

Resumo do Levantamento de Requisitos do Sistema de Gestão Hoteleira

Este sistema será uma **aplicação TUI (Text-based User Interface)** desenvolvida em **Java**, utilizando **PostgreSQL** como banco de dados, e operará localmente.

Objetivo Principal

O sistema visa o **gerenciamento abrangente do hotel**, facilitando a dinâmica tanto para os **hóspedes (clientes)** quanto para os **administradores e funcionários**, funcionando como um sistema de gestão completo para as operações hoteleiras.

Stakeholders

Os principais usuários do sistema serão:

- **Clientes/Hóspedes:** Pessoas que utilizam os serviços do hotel.
- **Funcionários:** Operadores do hotel (ex: recepcionistas).
- **Administradores:** Gerentes e responsáveis pela gestão total do hotel.

Requisitos Funcionais (O que o sistema deve fazer)

Perspectiva do Cliente/Hóspede (TUI)

O cliente interagirá via terminal e poderá:

- **Verificar Disponibilidade e Fazer Reservas:**
 - Consultar quartos disponíveis por datas.
 - Fazer novas reservas, inserindo seus dados.
 - Visualizar detalhes textuais dos tipos de quartos (sem fotos).
- **Registrar Pagamentos:** Informar e registrar o método de pagamento (sem processamento online).
- **Consultar Histórico:** Ver suas reservas passadas e futuras.
- **Acessar Informações do Hotel:** Visualizar dados básicos do hotel (endereço, contato, políticas, descrição de amenidades).

Perspectiva do Administrador/Funcionário (TUI)

Os administradores e funcionários, via terminal, poderão:

- **Gerenciar Reservas:**
 - Criar, modificar e cancelar reservas.
 - Pesquisar e visualizar todas as reservas.
 - Realizar check-ins e check-outs.
 - Visualizar um calendário de ocupação textual.

- **Gerenciar Quartos:**
 - Cadastrar e editar tipos e quartos individuais.
 - Atualizar o status de manutenção/limpeza dos quartos.
 - Verificar a disponibilidade de quartos.
- **Gerenciar Hóspedes:**
 - Cadastrar, editar e pesquisar informações de hóspedes.
 - Visualizar o histórico de reservas de cada hóspede.
- **Gerenciar Financeiro (Básico):**
 - Registrar pagamentos de reservas.
 - Visualizar saldos de reservas (pendências).
 - Gerar relatórios financeiros simples (textuais).
- **Gerenciar Funcionários/Usuários:**
 - Cadastrar e editar contas de funcionários.
 - Definir diferentes níveis de permissão.
 - Redefinir senhas de acesso.

Requisitos Não Funcionais (Como o sistema deve ser)

- **Segurança:**
 - Proteção rigorosa de dados sensíveis (pagamentos, informações pessoais de hóspedes, credenciais).
 - Autenticação segura (login com senha forte para *todos* os usuários).
 - Controle de acesso baseado em perfis (permissões).
 - Rotinas de backup de dados.
- **Usabilidade:**
 - **Extremamente intuitivo e fácil de usar** para o cliente (hóspedes) via TUI.
 - Clareza e poucos cliques para tarefas comuns para funcionários.
 - Mensagens claras de erro e sucesso.
- **Manutenibilidade:**
 - Código bem estruturado, documentado e modular para facilitar futuras modificações e adições de funcionalidades.
- **Escalabilidade:**
 - Apesar de TUI inicial, a arquitetura deve considerar que o hotel pode crescer em número de quartos e volume de dados.

Estrutura do Banco de Dados (PostgreSQL)

As principais entidades (tabelas) e seus atributos essenciais serão:

- **Usuario/Pessoa:**
 - `id_usuario` (PK)
 - `nome_completo`
 - `email`
 - `telefone`
 - `data_nascimento`
 - `role` (Hóspede, Funcionário, Administrador)
 - `senha_hash` (para todos os usuários)

- **Quarto:**
 - id_quarto (PK)
 - numero_quarto (Único)
 - tipo_quarto (Standard, Luxo, Suíte, etc.)
 - preco_diaria
 - estado_manutencao (Limpo, Sujo, Em Manutenção)
 - ocupado (Boolean)
- **Reserva:**
 - id_reserva (PK)
 - id_usuario (FK para Usuario/Pessoa)
 - id_quarto (FK para Quarto)
 - data_check_in
 - data_check_out
 - status_reserva (Confirmada, Pendente, Cancelada, etc.)
 - valor_total
 - status_pagamento (Pago, Pendente, Parcialmente Pago)
 - data_reserva