e classificando em átomos;

Comentários, espaços e outros caracteres de controle são ignorados;

Inicia a construção da tabela de símbolos.

Repassa os tokens (códigos internos) para o analisador sintático; e

Normalmente é implementado como uma subrotina que funciona sob o comando auxiliar do analisador sintático.

Imagine que o analisador léxico leia a seguinte linha em um arquivo fonte.

```
montante = deposito_inicial+taxa_de_juros*60;
```

Analizador léxico

Linha no arquivo fonte

```
montante = deposito_inicial + taxa_de_juros* 60;
```

Os tokens poderiam ser agrupados da seguinte forma:

identificador (id1): **montante**

símbolo atribuição: **=**

identificador (id2): **deposito_inicial**

operador adição: **+**

identificador (id3): **taxa_de_juros**

operador multiplicação: *****

número: **60**

Os espaços desses tokens são eliminados pelo analisador léxico.

Analizador léxico

Certos tokens serão enriquecidos por um valor léxico, por exemplo o identificador **taxa_de_juros**, além de ser gerado um token identificador, ele também é inserido na tabela de símbolos na entrada **id3**:

Tabela de símbolos

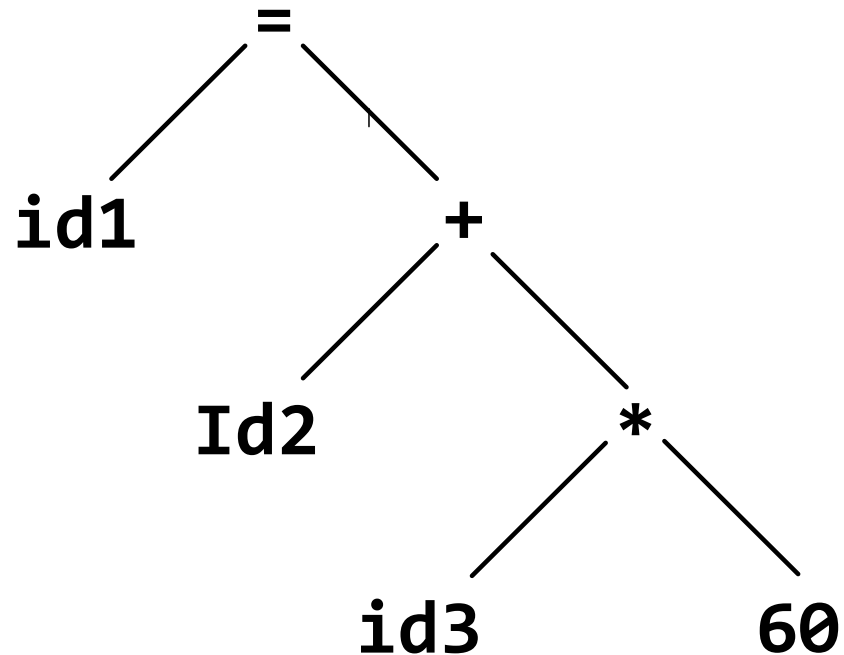
	identificador	categoria
id1	montante	VAR	
id2	deposito_inicial	VAR	
id3	taxa_de_juros	VAR	

Analizador léxico para Analizador sintático

O **analizador léxico** repassa para o **analizador sintático** a expressão da entrada convertida em uma representação interna com as entradas dos identificadores na tabela de símbolos:

id1 = id2+id3*60

A partir da entrada do analisado léxico o **analizador sintático** produz uma **árvore sintática** com a expressão.



Analizador sintático

O **analizador sintático** verifica se a estrutura gramatical do programa está correta. Isto é, se essa estrutura foi formada usando as regras gramaticais da linguagem;

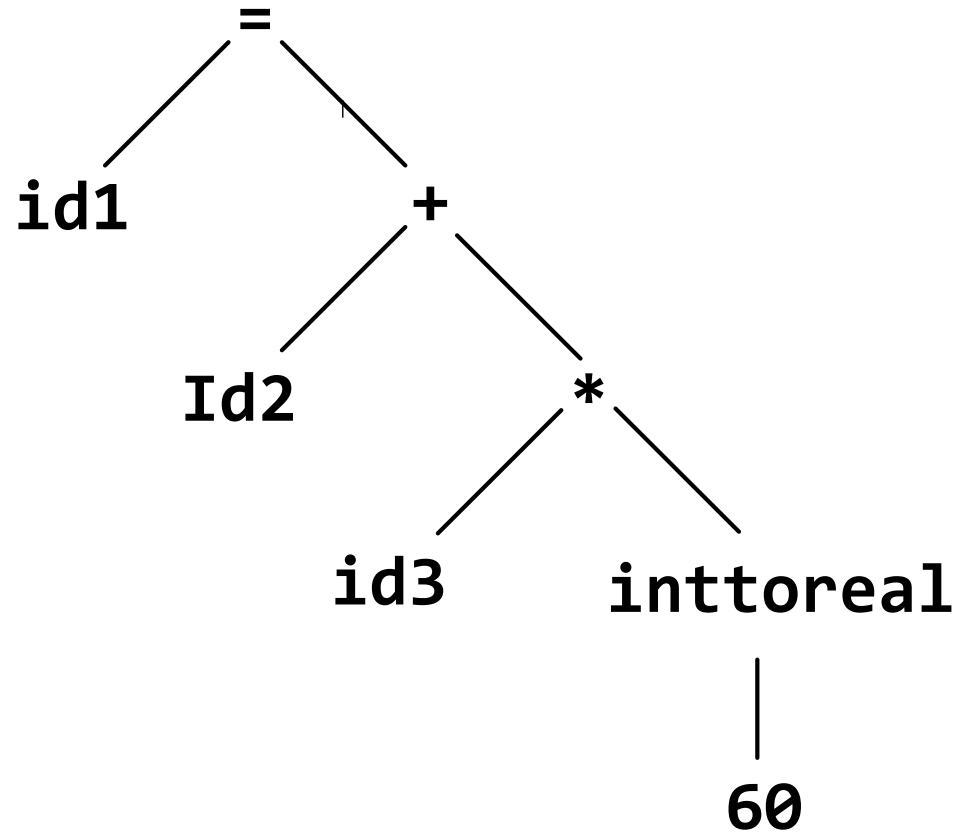
árvore, chamada **árvore sintática ou árvore de derivação**, é usada para detecção de erros de sintaxe; e por fim

O analisador sintático opera conjuntamente com o **analizador semântico**.

Analizador semântico

Verifica se as construções sintaticamente corretas possuem significado lógico dentro da linguagem;

Analizador semântico faz conversões que se fazem necessárias, e permitidas, para dar significado a uma sentença fazendo alterações na árvore sintática.



Gerador de código intermediário

Efetua o mapeamento do árvore sintática, enriquecida com as informações semânticas, para a linguagem intermediária;

Geração de código é feita para uma linguagem_intermediária de nível próximo ao código objeto. Na maioria das vezes temos um código de 3 endereços;

```
temp1 = inttoreal(60)
temp2 = id3 * temp1
temp3 = id2 + temp2
id1 = temp3
```

Não há preocupação com a escolha dos registradores oferecidas pela máquina onde o programa será executado.

Otimizador de código

Otimiza o código intermediário em termos de velocidade de execução e espaço de memória; e

Após a otimização o programa alvo deve continuar funcionando corretamente

Exemplos de otimizações: Atribuições redundantes, suprimir subexpressões comuns, melhor utilização dos registradores e etc...

```
temp1 = id3 * 60.0  
id1 = id2 + temp1
```

Gerador de código objeto

Geração do código alvo, normalmente um código de máquina;
As localizações de memória são selecionadas para cada uma das variáveis usadas pelo programa

Instruções intermediárias são traduzidas numa sequência de instruções de máquina que realizam a mesma tarefa.

```
MOVF id3, R2  
MULF #60.0, R2  
MOVF id2, R1  
ADDF R2, R1  
MOVF R1, id1
```

Programas relacionados a Compiladores

- Interpretadores:

Um interpretador é um processador de linguagens, assim como um compilador. A diferença é que o interpretador executa o programa-fonte de imediato, em vez de gerar um código-objeto. Os interpretadores compartilham muitas operações com os compiladores, podendo inclusive existir processadores híbridos.

Exemplo JVM??

- Pré-processadores:

É um programa separado que é chamado pelo compilador antes do início do processo de compilação. Ele pode apagar comentários, incluir arquivos e executar substituição de macros dentre outras coisas.

- Depuradores:

É utilizado para determinar erros de execução em um programa compilado, e costuma ser integrado com um IDE. Para efetuar suas funções, o depurador precisa receber informação simbólica apropriada do compilador.

Para saber mais

- Leia os capítulos 1 e 2 do livro AHO, A.V., LAM, M.S., SETHI, R., ULLMAN, J.D. **Compilers: Principles, Techniques and Tools**. 2.ed. New York: Addison-Wesley, 2006.

Exercícios

1. Explique as fases que compõe a parte de análise e as fases que compõe a fase de síntese em um processo de compilação.
2. Quais são as vantagens de um ter uma linguagem de máquina e uma linguagem interpretada ?
3. Procure exemplos de linguagens interpretadas e linguagens compiladas.
4. O que são compiladores *just-in-time* ? De um exemplo de linguagem de programação com essa característica.
5. O que é um pré-processador ? Explique como ele trabalha em conjunto com o compilador.
6. Explique cada uma das fases de um compilador e como se interagem entre si.
7. Explique para que serve o Montador, descrevendo como é feita a conversão do código de montagem para o código de máquina.
8. Explique o que são carregadores e editores de ligação.

Exercícios

9. Descreva o papel do analisador léxico no processo de compilação e liste um conjunto de funções desempenhadas usualmente pelo analisador léxico, descrevendo brevemente.
10. Quais são os componentes de um átomo (=token) extraído pelo analisador léxico, e em que aplicação são utilizadas no compilador ?
11. Qual é o papel dos átomos na linguagem descrita por uma gramática livre de contexto?
12. O que é uma tabela de símbolos ? Como é que o analisador léxico e analisador sintático podem participar de sua criação ?
13. O Que são palavras reservadas (=palavras chaves) ? Qual sua função nas linguagens de alto nível? Como é que o analisador léxico efetua o reconhecimento de palavras reservadas ?
14. Como o analisador léxico interage com o sistema operacional ?
15. Qual o papel do analisador sintático no processo de compilação ?
16. Comente a interação entre analisadores sintáticos e léxico em um compilador.

Fim