**Documentação de Funcionalidades**

**Sistema de Pizzaria em Python**

**1. Visão do Projeto**

* **Objetivo do Projeto**: Criar um sistema de gestão para uma pizzaria que automatize os processos de pedidos, controle de estoque, e gestão financeira, garantindo um atendimento eficiente para clientes e uma operação simplificada para funcionários e administradores.
* **Clientes/Usuários**: Funcionários (garçons, atendentes), administradores da pizzaria, e clientes finais.
* **Tecnologias Utilizadas**: Python 3.x, Tkinter ou PyQt (interface gráfica), SQLite ou MySQL (banco de dados).

**2. Histórias de Usuário**

**2.1 Cadastro de Pizzas**

* **ID da História de Usuário**: HU-001
* **Descrição**: Como administrador, eu quero cadastrar, editar e remover pizzas no menu para garantir que o cardápio esteja sempre atualizado.
* **Critérios de Aceitação**:
  + O administrador pode adicionar uma nova pizza com nome, descrição, preço, ingredientes e imagem (opcional).
  + O administrador pode editar informações de uma pizza existente.
  + O administrador pode remover pizzas do menu.
  + O sistema deve listar todas as pizzas cadastradas.
* **Prioridade**: Alta
* **Estimativa de Esforço**: 3 pontos

**2.2 Gestão de Pedidos**

* **ID da História de Usuário**: HU-002
* **Descrição**: Como atendente, eu quero gerenciar os pedidos dos clientes, desde a criação até a finalização, para garantir uma entrega eficiente.
* **Critérios de Aceitação**:
  + O atendente pode criar um novo pedido e adicionar pizzas.
  + O atendente pode excluir pizzas do pedido.
  + O sistema deve atualizar automaticamente o preço do pedido.
  + O atendente pode alterar o status do pedido (e.g., "Em Preparação", "Pronto para Entrega").
  + O pedido pode ser finalizado com o cálculo total (incluindo taxas de entrega, se aplicável).
* **Prioridade**: Alta
* **Estimativa de Esforço**: 5 pontos

**2.3 Controle de Estoque**

* **ID da História de Usuário**: HU-003
* **Descrição**: Como administrador, eu quero controlar o estoque de ingredientes para garantir que não faltem itens durante a preparação das pizzas.
* **Critérios de Aceitação**:
  + O administrador pode cadastrar novos ingredientes.
  + O estoque é automaticamente atualizado quando uma pizza é preparada.
  + O sistema deve emitir um alerta quando o estoque de um ingrediente estiver baixo.
  + O administrador pode visualizar o estoque de ingredientes.
* **Prioridade**: Alta
* **Estimativa de Esforço**: 4 pontos

**2.4 Cadastro de Clientes**

* **ID da História de Usuário**: HU-004
* **Descrição**: Como atendente, eu quero cadastrar e atualizar informações dos clientes para garantir um atendimento mais rápido e personalizado.
* **Critérios de Aceitação**:
  + O atendente pode adicionar e atualizar dados de clientes (nome, endereço, telefone).
  + O sistema permite consultar o histórico de pedidos de um cliente.
  + O atendente pode excluir clientes, se necessário.
* **Prioridade**: Média
* **Estimativa de Esforço**: 3 pontos

**2.5 Sistema de Pagamento**

* **ID da História de Usuário**: HU-005
* **Descrição**: Como atendente, eu quero registrar os pagamentos dos pedidos de forma segura e com diferentes opções de pagamento.
* **Critérios de Aceitação**:
  + O atendente pode registrar pagamentos via cartão de crédito, débito ou dinheiro.
  + O sistema deve aplicar descontos promocionais automaticamente, quando aplicáveis.
  + O sistema emite recibos detalhados após o pagamento.
  + O histórico de pagamentos pode ser visualizado pelo administrador.
* **Prioridade**: Alta
* **Estimativa de Esforço**: 5 pontos

**2.6 Relatórios e Gestão Financeira**

* **ID da História de Usuário**: HU-006
* **Descrição**: Como administrador, eu quero gerar relatórios financeiros para monitorar as vendas, lucros e estoque da pizzaria.
* **Critérios de Aceitação**:
  + O administrador pode gerar relatórios de vendas diárias, semanais e mensais.
  + O administrador pode calcular lucro líquido e bruto.
  + O sistema gera relatórios de estoque e itens mais consumidos.
* **Prioridade**: Média
* **Estimativa de Esforço**: 6 pontos

**2.7 Promoções e Descontos**

* **ID da História de Usuário**: HU-007
* **Descrição**: Como administrador, eu quero criar promoções e descontos para atrair mais clientes e aumentar as vendas.
* **Critérios de Aceitação**:
  + O administrador pode criar promoções (ex: 2 pizzas por preço promocional).
  + O sistema aplica os descontos automaticamente no momento do pagamento.
  + O administrador pode visualizar promoções ativas.
* **Prioridade**: Média
* **Estimativa de Esforço**: 4 pontos

**2.8 Entrega e Rastreamento**

* **ID da História de Usuário**: HU-008
* **Descrição**: Como atendente, eu quero gerenciar a entrega dos pedidos, atribuindo entregadores e monitorando o status de cada entrega.
* **Critérios de Aceitação**:
  + O cliente fornece o endereço de entrega durante o pedido.
  + O atendente ou sistema pode atribuir um entregador ao pedido.
  + O status da entrega pode ser monitorado em tempo real (em trânsito, entregue).
  + O sistema aplica a taxa de entrega, se houver.
* **Prioridade**: Média
* **Estimativa de Esforço**: 5 pontos

**2.9 Interface de Usuário (UI)**

* **ID da História de Usuário**: HU-009
* **Descrição**: Como usuário, eu quero uma interface gráfica amigável e responsiva para interagir com o sistema de forma intuitiva.
* **Critérios de Aceitação**:
  + A interface deve ter telas de login para funcionários e administradores.
  + As telas de gerenciamento de pedidos e estoque devem ser acessíveis.
  + A interface deve ser desenvolvida com Tkinter ou PyQt.
  + O sistema deve ter diferentes níveis de acesso (funcionários, atendentes e administradores).
* **Prioridade**: Alta
* **Estimativa de Esforço**: 6 pontos

**3. Plano de Iterações**

* **Iteração 1**: Implementação das funcionalidades básicas de cadastro de pizzas, gestão de pedidos e controle de estoque.
* **Iteração 2**: Sistema de pagamentos, relatórios financeiros e integração com a UI.
* **Iteração 3**: Promoções e descontos, entrega e rastreamento, melhorias na interface gráfica.

**4. Testes**

* **Testes de Unidade**: Testar a funcionalidade de cada módulo, como cadastro de pizzas, cálculo de preços e controle de estoque.
* **Testes de Aceitação**: Garantir que as histórias de usuário estão sendo entregues conforme os critérios de aceitação.
* **Testes Automatizados**: Testes de integração do sistema, garantindo que os módulos de pedidos, estoque e pagamento interajam corretamente.

**5. Métricas e Acompanhamento**

* **Burn-down Chart**: Acompanhar o progresso de cada iteração.
* **Velocity**: Monitorar o número de pontos de história completados por iteração.
* **Defeitos e Incidentes**: Registro de defeitos encontrados e solucionados.

**6. Reflexões e Melhoria Contínua**

* **Retrospectiva da Iteração**: Discutir o que funcionou bem e o que precisa ser melhorado.
* **Ajustes no Processo**: Adaptação contínua do processo de desenvolvimento conforme as necessidades da equipe e do projeto.

Esse modelo de documentação foi adaptado para atender aos princípios de **Extreme Programming (XP)**, priorizando a entrega de funcionalidades de valor para o cliente e a melhoria contínua do processo de desenvolvimento.

4o mini