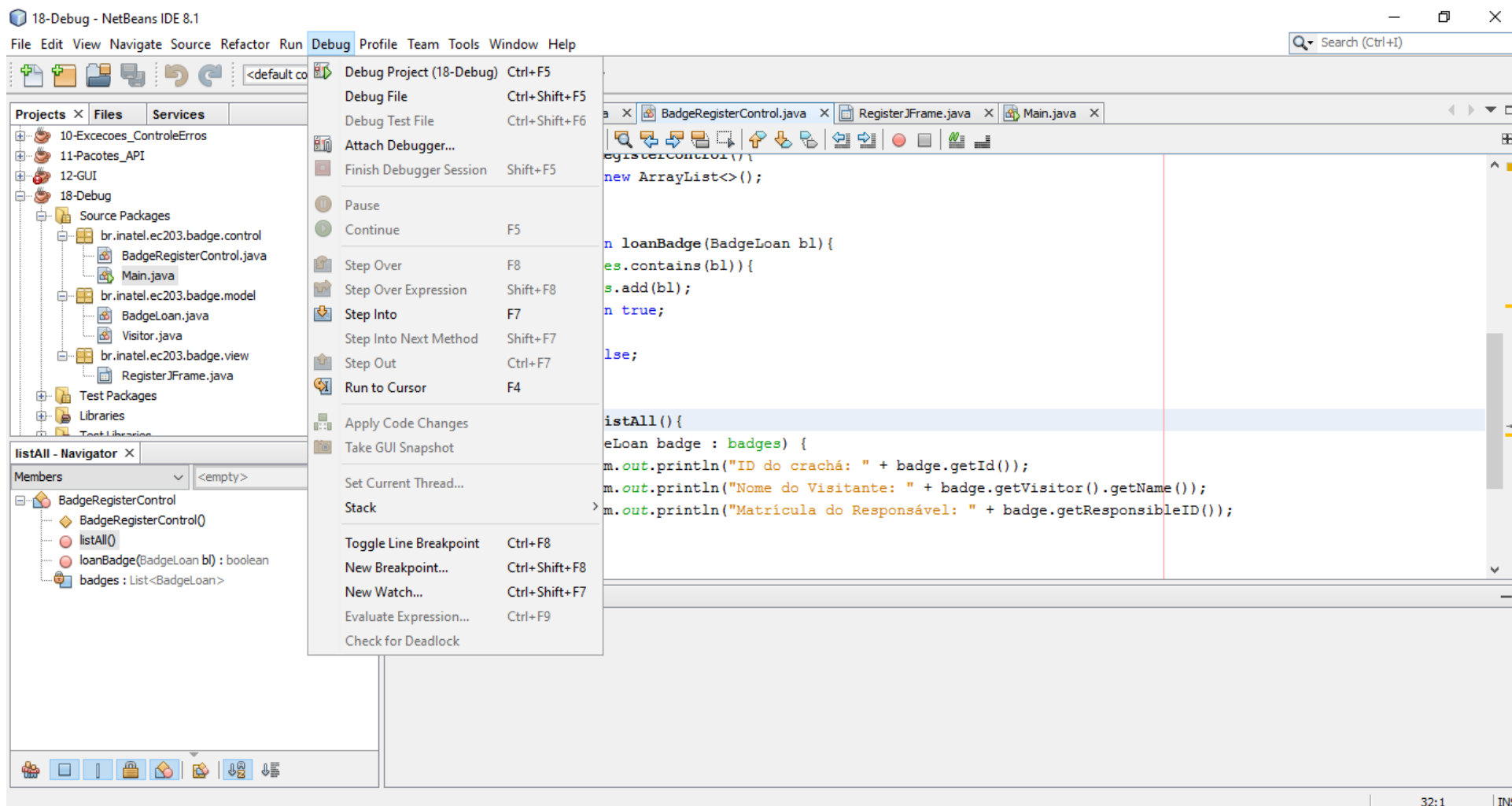


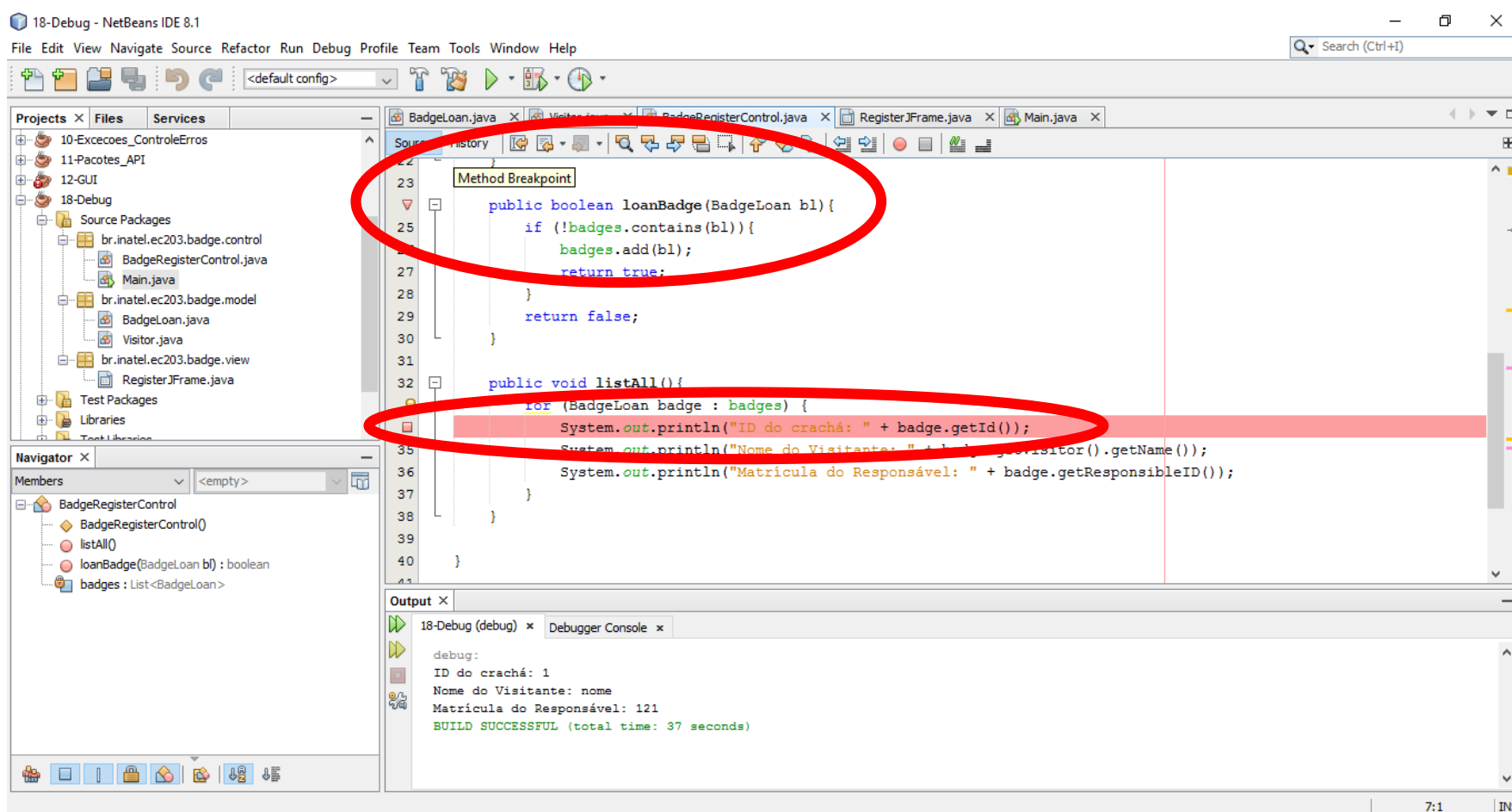
Projeto Final

Utilizando o Debbuger

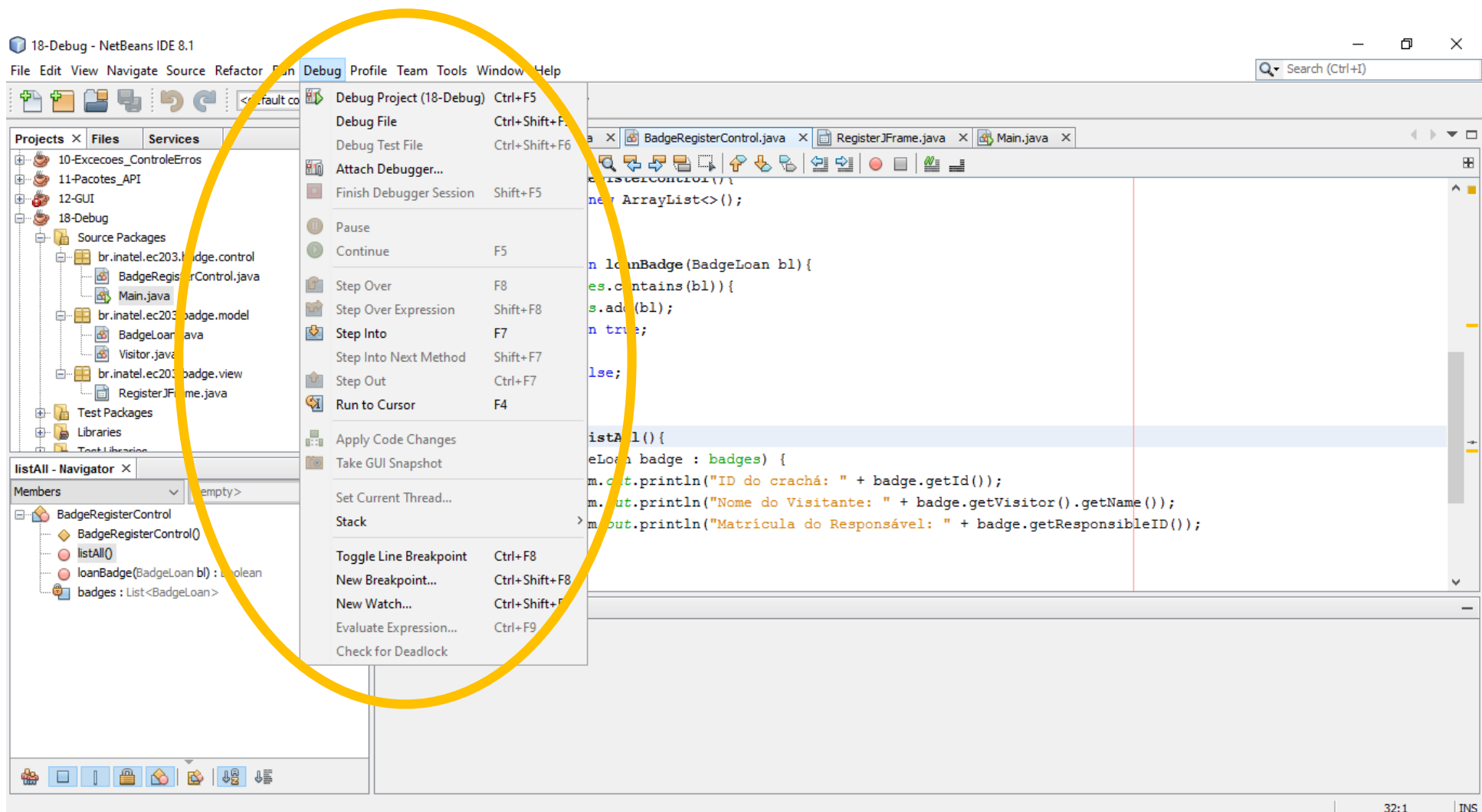
Alguns erros são difíceis de serem localizados, para isso podemos depurar nosso código utilizando o Debugger do Netbeans



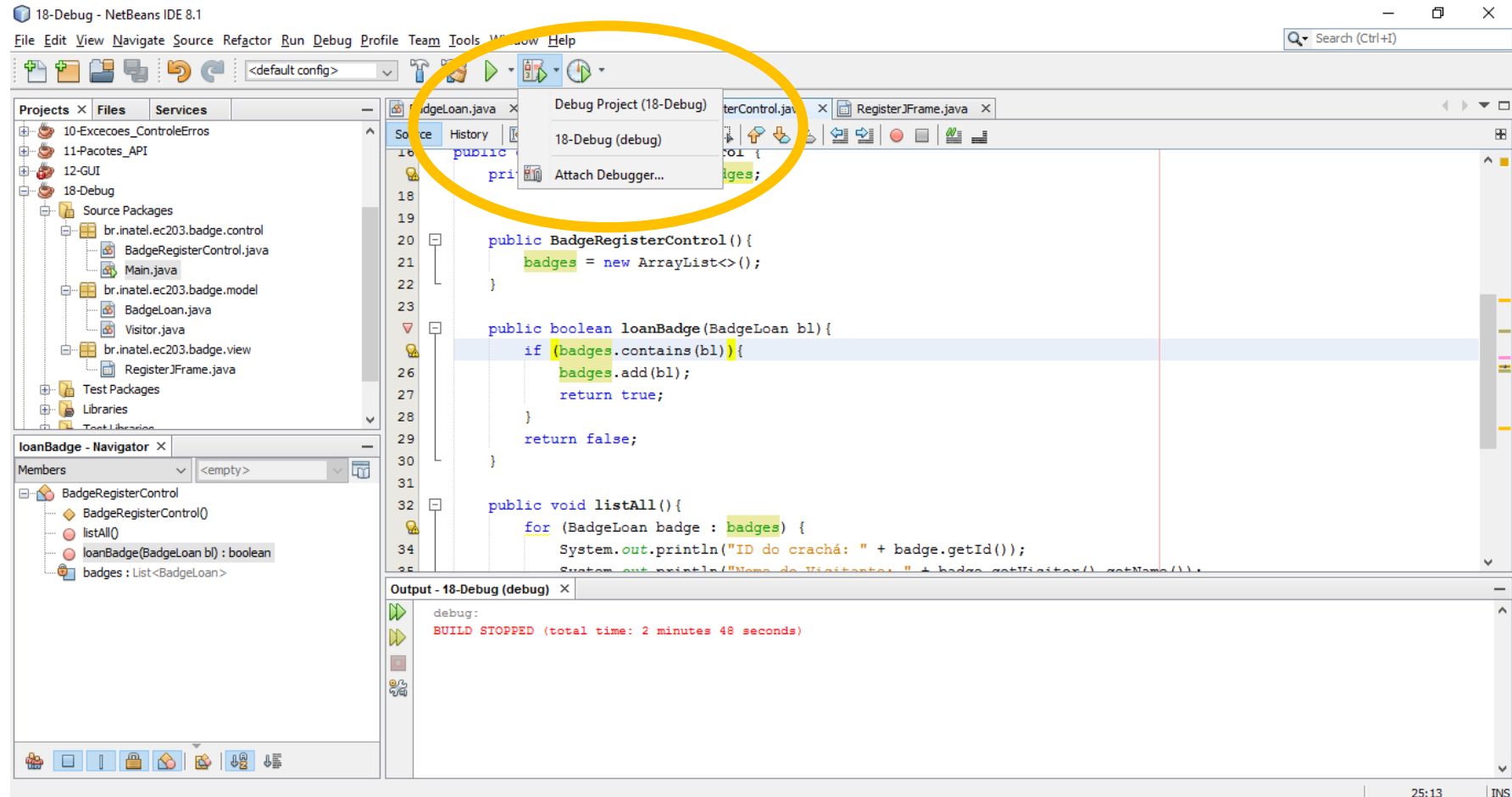
Para podermos localizar um erro precisamos marcar qual o trecho do código queremos analisar, esta marcação é o **breakpoint**.



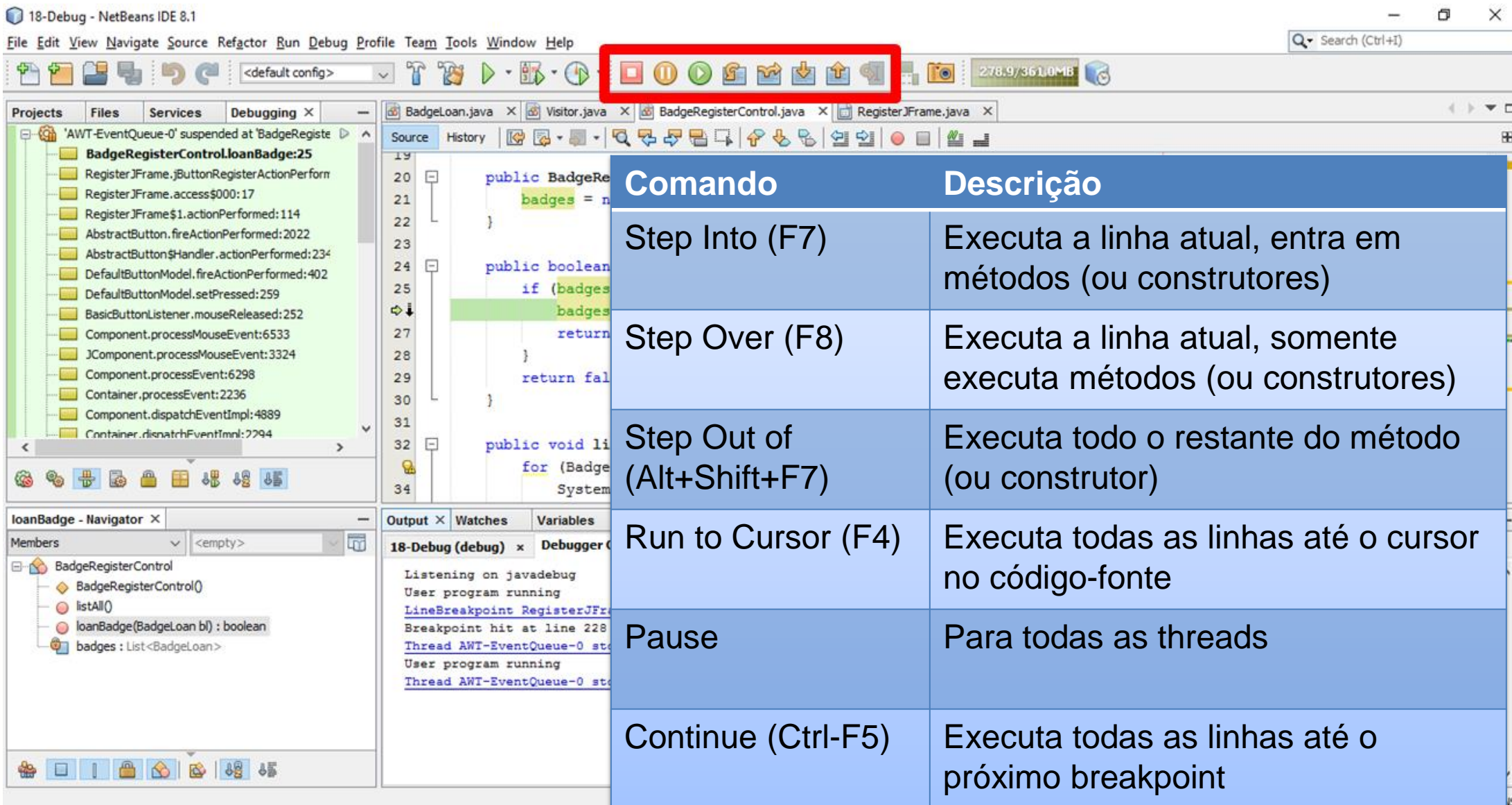
Após adicionarmos um **breakpoint** podemos depurar através do atalho **Ctrl + F5**



Também podemos depurar através do menu debug ou do ícone na tela abaixo



“Steps” comandos



18-Debug - NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

278.9/361.0MB

Projects Files Services Debugging

'AWT-EventQueue-0' suspended at 'BadgeRegisterControl'.

BadgeRegisterControl.loanBadge:25

RegisterJFrame.JButtonRegisterActionPerformed

RegisterJFrame.access\$000:17

RegisterJFrame\$1.actionPerformed:114

AbstractButton.fireActionPerformed:2022

AbstractButton\$Handler.actionPerformed:234

DefaultButtonModel.fireActionPerformed:402

DefaultButtonModel.setPressed:259

BasicButtonListener.mouseReleased:252

Component.processMouseEvent:6533

JComponent.processMouseEvent:3324

Component.processEvent:6298

Container.processEvent:2236

Component.dispatchEventImpl:4889

Container.dispatchEventImpl:7794

loanBadge - Navigator

Members

BadgeRegisterControl

BadgeRegisterControl()

listAll()

loanBadge(BadgeLoan bl): boolean

badges: List<BadgeLoan>

Source History

BadgeLoan.java Visitor.java BadgeRegisterControl.java RegisterJFrame.java

19

20 public BadgeRe

21 badges = n

22 }

23

24 public boolean

25 if (badges

26 badges

27 return

28 }

29 return fal

30 }

31

32 public void li

33 for (Badge

34 System

Output Watches Variables

18-Debug (debug) x Debugger

Listening on javadebug

User program running

LineBreakpoint RegisterJFr

Breakpoint hit at line 228

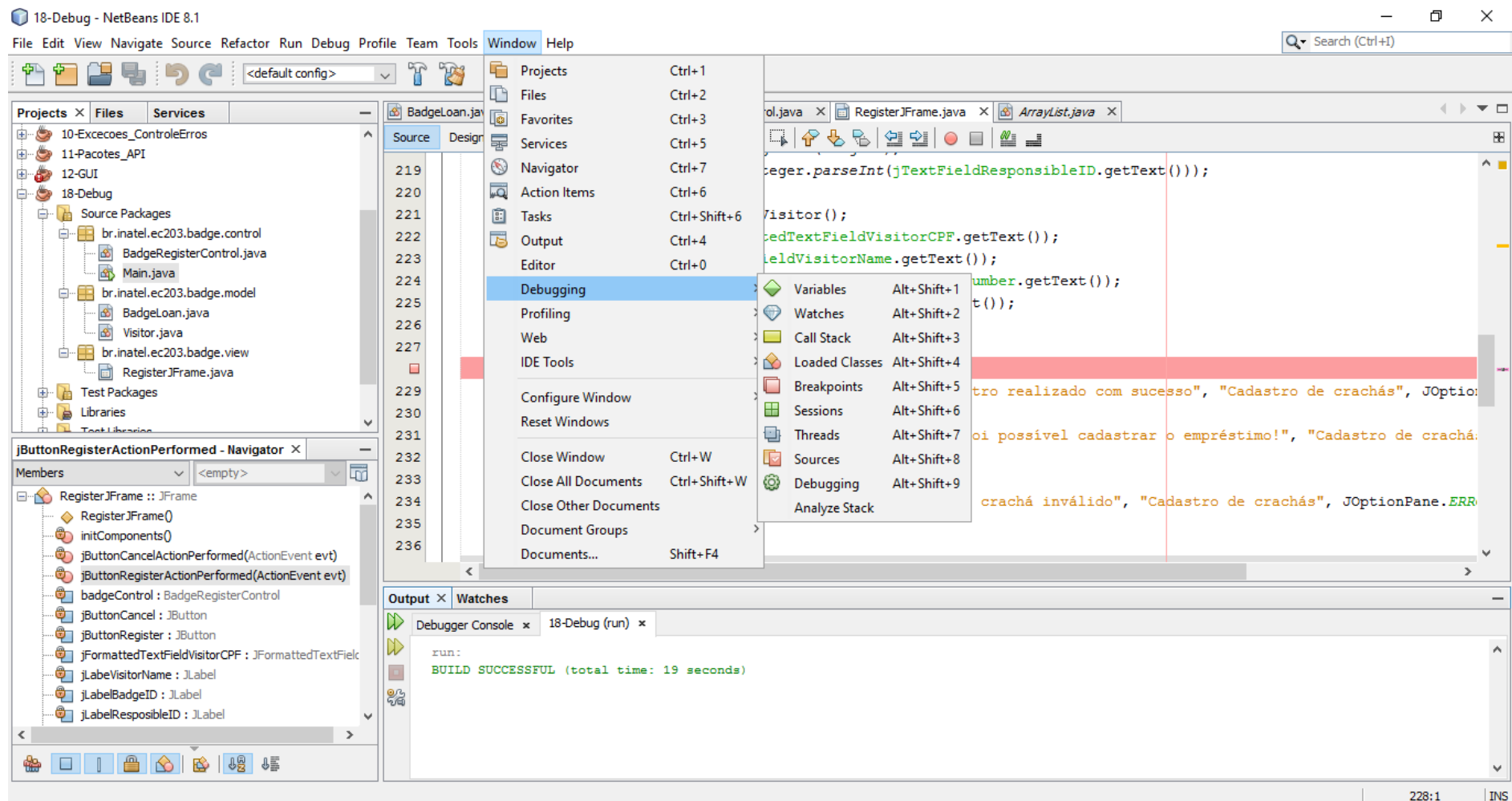
Thread AWT-EventQueue-0 st

User program running

Thread AWT-EventQueue-0 st

Comando	Descrição
Step Into (F7)	Executa a linha atual, entra em métodos (ou construtores)
Step Over (F8)	Executa a linha atual, somente executa métodos (ou construtores)
Step Out of (Alt+Shift+F7)	Executa todo o restante do método (ou construtor)
Run to Cursor (F4)	Executa todas as linhas até o cursor no código-fonte
Pause	Para todas as threads
Continue (Ctrl-F5)	Executa todas as linhas até o próximo breakpoint

Podemos escolher quais janelas visualizar durante a execução do debugger



Localizando um problema de lógica através do Watches

18-Debug - NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config> 220.2/358.5MB

Projects Files Services Debugging X

'AWT-EventQueue-0' suspended at 'BadgeRegisterControl loanBadge:25'

BadgeRegisterControl.loanBadge:25
RegisterJFrame.JButtonRegisterActionPerform
RegisterJFrame.access\$000:17
RegisterJFrame\$1.actionPerformed:114
AbstractButton.fireActionPerformed:2022
AbstractButton\$Handler.actionPerformed:234
DefaultButtonModel.fireActionPerformed:402
DefaultButtonModel.setPressed:259
BasicButtonListener.mouseReleased:252
Component.processMouseEvent:6533
JComponent.processMouseEvent:3324

loanBadge - Navigator X

Members <empty>

BadgeRegisterControl
BadgeRegisterControl()
listAll()
loanBadge(BadgeLoan bl) : boolean
badges : List<BadgeLoan>

```

16 public class BadgeRegisterControl {
17     private List<BadgeLoan> badges;
18
19
20     public BadgeRegisterControl() {
21         badges = new ArrayList<>();
22     }
23
24     public boolean loanBadge(BadgeLoan bl) {
25         if (badges.contains(bl)) {
26             badges.add(bl);
27             return true;
28         }
29         return false;
30     }
31
32     public void listAll() {
33         for (BadgeLoan badge : badges) {
34             System.out.println("ID do crachá: " + badge.getId());
35             System.out.println("Nome do Visitante: " + badge.getVisitor().getName());

```

Output Watches X Variables Evaluation Result Call Stack Breakpoints Threads

Name	Type	Value
badges	ArrayList	"size = 0"
badges.contains(bl)	boolean	false
<Enter new watch>		

18-Debug (debug) running... 26:1 INS

Verificando conteúdo de atributos/variáveis através do Variables

18-Debug - NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

235.6/360.5MB

Projects Files Services Debugging X

'AWT-EventQueue-0' suspended at 'BadgeRegisterControl.loanBadge:25'

- BadgeRegisterControl.loanBadge:25
- RegisterJFrame.JButtonRegisterActionPerformed
- RegisterJFrame.access\$000:17
- RegisterJFrame\$1.actionPerformed:114
- AbstractButton.fireActionPerformed:2022
- AbstractButton\$Handler.actionPerformed:234
- DefaultButtonModel.fireActionPerformed:402
- DefaultButtonModel.setPressed:259
- BasicButtonListener.mouseReleased:252
- Component.processMouseEvent:6533
- JComponent.processMouseEvent:3324

BadgeLoan.java Visitor.java BadgeRegisterControl.java RegisterJFrame.java

Source History

```

19
20 public BadgeRegisterControl() {
21     badges = new ArrayList<>();
22 }
23
24 public boolean loanBadge(BadgeLoan bl) {
25     if (badges.contains(bl)) {
26         badges.add(bl);
27         return true;
28     }
29     return false;
30 }
31
32 public void listAll() {
33     for (BadgeLoan badge : badges) {
34         System.out.println("ID do crachá: " + badge.getId());

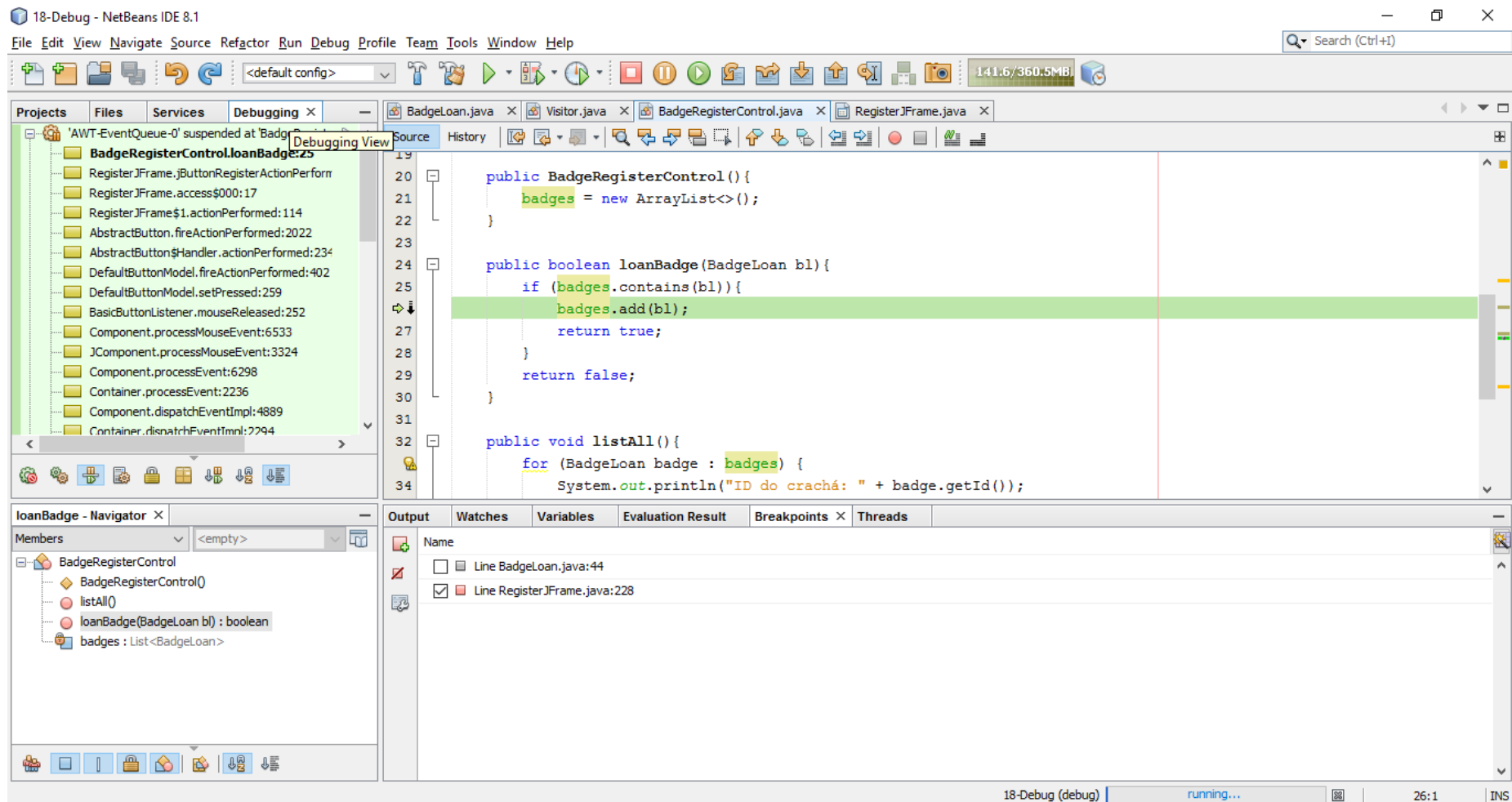
```

Output Watches Variables X Evaluation Result Call Stack Breakpoints Threads

Name	Type	Value
bl	BadgeLoan	#1456
id	int	1
responsibleID	int	2
visitor	Visitor	#1457
cpf	String	"111.111.111-11"
email	String	"email"
name	String	"felipe"
phoneNumber	String	"123"
this	BadgeRegisterControl	#1458

18-Debug (debug) running... 26:1 INS

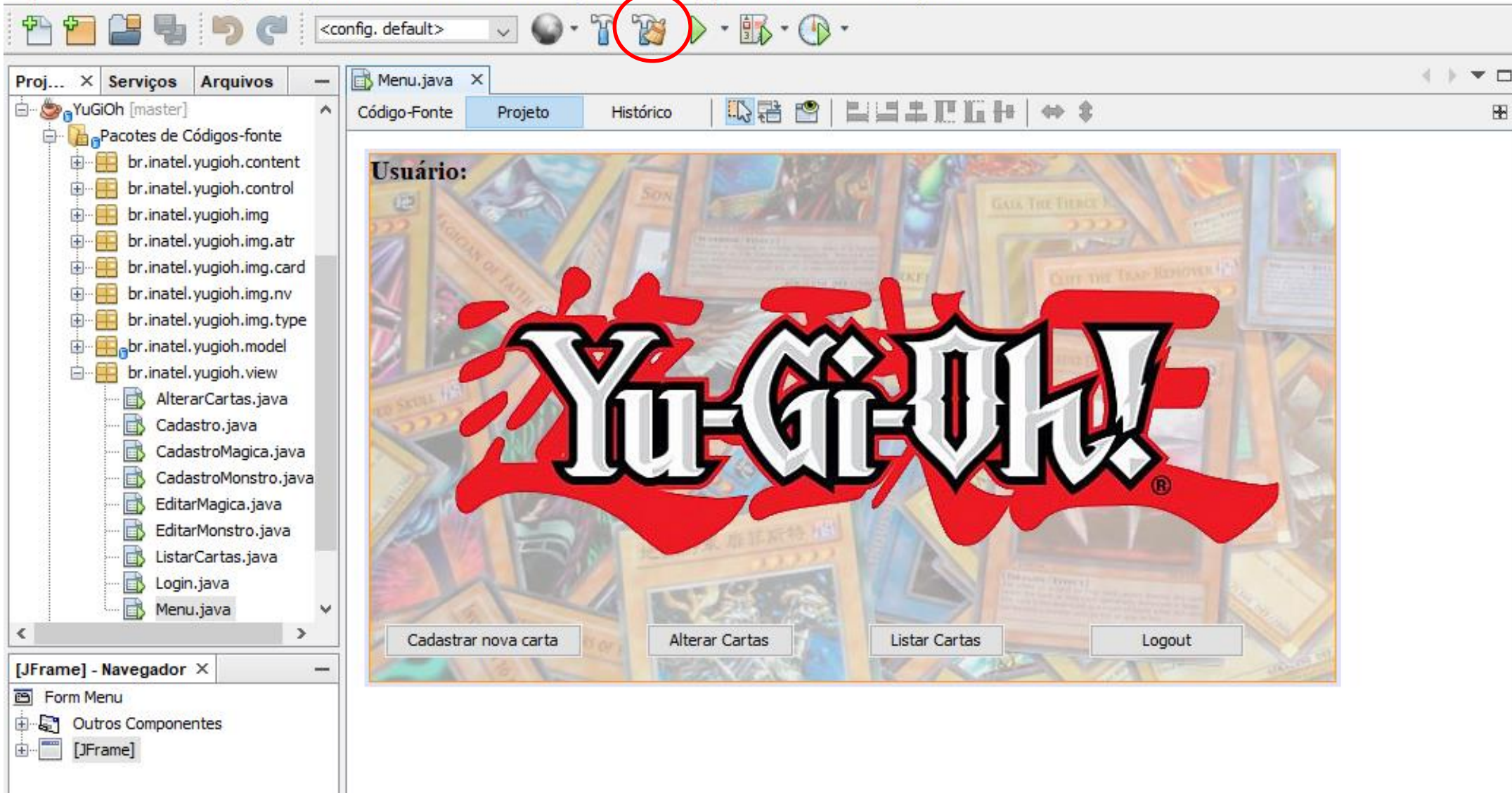
Verificando a pilha de chamada dos métodos através do **Debugging** (ou **Call Stack**)



Gerando arquivo .jar

YuGiOh - NetBeans IDE 8.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda



Gerando arquivo .jar

```
Saída - YuGiOh (clean,jar) X Menu.java X
ant -f C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh -Dnb.internal.action.name=rebuild clean jar
init:
deps-clean:
Updating property file: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build\\built-clean.properties
Deleting directory C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build
clean:
init:
deps-jar:
Created dir: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build
Updating property file: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build\\built-jar.properties
Created dir: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build\\classes
Created dir: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build\\empty
Created dir: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build\\generated-sources\\ap-source-output
Compiling 16 source files to C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build\\classes
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Copying 68 files to C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build\\classes
compile:
Created dir: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\dist
Copying 1 file to C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\build
Copy libraries to C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\dist\\lib.
Building jar: C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\dist\\YuGiOh.jar
To run this application from the command line without Ant, try:
java -jar "C:\\Users\\Samuel\\Documents\\EC205\\Laboratorio\\Projeto\\YuGiOh\\dist\\YuGiOh.jar"
jar:
CONSTRUÍDO COM SUCESSO (tempo total: 2 segundos)
```

Gerando [javadoc](#)

All Classes

Packages

br.inatel.yugioh.control
br.inatel.yugioh.model
br.inatel.yugioh.view

br.inatel.yugioh.model

Interfaces

Efeito
MagicCard

Classes

Carta
CartaMagica
CartaMonstro

OVERVIEWPACKAGECLASSUSETREEDEPRECATEDINDEXHELP

PREVNEXTFRAMESNO FRAMES

Packages

Package	Description
br.inatel.yugioh.control	
br.inatel.yugioh.model	
br.inatel.yugioh.view	

OVERVIEWPACKAGECLASSUSETREEDEPRECATEDINDEXHELP

PREVNEXTFRAMESNO FRAMES

Gerando [javadoc](#)

YuGiOh - NetBeans IDE 8.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

The screenshot displays the NetBeans IDE 8.2 interface. The 'Executar' menu is open, showing various options for running and building the project. The 'Gerar Javadoc (YuGiOh)' option is highlighted. The background shows a YuGiOh application interface with a 'Usuário:' label and buttons for 'Cadastrar nova carta', 'Alterar Cartas', 'Listar Cartas', and 'Logout'.

Executar Projeto (YuGiOh)	F6
Testar Projeto (YuGiOh)	Alt+F6
Construir Projeto (YuGiOh)	F11
Limpar e Construir Projeto (YuGiOh)	Shift+F11
Definir Configuração do Projeto	>
Definir Browser do Projeto	>
Definir Projeto Principal	>
Gerar Javadoc (YuGiOh)	
Executar Arquivo	Shift+F6
Testar Arquivo	Ctrl+F6
Compilar Arquivo	F9
Verificar Arquivo	Alt+F9
Validar Arquivo	Alt+Shift+F9
Repetir Construção/Execução: YuGiOh (javadoc)	Ctrl+F11
Interromper Construção/Execução	

Gerando javadoc

```
* <p>A {@code String} represents a string in the UTF-16 format
* in which <em>supplementary characters</em> are represented by <em>surrogate
* pairs</em> (see the section <a href="Character.html#unicode">Unicode
* Character Representations</a> in the {@code Character} class for
* more information).
* Index values refer to {@code char} code units, so a supplementary
* character uses two positions in a {@code String}.
* <p>The {@code String} class provides methods for dealing with
* Unicode code points (i.e., characters), in addition to those for
* dealing with Unicode code units (i.e., {@code char} values).
*
* @author Lee Boynton
* @author Arthur van Hoff
* @author Martin Buchholz
* @author Ulf Zibis
* @see java.lang.Object#toString()
* @see java.lang.StringBuffer
* @see java.lang.StringBuilder
* @see java.nio.charset.Charset
* @since JDK1.0
*/
```

```
public final class String
```

```
/**
 * Returns the {@code char} value at the
 * specified index. An index ranges from {@code 0} to
 * {@code length() - 1}. The first {@code char} value of the sequence
 * is at index {@code 0}, the next at index {@code 1},
 * and so on, as for array indexing.
 *
 * <p>If the {@code char} value specified by the index is a
 * <a href="Character.html#unicode">surrogate</a>, the surrogate
 * value is returned.
 *
 * @param index the index of the {@code char} value.
 * @return the {@code char} value at the specified index of this string.
 * The first {@code char} value is at index {@code 0}.
 * @exception IndexOutOfBoundsException if the {@code index}
 * argument is negative or not less than the length of this
 * string.
 */
public char charAt(int index) {
```

Gerando [javadoc](#)

Tag	Significado
@author	Especifica o autor da classe ou do método em questão.
@deprecated	Identifica classes ou métodos obsoletos. É interessante informar nessa tag, quais métodos ou classes podem ser usadas como alternativa ao método obsoleto.
@link	Possibilita a definição de um link para um outro documento local ou remoto através de um URL.
@param	Mostra um parâmetro que será passado a um método.
@return	Mostra qual o tipo de retorno de um método.
@see	Possibilita a definição referências de classes ou métodos, que podem ser consultadas para melhor compreender a ideia daquilo que está sendo comentada.
@since	Indica desde quando uma classe ou métodos foi adicionado na aplicação.
@throws	Indica os tipos de exceções que podem ser lançadas por um método.
@version	Informa a versão da classe.

Obrigado!