Documento de Levantamento de Requisitos

# 1. Identificação do Documento

|  |  |
| --- | --- |
| Nome do Projeto | Cardappio |
| Responsável | Equipe |
| Data de Criação | 07/04/2025 |
| Versão do Documento | 1.0 |
| Última Atualização | - |

# 2. Histórico de Versões

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Data | Autor | Alterações Realizadas |
| 1.0 | 07/04/2025 | Cássia | Criação inicial com requisitos base |

# 3. Lista de requisitos

3.1. Requisitos Funcionais

RF001 – Gerenciar cadastro de usuário

RF002 – Autenticação

RF003 – Recuperar senha

RF004 – Gerenciar cardápio

RF005 – Gerenciar comanda

RF006 – Gerenciar pedidos

RF007 – Gerenciar mesas

RF008 – Gerenciar pagamento

RF009 – Comunicação com a cozinha

RF010 – Notificações em tempo real / Visualizar status

RF011 – Recuperar histórico de pedidos

RF012 – Integração com sistema de cozinha e pagamento

RF013 – Exibir relatórios

3.2. Requisitos não-funcionais

RNF001 - Desempenho

RNF002 – Usabilidade

RNF003 – Segurança

RNF004 – Confiabilidade

RNF005 – Disponibilidade

RNF006 – Escalabilidade

RNF007 – Manutenibilidade

RNF008 – Integração

RNF009 – Resiliência

# 4. Requisitos Funcionais

# RF001 – Gerenciar cadastro de usuário

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF001 | Gerenciar cadastro de usuário | US001 – Gerenciar cadastro |
| **Descrição** | | |
| O sistema deve permitir o cadastro, edição, visualização e exclusão de usuários.  Campos obrigatórios: CPF e celular.  Campo opcional: senha. | | |
| **Critérios de Aceitação** | | |
| 01 | O cpf e o celular são validados. | |
| 02 | Os dados sensíveis são criptografados | |
| 03 | O usuário recebe mensagem de confirmação do cadastro | |
| 04 | Não é possível cadastrar um CPF ou celular já utilizados | |
| 05 | A senha deve conter pelo menos 8 caracteres, com letras, números e caracteres especiais. | |

# RF002 – Autenticação

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF002 | Autenticação no sistema | US002 – Logar no sistema |
| **Descrição** | | |
| O sistema deve permitir que usuários e administradores realizem autenticação com base em suas credenciais. O tipo de acesso será determinado conforme o perfil do usuário (usuário comum, administrador ou garçom). | | |
| **Critérios de Aceitação** | | |
| 01 | O usuário consegue realizar login com CPF e celular válidos | |
| 02 | Caso o usuário não esteja cadastrado, o sistema impede o login e exibe uma mensagem adequada. | |
| 03 | Tentativas de login com CPF ou celular inválidos devem ser bloqueados, com retorno de mensagem de erro. | |
| 04 | O administrador, após login bem-sucedido, deve ter acesso à interface administrativa, incluindo a visualização da lista de usuários cadastrados. | |
| 05 | A senha deve ser validada de forma segura, utilizando criptografia. | |
| 06 | O sistema deve encerrar sessões inativas após um tempo determinado por segurança. | |

# RF003 – Recuperar senha

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF003 | Recuperar senha | US003 – Recuperar senha  RF002 - Autenticação |
| **Descrição** | | |
| O sistema deve permitir que usuários cadastrados solicitem a redefinição de sua senha, por meio do envio de um link para o celuar previamente registrado. Após redefinir a senha, o usuário deve ser capaz de realizar login normalmente. | | |
| **Critérios de Aceitação** | | |
| 01 | O usuário consegue solicitar a redefinição de senha informando seu e-mail cadastrado. | |
| 02 | O sistema envia uma mensagem para o celular cadastrado com um link seguro e temporário para redefinição da senha. | |
| 03 | O link de redefinição tem validade limitada (por exemplo, 1 hora) e expira após o uso. | |
| 04 | O usuário consegue cadastrar uma nova senha seguindo as regras de segurança definidas (ex: mínimo de 8 caracteres, letras, números e caracteres especiais). | |
| 05 | Após cadastrar a nova senha, o usuário consegue realizar login normalmente. | |
| 06 | Se o celular informado não estiver cadastrado, o sistema deve informar que não foi possível enviar o link de redefinição. | |

# RF004 – Gerenciar cardápio

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF004 | Acessar cardápio | US006 – Visualizar cardápio  US017 – Gerenciar comandas |
| **Descrição** | | |
| O sistema deve permitir que usuários autenticados visualizem o cardápio do restaurante selecionado, contendo as seções organizadas (entradas, pratos principais, bebidas, sobremesas, etc.). O administrador deve ser capaz de adicionar, editar e excluir itens do cardápio de seu próprio restaurante. | | |
| **Critérios de Aceitação** | | |
| 01 | O usuário autenticado consegue visualizar o cardápio correspondente ao restaurante selecionado. | |
| 02 | O cardápio exibe itens organizados por categoria/seção (ex: entradas, pratos principais, bebidas, sobremesas). | |
| 03 | Os dados exibidos para cada item do cardápio incluem: nome, descrição, valor e disponibilidade. | |
| 04 | O administrador autenticado consegue adicionar novos itens ao cardápio, informando todos os campos obrigatórios. | |
| 05 | O administrador pode editar e excluir itens do cardápio de seu restaurante, e as alterações são refletidas imediatamente para os usuários. | |
| 06 | O sistema valida que apenas administradores do restaurante podem modificar seu respectivo cardápio. | |

# RF005 – Gerenciar comanda

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF005 | Gerenciar comanda | RF002 – Autenticação  US007 – Criar comanda  US011 - Fechar comanda  US016 – Visualizar comandas  US017 – Gerenciar comandas |
| **Descrição** | | |
| O sistema deve permitir que usuários autenticados abram uma comanda individual ao escolher um restaurante e mesa. O usuário poderá visualizar, reabrir (se ainda não paga), e fechar a comanda para iniciar o processo de pagamento. Administradores (ou usuários delegados com permissão) poderão visualizar todas as comandas do restaurante, inclusive por mesa, bem como realizar operações de gerenciamento (CRUD). | | |
| **Critérios de Aceitação** | | |
| 01 | Cada comanda é identificada por um código único e rastreável. | |
| 02 | O usuário autenticado consegue criar uma comanda individual vinculada a uma mesa e restaurante. | |
| 03 | O usuário consegue visualizar os itens adicionados à comanda em tempo real. | |
| 04 | O usuário consegue reabrir a comanda enquanto ela estiver com status "pendente" (antes do pagamento). | |
| 05 | O usuário consegue fechar a comanda e é redirecionado automaticamente à etapa de pagamento. | |
| 06 | O administrador consegue visualizar todas as comandas abertas por mesa, com seu respectivo status (aberta, fechada, paga, etc.). | |
| 07 | O administrador ou responsável autorizado pode abrir, editar ou fechar comandas, desde que ainda estejam abertas. | |
| 08 | O sistema bloqueia alterações em comandas já pagas. | |

# RF006 – Gerenciar pedidos

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF006 | Gerenciar pedidos | RF005 – Gerenciar comandas  US008 – Adicionar itens ao pedido  US009 – Gerenciar pedido  US010 – Confirmar pedido |
| **Descrição** | | |
| Após abrir uma comanda, o usuário deve conseguir adicionar, editar, excluir e visualizar itens no pedido, desde que ainda não tenham sido enviados para a cozinha. Após o envio, o sistema permitirá apenas a visualização. Poderá ser realizada uma solicitação ao administrador para que o pedido seja cancelado ou excluído. O administrador avaliará se é possível atender o pedido do usuário.  O usuário também pode adicionar observações e consultar o histórico e status dos pedidos feitos. | | |
| **Critérios de Aceitação** | | |
| 01 | O usuário consegue adicionar itens à comanda antes de enviá-la para a cozinha. | |
| 02 | O usuário pode editar ou remover itens do pedido enquanto o status for "não enviado". | |
| 03 | Os pedidos confirmados (com status “enviado”, “em preparo” ou “entregue”) somente podem ser cancelados pelo administrador. | |
| 04 | O sistema bloqueia alterações (edição/exclusão) em pedidos com status "enviado", "em preparo" ou "entregue", permitindo apenas que o usuário solicite a um administrador. | |
| 05 | O usuário consegue visualizar todos os pedidos realizados em sua comanda, incluindo o nome do item, quantidade, valor, observações e o status (em preparo, entregue, cancelado). | |
| 06 | O usuário consegue adicionar observações específicas a cada item do pedido antes do envio para a cozinha. | |
| 07 | O histórico de pedidos permanece disponível mesmo após o fechamento da comanda. | |

# RF007 – Gerenciar mesas

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF007 | Gerenciar mesas | RF005 – Gerenciar comanda  US015 – Visualizar mesas |
| **Descrição** | | |
| O sistema deve permitir que usuários visualizem as mesas disponíveis e possam selecionar uma para iniciar seu pedido. O administrador deve conseguir realizar o CRUD das mesas (criar, editar, excluir, listar) e visualizar o status atual de cada mesa (livre, ocupada, aguardando pagamento, etc.), com integração ao gerenciamento de comandas. | | |
| **Critérios de Aceitação** | | |
| 01 | O administrador consegue cadastrar, editar e excluir mesas através da interface de gerenciamento. | |
| 02 | O administrador consegue visualizar o status de cada mesa em tempo real, com base nas comandas associadas (livre, ocupada, aguardando pagamento, etc.). | |
| 03 | O sistema bloqueia a exclusão de mesas que tenham comandas abertas associadas. | |
| 04 | O usuário comum consegue visualizar as mesas disponíveis no restaurante selecionado. | |
| 05 | O usuário consegue selecionar uma mesa para iniciar a comanda (quando permitido pelo restaurante). | |

# RF008 – Gerenciar pagamento

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF008 | Gerenciar pagamento | RF005 – Gerenciar comanda  RF007 – Gerenciar mesas  US012 – Escolher método de pagamento  US013- Visualizar status de pagamento  US014 – Visualizar pagamentos realizados |
| **Descrição** | | |
| O sistema deve permitir que o usuário realize o pagamento online da comanda, escolhendo entre diferentes métodos (PIX, cartão de crédito, etc.), além de visualizar o status do pagamento em tempo real e consultar os pagamentos já realizados.  O usuário pode realizar o pagamento em dinheiro, devendo, neste caso, dirigir-se ao caixa ou fazê-lo diretamente ao garçom, a depender da regra de negócio, o qual será responsável por atualizar o status do pagamento.  O administrador deve poder acompanhar os pagamentos efetuados, verificar seu status (pago, pendente, falhado) e identificar o método utilizado. A funcionalidade deve garantir a integração com os módulos de comandas e mesas, refletindo corretamente os pagamentos associados a cada comanda/mesa. | | |
| **Critérios de Aceitação** | | |
| 01 | O usuário consegue selecionar o método de pagamento (PIX, cartão de crédito, dinheiro) durante o processo de fechamento da comanda. | |
| 02 | O pagamento pode ser realizado diretamente pelo app e é processado com segurança. | |
| 03 | O status do pagamento é atualizado automaticamente e exibido em tempo real para o usuário e o administrador. | |
| 04 | O usuário consegue acessar o histórico de pagamentos realizados, com valores, datas e formas de pagamento. | |
| 05 | O administrador consegue visualizar a lista de pagamentos realizados no sistema, com os detalhes de valor, comanda, mesa, status e forma de pagamento. | |
| 06 | Pagamentos com falha ou incompletos são identificados e notificados ao usuário. | |
| 07 | A funcionalidade está integrada ao fechamento de comandas e atualiza o status da comanda e mesa após a confirmação de pagamento. | |

# RF009 – Comunicação com a cozinha

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF009 | Comunicação com a cozinha | RF006 – Gerenciar pedidos  US010 – Confirmar pedidos |
| **Descrição** | | |
| O sistema deve possibilitar que os pedidos confirmados pelos usuários sejam automaticamente enviados para o painel da cozinha. A cozinha deve receber os pedidos com as informações relevantes (mesa, comanda, itens, observações), de forma clara e organizada, permitindo acompanhar o status de preparo. O envio deve ser imediato após a confirmação do pedido. | | |
| **Critérios de Aceitação** | | |
| 01 | O pedido é automaticamente enviado para o painel/cozinha após a confirmação do usuário. | |
| 02 | As informações enviadas para a cozinha incluem: número da mesa, número da comanda, itens pedidos, quantidades e observações adicionais. | |
| 03 | A cozinha consegue visualizar os pedidos em ordem cronológica de chegada e com status atualizado (ex: em preparo, pronto). | |
| 04 | Caso haja falha de comunicação entre app e cozinha, o sistema emite alerta para reenvio automático. | |
| 05 | O usuário não consegue modificar o pedido após o envio para a cozinha. | |
| 06 | O administrador consegue modificar o pedido, mesmo após o envio para cozinha. | |
| 07 | Os pedidos cancelados antes de serem preparados são sinalizados para a cozinha como "cancelado". | |

# RF010 – Notificações em tempo real / Visualizar status

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF010 | Notificações em tempo real / Visualizar status | RF006 – Gerenciar pedidos  RF008 – Gerenciar pagamentos  US010 – Confirmar pedido  US013 – Visualizar status do pagamento |
| **Descrição** | | |
| O sistema deve exibir atualizações em tempo real sobre o status do pedido (em preparo, pronto, entregue, cancelado) e do pagamento (pendente, processando, pago, falhou). O usuário deve receber essas informações de forma clara e imediata no app. O administrador também deve ter acesso a essas informações para acompanhamento da operação do restaurante. | | |
| **Critérios de Aceitação** | | |
| 01 | O sistema atualiza o status do pedido automaticamente conforme ele avança (ex: em preparo → pronto → entregue). | |
| 02 | O status do pagamento é atualizado conforme a confirmação da transação pelo meio de pagamento escolhido. | |
| 03 | O administrador pode visualizar o status atualizado de todos os pedidos e pagamentos do restaurante. | |
| 04 | A mudança de status é sincronizada com o painel da cozinha e com os dados da comanda/mesa. | |
| 05 | O usuário é notificado em caso de falha no pagamento. | |

# RF011 – Recuperar histórico de pedidos

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF011 | Recuperar histórico de pedidos | RF006 – Gerenciar pedidos  US014 – Visualizar pagamentos realizados |
| **Descrição** | | |
| O sistema deve permitir que o usuário visualize um histórico de pedidos e pagamentos realizados anteriormente, com detalhes como data, restaurante, itens pedidos, valores e status da comanda/pagamento. O histórico deve estar disponível tanto para usuários quanto para administradores (neste caso, para controle e análise das comandas e pagamentos processados). | | |
| **Critérios de Aceitação** | | |
| 01 | O usuário consegue visualizar os pedidos realizados anteriormente, com data, itens e valor total. | |
| 02 | O histórico mostra o status final da comanda (fechada, paga, cancelada). | |
| 03 | O histórico de pagamentos exibe o valor pago, forma de pagamento e data/hora. | |
| 04 | O histórico deve ser ordenado por data (do mais recente ao mais antigo) e permitir filtragem por período. | |
| 05 | O usuário não consegue modificar registros do histórico. | |

# RF012 – Integração com sistema de cozinha e pagamento

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF012 | Integração com sistema de cozinha e pagamento | RF006 – Gerenciar pedidos RF008 – Gerenciar pagamento RF010 – Notificações em tempo real / Visualizar status US010 – Confirmar pedido US013 – Visualizar status do pagamento |
| **Descrição** | | |
| O sistema deve se integrar com o painel da cozinha para o envio automático dos pedidos confirmados, permitindo o gerenciamento dos status (em preparo, pronto, entregue). Além disso, deve integrar-se com o sistema de pagamento para processar transações (Pix, cartão, etc.), garantindo retorno imediato sobre o status da operação. | | |
| **Critérios de Aceitação** | | |
| 01 | Após a confirmação do pedido pelo usuário, ele é enviado automaticamente ao painel da cozinha. | |
| 02 | A cozinha consegue visualizar o pedido com observações e identificar a mesa ou comanda associada. | |
| 03 | A cozinha consegue atualizar o status do pedido, que é refletido no app do cliente em tempo real. | |
| 04 | O sistema está integrado com pelo menos um meio de pagamento (ex: API de pagamentos) para processar as transações. | |
| 05 | O sistema retorna ao usuário o status da transação (pago, pendente, erro) logo após a tentativa de pagamento. | |
| 06 | Falhas de integração com a cozinha ou pagamento são tratadas e exibidas de forma amigável ao usuário e administrador. | |
| 07 | As integrações seguem padrões de segurança e estão documentadas para manutenção futura. | |

# RF013 – Exibir relatórios

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RF013 | Exibir relatórios | RF005 – Gerenciar comandas  RF006 - Gerenciar pedidos RF008 – Gerenciar pagamento |
| **Descrição** | | |
| O sistema deve permitir ao administrador gerar e visualizar relatórios gerenciais relacionados a comandas, pedidos e pagamentos. Os relatórios devem permitir filtros por período, status, itens, métodos de pagamento e usuários, com o objetivo de apoiar a gestão do restaurante. | | |
| **Critérios de Aceitação** | | |
| 01 | O administrador consegue gerar relatórios de comandas filtrando por status (aberta, fechada, paga, etc.) e por período. | |
| 02 | O administrador consegue gerar relatórios de pagamentos filtrando por método de pagamento (Pix, cartão, etc.), período e por usuário. | |
| 03 | O administrador consegue gerar relatórios de pedidos por itens mais pedidos, quantidade total e por usuário. | |
| 04 | Os relatórios podem ser exportados em formato PDF ou CSV. | |
| 05 | Os relatórios devem ser gerados com desempenho aceitável mesmo para grandes volumes de dados. | |

# 5. Requisitos Não Funcionais

# RNF001 - Desempenho

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN001 | Desempenho | RF004 – Gerenciar cardápio RF006 – Gerenciar pedidos RF008 – Gerenciar pagamento RF010 – Visualizar status em tempo real |
| **Descrição** | | |
| O sistema deve ser rápido e responsivo tanto no aplicativo mobile quanto na aplicação web. As ações do usuário não devem sofrer atrasos perceptíveis. | | |
| **Critérios de Aceitação** | | |
| 01 | O tempo de resposta do sistema deve ser inferior a 2 segundos para ações comuns (como abrir cardápio, enviar pedido ou visualizar status). | |
| 02 | O carregamento do aplicativo não deve ultrapassar 2 segundos. | |
| 03 | A performance deve ser consistente mesmo com até 100 usuários simultâneos. | |
| 04 | A atualização de status em tempo real (pagamento e pedidos) deve ocorrer sem atrasos perceptíveis ao usuário. | |

# RNF002 – Usabilidade

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RNF002 | Usabilidade | RF001 – Gerenciar cadastro de usuário RF004 – Gerenciar cardápio RF006 – Gerenciar pedidos RF007 – Gerenciar mesas RF008 – Gerenciar pagamento |
| **Descrição** | | |
| As interfaces do aplicativo mobile e da aplicação web devem ser simples, intuitivas e de fácil utilização para usuários finais e administradores. | | |
| **Critérios de Aceitação** | | |
| 01 | Os fluxos de uso devem seguir padrões de design conhecidos e acessíveis, como Material Design ou similares. | |
| 02 | As principais funcionalidades devem estar acessíveis em no máximo 3 toques/cliques. | |
| 03 | O sistema deve apresentar mensagens claras e orientações em caso de erro ou uso indevido. | |
| 04 | Deve haver consistência visual entre as telas, uso de cores e ícones padronizados. | |
| 05 | Deve ser possível utilizar o sistema sem a necessidade de treinamento técnico para usuários comuns. | |
| 06 | O app deve suportar acessibilidade básica, como contraste adequado e tamanho de fonte ajustável. | |

# RNF003 – Segurança

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN003 | Segurança | RF001 – Gerenciar cadastro de usuário RF002 – Autenticação no sistema RF003 – Recuperar senha RF008 – Gerenciar pagamento |
| **Descrição** | | |
| O sistema deve garantir a segurança dos dados dos usuários e das transações, protegendo contra acessos não autorizados e vazamentos de dados. | | |
| **Critérios de Aceitação** | | |
| 01 | Os dados sensíveis (como senhas e CPF) devem ser armazenados utilizando algoritmos de criptografia atualizados e seguros. | |
| 02 | O tráfego de dados entre cliente e servidor deve utilizar HTTPS/TLS. | |
| 03 | Autenticação deve seguir boas práticas, como verificação de e-mail e bloqueio temporário após tentativas incorretas consecutivas. | |
| 04 | Tokens de autenticação devem expirar após período de inatividade e serem renováveis de forma segura. | |
| 05 | Pagamentos online devem utilizar plataformas confiáveis de terceiros (ex: Mercado Pago, etc). | |

# RNF004 – Confiabilidade

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN004 | Confiabilidade | RF005 – Gerenciar comanda RF006 – Gerenciar pedidos RF008 – Gerenciar pagamento RF010 – Integração com cozinha |
| **Descrição** | | |
| O sistema deve operar de maneira estável e sem falhas, garantindo que os dados e transações sejam processados corretamente e sem perdas. | | |
| **Critérios de Aceitação** | | |
| 01 | Os pedidos devem ser registrados e atualizados corretamente, sem perda de informações. | |
| 02 | As comandas não devem ser duplicadas, perdidas ou inconsistentes após reabertura ou fechamento. | |
| 03 | O sistema deve tratar falhas de comunicação com a cozinha e registrar o status do pedido mesmo se houver perda temporária de conexão. | |
| 04 | A taxa de falhas não poderá ser superior a 1 a cada 100 mil requisições. | |
| 05 | Em caso de falha de pagamento, o sistema deve garantir que a transação não seja confirmada e exibir mensagem clara ao usuário. | |
| 06 | O sistema deve manter histórico dos pedidos e status mesmo após reinício ou atualização da aplicação. | |
| 07 | O sistema deve passar por testes automatizados de regressão a cada nova versão, cobrindo os principais fluxos críticos. | |

# RNF005 – Disponibilidade

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN005 | Disponibilidade | RF002 – Autenticação no sistema RF004 – Gerenciar cardápio RF005 – Gerenciar comanda RF006 – Gerenciar pedidos RF008 – Gerenciar pagamento |
| **Descrição** | | |
| O sistema, baseado em arquitetura de microsserviços, deve garantir alta disponibilidade, assegurando que falhas isoladas em serviços não comprometam a operação global do sistema. | | |
| **Critérios de Aceitação** | | |
| 01 | O sistema deve garantir uma taxa de disponibilidade mínima de 99,5% ao mês. | |
| 02 | A arquitetura de microsserviços deve isolar falhas, mantendo a disponibilidade dos demais serviços mesmo que um serviço específico esteja fora do ar. | |
| 03 | Cada microsserviço deve ser monitorado individualmente, com alertas automáticos em caso de falha ou degradação. | |
| 04 | O sistema deve possuir balanceamento de carga e orquestração de containers para garantir escalonamento automático e disponibilidade contínua. | |
| 05 | Em caso de indisponibilidade temporária de um serviço, o sistema deve apresentar mensagens amigáveis ao usuário e tentar retentativas automáticas. | |

# RNF006 – Escalabilidade

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN006 | Escalabilidade | RF004 – Gerenciar cardápio RF005 – Gerenciar comanda RF006 – Gerenciar pedidos RF008 – Gerenciar pagamento |
| **Descrição** | | |
| O sistema deve ser escalável horizontalmente, permitindo o aumento de capacidade por meio da adição de instâncias dos microsserviços conforme a demanda aumenta, sem prejudicar o desempenho. | | |
| **Critérios de Aceitação** | | |
| 01 | A infraestrutura deve permitir escalar individualmente cada microsserviço de acordo com a carga específica de uso. | |
| 02 | O sistema deve suportar aumento significativo no número de usuários, comandas e pedidos simultâneos sem impacto perceptível no tempo de resposta. | |

# RNF007 – Manutenibilidade

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN007 | Manutenibilidade | Todos os requistos funcionais |
| **Descrição** | | |
| O sistema deve ser fácil de manter e atualizar, permitindo mudanças em partes isoladas da aplicação com impacto mínimo nas demais funcionalidades, aproveitando os benefícios da arquitetura de microsserviços. | | |
| **Critérios de Aceitação** | | |
| 01 | Cada microsserviço deve possuir código desacoplado, com responsabilidades bem definidas e documentação atualizada. | |
| 02 | Alterações em um microsserviço não devem exigir modificações em outros, salvo casos específicos. | |
| 03 | Devem existir testes automatizados (unitários e de integração) que garantam o funcionamento após alterações no código. | |
| 04 | A infraestrutura de CI/CD deve permitir implantações rápidas e seguras, facilitando o deploy contínuo com rollback em caso de falhas. | |

# RNF008 – Integração

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN008 | Integração | RF006 – Gerenciar pedidos RF008 – Gerenciar pagamento RF010 – Comunicação com a cozinha |
| **Descrição** | | |
| O sistema deve permitir integração com sistemas internos e externos, como sistema da cozinha e meios de pagamento, de forma segura, padronizada e desacoplada. | | |
| **Critérios de Aceitação** | | |
| 01 | A comunicação com a cozinha deve ser em tempo real ou quase real, refletindo imediatamente os pedidos realizados ou alterados. | |
| 02 | Os módulos de pagamento devem permitir integração com provedores como Pix, cartão de crédito/débito, carteiras digitais, entre outros. | |
| 03 | A documentação das APIs deve estar disponível (por exemplo, via Swagger), facilitando futuras integrações. | |
| 04 | Devem existir testes de integração automatizados com os principais sistemas conectados. | |

# RNF009 – Resiliência

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Relacionamento** |
| RFN009 | Resiliência | RF005 – Gerenciar comandas  RF006 – Gerenciar pedidos RF008 – Gerenciar pagamento RF010 – Comunicação com a cozinha |
| **Descrição** | | |
| O sistema deve ser resiliente a falhas temporárias entre os microsserviços. Para isso, eventos críticos (como criação de pedidos, atualização de status, envio de pagamentos e comunicação com a cozinha) devem ser registrados em filas assíncronas. Dessa forma, caso um serviço esteja temporariamente indisponível, o evento poderá ser processado assim que o serviço for restaurado, sem impacto perceptível para o usuário. | | |
| **Critérios de Aceitação** | | |
| 01 | Os eventos críticos do sistema (como pedidos, pagamentos, atualizações de status e notificações à cozinha) são enviados para uma fila de mensagens. | |
| 02 | Em caso de falha de um microsserviço, o evento é mantido na fila e reprocessado automaticamente quando o serviço se recuperar. | |
| 03 | O usuário não deve perceber atrasos ou falhas no fluxo da aplicação devido a indisponibilidades temporárias de serviços internos. | |
| 04 | O sistema registra logs de falha e sucesso no processamento de eventos para auditoria e monitoramento. | |