LEVANTAMENTO DE REQUISITO PIE\_2024

## Requisitos Funcionais

* **Autenticação de Usuários:**
  + Relacionado à classe Usuario, que contém atributos como Login, Senha e Email. O sistema deve implementar uma funcionalidade de autenticação para validar essas credenciais.
* **Buscar hospitais no mapa:**
  + A classe Hospital possui informações como Endereco (associada à entidade Endereco) e contato. A integração com mapas não está diretamente detalhada no diagrama, mas os dados necessários estão presentes.
* **Consultar a lotação dos hospitais:**
  + Representado pelos atributos CapacidadePaciente e CapacidadeLeitos na classe Hospital. A entidade Leito detalha a ocupação por especialidade e status, como LEITO\_OCUPADO e LEITO\_ABERTO.
* **Solicitar Prontuário do Paciente**
  + Baseado na entidade Prontuario, o sistema deve possibilitar a solicitação e exibição de prontuários de pacientes armazenados no hospital.

## Requisitos Não Funcionais

* **Tipo de Sistema Operacional:**
  + Não está representado no diagrama, mas é compatível com o backend em Java (SpringBoot) e frontend em ReactJS, atendendo a sistemas Android, iOS, Windows, Linux.
* **Banco de dados:**
  + A estrutura do diagrama está bem organizada para ser implementada em um banco relacional como PostgreSQL.
* **APIs RESTful:**
  + O diagrama sugere possíveis endpoints baseados nas classes Paciente, Hospital, Leito e TransferenciaPaciente.
* **Dispositivos compatíveis:**
  + A modelagem não limita a interface, permitindo uso em dispositivos móveis e navegadores.
* **Arquitetura Monolito:**
  + A simplicidade do diagrama reflete uma modelagem compatível com essa arquitetura.

## Requisitos de Produto Final

* **Velocidade de execução, latência e taxa de erros:**
  + Não são refletidos no diagrama, mas dependem da implementação eficiente das entidades e APIs.
* **Segurança:**
  + A classe Usuario é um ponto inicial para a autenticação. A adição de criptografia para as senhas deve ser considerada.
* **Confiabilidade e consistência:**
  + O uso de atributos como Status e Especialidade assegura que o estado dos recursos seja bem definido.

## Requisitos Organizacionais

* **Infraestrutura:**
  + O sistema está modelado para se integrar com uma infraestrutura de banco de dados e APIs.
* **Conexão e criptografia:**
  + Os requisitos para RESTful APIs em HTTPS devem ser garantidos, mas não estão explícitos no diagrama.
* **Compatibilidade com outras linguagens:**
  + A modelagem pode se integrar com serviços existentes via APIs.

## Requisitos Externos

* **Localização geográfica:**
  + A entidade Endereco facilita a integração com serviços de mapas.
* **Legislação e proteção de dados:**
  + É essencial garantir conformidade com a LGPD para manipular dados sensíveis dos pacientes (CPF, Identidade, etc.).

### Fluxograma de Funcionalidades

1. **Início**
   1. O sistema é acessado.
2. **Login**
   1. Usuário insere credenciais.
      1. **Validação bem-sucedida**: O usuário acessa as funcionalidades.
      2. **Falha na validação**: Acesso é rejeitado.
3. **Busca de Hospitais**
   1. Usuário insere critérios de busca (localização, especialidade).
   2. O sistema retorna uma lista de hospitais no mapa com detalhes.
4. **Consulta de Lotação**
   1. Usuário seleciona um hospital.
   2. O sistema exibe detalhes da lotação (número de leitos abertos/ocupados).
5. **Transferência de Paciente**
   1. Usuário registra o paciente no sistema.
   2. Define hospital de saída e entrada.
   3. Aloca uma ambulância para o transporte.
6. **Aceitar/Recusar Transferência**
   1. Hospitais recebem pedidos de transferência.
   2. Decidem aceitar ou negar a entrada do paciente.
7. **Solicitação de Prontuários**
   1. Usuário verifica se o prontuário do paciente existe.
   2. Caso exista, é possível solicitar o documento.