

## **Plano de Teste**

**Nome do Projeto: ConsulFácil**

**Equipe: Filipe Silva Bravos Valadares, Jhéffrey Thúlyo dos Santos, Kerley Pablo Cordeiro de Padua e Raphael Carlos Botelho Cardoso.**

### **1. Introdução**

O objetivo do teste foi identificar e sanar possíveis erros no código do aplicativo para que seu funcionamento esteja correto e todas as funções sendo executadas sem erro. Foram testadas as funcionalidades: Alergia, Agenda, Arquivo, Consulta, Consultório, Convênio, Especialidade, Medicamento, Médico, Paciente, Prontuário e Usuário.

Se uma dessas funcionalidades não estiver funcionando corretamente, podem ocasionar erros críticos tanto para os usuários Paciente e Médico.

Para o usuário Paciente os riscos envolvidos com o não funcionamento das funções podem ocasionar consultas não marcadas, datas de agendamentos não disponíveis, falta de lista dos médicos associados ao aplicativo, lista de consultórios associados ao aplicativo, falta da lista de Especialidades médicas para o paciente. Para o usuário médico erros podem ocasionar consultas não marcadas, datas de agendamentos não disponíveis, lista de convênios, lista de medicamentos que pacientes usam, os dados do paciente e prontuário.

### **2. Escopo de Teste**

Foram testadas as classes: Alergia, Agenda, Arquivo, Consulta, Consultório, Convênio, Especialidade, Medicamento, Médico, Paciente, Prontuário e Usuário.

Os métodos testados em cada classe foram:

- Agenda: Consultório, Data, Médico e paciente.
- Alergia: Nome.
- Arquivo: Médico, Nota, Paciente, Pedido de exame, Receita.
- Consulta: Agenda, Consultório, Convênio, Médico, Paciente, Tipo, Valor.
- Consultório: Endereço, Médico, Nome, Telefone.
- Convênio: Nome.
- Especialidade: Nome.
- Medicamento: Nome medicamento.

- Médico: Agenda, Arquivo, CNPJ, Consultório, CRM, Especialidade, Paciente.
- Paciente: Agenda, Arquivo, Prontuário.
- Prontuário: Alergia, Medicamentos, Tipo sanguíneo, Usuário.
- Usuário: Data de Nascimento, E-mail, Nome, Senha, Telefone.

### **3. Fora do Escopo de Teste**

Apenas a Interface não foi testada.

### **4. Considerações de Infraestrutura**

Foram usadas plataforma Junit para a realização dos testes, usado NetBeans para criação dos códigos testados e usado computador para realização de todos os procedimentos.

### **5. Premissas**

Foi utilizado como premissa a declaração de todas as variáveis como private, nos testes realizados.

### **6. Riscos**

Os testes realizados não validaram regras relacionadas a banco de dados.

### **7. Conclusão**

Ao final dos testes, foram verificados alguns erros em duas classes relacionadas a data, mostrando que o código necessitava de alguns ajustes, para o seu devido funcionamento. Nos de mais testes realizados foram concluídos com êxito.