**Documento de Arquitetura de Software**

* **Sistema de Gerenciamento de Academia XYZ**

1. **Visão Geral do Sistema**
   * **Objetivo do Sistema**: O sistema permitirá que os membros da "Academia XYZ" se cadastrem, agendem aulas, realizem pagamentos e acessem relatórios sobre frequência e desempenho.
   * **Principais Funcionalidades**:
     + Cadastro e gerenciamento de membros.
     + Agendamento e cancelamento de aulas.
     + Controle e registro de pagamentos.
     + Geração de relatórios sobre frequência e modalidades.
   * **Requisitos Não Funcionais**:
     + **Desempenho**: Capacidade de atender até 1000 membros simultaneamente.
     + **Segurança**: Implementação de segurança robusta para proteger dados pessoais e financeiros.
     + **Usabilidade**: Interface intuitiva e compatibilidade com dispositivos móveis.
2. **Descrição dos Componentes**
   * **Componentes Principais**:
     + **Frontend (UI/UX)**: Interface para membros realizarem cadastro, agendarem aulas e acessarem relatórios.
       - **Tecnologia**: React.js
       - **Função**: Proporcionar uma experiência amigável ao usuário e interagir com o backend.
     + **Módulo de Cadastro de Membros**: Gerencia informações dos membros, incluindo cadastro, atualização e exclusão.
       - **Tecnologia**: Node.js (Express)
       - **Função**: API para gerenciar dados pessoais dos membros.
     + **Módulo de Agendamento de Aulas**: Permite que os membros agendem, cancelem ou reajam suas aulas.
       - **Tecnologia**: Node.js (Express)
       - **Função**: Controlar as reservas de aulas e verificar disponibilidade.
     + **Módulo de Pagamentos**: Registra e controla os pagamentos dos membros, gerando alertas para pendências.
       - **Tecnologia**: Java (Spring Boot)
       - **Função**: Processar pagamentos e gerenciar o status das mensalidades.
     + **Módulo de Relatórios**: Gera relatórios sobre frequência e utilização das aulas.
       - **Tecnologia**: Python (Django)
       - **Função**: Analisar dados e apresentar informações para a administração.
     + **Banco de Dados**: Armazena informações sobre membros, aulas, pagamentos e relatórios.
       - **Tecnologia**: PostgreSQL
       - **Função**: Persistir dados de forma segura e acessível.
3. **Diagramas de Componentes**
   * **Diagrama de Componentes do Sistema de Gerenciamento de Academia**: Visualização da interação entre os componentes do sistema.
   * **Explicação do Diagrama**: Descrição das interações entre frontend, backend e banco de dados, e como cada módulo se comunica.
4. **Definição de Interfaces e Protocolos de Comunicação**
   * **Interfaces do Sistema**:
     + Frontend → Backend: Comunicação via API REST em formato JSON.
     + **Protocolos de Comunicação**:
       - **Protocolo de Segurança**: Uso de HTTPS para proteger dados durante a transmissão.
       - **Autenticação**: Implementação de autenticação segura para garantir acesso autorizado.
5. **Aspectos de Segurança, Escalabilidade e Manutenibilidade**
   * **Segurança**: Proteção de dados sensíveis e controle de acesso. Implementação de autenticação forte e criptografia de dados.
   * **Escalabilidade**: Sistema preparado para expandir horizontalmente e atender a um número crescente de membros.
   * **Manutenibilidade**: Arquitetura modular que facilita a atualização e manutenção contínua do sistema, além de testes automatizados.
6. **Conclusão**
   * O documento de arquitetura define a estrutura do sistema de gerenciamento da "Academia XYZ", destacando a interconexão dos componentes, as funcionalidades essenciais e as práticas de segurança. A abordagem modular e a interface intuitiva garantem que o sistema atenda às necessidades dos membros e da administração de forma eficiente.