



**FACULDADE ZACARIAS DE GÓES**  
**SISTEMAS DE INFORMAÇÃO**  
**ADRIEL ALMEIDA CAFÉ**

## **PROCESSO DE COMPILAÇÃO**

**Valença – Bahia**  
**Novembro 2010**

ADRIEL ALMEIDA CAFÉ

## **PROCESSO DE COMPILAÇÃO**

Trabalho apresentado como requisito parcial para AVII da disciplina Sistemas Operacionais do curso de Sistema de Informação da Faculdade Zacarias de Góes, sob orientação do professor Adonai Medrado.

Valença - Bahia  
Novembro 2010

## INDICE

1 INTRODUÇÃO	3
2 PROCESSO DE COMPILAÇÃO	4
2.1 FASES DA COMPILAÇÃO	4
2.2 REPRESENTAÇÃO GRÁFICA	5
3 COMPILAÇÃO NO JAVA	5
4 COMPILAÇÃO NO C#	6
5 REFERÊNCIAS	7

## 1 INTRODUÇÃO

Quando se fala em programas, geralmente se está falando dos códigos-fonte escritos em linguagens de programação de alto nível, ou seja, com um nível de abstração muito grande, mais próximo do modo “humano” de se pensar. Em contrapartida, existe o código executável que é um código feito numa linguagem de baixo nível de abstração, muito mais próxima do modo de funcionamento das máquinas do que do raciocínio humano.

Existem duas formas de se transformar um programa escrito em uma linguagem de alto nível em um código capaz de ser executado por um dispositivo computacional, são elas: o compilador e o interpretador.

Este trabalho tem como objetivo introduzir o conceito de compilação, explicando os passos necessários para que um programa possa ser compilado. Duas linguagens, Java e C#, foram escolhidas para demonstrarem como o processo de compilação pode ser feito de diferentes formas.

## 2 PROCESSO DE COMPILAÇÃO

Compilação é o processo de tradução de um código fonte escrito normalmente em uma linguagem de alto nível (de fácil entendimento por parte do programador) para uma linguagem de baixo nível. Um compilador é um programa responsável por executar os processos de compilação descritos a seguir.

### 2.1 FASES DA COMPILAÇÃO

#### 1. Análise Léxica

Esta é a primeira etapa da compilação. A função do analisador léxico, também conhecido como scanner, é analisar todo o código fonte e produzir símbolos (tokens) que podem ser manipulados na etapa seguinte. Nesta etapa são eliminados os espaços em branco e comentários.

#### 2. Análise Sintática

O analisador sintático (parsing) é quem dá significado às sequências de tokens criadas anteriormente.

#### 3. Análise Semântica

Esta etapa é responsável por analisar a semântica, ou significado, de cada elemento do código. É ele quem encontra erros como, por exemplo, uma multiplicação entre tipos de dados diferentes.

#### 4. Geração do Código Intermediário

Nesta etapa ocorre a conversão da árvore sintática, criada na etapa 2, em uma representação intermediário do código fonte.

#### 5. Otimização do Código

Nesta etapa o código é otimizado para uma determinada arquitetura (hardware e sistema operacional específico).

#### 6. Geração de Código Final

Nesta última etapa da compilação, o arquivo executável (.exe) é criado, otimizado para aquela arquitetura.

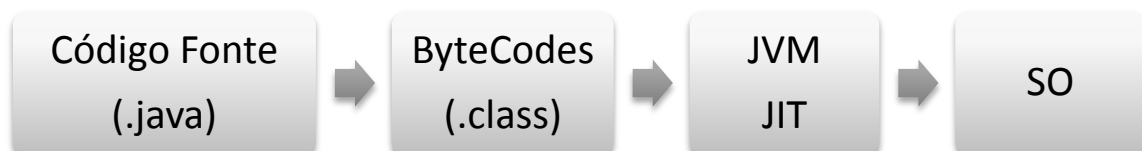
## 2.2 REPRESENTAÇÃO GRÁFICA



## 3 COMPILAÇÃO NO JAVA

Na linguagem Java o processo de compilação ocorre da seguinte maneira:

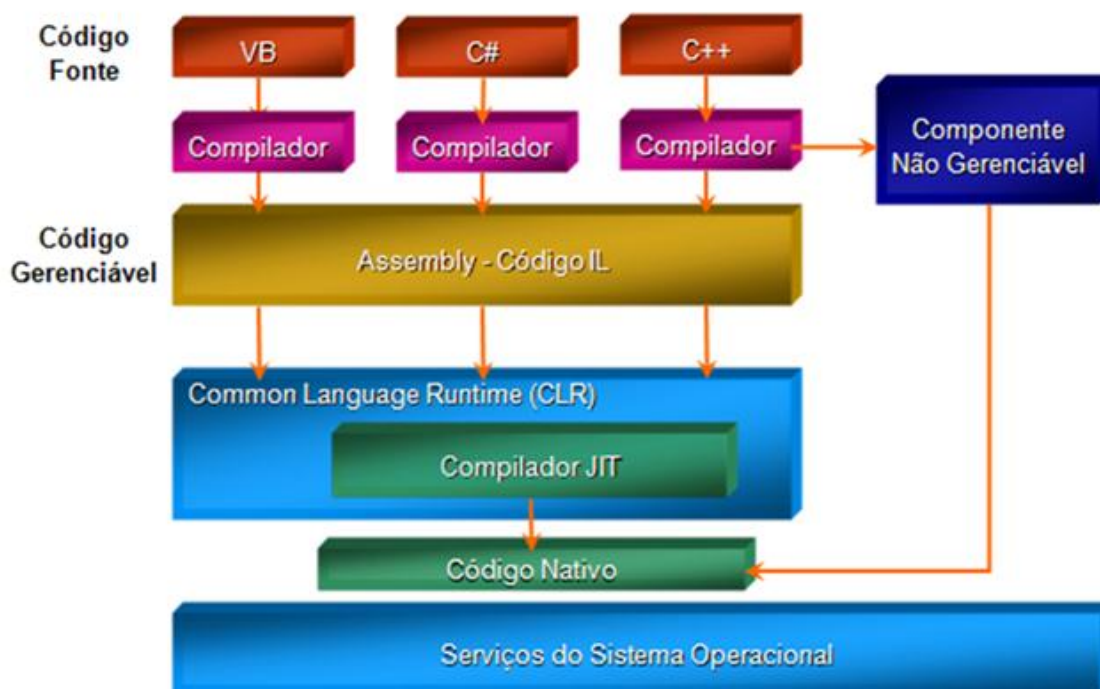
1. O código fonte (extensão .java) é compilado utilizando o javac (Java Compiler) e armazenado em um arquivo de extensão .class, também conhecido como bytecode.
2. Em seguida a JVM (Java Virtual Machine), interpreta os bytecodes produzidos pelo compilador.
3. O JIT (Just-In-Time) converte, em tempo de execução, o código em bytecode para código de máquina.
4. Por último, fica a cargo do Sistema Operacional em questão executar o programa.



## 4 COMPILAÇÃO NO C#

Na linguagem C# o processo de compilação é bastante semelhante ao do Java. Uma das grandes diferenças é a possibilidade de utilizar várias linguagens distintas em um mesmo programa, como pode ser observado mais abaixo.

1. O código-fonte (extensão .cs) é compilado em uma linguagem intermediária (IL). “O código IL, juntamente com recursos como bitmaps e sequências de caracteres, são armazenados em disco em um arquivo executável chamado de assembly, normalmente com uma extensão .exe ou .dll.” (MSDN)
2. Diferente do Java, que utiliza a JVM, o C# utiliza o .NET Framework que trás consigo o CLR (Common Language Runtime), o componente encarregado de gerenciar aplicações desenvolvidas em .NET. No CLR existe o JIT que é o responsável por converter, em tempo de execução, o código em IL para código de máquina, bem parecido com o Java.
3. Com o código nativo gerado, fica a cargo do Sistema Operacional executar o programa.



Fonte: <<http://denmartins.files.wordpress.com/2010/08/compilation5b75d.png>>

## 5 REFERÊNCIAS

- MOHR, Jonathan. **The Compilation Process**. Disponível em: <  
[http://www.augustana.ab.ca/~mohrj/courses/common/csc370/lecture\\_notes/compilation.html](http://www.augustana.ab.ca/~mohrj/courses/common/csc370/lecture_notes/compilation.html)>. Acesso em: 18 nov. 2010.
- MSDN. **Introdução à linguagem C# e ao Framework .NET**. Disponível em: <  
[http://msdn.microsoft.com/pt-br/library/z1zx9t92\(v=VS.100\).aspx](http://msdn.microsoft.com/pt-br/library/z1zx9t92(v=VS.100).aspx)>. Acesso em: 18 nov. 2010.
- SCHNEIDER, Bertoldo. **Resumo sobre o conceito de Compilação**. Disponível em: <  
<http://www.pessoal.utfpr.edu.br/bertoldo/Downloads/Compilador.pdf> >. Acesso em: 18 nov. 2010.
- SILVEIRA, I. F.. **Linguagem JAVA**. Disponível em: <  
<http://www.infowester.com/lingjava.php>>. Acesso em: 18 nov. 2010.
- WIKIPEDIA. **Compilador**. Disponível em: <  
<http://pt.wikipedia.org/wiki/Compilador>>. Acesso em: 18 nov. 2010.