


Commit

 This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Mensagem de commit descrevendo as alterações

[Browse files](#)

 hannaranielly committed on Jun 26

1 parent [56b07a0](#) commit [2cb1406](#)

Showing 7 changed files with 106 additions and 43 deletions.

[Whitespace](#)[Ignore whitespace](#)[Split](#)[Unified](#)

58


...ema-de-Gestao-Consultas/src/br/edu/ufersa/controlConsult/gui/ConfirmarDataDaConsulta.java

```
285      285      }
286      286
287      287      private void confirmar(HorarioAtendimento horario, Date agendada) {
288          -      int numeroConsultas = 3;//O número de consultas que o médico tem marcadas
naquele horário
289          -      if (numeroConsultas < calcula(horario)) {
290              -      String mensagem;
291              -      mensagem = "" + (numeroConsultas + 1) + "-esima pessoa na fila";
292              -      /*
293              -      ConsultaValida tela = new
ConsultaValida(horario.getInicio(),horario.getFim(),medico,agendada,mensagem);
294              -      tela.setVisible(true);
295              -      tela.setLocationRelativeTo(null);
296              -      this.dispose();*/
288          +      int numeroConsultas = getNumeroConsultasMarcadas(horario); // O número de
consultas que o médico tem marcadas naquele horário
289          +      int limiteConsultas = calcula(horario);
290          +      if (numeroConsultas < limiteConsultas) {
291              +      String mensagem = (numeroConsultas + 1) + "-ésima pessoa na fila";
292              +      exibirConsultaValida(horario, agendada, mensagem);
297          293      } else {
298          294      String erro = "Horário lotado";
299          -      /*
300          -      ConsultaInvalida tela = new ConsultaInvalida(medico,erro);
```

```

301 -         tela.setVisible(true);
302 -         tela.setLocationRelativeTo(null);
303 -         this.dispose();*/
295 +         exibirConsultaInvalida(erro);
304 296     }
305 297 }
306 298
299 + private int getNumeroConsultasMarcadas(HorarioAtendimento horario) {
300 +     // Lógica para obter o número de consultas marcadas
301 +     // Retorne o valor correto
302 +     return 3;
303 + }
304 +
305 +
306 + private void exibirConsultaValida(HorarioAtendimento horario, Date agendada,
String
307 +     mensagem) {
308 +     // Exibir tela de consulta válida com os parâmetros informados
309 +     /*
310 +     ConsultaValida tela = new
311 +     ConsultaValida(horario.getInicio(),horario.getFim(),medico,agendada,mensagem);
312 +     tela.setVisible(true);
313 +     tela.setLocationRelativeTo(null);
314 +     this.dispose();
315 +     */
316 + }
317 +
318 +
319 + private void exibirConsultaInvalida(String erro) {
320 +     // Exibir tela de consulta inválida com a mensagem de erro informada
321 +     /*
322 +     ConsultaInvalida tela = new ConsultaInvalida(medico, erro);
323 +     tela.setVisible(true);
324 +     tela.setLocationRelativeTo(null);
325 +     this.dispose();
326 +     */
327 + }
328 +
329 + private static final int CONSULTAS_POR_HORA = 6;
330 +
307 331 private int calcula(HorarioAtendimento horario) {
308 -     return ((horario.getInicio().getHours()) - (horario.getFim().getHours())) *
6;
332 +     int horasInicio = horario.getInicio().getHours();
333 +     int horasFim = horario.getFim().getHours();
334 +     int diferencaHoras = horasInicio - horasFim;
335 +     int consultas = diferencaHoras * CONSULTAS_POR_HORA;
336 +     return consultas;
309 337 }
310 338
311 339 }

```

 29 

 ...-de-Gestao-Consultas/src/br/edu/ufersa/controlConsult/gui/avaliacao/MostraDesempenho.java 


```

44      tempoMedio();
45  }
46
47  +     private static final int MINUTOS_POR_HORA = 60;
48  +     private static final int MILISEGUNDOS_POR_MINUTO = 1000 * 60;
49  +     private static final int MILISEGUNDOS_POR_HORA = MILISEGUNDOS_POR_MINUTO *
MINUTOS_POR_HORA;
50  +
51  private void tempoMedio() {
52      List<Consulta> consultas = findporMedico(pessoa.getMedico());
53  -     long tempo_total = 0L;
54  +     long tempoTotal = 0L;
55  +
56  +     for (Consulta c : consultas) {
57  -         tempo_total = tempo_total + (c.getData_fim().getTime() -
c.getData_inicio().getTime());
58  +         tempoTotal += c.getData_fim().getTime() - c.getData_inicio().getTime();
59  +     }
60  +     if (consultas.isEmpty()) {
61  +         mt.setText("0 minutos");
62  +     } else {
63  -         long tempo_medio = tempo_total / consultas.size();
64  -         long diffMinutesmedio = tempo_medio / (60 * 1000) % 60;
65  -         mt.setText(valueOf(diffMinutesmedio) + " minutos");
66  +         long tempoMedio = tempoTotal / consultas.size();
67  +         long diffMinutesMedio = tempoMedio / MILISEGUNDOS_POR_MINUTO %
MINUTOS_POR_HORA;
68  +         mt.setText(String.valueOf(diffMinutesMedio) + " minutos");
69  +     }
70  +
71  +     List<Frequencia> frequencias = porMedico(pessoa.getMedico());
72  -     tempo_total = 0L;
73  +     tempoTotal = 0L;
74  +
75  +     for (Frequencia f : frequencias) {
76  -         tempo_total = tempo_total + (f.getData_saida().getTime() -
f.getData_entrada().getTime());
77  +         tempoTotal += f.getData_saida().getTime() -
f.getData_entrada().getTime();
78  +     }
79  +
80  +     long diffHoras = tempoTotal / MILISEGUNDOS_POR_HORA;
81  +     ht.setText(String.valueOf(diffHoras) + " horas");
82  + }
83  -
84  -     long diffHoras = tempo_total / (60 * 60 * 1000);
85  -     ht.setText(valueOf(diffHoras) + " horas");
86  -
87  - }

```

69	77	+	
70	78		
71	79	/**	
	80		* This method is called from within the constructor to initialize the form.

21  


Sistema-de-Gestao-Consultas/src/br/edu/ufersa/controlConsult/model/Medico.java 

```

116      116      return listaHorario;
117      117      }
118      118
119      119      +
120      120      + private static final int LIMITE_CARGA_HORARIA = 60;
121      121      +
119      122      public void setCargaHoraria(Integer cargaHoraria)
120      123      throws CampoObrigatorioException, CampoInvalidoException {
121      124      if (cargaHoraria > 0) {
122      125      - if (cargaHoraria <= 60) {
123      126      + if (cargaHoraria <= LIMITE_CARGA_HORARIA) {
124      127      this.cargaHoraria = cargaHoraria;
125      128      } else {
126      129      - throw new CampoInvalidoException("Carga horária excedeu o limite de
127      130      60 horas.");
128      131      + throw new CampoInvalidoException("Carga horária excedeu o limite
129      132      de" + LIMITE_CARGA_HORARIA + "horas.");
130      133      }
131      134      } else {
132      135      throw new CampoObrigatorioException("Campo Carga Horária é
133      136      obrigatório");
134      137
135      138      return crm;
136      139      }
137      140
138      141      + private static final int MAX_CARACTERES_CRM = 15;
139      142      +
140      143      public void setCrm(String crm) throws CampoObrigatorioException,
141      144      CampoLimiteStringException {
142      145      - if (crm != null && crm.length() > 0) {
143      146      - if (crm.length() > 15) {
144      147      - throw new CampoLimiteStringException("CRM excedeu o máximo de
145      148      caracteres.");
146      149      +
147      150      + if (crm != null && !crm.isEmpty()) {
148      151      + if (crm.length() <= MAX_CARACTERES_CRM) {
149      152      + this.crm = crm;
150      153      + } else {
151      154      + throw new CampoLimiteStringException("Campo CRM excedeu o limite de
152      155      caracteres.");
153      156      }
154      157      - this.crm = crm;
155      158      } else {
156      159      - throw new CampoObrigatorioException("Campo CRM é obrigatório.");
157      160


```

	275	+	<code>throw new CampoObrigatorioException("Campo CRM é obrigatório");</code>
269	276		<code>}</code>
270	277		<code>}</code>
271	278		

6  Sistema-de-Gestao-Consultas/src/br/edu/uferasa/controlConsult/model/Pessoa.java 

351	351		<code>}</code>
352	352		
353	353		<code>public void setEmail(String email) throws CampoLimiteStringException {</code>
354	-		<code>if (email != null) {</code>
355	-		<code>if (email.length() > 50) {</code>
	354	+	<code>if (email != null && !email.isEmpty()) {</code>
	355	+	<code>final int MAX_EMAIL_LENGTH = 50;</code>
	356	+	<code>if (email.length() > MAX_EMAIL_LENGTH) {</code>
356	357		<code>throw new CampoLimiteStringException("E-mail excedeu máximo de</code>
			<code>caracteres.");</code>
357	358		<code>}</code>
358	359		<code>}</code>
359	360		<code>this.email = email;</code>
360	361		<code>}</code>
361	362		
	363	+	
362	364		<code>public int getNumCasa() {</code>
363	365		<code>return numCasa;</code>
364	366		<code>}</code>

5 

Sistema-de-Gestao-Consultas/src/br/edu/uferasa/controlConsult/model/Usuario.java 

97	97		<code>return instance.pegar_Banco(usuario);</code>
98	98		<code>}</code>
99	99		
	100	+	<code>private static final int MAX_PASSWORD_LENGTH = 32;</code>
	101	+	
100	102		<code>public void setPassword(char[] password_raw) throws IllegalArgumentException {</code>
101	103		<code>char[] password_hash = Criptografia.criptografar(password_raw);</code>
102	-		<code>if (password_hash.length > 32) {</code>
	104	+	<code>int password_length = password_hash.length;</code>
	105	+	<code>if (password_length > MAX_PASSWORD_LENGTH) {</code>
103	106		<code>throw new IllegalArgumentException("Senha maior que 32 caracteres.");</code>
104	107		<code>}</code>
105	108		<code>this.password = password_hash;</code>

19 

Sistema-de-Gestao-Consultas/src/br/edu/uferasa/controlConsult/model/validacao/Cpf_Util.java 

70	70		<code>return true;</code>
71	71		<code>}</code>
72	72		

```

73 +     private static final int POSICAO_PRIMEIRO_PONTO = 3;
74 +     private static final int POSICAO_SEGUNDO_PONTO = 6;
75 +     private static final int POSICAO_TRACO = 9;
76 +     private static final int TAMANHO_CPF = 11;
77 +
73 78     public static String imprimeCPF(String CPF) {
74     -         return (CPF.substring(0, 3) + "." + CPF.substring(3, 6) + "."
75     -             + CPF.substring(6, 9) + "-" + CPF.substring(9, 11));
76
77 79 +
80 +         if (CPF.length() != TAMANHO_CPF) {
81 +             throw new IllegalArgumentException("CPF inválido: tamanho incorreto");
82 +         }
83 +
84 +         StringBuilder sb = new StringBuilder();
85 +         sb.append(CPF.substring(0, POSICAO_PRIMEIRO_PONTO)).append(".");
86 +
87 +         sb.append(CPF.substring(POSICAO_PRIMEIRO_PONTO, POSICAO_SEGUNDO_PONTO)).append(".");
88 +         sb.append(CPF.substring(POSICAO_SEGUNDO_PONTO, POSICAO_TRACO)).append("-");
89 +         sb.append(CPF.substring(POSICAO_TRACO));
90 +
91 +         return sb.toString();
76 91     }
77 92 }

```

11

...ma-de-Gestao-Consultas/src/br/edu/ufrsa/controlConsult/model/validacao/Criptografia.java

```

28 28     }
29 29
30 30     private static char[] hexCodes(byte[] text) {
31     -         char[] hexOutput = new char[text.length * 2];
32     -         String hexString;
33     -
34 31 +         final int hexCharsPerByte = 2;
35 32 +         char[] hexOutput = new char[text.length * hexCharsPerByte];
36 33 +         for (int i = 0; i < text.length; i++) {
37     -             hexString = "00" + Integer.toHexString(text[i]);
38     -             hexString.toUpperCase().getChars(hexString.length() - 2,
39     -                 hexString.length(), hexOutput, i * 2);
40 34 +             String hexString = String.format("%02X", text[i]);
41 35 +             hexString.getChars(hexString.length() - hexCharsPerByte,
42 36 +                 hexString.length(), hexOutput, i * hexCharsPerByte);
43 37         }
44 38     return hexOutput;
45 39 }

```

0 comments on commit 2cb1406

Please [sign in](#) to comment.