

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

HUGO DANIEL BOSADA RODRIGUES

TIAGO MELLO

VINICIUS NUNES

THOMAS FIDELIS

LUCAS PRESENDO CANHETE

RELATÓRIO DE TESTE CAIXA BRANCA:

Método `contemCaracterEspecial()` da classe `Senha`

CAMPINAS

2024

1) Defina as classes / partições – entrada e saída

Classes de Entrada:

Senha: String senha recebida para validação

Classes de Saída:

True ou false (Se é válida ou não)

2) Defina os domínios

Classes de Entrada:

1. Senha

Strings com **apenas um caractere especial**:

- Exemplo: "@", "!"

Strings com caracteres normais e **um ou mais caracteres especiais**:

- Exemplo: "senha@123", "!abcDEF", "123#456"

Strings compostas **somente por caracteres especiais**:

- Exemplo: "!@#\$", "*%^&%"

Classes de Saída:

- True ou false

3) Efetue uma análise BVA. Especifique os valores típicos (Válidos e não válidos).

VARIÁVEIS E VALORES RELEVANTES	TIPO DE DADO	PARÂMETRO E/OU VALOR DE REFERÊNCIA	CLASSES - PARTIÇÕES VÁLIDAS	CLASSES - PARTIÇÕES NÃO VÁLIDAS
			Senha	Senha
TODAS AS ENTRADAS: Senha	String	TODAS AS SAÍDAS: True e false	"@123" "senha#valida" "abc!" "*&^%\$"	"senha123" "normal"

VALORES VÁLIDOS:

1. Senha

- "@123" - Contém o caractere especial @ com números.
- "senha#valida" - Contém o caractere especial # com letras.
- "abc!" - Contém o caractere especial ! no final da string.
- "*&^%\$" - Contém apenas caracteres especiais.

VALORES NÃO VÁLIDOS:

1. Senha

- "senha123" - Não contém nenhum caractere especial.
- "normal" - Apenas letras, sem caracteres especiais.

4) Defina o dataset para testes.

1. Senha

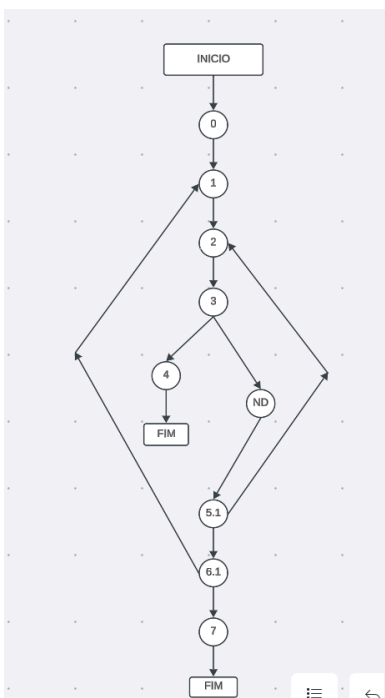
Valores Válidos Seleccionados Após Exclusão

1. "@123" - **Mantido**: Combina caracteres especiais e numéricos, sendo um exemplo típico de caso válido.
2. "*&^%\$" - **Mantido**: É composto apenas por caracteres especiais, o que valida o método para esse tipo de string.

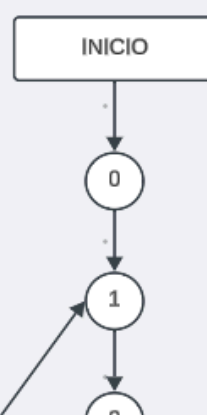
Valores Válidos Excluídos e Justificativas

1. "senha#valida" - **Excluído**: Já temos um exemplo ("@123") que combina caracteres especiais com texto alfanumérico, então este é redundante.
2. "abc!" - **Excluído**: Testa caracteres especiais no final da string, mas já validamos casos de posição com "*&^%\$".

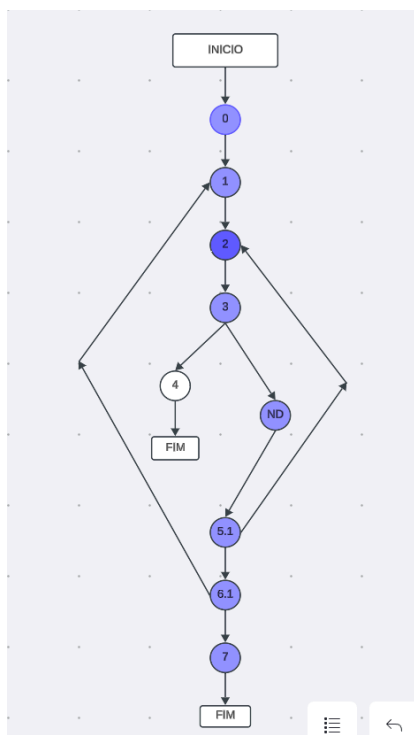
5) Definição do(s) GFC(s) - Grafo de Fluxo de Controle



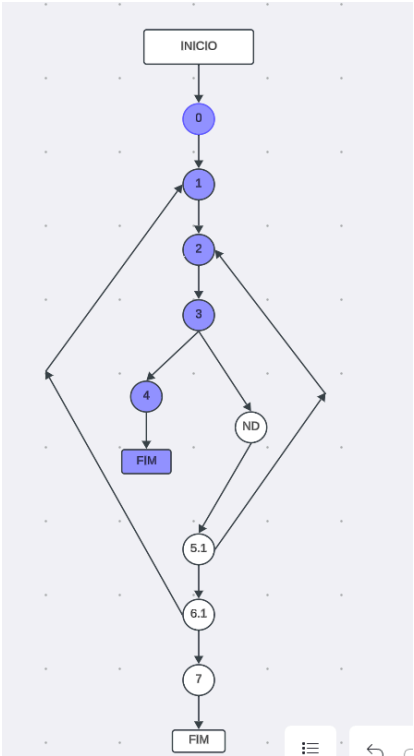
```
0 public boolean contemCaracterEspecial(){
0     char[] caracteresEspeciais = {'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '=',
0                                     '+', '[', ']', '{', '}', '|', ':', ';', '"', '<', '>', ',', '.',
0                                     '?', '/'};
1     for(int i = 0; i < this.senha.length(); i++){
2         for(int j = 0; j < caracteresEspeciais.length; j++){
3             if(this.senha.charAt(i) == caracteresEspeciais[j])
4                 return true;
5.1         }
6.1     }
7     return false;
}
```



6) Especificação dos fluxos do GFC



```
0 public boolean contemCaracterEspecial(){
0     char[] caracteresEspeciais = {'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '=',
0         '+', '[', ']', '{', '}', '|', ':', ';', '"', '<', '>', ',', '.',
0         '?', '/'};
1     for(int i = 0; i < this.senha.length(); i++){
2         for(int j = 0; j < caracteresEspeciais.length; j++){
3             if(this.senha.charAt(i) == caracteresEspeciais[j])
4                 return true;
5.1         }
6.1     }
7     return false;
}
```



```
0 public boolean contemCaracterEspecial(){
0     char[] caracteresEspeciais = {'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '=',
0         '+', '[', ']', '{', '}', '|', ':', ';', '"', '<', '>', ',', '.', '/',
0         '?', '\\'};
1     for(int i = 0; i < this.senha.length(); i++){
2         for(int j = 0; j < caracteresEspeciais.length; j++){
3             if(this.senha.charAt(i) == caracteresEspeciais[j])
4                 return true;
5.1         }
5.1     }
6.1     return false;
7 }
```

7) Protocolos de testes incorporando os GFCs

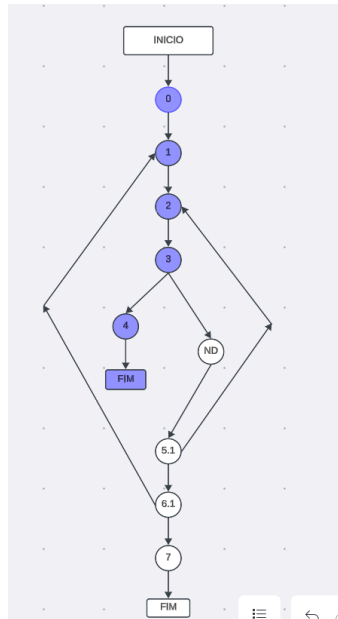
NOME DO TESTE:	Teste de Software		NÚMERO:	TV001
RESPONSÁVEL:	Lucas			
MÓDULO:				
OBJETIVOS:	Os testes com valores válidos têm como objetivo principal verificar o comportamento do software.			
DATA DE CRIAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas	
DATA DA ÚLTIMA ALTERAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas	
DESCRIÇÃO:				

Teste para intervalo de valores válidos para senha. Deve reootrnar: return true

Indicar Valores de Entrada

- **Senha: “@123”**

FLUXO GFC: 0 1 2 3 ND 5.1 6.1 4 FIM



```
0 public boolean contemCaracterEspecial(){
0     char[] caracteresEspeciais = {'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '=',
0         '+', '[', ']', '{', '}', '|', ':', ';', '"', '<', '>', ',', '.',
0         '?', '/'};
1     for(int i = 0; i < this.senha.length(); i++){
2         for(int j = 0; j < caracteresEspeciais.length; j++){
3             if(this.senha.charAt(i) == caracteresEspeciais[j])
4                 return true;
5.1         }
6.1     }
7     return false;
}
```

FLUXO DE EXECUÇÃO DO TESTE:

- **Abrir o VSCode.**
- **No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.**
- **Abrir um terminal com o atalho “ctrl + `”.**
- **Para rodar o programa, no terminal digite: “java SenhaTeste.java”.**
- **Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“@123”).**
- **Em seguida, o programa retorna true ou false se a senha é válida.**

OBSERVAÇÕES:

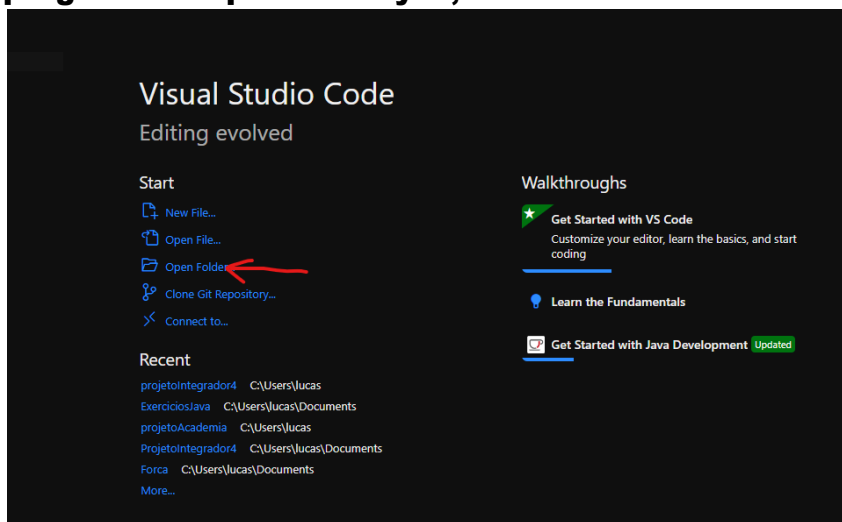
Nenhuma observação.

Fluxo de execução do teste TV001:

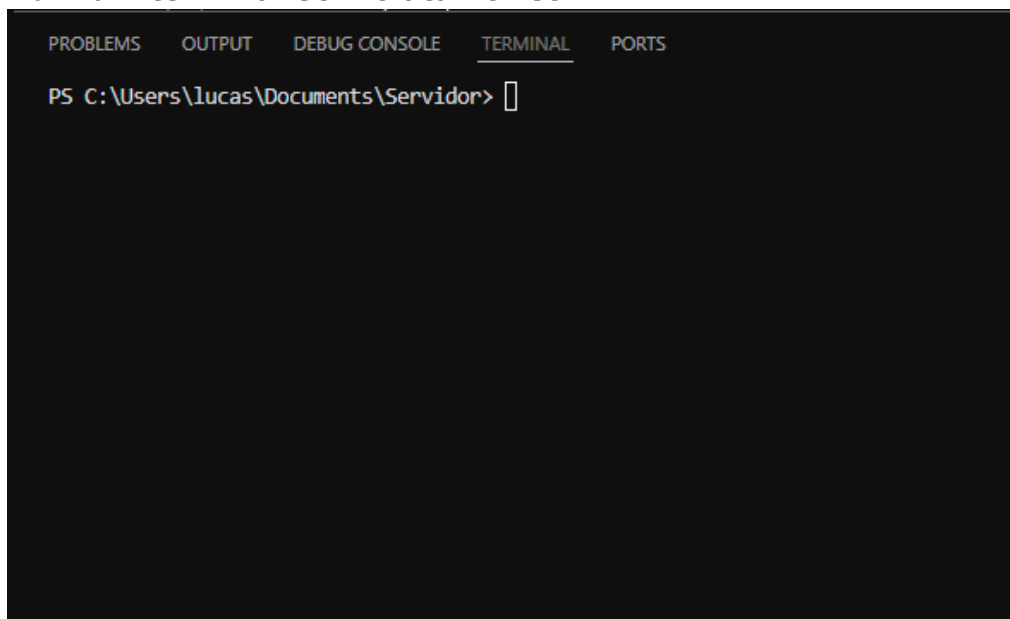
Valores de entrada:

senha: “@123”

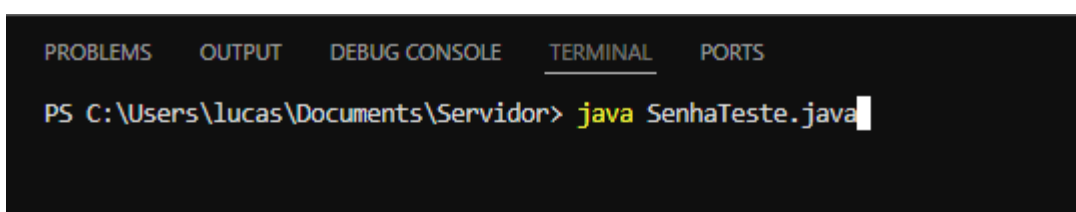
- **Abrir o VSCode.**
- **No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.**



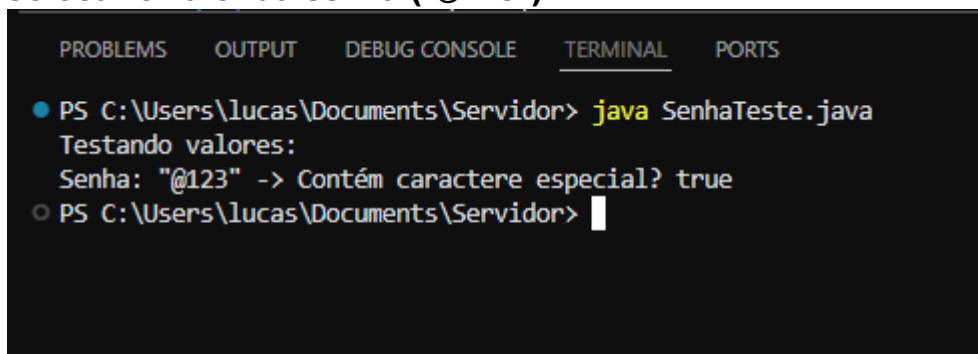
- **Abrir um terminal com o atalho “ctrl + `”.**



- **Para rodar o programa, no terminal digite: “java SenhaTeste.java”.**



- **Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“@123”).**



The screenshot shows a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. Below the tabs, the terminal shows the following text:

```
PS C:\Users\lucas\Documents\Servidor> java SenhaTeste.java
Testando valores:
Senha: "@123" -> Contém caractere especial? true
PS C:\Users\lucas\Documents\Servidor> 
```

- **Em seguida, o programa retorna true ou false se a senha é válida.**

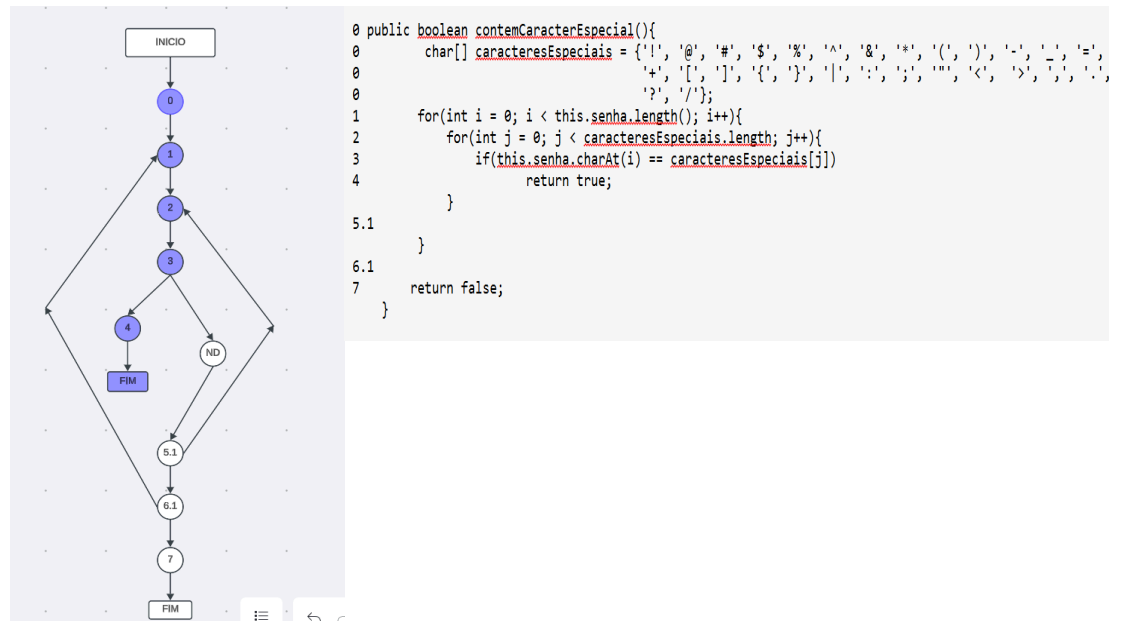
NOME DO TESTE:	Teste de Software	NÚMERO:	TV002
RESPONSÁVEL:	Lucas		
MÓDULO:			
OBJETIVOS:	Os testes com valores válidos têm como objetivo principal verificar o comportamento do software.		
DATA DE CRIAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas
DATA DA ÚLTIMA ALTERAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas
DESCRIÇÃO:			

Teste para intervalo de valores válidos para senha. Deve retornar: return true

Indicar Valores de Entrada

- **Senha: “* & ^ % \$”**

FLUXO GFC: 0 1 2 3 ND 5.1 6.1 4 FIM



FLUXO DE EXECUÇÃO DO TESTE:

- **Abrir o VSCode.**
- **No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.**
- **Abrir um terminal com o atalho “ctrl + `”.**
- **Para rodar o programa, no terminal digite: “java SenhaTeste.java”.**
- **Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“* & ^ % \$”).**
- **Em seguida, o programa retorna true ou false se a senha é válida.**

OBSERVAÇÕES:

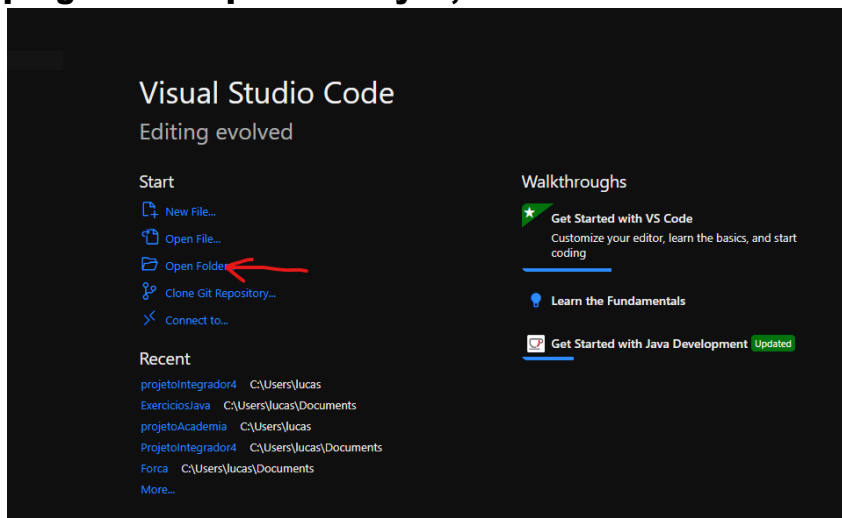
Nenhuma observação.

Fluxo de execução do teste TV002:

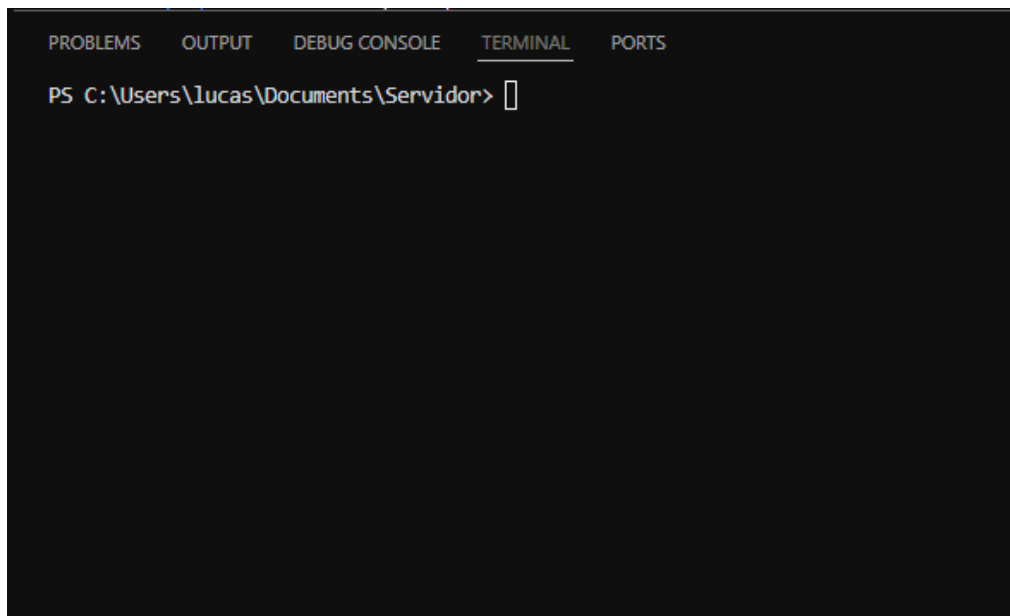
Valores de entrada:

senha: “* & ^ % \$”

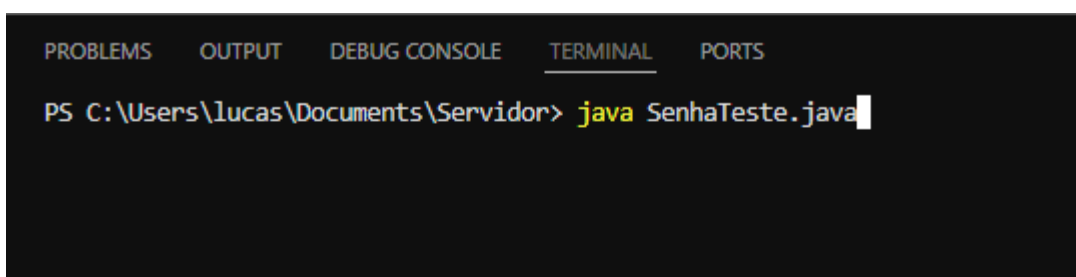
- **Abrir o VSCode.**
- **No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.**



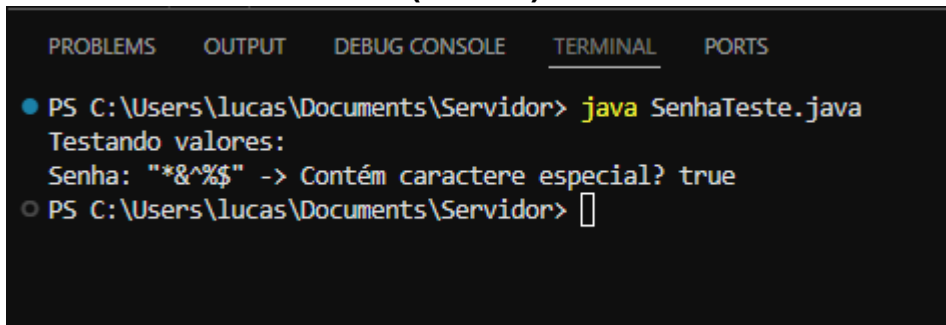
- **Abrir um terminal com o atalho “ctrl + `”.**



- **Para rodar o programa, no terminal digite: “java SenhaTeste.java”.**



- **Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“@123”).**



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS C:\Users\lucas\Documents\Servidor> java SenhaTeste.java
Testando valores:
Senha: "*&^%$" -> Contém caractere especial? true
○ PS C:\Users\lucas\Documents\Servidor> 
```

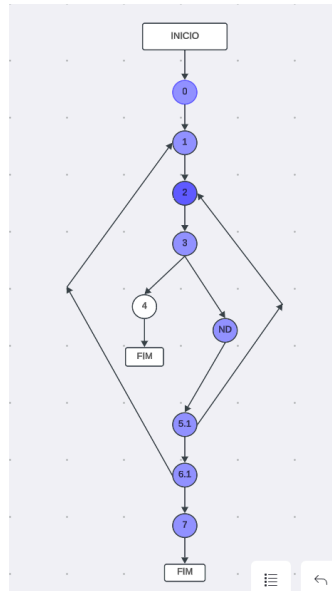
- **Em seguida, o programa retorna true ou false se a senha é válida.**

NOME DO TESTE:	Teste de Software	NÚMERO:	TNV001
RESPONSÁVEL:	Lucas		
MÓDULO:			
OBJETIVOS:	Os testes com valores não válidos têm como objetivo principal verificar o comportamento do software.		
DATA DE CRIAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas
DATA DA ÚLTIMA ALTERAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas
DESCRIÇÃO:	Teste para intervalo de valores válidos para senha. Deve retornar: return false		

Indicar Valores de Entrada

- **Senha: “senha123”**

FLUXO GFC: 0 1 2 3 ND 5.1 6.1 7 FIM



```
0 public boolean contemCaracterEspecial(){
0     char[] caracteresEspeciais = {'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '=',
0         '+', '[', ']', '{', '}', '|', ':', ';', '"', '<', '>', ',', '.',
0         '?', '/'};
1     for(int i = 0; i < this.senha.length(); i++){
2         for(int j = 0; j < caracteresEspeciais.length; j++){
3             if(this.senha.charAt(i) == caracteresEspeciais[j])
4                 return true;
5.1         }
6.1     }
7     return false;
}
```

FLUXO DE EXECUÇÃO DO TESTE:

- **Abrir o VSCode.**
- **No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.**
- **Abrir um terminal com o atalho “ctrl + `”.**
- **Para rodar o programa, no terminal digite: “java SenhaTeste.java”.**
- **Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“senha123”).**
- **Em seguida, o programa retorna true ou false se a senha é válida.**

OBSERVAÇÕES:

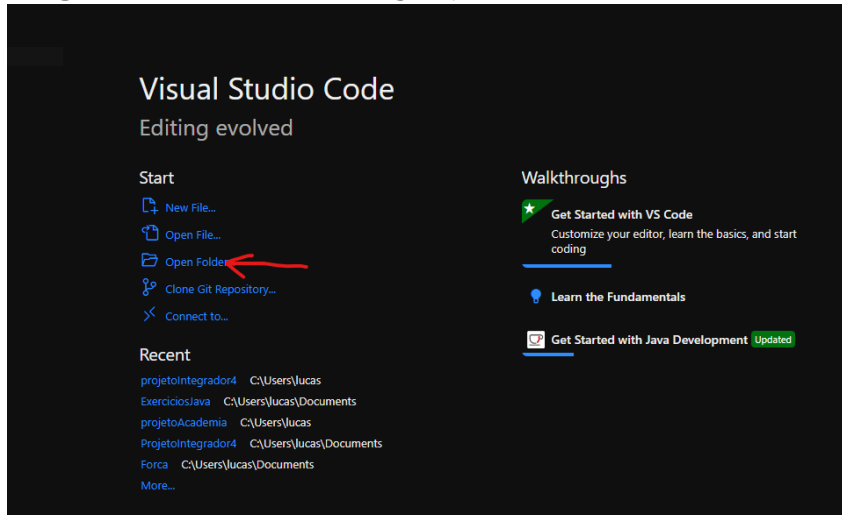
Nenhuma observação.

Fluxo de execução do teste TNV001:

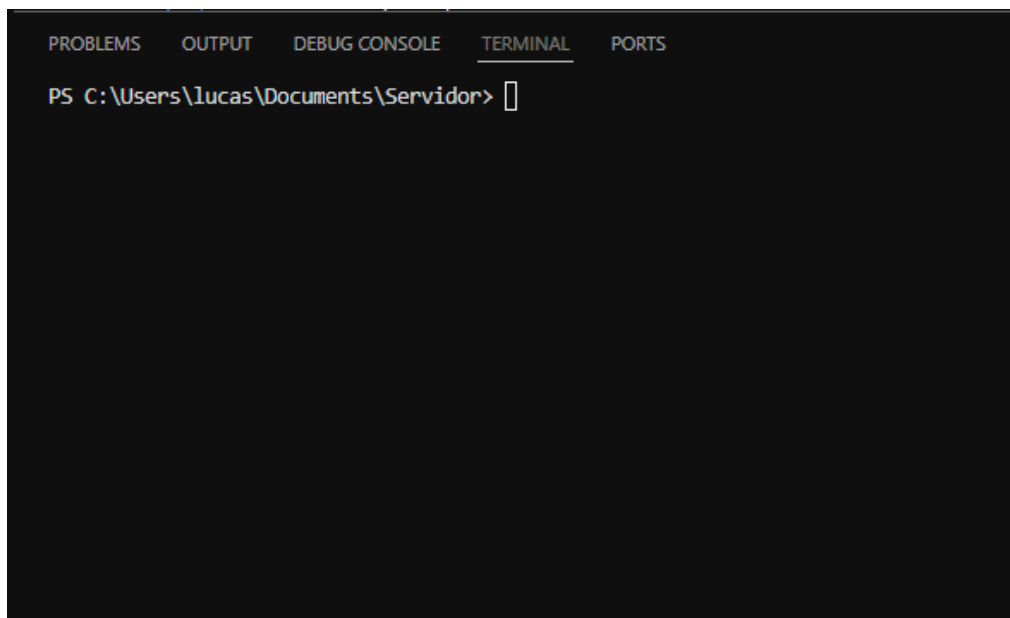
Valores de entrada:

senha: “senha123”

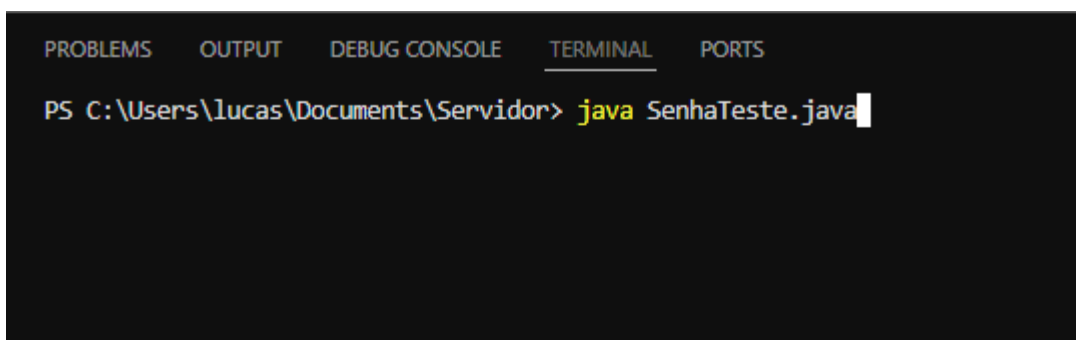
- **Abrir o VSCode.**
- **No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.**



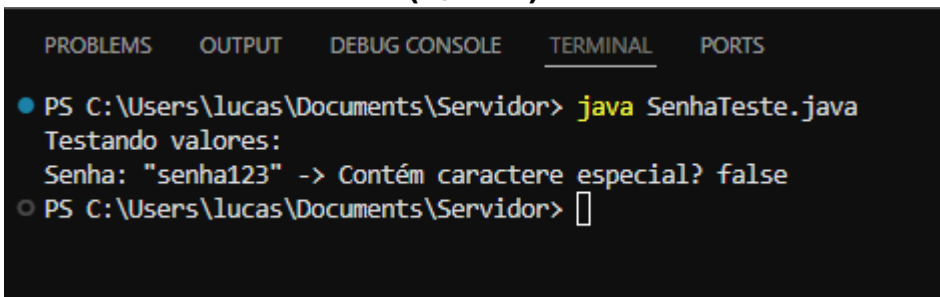
- **Abrir um terminal com o atalho “ctrl + `”.**



- **Para rodar o programa, no terminal digite: “java SenhaTeste.java”.**



- **Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“@123”).**



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS C:\Users\lucas\Documents\Servidor> java SenhaTeste.java
Testando valores:
Senha: "senha123" -> Contém caractere especial? false
○ PS C:\Users\lucas\Documents\Servidor> 
```

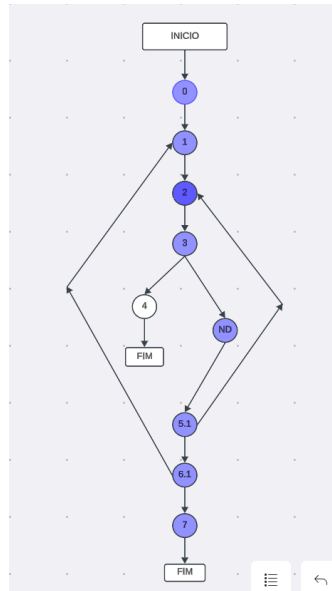
- **Em seguida, o programa retorna true ou false se a senha é válida.**

NOME DO TESTE:	Teste de Software	NÚMERO:	TNV002
RESPONSÁVEL:	Lucas		
MÓDULO:			
OBJETIVOS:	Os testes com valores não válidos têm como objetivo principal verificar o comportamento do software.		
DATA DE CRIAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas
DATA DA ÚLTIMA ALTERAÇÃO:	25/11/2024	RESPONSÁVEL:	Lucas
DESCRIÇÃO:	Teste para intervalo de valores válidos para senha. Deve retornar: return false		

Indicar Valores de Entrada

- Senha: “normal”

FLUXO GFC: 0 1 2 3 ND 5.1 6.1 7 FIM



```
0 public boolean contemCaracterEspecial(){
0     char[] caracteresEspeciais = {'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '_', '=',
0                                     '+', '[', ']', '{', '}', '|', ':', ';', '"', '<', '>', ',', '.',
0                                     '?', '/'};
1     for(int i = 0; i < this.senha.length(); i++){
2         for(int j = 0; j < caracteresEspeciais.length; j++){
3             if(this.senha.charAt(i) == caracteresEspeciais[j])
4                 return true;
5.1         }
6.1     }
7     return false;
}
```

FLUXO DE EXECUÇÃO DO TESTE:

- Abrir o VSCode.
- No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.
- Abrir um terminal com o atalho “ctrl + ”.
- Para rodar o programa, no terminal digite: “java SenhaTeste.java”.
- Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“normal”).
- Em seguida, o programa retorna true ou false se a senha é válida.

OBSERVAÇÕES:

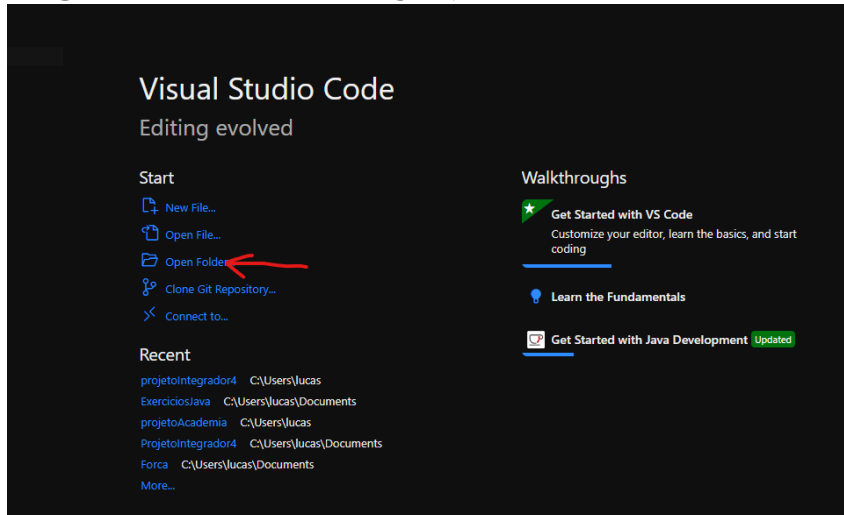
Nenhuma observação.

Fluxo de execução do teste TNV002:

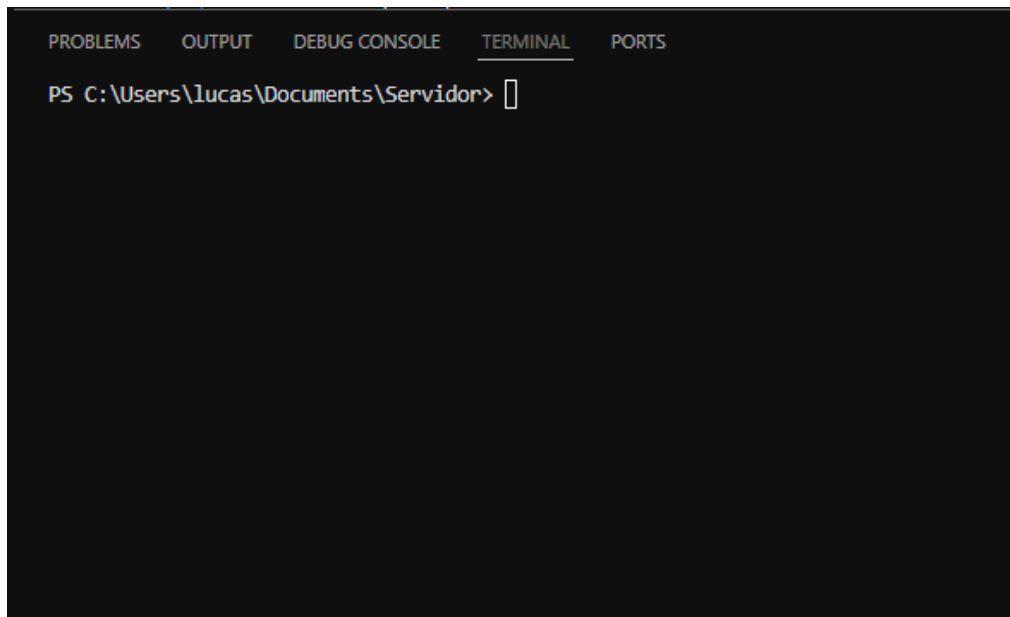
Valores de entrada:

senha: “normal”

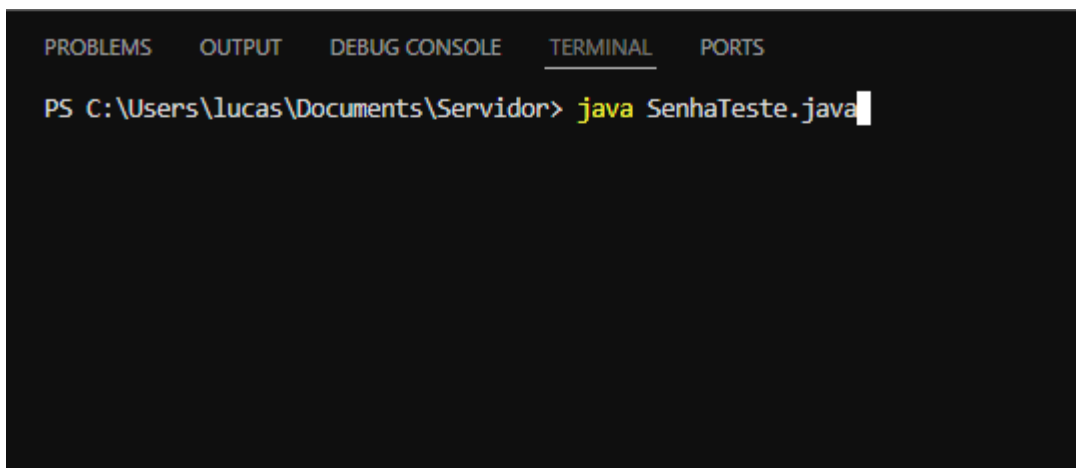
- **Abrir o VSCode.**
- **No menu localizado no canto superior esquerdo selecionar file -> Open folder... -> Selecionar a pasta que está o código fonte do programa -> aperte em “yes, I trust the authors”.**



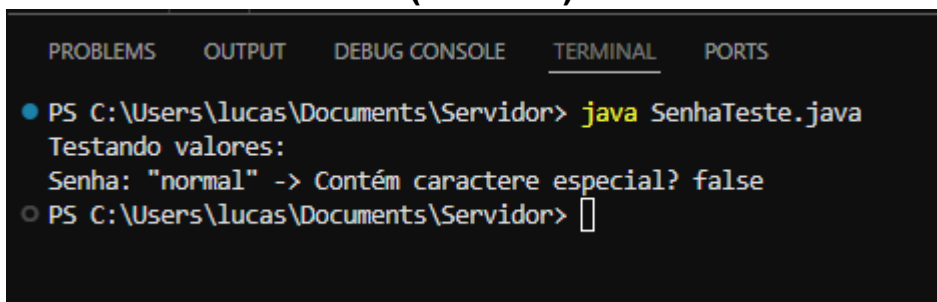
- **Abrir um terminal com o atalho “ctrl + `”.**



- **Para rodar o programa, no terminal digite: “java SenhaTeste.java”.**



- **Rodando o programa, a primeira informação que ele pede é a senha: Colocar o valor da senha (“normal”).**



The screenshot shows a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. The terminal content shows a command prompt where the user has run 'java SenhaTeste.java'. The program outputs 'Testando valores:' followed by a prompt for a password. The user has entered 'normal', and the program has responded with 'Senha: "normal" -> Contém caractere especial? false'. The prompt is now ready for the next command.

```
● PS C:\Users\lucas\Documents\Servidor> java SenhaTeste.java
Testando valores:
Senha: "normal" -> Contém caractere especial? false
○ PS C:\Users\lucas\Documents\Servidor> 
```

- **Em seguida, o programa retorna true ou false se a senha é válida.**