Trabalho interdisciplinar entre Fundamentos de Engenharia de Software e AEDs1

Grupo: Pedro Henrique Gonçalves Barcelos 689803

Olá prezados professores, como vão? Espero que bem.  
Eu não curso AEDs1 e TI1, na verdade estou no quinto período do curso.

Solução:

Criei um projeto Spring Boot, e nele criei as classes obrigatórias (Paciente, Consulta e Médico).

Hospedei o banco de dados na Azure, para facilitar o teste por parte dos professores avaliadores, visto que utilizei o Docker na fase de desenvolvimento. Dessa forma basta rodar o código e acessar a <http://localhost:8080/> que ele já está apontando para o banco na nuvem.

Implementei as seguintes funcionalidades:

1. Criar Paciente
2. Criar Médico
3. Criar Consulta
4. Cancelar Consulta

Todas essas funcionalidades possuem validação de campos, por exemplo:

* Não se pode criar uma consulta caso não haja médicos e pacientes cadastrados.
* Não se pode criar Paciente sem antes informar todos os campos, com seus respectivos tamanhos mínimos (Ex.: telefone deve ter exatamente 11 caracteres)

Todas as operações exibem feedbacks ao usuário, confirmando que sua intenção inicial foi concretizada. Isto foi feito com Javascript

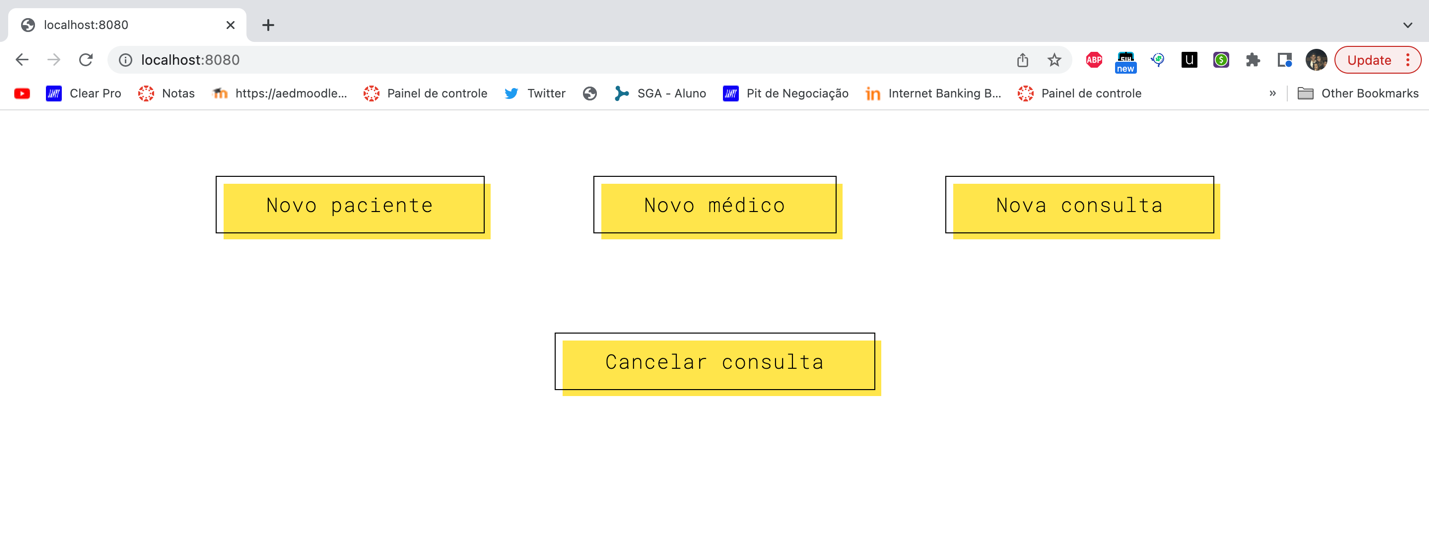


Figura 1 – Tela inicial

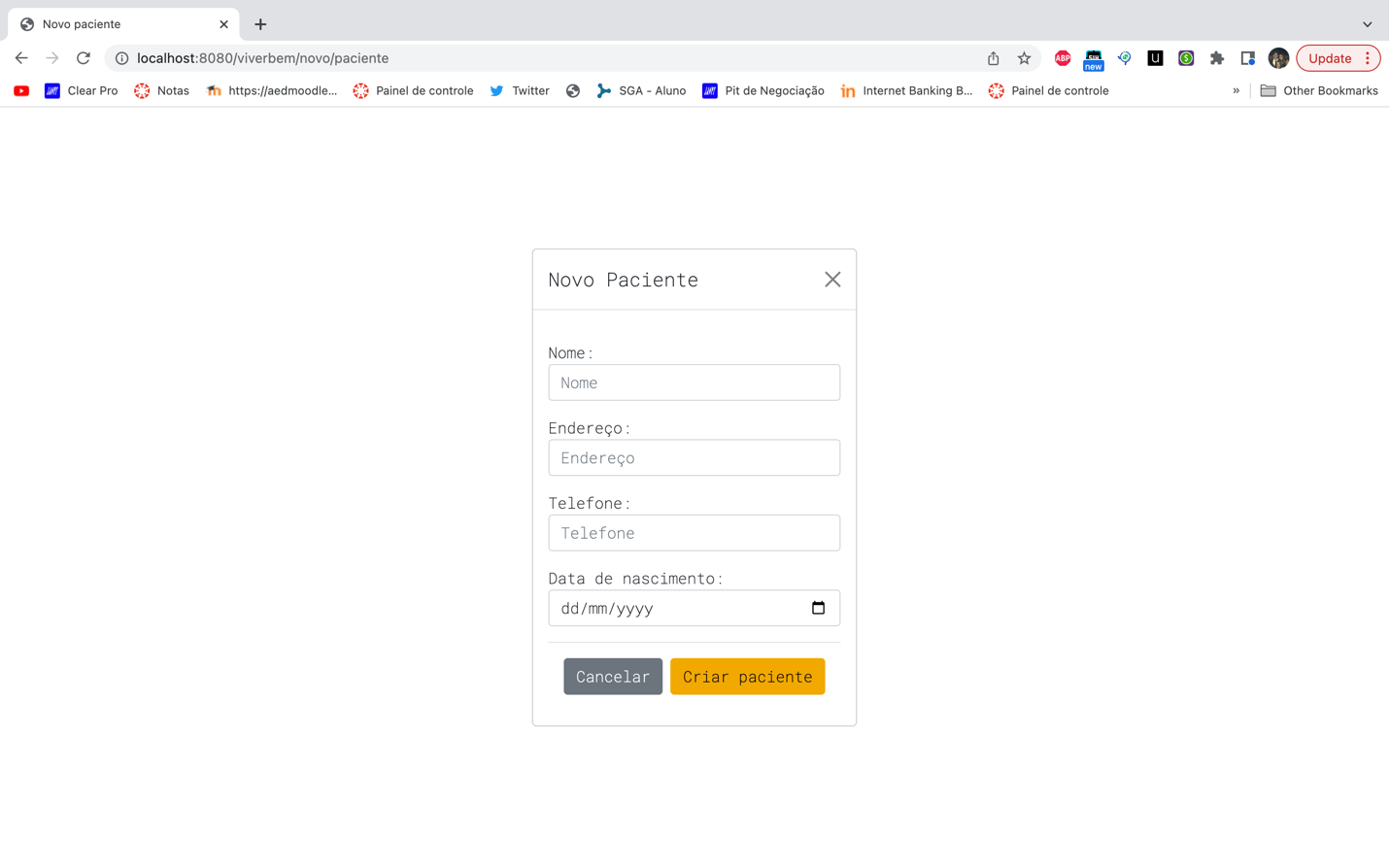


Figura 2 – Tela de Cadastro de paciente

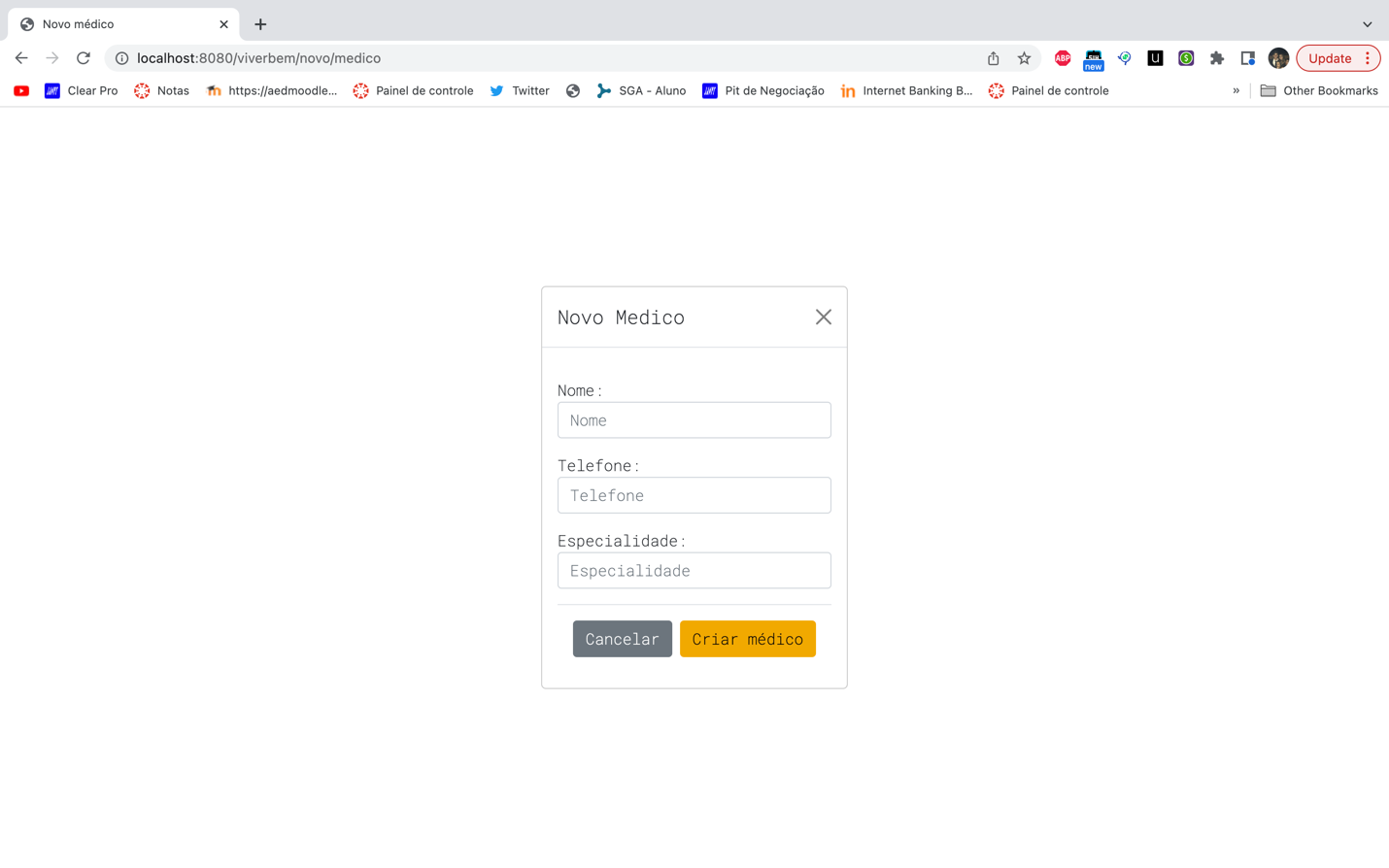


Figura 3 – Tela de cadastro de médico

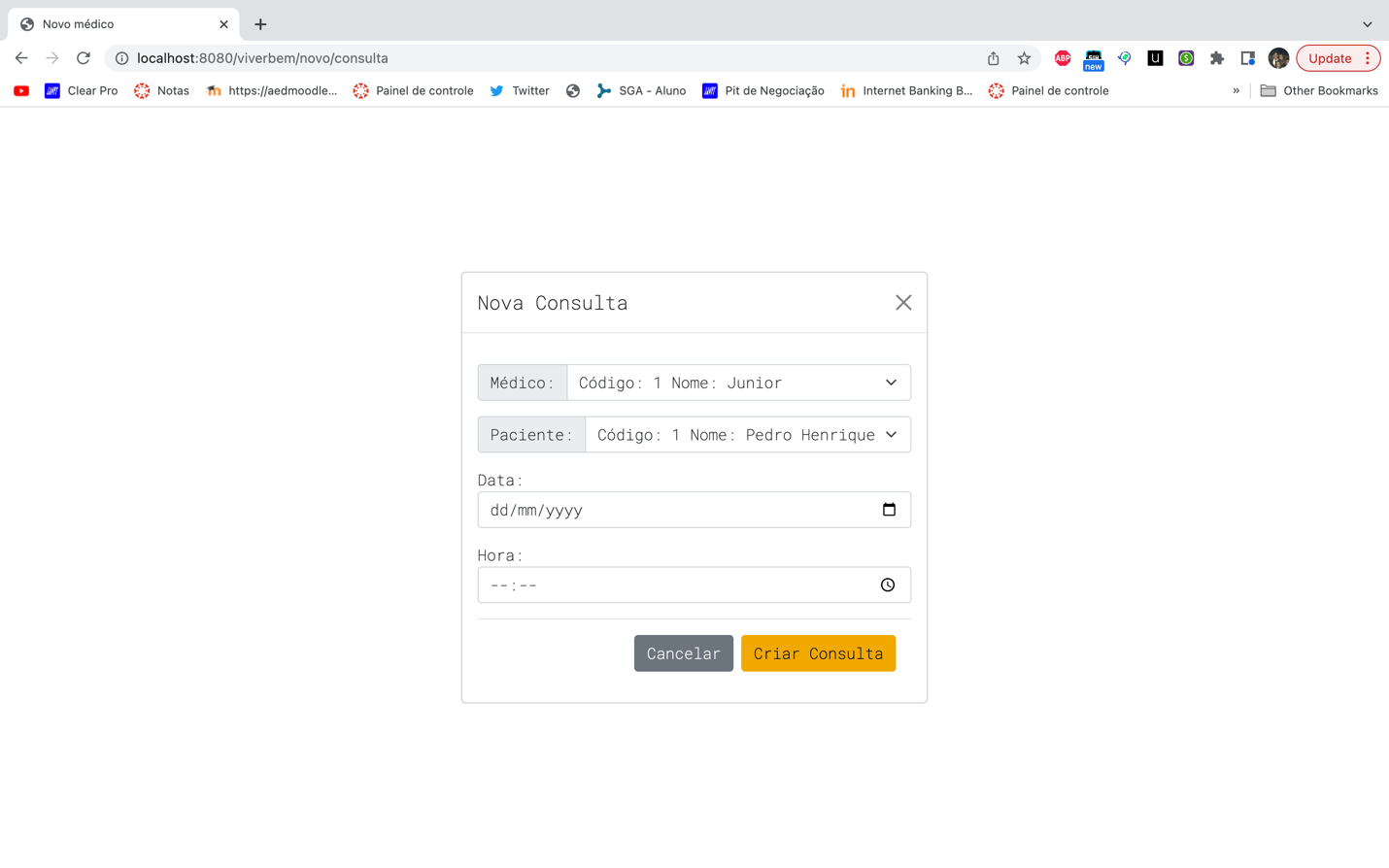


Figura 4 – Tela de cadastro de consulta

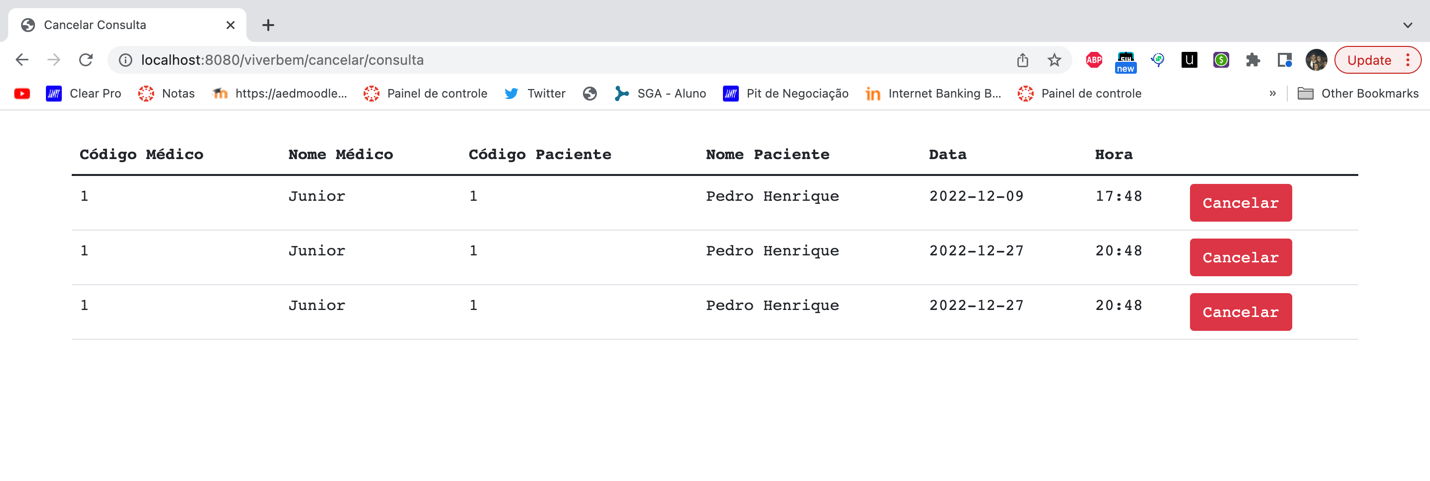


Figura 5 – Tela de cancelamento de consultas

Casos de teste:

1. Testar se o número de registros na tabela de pacientes aumenta em 1 após cadastrar um paciente:

@Test  
void testaSeCadastraPaciente() throws Exception {  
  
 //Tamanho da tabela antes do nosso teste  
 Long tamanho\_da\_tabela = pacienteRepo.count();  
  
 //Instanciar um paciente novo  
 Paciente paciente = new Paciente("Izabella", "Rua 1", "11111111111", "08/12/2022");  
  
 //Cadastrar o paciente novo na tabela por meio de um POST  
 mockMvc.perform(*post*("http://localhost:8080/viverbem/salvar/paciente")  
 .contentType("application/json")  
 .content(objectMapper.writeValueAsString(paciente)))  
 .andExpect(*status*().is(302));  
  
 //testar se o tamanho da tabela aumentou em 1  
 Assertions.*assertTrue*(pacienteRepo.count() == tamanho\_da\_tabela + 1);  
}

1. Testar se o número de registros na tabela de medicos aumenta em 1 após cadastrar um medico:

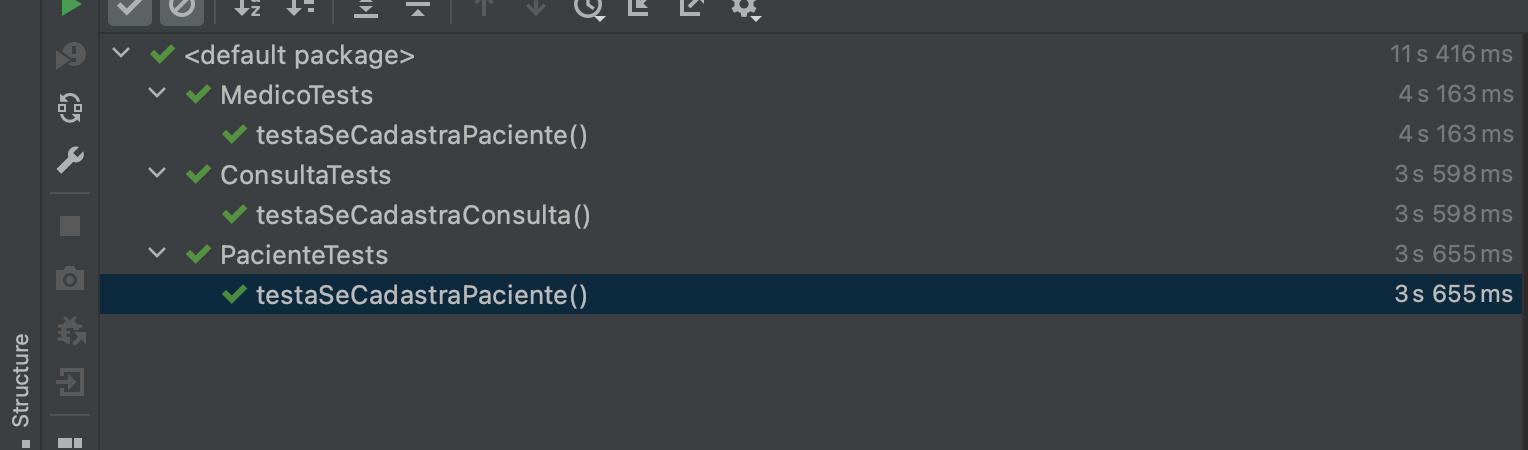
@Test  
void testaSeCadastraMedico() throws Exception {  
  
 //Tamanho da tabela antes do nosso teste  
 Long tamanho\_da\_tabela = medicoRepo.count();  
  
 //Instanciar um paciente novo  
 Medico medico = new Medico("Medico joao");  
  
 //Cadastrar o paciente novo na tabela por meio de um POST  
 mockMvc.perform(*post*("http://localhost:8080/viverbem/salvar/medico")  
 .contentType("application/json")  
 .content(objectMapper.writeValueAsString(medico)))  
 .andExpect(*status*().is(302));  
  
 //testar se o tamanho da tabela aumentou em 1  
 Assertions.*assertTrue*(medicoRepo.count() == tamanho\_da\_tabela + 1);  
}

1. Testar se o número de registros na tabela de consultas aumenta em 1 após cadastrar uma consulta:

@Test  
void testaSeCadastraConsulta() throws Exception {  
  
 //Tamanho da tabela antes do nosso teste  
 Long tamanho\_da\_tabela = consultaRepo.count();  
  
 //Instanciar um paciente, um medico e uma consulta  
 Consulta consulta = new Consulta("08/10/2012");  
  
 //Cadastrar o paciente novo na tabela por meio de um POST  
 mockMvc.perform(*post*("http://localhost:8080/viverbem/salvar/consulta")  
 .contentType("application/json")  
 .content(objectMapper.writeValueAsString(consulta)))  
 .andExpect(*status*().is(302));  
  
 //testar se o tamanho da tabela aumentou em 1  
 Assertions.*assertTrue*(consultaRepo.count() == tamanho\_da\_tabela + 1);  
}

Resultados:

Todos os testes obtiverem êxito, isto é, obtiveram o resultado esperado:



Conclusão:

Deu bastante trabalho desenvolver a aplicação em Spring, por mais que eu já esteja um pouco familiarizado com o framework. Isso foi o que consegui fazer espero que tenham gostado.  
  
O código pode ser encontrado neste repositório do git: <https://github.com/pedrobarcelos/ViverBem>