Especificação Formal: Sistema de Agendamento de Serviços Acadêmicos

Versão 1.1

2024

Sumário

1. Relatório de especificação do ASA

A partir das definições e métodos definidos a partir do plano de especificação, o seguinte documento fornece uma visão dos resultados obtidos com a realização das especificações. Com base nele, será possível avaliar os métodos do sistema, verificando se é seguido um comportamento adequado. Isso foi realizado através das definições de pré-condições e pós-condições nos métodos e classes do sistema.

2. Pessoas envolvidas

- LAVINIA DANTAS DE MESQUITA
- CRISTIANA DE PAULO
- ANTONIO CAUE OLIVEIRA MORAIS
- ERIKY ABREU VELOSO

3. Especificação Formal

3.1. Requisitos Funcionais

Agendamento

0. [RF] - O sistema deve permitir que usuários **façam login** em suas contas.

☑ Alto	☐ Médio	☐ Baixo

Disponível em DiscenteService.cs:

Pré-condições Login Discente

```
C/C++
Contract.Requires(login.Email != null, "Email não pode ser nulo ou vazio.");
Contract.Requires(login.Senha != null, "Senha não pode ser nula ou vazia.");
```

Pré-condições Login Profissional

```
C/C++
Contract.Requires(login.Email != null, "Email não pode ser nulo ou vazio.");
Contract.Requires(login.Senha != null, "Senha não pode ser nula ou vazia.");
```

1. [RF] - O sistema deve permitir que usuários do tipo discente **solicitem** o agendamento dos serviços.

☑ Alto	☐ Médio	☐ Baixo

Disponível em AgendamentoService.cs:

Pré-condições Solicitar Agendamento

```
C/C++
Contract.Requires(dto != null, "O objeto de solicitação de agendamento não pode ser nulo.");
Contract.Requires(dto.DiscenteId > 0, "O ID do discente deve ser válido.");
Contract.Requires(dto.HorarioId > 0, "O ID do horário deve ser válido.");
Contract.Requires(dto.ProfissionalId > 0, "O ID do profissional deve ser válido.");
Contract.Requires(dto.ServicoId > 0, "O ID do serviço deve ser válido.");
Contract.Requires(dto.Data != default(DateTime), "A data do agendamento deve ser válida.");
Contract.Requires(!string.IsNullOrEmpty(dto.Status), "O status do agendamento não pode ser nulo ou
vazio.");
```

Pós-condições Solicitar Agendamento

```
C/C++
Contract.Ensures(novoAgendamento != null, "O novo agendamento deve ter sido criado com sucesso.");
```

2. [RF] - O sistema deve permitir que os usuários **visualizem** os detalhes do agendamento cadastrado.

☐ Alto	☑ Médio	☐ Baixo

Disponível em AgendamentoService.cs:

Pré-condições Listar Agendamentos por Discente

```
C/C++
Contract.Requires(discenteId > 0, "O ID do discente deve ser maior que zero.");
```

3. [RF] - O sistema deve permitir que usuários do tipo discente e do tipo profissional **cancelem** um agendamento antes da data marcada.

☐ Alto	☑ Médio	☐ Baixo

Disponível em AgendamentoService.cs:

Pré-condições Cancelar Agendamento

```
C/C++
Contract.Requires(agendamentoId > 0, "O ID do agendamento deve ser maior que zero.");
```

Pós-condições Cancelar Agendamento

```
C/C++
Contract.Ensures(agendamento == null, "O agendamento deve ter sido removido do banco de dados.");
```

4. [RF] - O sistema deve **notificar** o cancelamento do agendamento para as partes envolvidas na consulta. (**Esse requisito não foi implementado**)

☐ Alto	☐ Médio	☑ Baixo

5. [RF] - O sistema deve **notificar** a confirmação do agendamento para o discente no ato da aceitação e 24h antes da data marcada. (**Esse requisito não foi implementado**)

☐ Alto	☐ Médio	☑ Baixo

6. [RF] - O sistema deve permitir que usuários do tipo discente **visualizem** os horários disponíveis para cada serviço ofertado.

☑ Alto	☐ Médio	☐ Baixo

Disponível em AgendamentoService.cs:

Pré-condições Listar Horários

```
C/C++
Contract.Requires(profissionalId > 0, "O ID do profissional deve ser maior que zero.");
```

Pré-condições Buscar Agendamentos

```
C/C++
Contract.Requires(profissionalId > 0, "O ID do profissional deve ser maior que zero.");
```

Serviço

7. [RF] - O sistema deve **conter** os serviços de psicólogo, pedagogo, nutricionista e assistente social para a comunidade acadêmica. (**Esse requisito depende do registro de tais profissionais no Banco de Dados)**

☑ Alto	☐ Médio	☐ Baixo

8. [RF] - Os serviços de psicólogo e nutricionista devem **disponibilizar** as opções de consulta e retorno. (**Esse requisito depende do registro de tais informações no Banco de Dados**)

☐ Alto	☑ Médio	☐ Baixo

9. [RF] - Um serviço poderá ter mais de um profissional disponível. (**Esse requisito depende do registro de tais profissionais no Banco de Dados**)

☐ Alto	☐ Médio	☑ Baixo

Usuários

10. [RF] - O sistema deve permitir o **cadastro** de usuários por meio de um e-mail e de uma senha.

☑ Alto	☐ Médio	☐ Baixo

11. [RF] - O sistema deve permitir após a aprovação do e-mail e senha, realizar o **cadastro** das informações de Nome, Sobrenome, Matrícula e Telefone (opcional).

☑ Alto	☐ Médio	☐ Baixo

Disponível em DiscenteService.cs:

Pré-condições Registrar Discente

```
C/C++
Contract.Requires(registro.Nome != null, nameof(registro.Nome));
Contract.Requires(registro.Email != null, nameof(registro.Email));
Contract.Requires(
      registro.Email.EndsWith("@alunos.ufersa.edu.br"),
            "O Email deve conter o domínio '@alunos.ufersa.edu.br'.");
Contract.Requires(registro.Senha != null, nameof(registro.Senha));
Contract.Requires(registro.Matricula != null, nameof(registro.Matricula));
```

Pós-condições Registrar Discente

```
C/C++
Contract.Ensures(profissional != null, "O objeto Discente não deve ser nulo ao final do método.");
Contract.Ensures(profissional.Email.EndsWith("@alunos.ufersa.edu.br"), "O email deve conter o domínio correto após a inserção.");
```

Pré-condições Registrar Profissional

```
C/C++
Contract.Requires(registro.Nome != null, nameof(registro.Nome));
Contract.Requires(registro.Email != null, nameof(registro.Email));
Contract.Requires(
      registro.Email.EndsWith("@ufersa.edu.br"),
      "O Email deve conter o domínio '@ufersa.edu.br'.");
Contract.Requires(registro.Senha != null, nameof(registro.Senha));
Contract.Requires(registro.ServicoId != null, nameof(registro.ServicoId));
```

Pós-condições Registro Profissional

```
C/C++
Contract.Ensures(profissional != null, "O objeto Profissional não deve ser nulo ao final do método.");
Contract.Ensures(profissional.Email.EndsWith("@ufersa.edu.br"), "O email deve conter o domínio correto após a inserção.");
```

12. [RF] - O sistema deve permitir que usuários, uma vez cadastrados, possam **alterar** a senha e seu perfil.

☐ Alto	☑ Médio	☐ Baixo

Disponível em DiscenteService.cs:

Pré-condições Atualizar Perfil

```
C/C++
Contract.Requires(atualizarPerfil != null, "O objeto atualizarPerfil não pode ser nulo.");
```

Pós-condições Atualizar Perfil

```
C/C++
Contract.Ensures(usuarioDiscente != null, "O perfil do usuário foi atualizado com sucesso.");
Contract.Ensures(usuarioDiscente.Nome != null, "O perfil do usuário foi atualizado com sucesso.");
```

Pré-condições Alterar Senha

```
C/C++
Contract.Requires(alterarSenha != null, "O objeto alterarSenha não pode ser nulo.");
```

Pós-condições Alterar Senha para discente

```
C/C++
Contract.Ensures(usuarioDiscente.Senha != null, "A senha do usuário foi
atualizada com sucesso.");
```

Pós-condições Alterar Senha para profissional

```
C/C++
Contract.Ensures(usuarioProfissional.Senha != null, "A senha do usuário foi
atualizada com sucesso.");
```

13. [RF] - Cada usuário do tipo profissional deve estar ligado a um único serviço específico. (**Esse requisito é atendido através da estrutura do Banco de Dados**)

☐ Alto	☑ Médio	☐ Baixo

Recursos

14. [RF] - O sistema deve permitir que usuários do tipo profissional **cadastrem** informações acerca do serviço prestado.

☑ Alto	☐ Médio	☐ Baixo

Disponível em ServicoController.cs:

Pré-condições Cadastrar Serviço

```
C/C++
Contract.Requires(!string.IsNullOrEmpty(servicoDto.Tipo), "O tipo do serviço nãoo pode ser nulo ou vazio.");
Contract.Requires(!string.IsNullOrEmpty(servicoDto.TipoAtendimento), "O tipo de atendimento não pode ser nulo ou vazio.");
```

15. [RF] - Os horários disponíveis para atendimento devem ser definidos pelo profissional. (**Esse requisito depende do registro de tais informações no Banco de Dados**)

☑ Alto	☐ Médio	☐ Baixo