

# 1. DOCUMENTO DE REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

## 1.1. Requisitos Funcionais (RF)

- **RF01.** O sistema deve permitir o cadastro e autenticação de usuários via JWT.
- **RF02.** O sistema deve permitir que o usuário agende refeições (almoço ou jantar) para datas futuras.
- **RF03.** O sistema deve controlar a presença do usuário no restaurante.
- **RF04.** O sistema deve aplicar penalidades para agendamentos não cumpridos.
- **RF05.** O sistema deve permitir ao administrador publicar opções de cardápio para votação.
- **RF06.** O sistema deve permitir aos usuários votar nas opções de cardápio.
- **RF07.** O sistema deve definir automaticamente o cardápio futuro com base nos votos.
- **RF08.** O sistema deve permitir que usuários avaliem refeições e o serviço prestado.
- **RF09.** O sistema deve controlar os créditos de refeição por semestre letivo.
- **RF10.** O sistema deve gerar relatórios de agendamentos, presenças, penalidades e avaliações.

## 1.2. Requisitos Não Funcionais (RNF)

- **RNF01.** A API deve ser desenvolvida em TypeScript com modo estrito ativado.
- **RNF02.** O sistema deve adotar Clean Architecture e boas práticas de Clean Code.
- **RNF03.** O sistema deve ser coberto com testes automatizados (TDD).
- **RNF04.** A API deve estar documentada via OpenAPI (Swagger).
- **RNF05.** O sistema deve persistir os dados em um banco de dados relacional (PostgreSQL).

# 2. ROTEIRO DIDÁTICO

## Fundamentos

- Apresentação da proposta do sistema.
- Conceitos de Clean Code e SOLID.
- Configuração de ambiente e estrutura base.

## Camada de Domínio

- Modelagem de entidades e value objects.

- Casos de uso com TDD: autenticação e registro.

## **Agendamento e Presença**

- Casos de uso: agendar refeição, confirmar presença, aplicar penalidade.
- Validações de regras de negócio.

## **Cardápios e Votação**

- Publicação de opções de cardápio.
- Lógica de votação e encerramento.

## **Avaliações e Créditos**

- Registro de avaliações com restrição de acesso.
- Controle de créditos e relação com agendamentos e penalidades.

## **Relatórios e Integração Final**

- Relatórios por perfil admin.
- Testes integrados e deploy local com Docker.

# **3. CASOS DE USO**

### **UC01. Autenticar Usuário**

- Ator: Usuário
- Descrição: Permite login com e-mail e senha. Retorna token JWT.

### **UC02. Agendar Refeição**

- Ator: Usuário
- Descrição: Agenda refeição futura para um turno específico, se houver crédito.

### **UC03. Confirmar Presença**

- Ator: Usuário
- Descrição: Registra comparecimento no dia agendado.

### **UC04. Publicar Cardápio e Opções**

- Ator: Administrador

- Descrição: Cria opções de cardápio para votação futura.

#### **UC05. Votar no Cardápio**

- Ator: Usuário
- Descrição: Seleciona uma entre as opções publicadas para o turno agendado.

#### **UC06. Avaliar Refeição**

- Ator: Usuário
- Descrição: Envia nota e comentário após presença registrada.

#### **UC07. Gerar Relatórios**

- Ator: Administrador
- Descrição: Consulta indicadores do restaurante (número de agendamentos, ausências, avaliações etc.).

## **4. GLOSSÁRIO (LINGUAGEM UBÍQUA)**

- **Usuário:** Pessoa autorizada a utilizar o RU.
- **Turno:** Período de funcionamento da refeição: Almoço ou Jantar.
- **Agendamento:** Intenção registrada de comparecer a uma refeição.
- **Presença:** Confirmação do comparecimento.
- **Penalidade:** Restrição aplicada a um usuário por comportamento inadequado.
- **Crédito:** Direito de agendar refeições. Gerenciado por semestre.
- **Cardápio:** Conjunto de opções alimentares a serem escolhidas.
- **Avaliação:** Feedback de qualidade da comida e do serviço.

## **5. FLUXO DE ESTADO: CICLO DE VIDA DE UM AGENDAMENTO**

Estados:

- **Criado:** Usuário agenda uma refeição.
- **Cancelado:** Usuário cancela antes do prazo limite.
- **Confirmado:** Presença registrada no sistema.
- **Ausente:** Data da refeição passou sem confirmação.
- **Penalizado:** Ausência resultou em penalidade.

Transições:

- Criado → Cancelado (antes do prazo)
- Criado → Confirmado (presença registrada)
- Criado → Ausente → Penalizado (pelo sistema após vencimento do horário)

## 6. DETALHAMENTO DOS CASOS DE USO E FLUXOS DE TESTE

### UC01. Autenticar Usuário

- **Ator:** Usuário
- **Pré-condições:**
  - Usuário cadastrado com e-mail e senha válidos.
- **Fluxo Principal:**
  - i. Usuário informa e-mail e senha.
  - ii. Sistema verifica se e-mail existe.
  - iii. Sistema valida senha fornecida.
  - iv. Sistema retorna token JWT válido com informações do usuário.
- **Fluxo Alternativo 1 - E-mail inexistente:**
  - Sistema retorna erro 401 "Usuário não encontrado".
- **Fluxo Alternativo 2 - Senha incorreta:**
  - Sistema retorna erro 401 "Credenciais inválidas".
- **Pós-condições:**
  - Token JWT é emitido e pode ser usado para acessar endpoints protegidos.
- **Fluxos de Teste:**
  - Cenário 1: Login bem-sucedido com credenciais corretas.
  - Cenário 2: Erro por e-mail inexistente.
  - Cenário 3: Erro por senha incorreta.

### UC02. Agendar Refeição

- **Ator:** Usuário
- **Pré-condições:**
  - Usuário deve estar autenticado.
  - Deve possuir ao menos 1 crédito disponível para o turno solicitado.
  - O agendamento deve ser feito antes do prazo limite estabelecido pelo RU.

- O turno solicitado (almoço ou jantar) deve estar disponível na data especificada.
- **Fluxo Principal:**
  - i. O usuário solicita o agendamento para uma data e turno específico.
  - ii. O sistema verifica se já existe um agendamento para o mesmo usuário, data e turno.
  - iii. O sistema verifica se o usuário possui créditos suficientes.
  - iv. O sistema valida se o agendamento respeita o prazo mínimo exigido.
  - v. O sistema registra o agendamento com status "Criado".
  - vi. O sistema retorna os dados do agendamento ao usuário.
- **Fluxo Alternativo 1 - Já existe agendamento para o turno:**
  - O sistema retorna erro informando que já existe agendamento para o turno escolhido.
- **Fluxo Alternativo 2 - Créditos insuficientes:**
  - O sistema retorna erro de saldo insuficiente para agendamento.
- **Fluxo Alternativo 3 - Fora do prazo de agendamento:**
  - O sistema rejeita a solicitação por estar fora do prazo permitido (ex: agendamento para o mesmo dia após o limite de hora).
- **Fluxo Alternativo 4 - Turno indisponível:**
  - O sistema informa que o turno não estará ativo na data solicitada (ex: feriado).
- **Pós-condições:**
  - O agendamento é registrado no sistema com status "Criado".
  - Nenhum crédito é consumido neste momento; o consumo ocorrerá na confirmação da presença ou penalização por ausência.
- **Fluxos de Teste:**

#### **Cenário 1 - Agendamento bem-sucedido:**

- Dado: Usuário autenticado, com crédito disponível e sem agendamento para a data e turno.
- Quando: Solicita agendamento para uma data válida.
- Então: Agendamento é criado com status "Criado".

#### **Cenário 2 - Agendamento duplicado:**

- Dado: Usuário já agendou para o mesmo turno e data.
- Quando: Tenta novo agendamento.
- Então: Sistema rejeita com erro "Agendamento duplicado".

#### **Cenário 3 - Sem crédito disponível:**

- Dado: Usuário está com saldo zerado.
- Quando: Solicita agendamento.
- Então: Sistema retorna erro de crédito insuficiente.

#### **Cenário 4 - Fora do prazo:**

- Dado: Agendamento é para hoje às 11h, e o prazo era até 8h.
- Quando: Usuário solicita agendamento às 10h.
- Então: Sistema rejeita por fora do prazo.

#### **Cenário 5 - Turno indisponível:**

- Dado: Dia é feriado e turno está desativado.
- Quando: Usuário solicita agendamento.
- Então: Sistema informa indisponibilidade do turno.

### **UC03. Confirmar Presença**

- **Ator:** Usuário
- **Pré-condições:**
  - Usuário tem agendamento para o dia e turno atual.
- **Fluxo Principal:**
  - i. Usuário autentica-se e acessa a funcionalidade de confirmação.
  - ii. Sistema localiza agendamento correspondente.
  - iii. Sistema registra presença como "Confirmado".
- **Fluxo Alternativo 1 - Sem agendamento:**
  - Sistema retorna erro informando que não há agendamento ativo.
- **Pós-condições:**
  - Presença é registrada. Crédito é consumido.
- **Fluxos de Teste:**
  - Cenário 1: Confirmação bem-sucedida.
  - Cenário 2: Erro ao tentar confirmar sem agendamento.

### **UC04. Publicar Cardápio e Opções**

- **Ator:** Administrador
- **Pré-condições:**
  - Administrador autenticado.
- **Fluxo Principal:**
  - i. Admin informa data e turno do cardápio.
  - ii. Admin define opções de refeição.

iii. Sistema salva opções e inicia período de votação.

- **Fluxo Alternativo 1 - Data passada:**
  - Sistema rejeita cadastro de cardápio para data passada.
- **Pós-condições:**
  - Cardápio e opções ficam disponíveis para voto.
- **Fluxos de Teste:**
  - Cenário 1: Publicação bem-sucedida de opções.
  - Cenário 2: Rejeição por data inválida.

## UC05. Votar no Cardápio

- **Ator:** Usuário
- **Pré-condições:**
  - Usuário autenticado.
  - Opções de cardápio disponíveis para data e turno.
- **Fluxo Principal:**
  - Usuário visualiza opções.
  - Escolhe uma opção e confirma voto.
  - Sistema registra voto associado ao usuário e opção.
- **Fluxo Alternativo 1 - Votação encerrada:**
  - Sistema rejeita voto e informa encerramento.
- **Pós-condições:**
  - Voto computado e bloqueado para nova votação no mesmo turno.
- **Fluxos de Teste:**
  - Cenário 1: Voto bem-sucedido.
  - Cenário 2: Tentativa de voto após encerramento.
  - Cenário 3: Voto duplicado no mesmo turno (erro).

## UC06. Avaliar Refeição

- **Ator:** Usuário
- **Pré-condições:**
  - Presença registrada na refeição avaliada.
- **Fluxo Principal:**
  - Usuário acessa funcionalidade de avaliação.

- ii. Informa nota e comentários.
- iii. Sistema salva avaliação vinculada à refeição e ao usuário.
- **Fluxo Alternativo 1 - Tentativa sem presença:**
  - Sistema bloqueia e informa necessidade de presença.
- **Pós-condições:**
  - Avaliação computada e agregada em relatórios.
- **Fluxos de Teste:**
  - Cenário 1: Avaliação bem-sucedida.
  - Cenário 2: Erro ao tentar avaliar refeição sem comparecimento.

## UC07. Gerar Relatórios

- **Ator:** Administrador
- **Pré-condições:**
  - Admin autenticado.
- **Fluxo Principal:**
  - i. Admin solicita relatório de agendamentos, presenças ou avaliações.
  - ii. Sistema executa consultas agregadas.
  - iii. Sistema retorna dados consolidados.
- **Fluxo Alternativo 1 - Parâmetros inválidos:**
  - Sistema retorna erro com motivo (ex: intervalo de datas inválido).
- **Pós-condições:**
  - Dados são exibidos em formato analítico para tomada de decisão.
- **Fluxos de Teste:**
  - Cenário 1: Geração correta de relatório por período.
  - Cenário 2: Erro ao informar intervalo inválido.
  - Cenário 3: Relatório vazio (sem dados no período).