

# 1. Descrição Geral

## 1.1 Nome do Sistema

Sistema de Delivery Rápido (SDR)

## 1.2 Visão Geral do Projeto

O SDR é um aplicativo para facilitar o processo de compras e entregas de produtos de diversos estabelecimentos (restaurantes, supermercados, farmácias etc.). O objetivo principal é permitir que usuários façam pedidos, acompanhem o status da entrega em tempo real e avaliem o serviço, enquanto os fornecedores podem gerenciar os pedidos, organizar rotas de entrega e manter seu catálogo de produtos atualizado.

## 1.3 Objetivos do Sistema

- **Facilitar** o processo de compras online, reduzindo a necessidade do cliente de se deslocar fisicamente.
- **Acompanhar** todo o processo de entrega, oferecendo transparência e segurança ao cliente.
- **Gerenciar** de forma integrada o estoque, catálogo de produtos e pedidos.
- **Fornecer** relatórios e estatísticas de vendas e entregas para os estabelecimentos.

## 1.4 Escopo

A primeira versão do SDR abrange as funcionalidades essenciais para cadastro de usuários, cadastro de estabelecimentos, visualização de produtos, inserção de pedidos e geração de rotas para entrega. Futuramente, pretende-se expandir para funcionalidades avançadas, como sistema de pontos/fidelidade, integrações com serviços de pagamento e análise preditiva de demanda.

# 2. Glossário

Termo	Definição
Usuário	Pessoa que realiza pedidos e recebe entregas por meio do sistema.
Fornecedor	Estabelecimento que vende produtos/serviços por meio do SDR.
Entregador	Profissional ou colaborador responsável por recolher o pedido no fornecedor e entregá-lo ao usuário.
Pedido	Solicitação de compra de produtos/serviços feita por um usuário dentro do sistema.
Rota	Percurso definido entre o fornecedor e o usuário para efetuar a entrega de um pedido.
Catálogo	Lista de produtos ou serviços disponíveis para compra em cada fornecedor.
Checkout	Etapa em que o usuário confirma o endereço, forma de pagamento e finaliza o pedido.
Dashboard	Tela onde o fornecedor pode gerenciar pedidos, atualizar o catálogo e acompanhar o status de entregas.

# 3. Documentação do Uso de 2 Técnicas de Elicitação

Nesta seção, descrevem-se duas técnicas de elicitação utilizadas para compreender melhor as necessidades e requisitos do SDR.

### 3.1 Técnica 1: Entrevista com Stakeholders

#### Objetivo da Técnica:

Obter informações detalhadas sobre as necessidades, expectativas e problemas enfrentados por usuários finais, fornecedores e entregadores.

#### Procedimento:

1. **Seleção de entrevistados:** Foram escolhidos 3 usuários habituais de aplicativos de entrega, 2 representantes de restaurantes e 2 entregadores.
2. **Roteiro de perguntas:** Preparou-se um roteiro abordando temas como funcionalidades desejadas, pontos de melhoria em sistemas existentes, dificuldades de logística e usabilidade.
3. **Condução das entrevistas:** As entrevistas ocorreram presencialmente e por videoconferência, com duração média de 30 minutos.
4. **Registro das informações:** As respostas foram gravadas e transcritas para posterior análise e organização em requisitos.

#### Principais Resultados:

- Necessidade de **notificações em tempo real** para cada etapa do pedido.
- Requerimento de **interface simples** para atrair usuários inexperientes.
- Importância de um **controle de estoque** unificado para estabelecimentos.
- Desejo de **opção de rastreamento** (mapa em tempo real) e estimativa de tempo de entrega.
- **Reclamações sobre taxas** de entrega excessivas em alguns sistemas concorrentes.

### 3.2 Técnica 2: Brainstorming com Equipe de Desenvolvimento

#### Objetivo da Técnica:

Gerar o máximo de ideias sobre funcionalidades e requisitos do sistema em um curto espaço de tempo, permitindo a equipe de desenvolvimento explorar diferentes soluções de forma criativa.

#### Procedimento:

1. **Preparação:** Reunião virtual com a equipe de desenvolvimento, Product Owner e designer UX.
2. **Sessão de brainstorm:** Cada participante sugeriu funcionalidades, melhorias e requisitos que considerassem relevantes.
3. **Agrupamento de ideias:** As sugestões foram agrupadas por tema (usabilidade, integrações, logística, pagamento etc.).
4. **Priorização:** Foi utilizada a matriz de valor vs. esforço para identificar quais ideias seriam desenvolvidas primeiro.

#### Principais Resultados:

- Proposta de um **sistema de avaliação** (rating) ao final de cada entrega.
  - Ideia de **integração** com gateways de pagamento e carteiras digitais.
  - Sugestão de **sistema de cupons** e promoções.
  - Recomendação de **dashboard analítico** para fornecedores acompanharem desempenho de vendas e entregas.
-

## 4. Requisitos Funcionais (RF) com 20% Mais Importantes

Abaixo, lista-se um conjunto de Requisitos Funcionais, enfatizando os que são considerados o “núcleo” do sistema (o 20% mais importante). Esses requisitos críticos aparecem marcados com asterisco (\*).

### 1. RF001 – Cadastro de Usuário \*

- O sistema deve permitir o **cadastro de usuários** com dados básicos (nome, e-mail, senha, endereço).
- O cadastro deve ser validado via e-mail.

### 2. RF002 – Autenticação e Autorização \*

- O sistema deve fornecer **login e logout** seguros.
- Diferentes perfis de acesso (usuário, fornecedor, entregador) devem ter permissões específicas.

### 3. RF003 – Cadastro de Fornecedor \*

- O sistema deve permitir o **cadastro de fornecedores** (restaurantes, lojas etc.), incluindo informações de CNPJ, endereço e cardápio/portfólio inicial.

### 4. RF004 – Catálogo de Produtos/Serviços \*

- O fornecedor deve conseguir **incluir, editar e remover itens** do seu catálogo.
- O catálogo deve exibir nome do produto, descrição, preço e disponibilidade em estoque.

### 5. RF005 – Inserção de Pedido

- O usuário deve poder **adicionar produtos** de um ou mais fornecedores ao carrinho de compras.
- Deve ser possível **finalizar o pedido** informando endereço de entrega e forma de pagamento.

### 6. RF006 – Notificações em Tempo Real

- O sistema deve enviar **notificações** (push, e-mail, SMS) ao usuário a cada mudança de status do pedido (confirmado, em preparo, em rota, entregue).

### 7. RF007 – Rastreamento da Entrega

- O usuário deve ter acesso a um **mapa em tempo real**, exibindo a localização do entregador e a previsão de chegada.

### 8. RF008 – Dashboard de Pedidos para o Fornecedor

- Deve haver uma tela onde os fornecedores possam **visualizar todos os pedidos**, atualizarem status e organizarem as entregas.

### 9. RF009 – Sistema de Avaliação

- Ao final da entrega, o usuário deve poder **avaliar** o serviço do entregador e/ou do fornecedor, atribuindo notas e comentários.

### 10. RF010 – Geração de Relatórios

- O sistema deve permitir a **geração de relatórios** de vendas, entregas e avaliações, filtrados por período e fornecedor.

## 5. Requisitos Não Funcionais (RNF) – Exemplos

Nesta seção, listam-se requisitos não funcionais relacionados a diferentes categorias, tais como desempenho, segurança, usabilidade, compatibilidade, entre outras.

### 1. RNF001 – Desempenho

- O tempo de resposta para carregar o catálogo de produtos deve ser **inferior a 2 segundos** com até 10.000 itens cadastrados.

### 2. RNF002 – Confiabilidade

- O sistema deve ter **tolerância a falhas**, garantindo que, em caso de queda de um servidor, o serviço seja retomado em no máximo 5 minutos.

### 3. RNF003 – Segurança

- As senhas devem ser **armazenadas utilizando hashing seguro** e sal.
- O acesso aos dados sensíveis (dados de pagamento) deve ser **criptografado**.

### 4. RNF004 – Usabilidade

- A interface do sistema deve ser **intuitiva**, permitindo que um usuário sem treinamento faça um pedido em até 3 minutos.
- O aplicativo móvel deve seguir as diretrizes de design (Material Design ou Human Interface Guidelines) para **navegação consistente**.

### 5. RNF005 – Portabilidade/Compatibilidade

- O sistema deve ser acessível via **navegadores modernos** (Chrome, Firefox, Safari) e possuir **aplicativo móvel** para Android e iOS.

### 6. RNF006 – Manutenibilidade

- O código deve seguir **padrões de desenvolvimento** (ex.: Clean Code, SOLID) e contar com uma **documentação mínima** que facilite futuras manutenções.

### 7. RNF007 – Escalabilidade

- A arquitetura deve permitir **adição de novos servidores** ou serviços em nuvem conforme a demanda aumentar, sem interrupções significativas.