



Lab3 CSI-28

Equipe:

Hyuri Fragoso

Iago Jacob

Matheus Militão

Pedro Ulisses

Welberson Franklin

Nome do Sistema: Placebo

Instituto Tecnológico de Aeronáutica – ITA / IEC

Disciplina: Fundamentos da Engenharia de Software - CSI-28

1 Sobre o Sistema

a) Descrição do Sistema em desenvolvimento

Será um sistema de informação que servirá para o controle de remédios, consultas e dados de saúde de um paciente, com o disparo de alarmes nos horários em que ele deve tomar os seus remédios. O interesse neste sistema surge do fato de que o mercado carece de ferramentas, de utilização simples, para registrar e acompanhar a saúde de pessoas idosas.

Além de servir de alarme, o sistema terá um banco de dados on-line no qual pode ser atualizada a lista de remédios e a disposição deles na rotina de um paciente. Para tanto, pretende-se conceder diferentes níveis de acesso ao sistema, ambos no mesmo aplicativo de celular: o primeiro nível, mais simples, seria puramente o aplicativo com o alarme dos remédios; o segundo, em contrapartida, seria para cuidadores, responsáveis e/ou médicos do paciente, que poderiam, a partir de seus próprios celulares, acrescentar, cortar ou remanejar remédios da lista dos pacientes. Por haver um banco de dados on-line, seria também possível aos responsáveis, por mais que à distância do paciente, verificar se ele está realmente tomando os remédios e, caso sim, se está tomando nas horas certas.

b) Modelo de Processo a ser adotado no desenvolvimento do projeto

O modelo a ser adotado é o incremental. Decidiu-se utilizar esse modelo para que o desenvolvimento seja dado em pacotes, estes sendo, nesta ordem: projeto de frontend com Kivy, conexão do frontend com o backend por meio do módulo UrlRequests do Kivy, desenvolvimento do backend com Django, desenvolvimento e aplicação e integração do banco de dados (com o gerenciador Postgres) ao backend.

c) Diagrama que representa a arquitetura do sistema

Decidiu-se que a arquitetura mais apropriada ao nosso sistema seria a em três camadas, visto que as ferramentas que utilizaremos se adequam bem a essa organização, isto é, a interface gráfica com Kivy, a lógica de negócio com Django e o banco de dados com o SGBD Postgres. Abaixo, na Figura 1, é mostrado um diagrama ilustrativo da arquitetura escolhida.

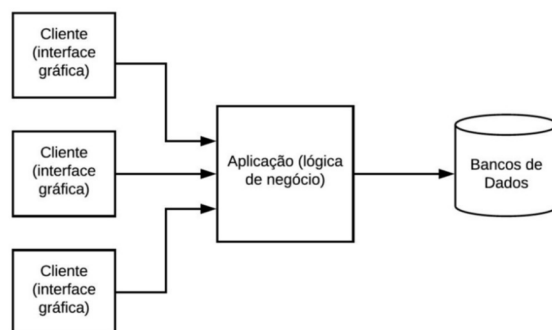


Figura 1: Diagrama ilustrativo da Arquitetura em 3 camadas.

d) Diagramas de Classe

Na Figura 2, é mostrado o diagrama de classes correspondente ao frontend, e, na Figura 3, os de backend.

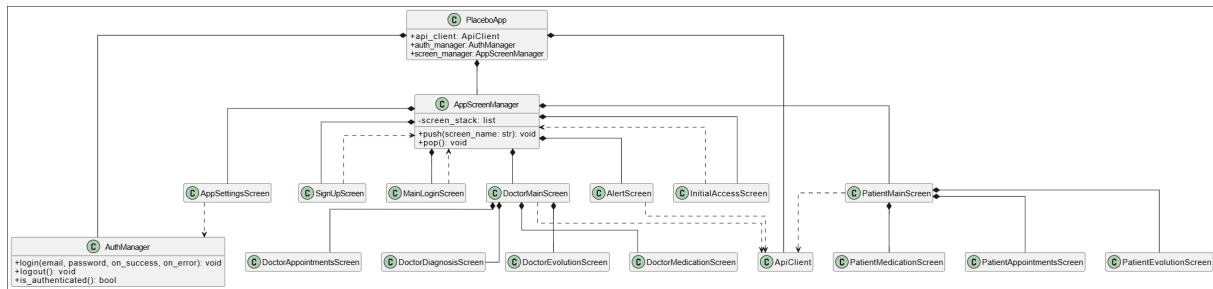


Figura 2: Diagrama de classes correspondente aos principais elementos gráficos do frontend.

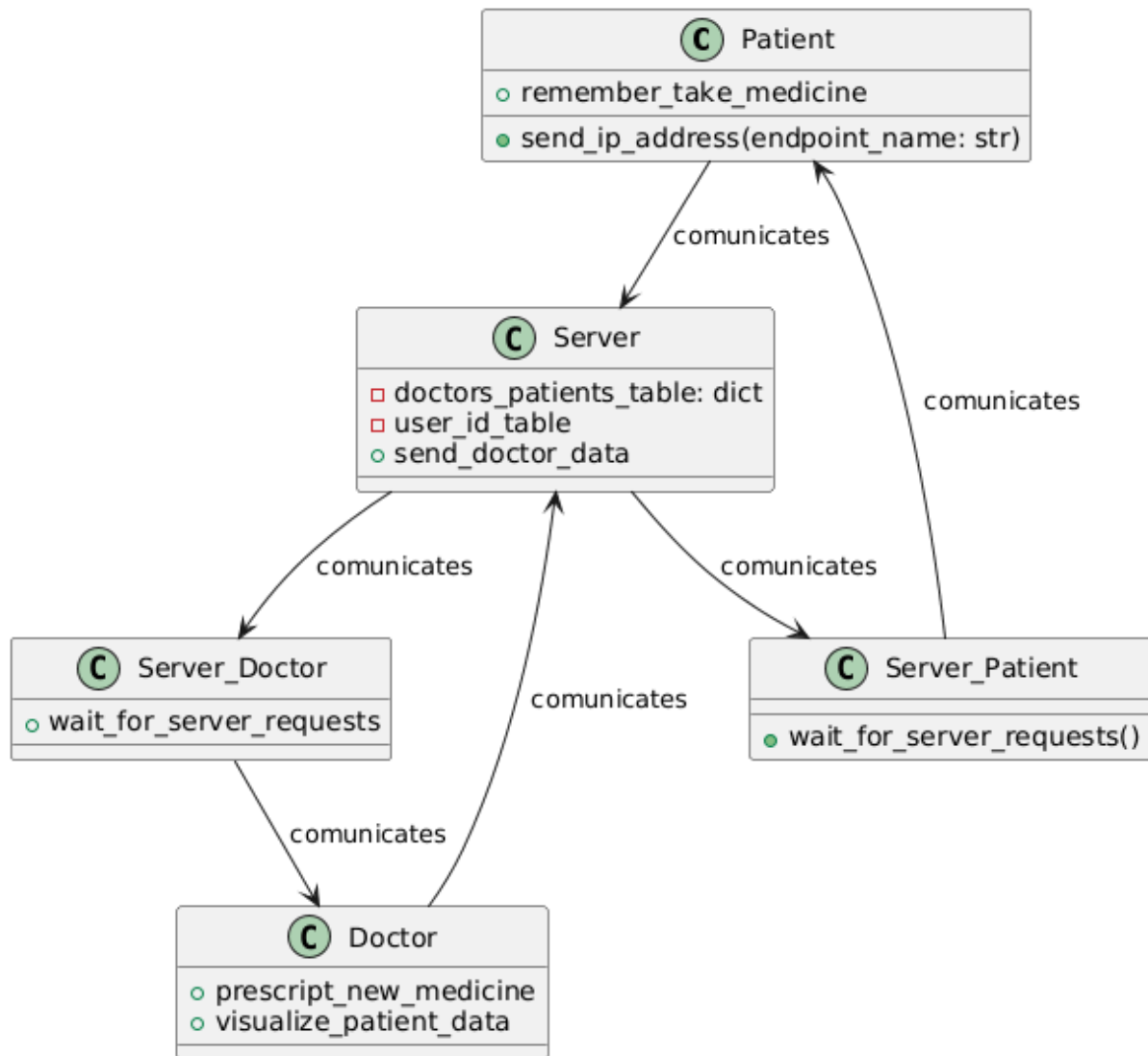


Figura 3: Diagrama de classes correspondente aos principais elementos web do backend.

2 Sobre o Projeto

a) Divisão de tarefas da equipe

A divisão considera as responsabilidades principais e as dedicações/competências/experiências relevantes de cada membro, alinhadas ao processo incremental e à arquitetura em três camadas já definidas pelo grupo.

Membro	Responsabilidades	Dedicações/Competências/Experiências Relevantes
Pedro Ulisses	Frontend (Kivy)	Prototipação de telas e fluxos; criação de widgets e layouts Kivy; boas práticas de UX; integração com serviços assíncronos; navegação entre telas.
Iago	Frontend (Kivy)	Componentização e navegação; validação de entradas; telas de agenda/alarme; empacotamento mobile; ajustes visuais e feedbacks de rede.
Hyuri	Backend (Django) + API	Django/Django REST; definição de modelos e lógica de negócio; autenticação/autorizações; endpoints REST; testes básicos e documentação curta de API.
Welberson	Backend (Django) + UrlRequests	Integração Kivy-API via <code>UrlRequest</code> ; serialização JSON; tratamento de erros/timeouts; versionamento simples de endpoints; logs básicos.
Matheus	Banco de Dados (PostgreSQL)	Modelagem relacional; migrações e integridade referencial; consultas/índices essenciais; scripts de carga/seed para demonstração; consistência de dados.

b) Cronograma de desenvolvimento do protótipo inicial

O cronograma parte da **2ª semana** e vai até a **8ª semana** (encerramento do protótipo), contemplando apenas as etapas essenciais previstas pelo grupo (Kivy, UrlRequests, Django e Postgres) no processo incremental.

Semana	Entregáveis Principais	Atividades-Chave (com responsáveis)
2	Versão inicial do Relatório Descritivo	Consolidação de escopo mínimo do protótipo; priorização de requisitos; alinhamento de tecnologias e arquitetura (todos).
3	<i>UI skeleton</i> navegável (mock)	Protótipos de telas/fluxos no Kivy (Pedro, Iago); esboço dos contratos da API (Hyuri, Welberson); rascunho do modelo relacional (Matheus).
4	API esqueleto + handshake app↔API	Projeto Django com rotas básicas e respostas <i>mock</i> (Hyuri); teste de integração Kivy via <code>UrlRequest</code> (Welberson, Pedro, Iago).
5	Persistência inicial funcional	Modelo no Postgres com migrações (Matheus); CRUD mínimo nos endpoints (Hyuri); frontend consumindo dados reais nas telas prioritárias (Iago, Pedro); tratamento básico de erros/timeouts (Welberson).
6	Fluxos centrais do protótipo (<i>MVP</i>)	Alarme de remédios e lista/agenda ponta-a-ponta; autenticação simples no backend (Hyuri); ajustes de UX e estados de carregamento no app (Pedro, Iago, Welberson); ajustes pontuais no esquema/índices (Matheus).
7	Testes e estabilização	Testes de sistema e correções; <i>seed</i> de dados para demonstração (Matheus); documentação curta dos endpoints (Hyuri) e de integração no app (Welberson); polimento visual (Pedro, Iago).
8	Versão final pronta para entrega	Gerar a versão final do app para demonstrar; salvar uma cópia do banco de dados para quem for testar; preparar um passo a passo simples de como rodar; montar um guia rápido de uso e um roteiro de apresentação; revisão final do relatório (todos).