# Sistema de Administração de Farmácia

## 1. Metodologias de Desenvolvimento

**Metodologia Escolhida:** Ágil (Scrum)  
**Justificativa:** Flexibilidade e adaptação contínua com entregas incrementais a cada sprint, permitindo evolução conforme o feedback dos stakeholders.

## 2. Apresentação do Projeto

**Objetivo:** Desenvolver um sistema web para gestão de remédios e compras, com funcionalidades de cadastramento, autenticação e autorização.

### Objetivos SMART:

1. **Específico:**
   * CRUD completo de remédios.
   * Interface intuitiva para visualização e compra de remédios.
   * Cadastramento e autenticação de usuários farmacêuticos e clientes.
2. **Mensurável:**
   * CRUD com no máximo 2 cliques por operação.
   * Suporte para 100 usuários simultâneos.
   * Processo de compra em até 5 minutos.
3. **Atingível:**
   * Equipe de programadores, analistas e gerente de projeto.
   * Utilização de tecnologias confiáveis para segurança e escalabilidade.
4. **Relevante:**
   * Melhoria na gestão de estoque e eficiência operacional.
   * Experiência de compra aprimorada para aumentar vendas e satisfação.
5. **Temporal:**

* **Planejamento e Design (1 mês):**
  + Fase inicial de levantamento de requisitos, wireframes, design de interface, e arquitetura do sistema.
  + Reuniões semanais para revisão de progresso e ajustes.
* **Desenvolvimento e Testes (Sprints de 2 semanas):**
  + **Sprint 1**: Configuração inicial do projeto, desenvolvimento do sistema de login e cadastro.
  + **Sprint 2**: Implementação do CRUD de remédios e gerenciamento de estoque.
  + **Sprint 3**: Desenvolvimento do carrinho de compras e fluxo de checkout.
  + **Sprint 4**: Testes de segurança e integração com serviços de pagamento.
  + **Sprint 5**: Otimização de desempenho e testes de carga.
* **Lançamento (6 meses):**
  + **Beta Público**: Lançamento para um grupo restrito de usuários para feedback.
  + **Lançamento Completo**: Disponibilização para todos os usuários, seguido por monitoramento e suporte pós-lançamento.

### ****Extras:****

* **Documentação**: Documentar todas as APIs, fluxos de trabalho, e criar manuais para usuários e administradores.
* **Treinamento**: Oferecer treinamento à equipe de suporte para garantir uma transição suave após o lançamento.

## 3. Escopo

### Farmacêutico:

* **Gestão Avançada de Remédios:**
  + CRUD completo e eficiente.
  + Controle dinâmico de estoque com ajuste automático.
  + Gestão de categorias e preços com promoções.
  + Relatórios detalhados sobre vendas e estoque.
* **Administração do Sistema:**
  + Gestão e controle de usuários farmacêuticos e clientes.
  + Configuração e ajuste de permissões e roles.

### Cliente:

* **Visualização Avançada de Remédios:**
  + Pesquisa e filtros detalhados.
  + Exibição de detalhes do produto com avaliações e imagens de alta resolução.
* **Compra Eficiente:**
  + Carrinho de compras dinâmico e checkout avançado.
  + Opções de pagamento variadas e suporte a cupons de desconto.
  + Histórico de compras e rastreamento de pedidos.

### Sistema:

* **Cadastramento e Autenticação:**
  + Registro seguro e autenticação multifatorial.
  + Gerenciamento de sessões e controle de atividades.
* **Funcionalidades Adicionais:**
  + Notificações e alertas para atualizações e ofertas.
  + Integração com gateways de pagamento.
  + Responsividade e acessibilidade em múltiplos dispositivos.

## 4. Cronograma

## ****1. Planejamento e Design (1 mês)****

* **Semana 1-2:**
  + **Levantamento de Requisitos:** Reuniões com stakeholders e documentação dos requisitos.
  + **Design de Interface:** Criação de wireframes e protótipos iniciais.
  + **Arquitetura do Sistema:** Definição das tecnologias e estrutura do sistema.
* **Semana 3-4:**
  + **Revisão e Ajustes:** Feedback dos stakeholders e ajustes no design e arquitetura.
  + **Preparação para Desenvolvimento:** Configuração do ambiente e planejamento das sprints.

## ****2. Desenvolvimento e Testes (Sprints de 2 semanas)****

* **Sprint 1-2 (Semanas 5-8):**
  + **Login e Cadastro:** Implementação do sistema de autenticação.
  + **CRUD de Remédios:** Desenvolvimento e testes das funcionalidades de gerenciamento de remédios.
* **Sprint 3-4 (Semanas 9-12):**
  + **Carrinho de Compras e Checkout:** Implementação do fluxo de compra e integração com serviços de pagamento.
  + **Testes de Segurança e Desempenho:** Otimização e testes de carga.

## ****3. Lançamento (6 meses)****

* **Semanas 13-16:**
  + **Beta Público:** Lançamento para um grupo restrito de usuários para feedback e ajustes.
  + **Lançamento Completo:** Disponibilização do sistema para todos os usuários, seguido por suporte pós-lançamento.

## 5. Recursos

* **Equipe de Desenvolvimento:** Programadores, analistas e gerente de projeto.
* **Infraestrutura:** Servidores em nuvem com alta disponibilidade e backup.
* **Frontend:** HTML, CSS, JavaScript, Bootstrap
* **Backend:** Laravel, MySQL
* **Infraestrutura:** AWS (Servidores em Nuvem)
* **Controle de Versão:** Git, GitHub
* **Metodologia:** Ágil (Scrum)

## 6. Análise de Riscos

**Riscos Identificados:**

* **Segurança:** Acesso não autorizado aos dados.
* **Indisponibilidade:** Queda do servidor.
* **Usabilidade:** Problemas na interface.

**Mitigação:**

* **Segurança:** Criptografia e autenticação em duas etapas.
* **Infraestrutura:** Servidores em nuvem e backup regular.
* **Usabilidade:** Testes contínuos e feedback dos usuários.

## 7. Diagramas de Desenvolvimento

### Diagrama de Casos de Uso:

* **Farmacêutico:** Cadastro, Login, criar, deletar, atualizar, visualizar remédios.
* **Cliente:** Cadastro, login, visualização e compra de remédios.
* **Qualquer usuário:** Home (produtos em promoção), Ajuda (acesso ao manual), sobre a empresa.

### Diagrama de Fluxo:

1. **Acesso ao Sistema:**
   * **Página Inicial:** Apresenta produtos em promoção, opção de login, e links para Ajuda e Sobre a Empresa.
   * **Login:**
     + **Usuário seleciona Login.**
     + **Verificação de Credenciais:**
       - **Sucesso:**
         * **Farmacêutico:** Redirecionado para o Dashboard Farmacêutico.
         * **Cliente:** Redirecionado para o Dashboard do Cliente.
       - **Falha:** Retorno à página de login com mensagem de erro.
2. **Farmacêutico:**
   * **Dashboard Farmacêutico:**
     + **Acesso a CRUD de Remédios:**
       - **Criar:** Adiciona novo remédio ao sistema.
       - **Atualizar:** Modifica informações de um remédio existente.
       - **Deletar:** Remove um remédio do sistema.
       - **Visualizar:** Consulta detalhes dos remédios cadastrados.
     + **Logout:** Encerramento da sessão.
3. **Cliente:**
   * **Dashboard do Cliente:**
     + **Visualização de Remédios:** Acesso à lista de remédios disponíveis.
     + **Compra de Remédio:**
       - **Seleção de Remédio:** Cliente escolhe um remédio.
       - **Adicionar ao Carrinho:** Remédio é adicionado ao carrinho de compras.
       - **Confirmação:** Cliente revisa os itens no carrinho e confirma a compra.
       - **Finalização da Compra:** Conclusão da transação, com detalhes de pagamento e envio.
     + **Logout:** Encerramento da sessão.
4. **Qualquer Usuário:**
   * **Acesso sem Login:**
     + **Página Inicial:** Visualização de produtos em promoção.
     + **Ajuda:** Acesso ao manual do sistema.
     + **Sobre a Empresa:** Informações sobre a farmácia.

### Diagrama de Classes:

#### 1. Classe: **Produto**

* **Atributos:**
  + ID: int — Identificador único do produto.
  + Nome: string — Nome do produto.
  + Descrição: string — Descrição do produto.
  + Preço: decimal — Preço do produto.
  + Quantidade: int — Quantidade disponível em estoque.
  + Imagem: string — Caminho para a imagem do produto (opcional).
* **Métodos:**
  + criarProduto()
  + atualizarProduto()
  + removerProduto()
  + listarProdutos()

#### 2. Classe: **Cliente**

* **Atributos:**
  + ID: int — Identificador único do cliente.
  + Nome: string — Nome do cliente.
  + Email: string — Endereço de e-mail do cliente.
  + Senha: string — Senha para autenticação.
  + Endereço: string — Endereço de entrega (opcional).
* **Métodos:**
  + registrar()
  + atualizarPerfil()
  + comprarRemedio(produto: Produto, quantidade: int)
  + consultarHistoricoCompras()

#### 3. Classe: **Venda**

* **Atributos:**
  + ID: int — Identificador único da venda.
  + ID\_produto: int — Identificador do produto vendido (referência a Produto).
  + ID\_cliente: int — Identificador do cliente que fez a compra (referência a Cliente).
  + Quantidade: int — Quantidade de produtos vendidos.
  + Status: string — Status da venda (e.g., "Pendente", "Concluída", "Cancelada").
  + Data: datetime — Data e hora da venda.
* **Métodos:**
  + registrarVenda()
  + atualizarStatus()
  + listarVendas()

### Relacionamentos:

* **Produto** e **Venda**:
  + Um Produto pode estar associado a muitas Vendas.
  + Cada Venda refere-se a um Produto.
* **Cliente** e **Venda**:
  + Um Cliente pode fazer muitas Vendas.
  + Cada Venda é associada a um Cliente.

### Diagrama Visual (Descrição):

1. **Produto**:
   * Atributos e métodos descritos acima.
   * Relacionado a **Venda** (um-para-muitos).
2. **Cliente**:
   * Atributos e métodos descritos acima.
   * Relacionado a **Venda** (um-para-muitos).
3. **Venda**:
   * Atributos e métodos descritos acima.
   * Relacionado a **Produto** e **Cliente** (muitos-para-um).