

Netbeans

Projetos

Bibliotecas de Classes

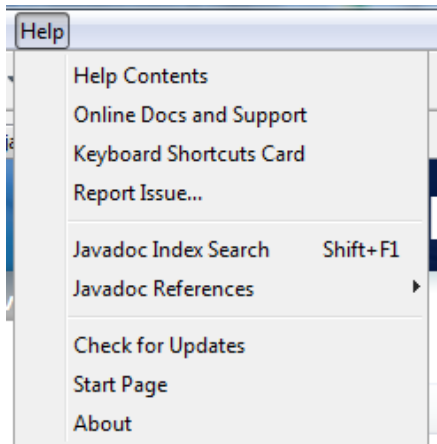
Ficheiro JAR Executável

Atalhos

Templates

- [Referências](#)
- [Projetos](#)
- [Bibliotecas de Classes](#)
- [Ficheiro JAR](#)
- [Atalhos do Teclado](#)
- [Templates](#)

- Menu Help do Netbeans



- Netbeans Home Page

- <http://netbeans.org/index.html>

Projetos

- [Noção de Projeto](#)
- [Ficheiros de um Projeto](#)
- [Definições](#)

Projeto Netbeans

- É um conjunto de **ficheiros** e de **definições** que permitem

- Construção (build)
 - Execução (run)
 - Teste (test)
 - Deteção de Erros (debug)
- do projeto

Projeto
=
Ficheiros + Definições

- Exemplos



- Quais são esses ficheiros e definições?

Tipos de Ficheiro

- Fonte (.java) // contêm código Java ... do programa pretendido
- Teste (.java) // código Java
- Executável (.class) // bytecode
- (.jar) // executável (.class) + código Java (opcional)

Organização dos Ficheiros (segundo finalidade)

- Programa (aplicação)

Programa (aplicação)

Ficheiros internos

Ficheiros externos



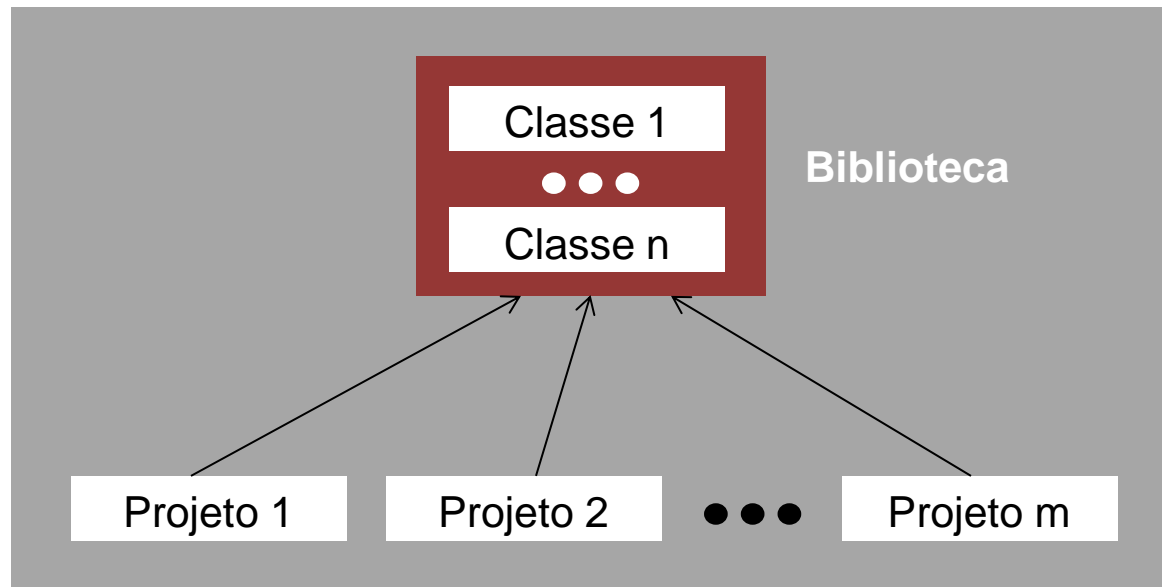
- Sources Packages // agrupa ficheiros fonte
- Libraries // ficheiros externos (fonte e/ou executáveis): bibliotecas + ...
// concretamente, são **referências** de ficheiros
// precisam de ser **importados**
- Teste
 - Test Packages // agrupa ficheiros de teste
 - Test Libraries // ficheiros externos (fonte e/ou executáveis), só para teste

- **Classpath**
 - Define caminhos para compilador procurar classes do projeto durante a construção

Bibliotecas de Classes

- [Noção de Biblioteca](#)
- [Criação de Biblioteca](#)
- [Uso de Biblioteca](#)

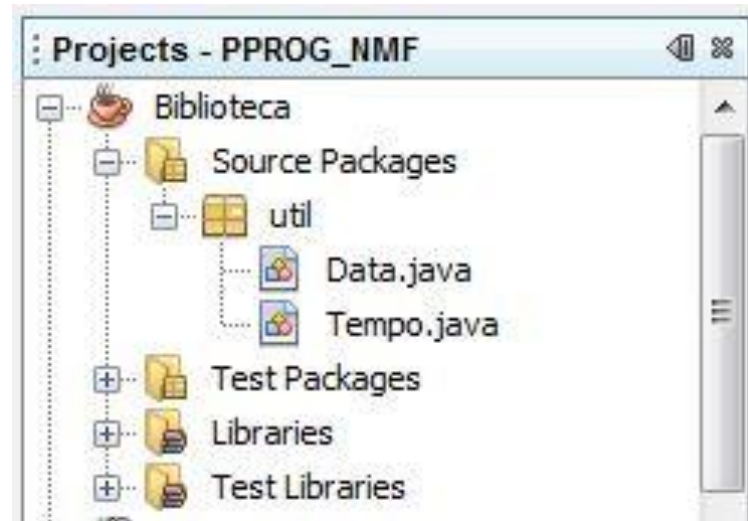
- Noção
 - Conjunto de classes que podem ser **partilhadas** por diferentes **projetos**



- Interesse
 - Permitir a **reutilização** de código (**classes** em POO)
- Como **criar** e **usar** bibliotecas?

- **Exemplo**

- Biblioteca com classes Data e Tempo



- **Procedimento Genérico**

1. Criar um projeto do tipo **Java Class Library** // Exemplo: Biblioteca
2. Criar uma **nova** source **package** // Exemplo: util
3. **Adicionar classes** à package criada // Exemplo: Data e Tempo

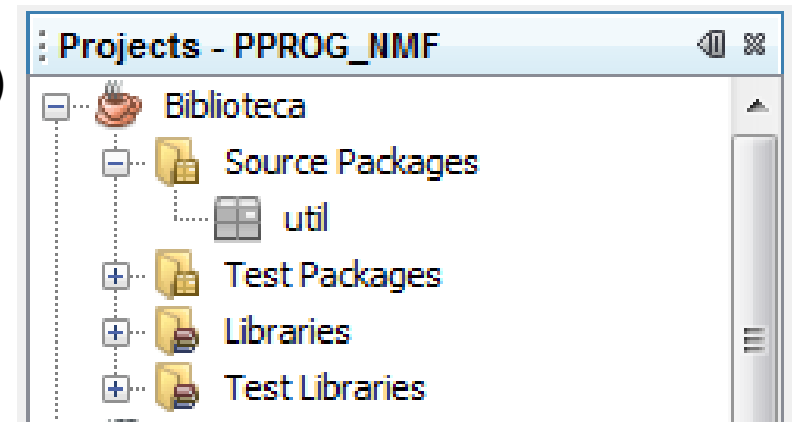
- **Procedimento detalhado**

- Slides seguintes

1. Criar um Projeto do tipo **Java Class Library**
 1. Clique no **menu File**
 2. Clique em **New Project ...**
 3. Na caixa **Categories**, clique em **Java**
 4. Na caixa **Projects**, clique em **Java Class Library**
 5. Clique no botão **Next**
 6. Indique o **nome** da biblioteca (ex: Biblioteca)
 7. Indique a **localização** da biblioteca
 8. Clique no botão **Finish**

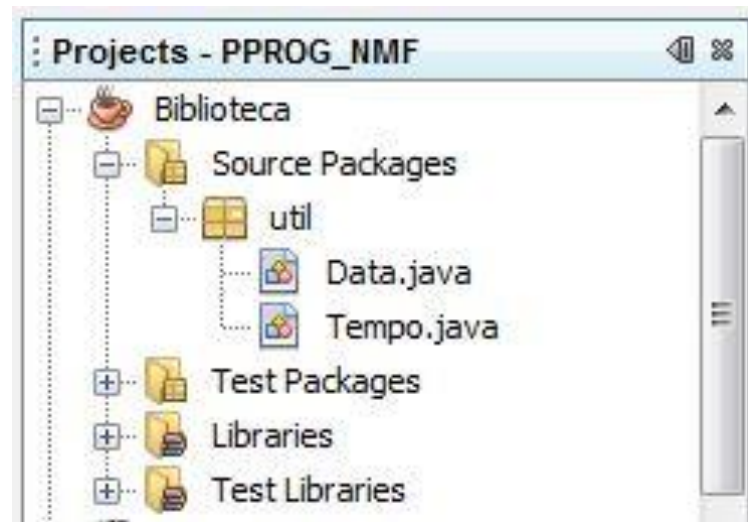


2. Criar uma **nova source package**
 1. Clique direito sobre o **nó do projeto** (Ex: Biblioteca)
 2. Clique **New**
 3. Clique **Java Package ...**
 4. Insira o **nome** da nova source package (ex: util)
 5. Clique no botão **Finish**

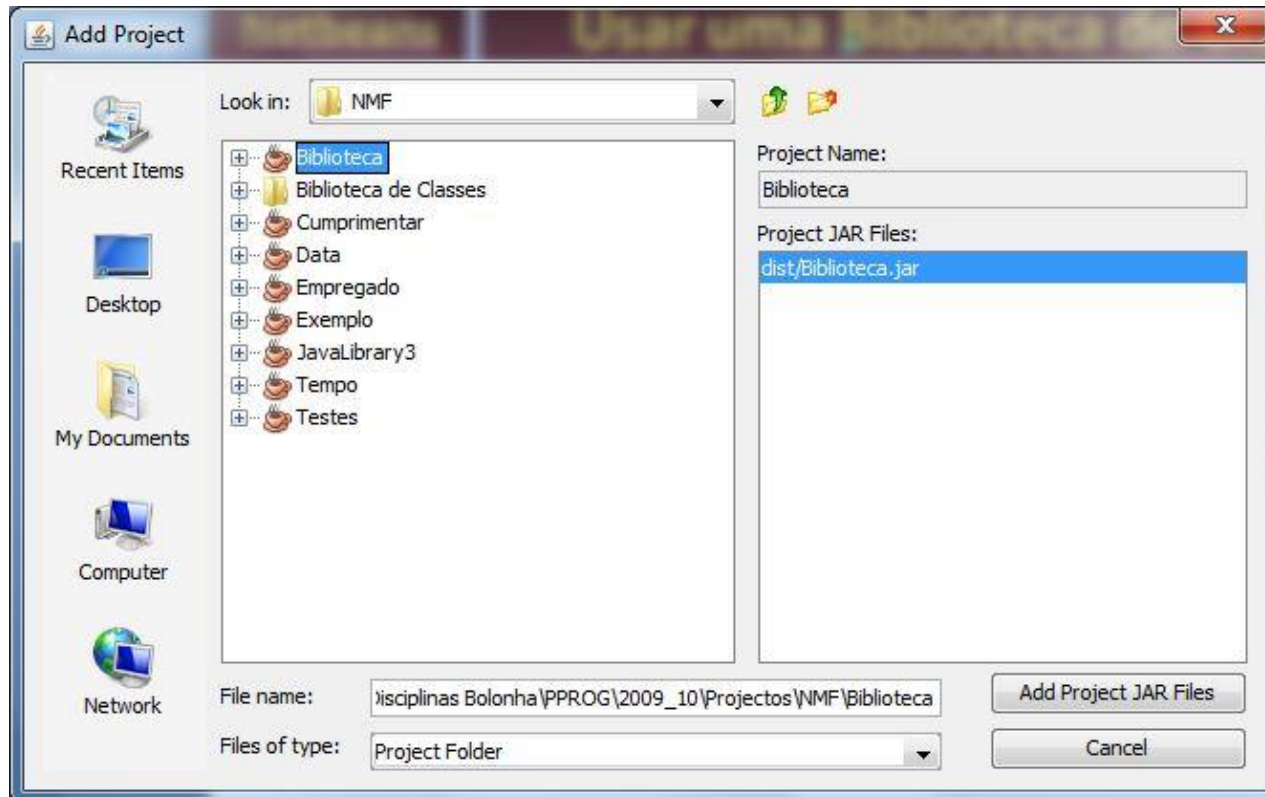
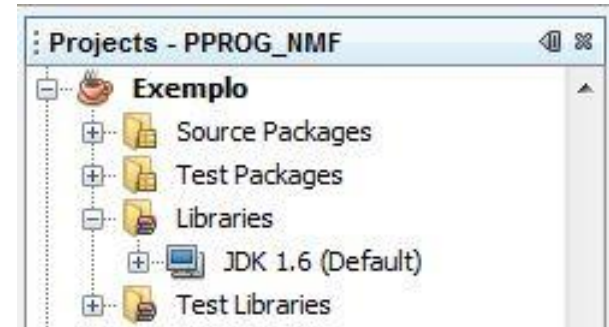


3. Adicionar classes à *package* criada

- Exemplo
 - Copiar classes de outros projetos

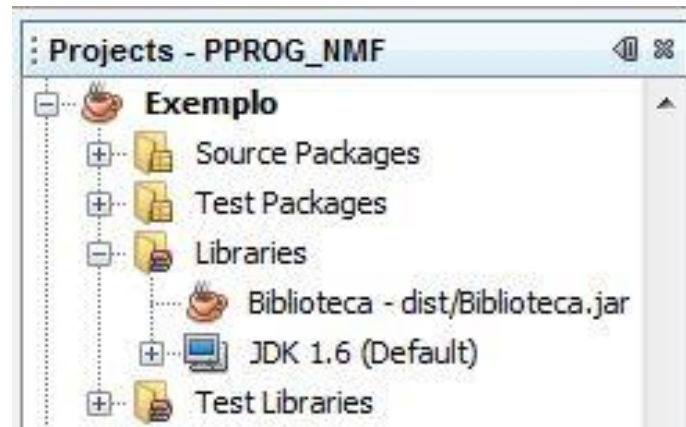


1. Clique direito na pasta **Libraries** do projeto
2. Clique **Add Project...**
3. Clique no **nome do projeto** da biblioteca a adicionar



4. Clique no botão **Add Project JAR Files**

É adicionada a biblioteca à pasta Libraries



5. Importar classes da biblioteca num ficheiro fonte do projeto

■ Exemplos

```
package clinica;

import util.Data;
import util.Tempo;

public class Consulta {

    private Data data;
    private Tempo hora;
    ...
}
```

```
package clinica;

import util.*;

public class Consulta {

    private Data data;
    private Tempo hora;
    ...
}
```


Ficheiro JAR Executável

- [Referências](#)
- [Noção de Ficheiro JAR](#)
- Ficheiro JAR Executável
 - [Criação](#)
 - [Localização](#)
 - [Execução](#)

- **Tutorial Netbeans**

- Packaging and Distributing Java Desktop Applications
- <http://netbeans.org/kb/docs/java/javase-deploy.html>

- **Tutorial Java**

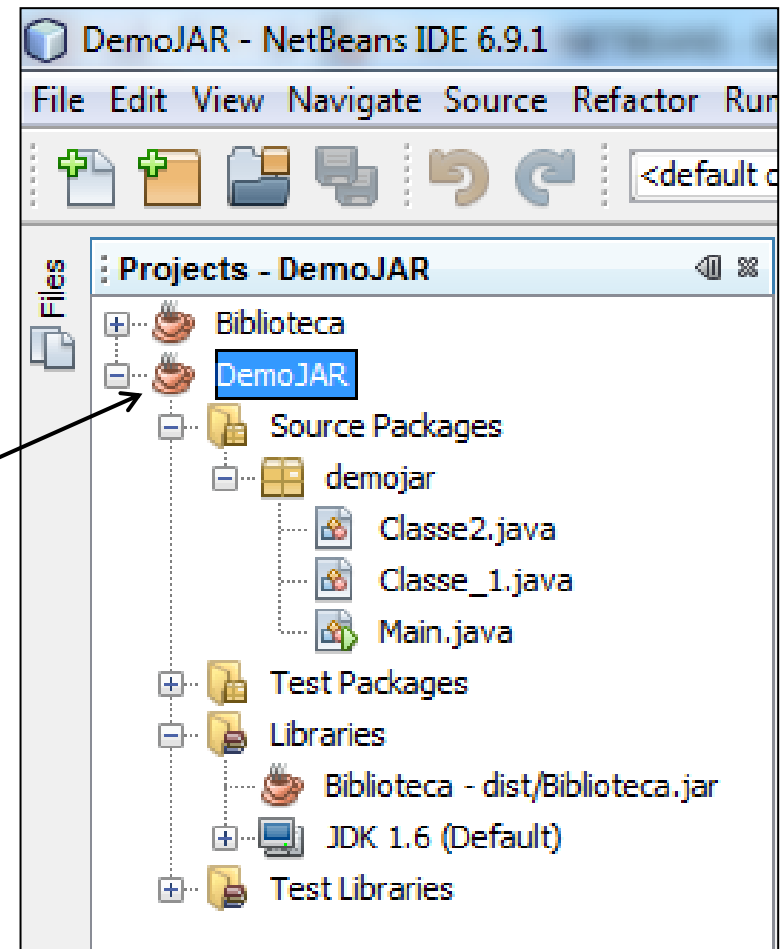
- Lesson: Packaging Programs in JAR Files
- <http://download.oracle.com/javase/tutorial/deployment/jar/index.html>

- **Significado JAR**
 - Java ARchive
- **É um ficheiro de arquivo**
 - Pode conter **múltiplos**
 - Ficheiros
 - Pastas
- **Tem formato ZIP**
 - Semelhante ao ficheiro ZIP
 - Tem **propriedades adicionais** para a distribuição de aplicações Java, tais como:
 - Assinatura digital
 - Compressão adicional
 - Compatibilidade multiplataforma
- **Interesse**
 - Geral: arquivar ficheiros de forma comprimida
 - Particular: ser ficheiro **executável** de programas Java

▪ Procedimento

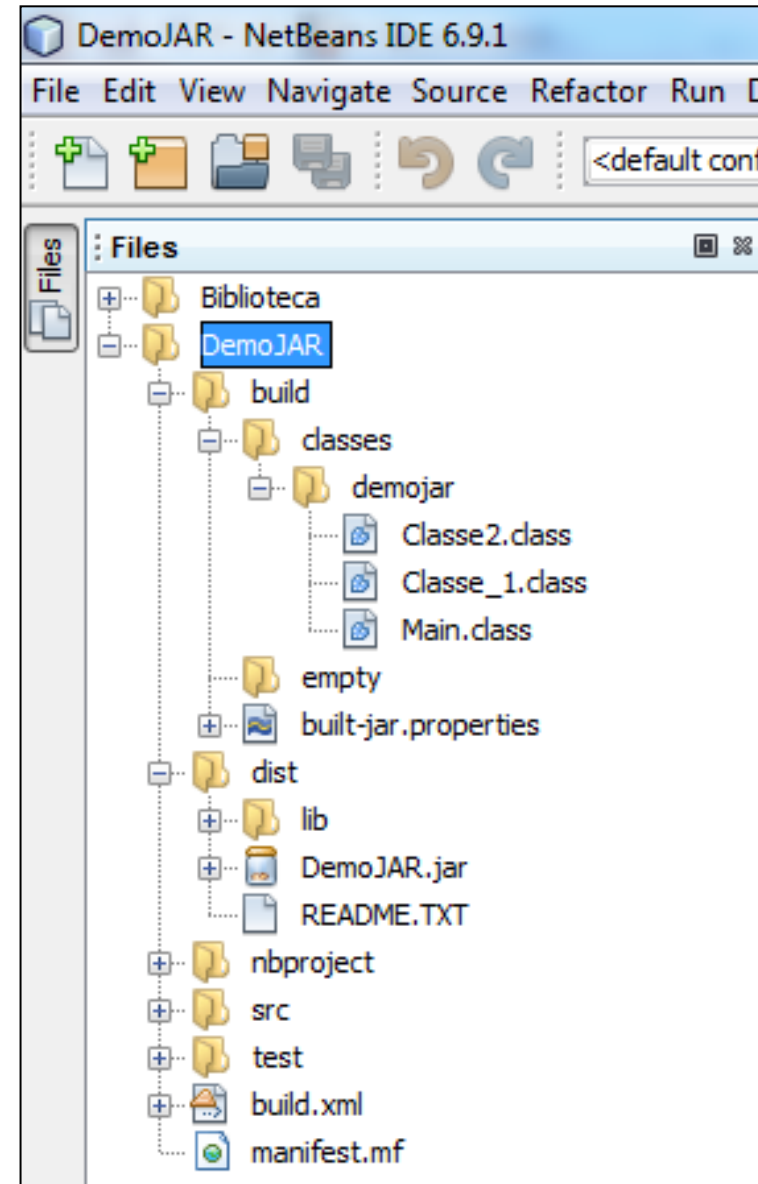
1. Clique na Janela Projects
2. Clique direito no nó do Projecto
3. Clique na opção Set as Main Project
4. Novo clique direito sobre o nó do Projecto
5. Clique na opção Clean and Build Main Project

Nó de Projeto



- Resultado da Construção de um Projeto

- Pastas adicionadas ao projeto
 - build
 - dist
- Todos os ficheiros fonte são:
 - Compilados em ficheiros .class
 - Guardados na pasta build
- Criado um ficheiro JAR executável dentro da pasta dist
 - Por omissão, contém apenas os ficheiros .class do projecto
- Se projecto tiver bibliotecas, para além do JDK:
 - Criada pasta lib na pasta dist
 - Bibliotecas copiadas para pasta lib



- Duplo-clique sobre o ficheiro JAR executável
 - Dentro da pasta dist
- Nota
 - Pasta **dist** pode ser movida/copiada para qualquer lugar

Atalhos do Teclado

■ Formatar Código

- Completo: ALT + SHIFT + f (ou Menu Source > Format)
- Parcial: Selecionar código > (comando anterior)

■ Copiar Linha/Linhas Selecionadas

- Para Baixo: CTRL + SHIFT + ↓
- Para Cima: CTRL + SHIFT + ↑

■ Mover Linha/Linhas Selecionadas

- Baixo: ALT + SHIFT + ↓
- Cima: ALT + SHIFT + ↑

■ Eliminar Linha/Linhas Selecionadas

- CTRL + e

■ Completar Código com Código Existente

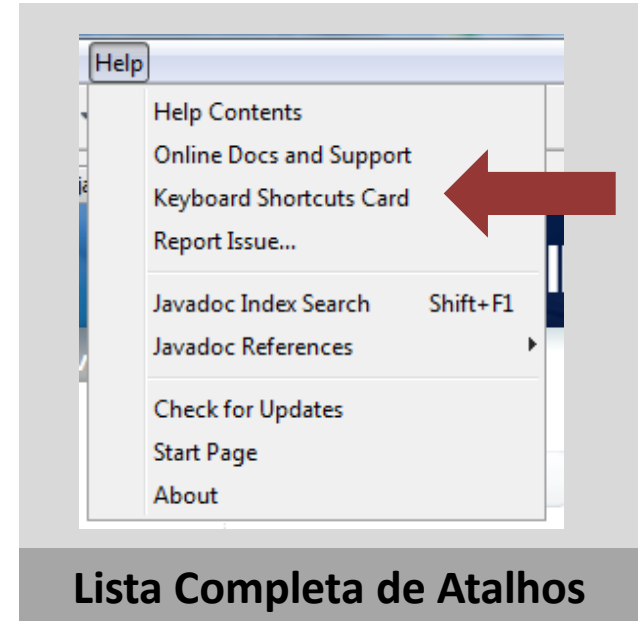
- CTRL + ESPAÇO

// nomes: variáveis, métodos, construtores, classes, etc.

■ Mostrar Lista de Parâmetros de Método

- CTRL + p

// com cursor entre parêntesis de método



Lista Completa de Atalhos

- **Importar Classe**
 - CTRL + SHIFT + i

- **Inserir Automaticamente Métodos de Consulta (get) e de Modificação (set)**
 - Menu Refactor > Encapsulate Fields ...
 - ALT + INSERT + Getter and Setter...
 - ALT + INSERT + Setter...
 - ALT + INSERT + Getter...

- **Mostrar Declaração de Método/Construtor/Variável/Instância/Classe**
 - CTRL + Clique no nome desse elemento

- **Corpo de Métodos**
 - Esconder
 - Todos: CTRL + SHIFT + -
 - Com Cursor: CTRL + -
 - Mostrar
 - Todos: CTRL + SHIFT + +
 - Com Cursor: CTRL + +

- **Ampliar/Reduzir Tamanho da Letra**
 - **Editor**
 - ALT + Mover Roda do Rato
 - **Output**
 - ALT + Mover Roda do Rato
 - CTRL + Mover Roda do Rato

Templates

- Eliminar comentários iniciais

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package pl6_demo2;

/**
 *
 * @author Nelson
 */
public class Exemplo {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```



```
package pl6_demo2;

public class Exemplo {

    public static void main(String[] args) {

    }

}
```

- Eliminar comentários iniciais
 - Abrir Template
 - Menu Tools > Templates > Pasta Java > Java Main Class > Open in Editor
 - Eliminar linhas

```
1 <#assign licenseFirst = "/*">
2 <#assign licensePrefix = " * ">
3 <#assign licenseLast = " */">
4 <#include "../Licenses/license-${project.license}.txt">
5
6 <#if package?? && package != "">
7 package ${package};
8
9 </#if>
10 /**
11  *
12  * @author ${user}
13  */
14 public class ${name} {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21     }
22
23 }
```

Template Original



```
package ${package};

public class ${name} {

    public static void main(String[] args) {

    }

}
```

Template Modificado