

Hospedar

Versão 0.1 - Desenvolvimento

Versão: v0.1.

Data de Início: 16 de novembro de 2024

Descrição: Esta é a versão inicial do projeto, em fase de desenvolvimento. Inclui a documentação preliminar com a visão geral, requisitos funcionais e estrutura básica do design.

Estado: Em desenvolvimento, sujeito a alterações constantes.

Observação: Por se tratar de um projeto pessoal, não há previsão de lançamento oficial. Atualizações serão realizadas conforme a evolução do projeto

Data de Atualização: 30 de novembro de 2024

Autor: Silas Sefas de Souza Aquino

1. Visão Geral
 - 1.1. Objetivo do Projeto
 - 1.2. Usuários
 - 1.3. Escopo
 - 1.3.1. Gestão do Hotel
 - 1.3.2. Gestão de Pessoal
 - 1.4. Stakeholders
 - 1.5. Prazos e recursos
2. Requisitos
 - 2.1. Requisitos Funcionais
 - 2.2. Requisitos Não Funcionais
 - 2.3. Casos de Uso/User Stories
3. Design
 - 3.1. Arquitetura do Sistema
 - 3.2. Estrutura de Dados
 - 3.3. Fluxo de Processos
4. Documentação Técnica
 - 4.1. Configuração do Ambiente
 - 4.2. Instruções de Deploy
 - 4.3. APIs e Integrações
5. Testes
 - 5.1. Plano de Testes
 - 5.2. Casos de Teste
 - 5.3. Relatórios de Bugs
6. Manual do Usuário
 - 6.1. Instruções para Usuários Finais
 - 6.2. FAQ
 - 6.3. Guia de Solução de Problemas
7. Manutenção
 - 7.1. Registro de Alterações
 - 7.2. Planos de Atualização
 - 7.3. Backup e Recuperação

Visão Geral

1.1- Objetivo do Projeto

O **Hospedar** é uma plataforma web destinada ao gerenciamento de hotéis, com o objetivo de otimizar e automatizar processos administrativos e operacionais. O sistema permitirá que os administradores do hotel possam:

- Cadastrar e gerenciar acomodações, como quartos e suítes.
- Gerenciar reservas feitas por hóspedes e colaboradores.
- Gerenciar hóspedes, incluindo cadastro, histórico de estadias e check-in/check-out.
- Gerenciar insumos e estoque de itens utilizados nos quartos e nas áreas comuns.
- Auxiliar na administração geral, como gestão de pessoal e controle financeiro.

A plataforma permitirá que os administradores controlem a disponibilidade dos quartos, otimizem as operações de recepção, e tenham uma visão ampla da administração do hotel por meio de relatórios e indicadores.

1.2 - Usuários

1. Administradores e gestores de hotéis
2. Recepcionistas
3. Clientes e Hospedes de hotéis

1.3 - Escopo

1.3.1 Gestão do Hotel

1. Cadastro e Gerenciamento de Acomodações:
 - 1.1. Cadastro de diferentes tipos de acomodações.
 - 1.2. Quarto Individual
 - 1.3. Quarto Duplo
 - 1.4. Quarto Triplo
 - 1.5. Suíte
 - 1.6. Família
 - 1.7. Personalizado
2. Fazer o quantitativo de refeições com base no número de check-in e check-out
3. Registro de check-in e check-out, atualizando automaticamente alterando o status dos quartos.
 - 3.1. Status Possíveis
 - 3.1.1. Disponível: O quarto está livre e pode ser reservado.
 - 3.1.2. Ocupado: O quarto já está sendo utilizado por um hóspede.
 - 3.1.3. Reservado: O quarto foi reservado, mas o hóspede ainda não chegou.
 - 3.1.4. Em Limpeza: O quarto está sendo limpo ou preparado para um novo hóspede.
 - 3.1.5. Em Manutenção: O quarto não está disponível devido a problemas de manutenção.
 - 3.1.6. Check-in pendentes: O hóspede ainda não fez o check-in.
 - 3.1.7. Check-out pendentes: O hóspede está prestes a fazer o check-out.
 - 3.1.8. Bloqueado: O quarto foi bloqueado para não ser vendido, geralmente por motivos internos.
4. Gerenciamento de Reservas:
 - 4.1. Reservas por telefone
 - 4.2. Check-in sem reservas, direto na recepção
 - 4.3. Gerenciar, confirmar, modificar e cancelar reservas.

5. Cadastro e Gerenciamento de Hóspedes:

5.1. Dados do hospede

5.1.1.	Nome	5.1.5.	Nacionalidade
5.1.2.	CPF	5.1.6.	Telefone
5.1.3.	RG	5.1.7.	E-mail
5.1.4.	Sexo	5.1.8.	Histórico

6. Gestão de Insumos e Estoque:

6.1. Cadastro de Itens para refeição

- 6.1.1. Nome
- 6.1.2. Estoque
- 6.1.3. Estoque Mínimo
- 6.1.4. Valor de compra

6.2. Controle de Itens de limpeza e manutenção

- 6.2.1. Quantidade de Roupas de cama disponível
- 6.2.2. Gestão de produtos de limpeza em geral
- 6.2.3. Nome
- 6.2.4. Estoque
- 6.2.5. Estoque Mínimo
- 6.2.6. Valor de compra

6.3. Relatório de estoque

6.4. Indicar necessidade de compra

7. Relatórios de consumo e alertas de estoque baixo.

- 7.1. Permitir emitir relatório de estoque de produtos
- 7.2. Emitir alerta de quando o produto atingir o estoque mínimo

8. Visualização da taxa de ocupação do hotel por data.

- 8.1. Emitir relatório de ocupação
- 8.2. Mostrar status de ocupação do hotel

9. Controle de preço, fotos e descrições.

- 9.1. Permitir criar demonstrativo de quartos para integração web e futuras

1.3.2 Gestão de Pessoal

1. Cadastro de funcionários e controle de escalas.

- 1.1. Cadastro de Pessoal (funcionários)

- 1.1.1. Nome completo
- 1.1.2. Data de nascimento
- 1.1.3. Número de CPF
- 1.1.4. Número de identidade (RG)
- 1.1.5. Estado civil
- 1.1.6. Sexo
- 1.1.7. Nacionalidade
- 1.1.8. Naturalidade (cidade e estado de nascimento)
- 1.1.9. Nome dos pais
- 1.1.10. Endereço completo
 - 1.1.10.1. Rua
 - 1.1.10.2. Número
 - 1.1.10.3. Bairro
 - 1.1.10.4. Cidade
 - 1.1.10.5. Estado
 - 1.1.10.6. CEP
- 1.1.11. Telefone de contato
- 1.1.12. E-mail
- 1.1.13. Cargo/ função
- 1.1.14. Data de admissão
- 1.1.15. Salário ou remuneração
- 1.1.16. Tipo de contrato
- 1.1.17. Jornada de trabalho
- 1.1.18. Departamento/setor
- 1.1.19. Forma de pagamento
 - 1.1.19.1. Mensal
 - 1.1.19.2. Quinzenal
 - 1.1.19.3. Semanal
- 1.1.20. Banco e agência para depósito de salário
- 1.1.21. Conta bancária (para pagamento)
- 1.1.22. Número de PIS/PASEP
- 1.1.23. Carteira de Trabalho e Previdência Social (CTPS) –
 - 1.1.23.1. Número
 - 1.1.23.2. Série
- 1.1.24. Informações sobre dependentes (se houver)
- 1.1.25. Informações sobre benefícios

- 1.1.25.1. Vale-transporte
- 1.1.25.2. Vale-refeição
- 1.1.25.3. Plano de saúde, etc.)
- 1.1.26. Imposto de Renda (se aplicável)
- 1.1.27. Acompanhamento de desempenho
- 1.1.28. Histórico de treinamentos realizados
- 1.1.29. Férias (datas e saldo de dias)
- 1.1.30. Histórico de Licenças (médicas, maternidade, etc.)
- 1.1.31. Histórico de Afastamentos (se houver)

1.4- Stakeholders

1 Desenvolvedor (Silas Sefas de Souza Aquino)

1.1. **Descrição:** Responsável pelo desenvolvimento, documentação e evolução do projeto.

1.2. **Interesse:**

- Aprendizado e consolidação de habilidades em desenvolvimento web.
- Criação de um portfólio para demonstrar competências técnicas.
- Possível uso comercial no futuro.

2. Stakeholders indiretos

2.1. **Usuários Futuros** (Administradores Hoteleiros)

2.2. **Descrição:** Caso o projeto seja comercializado, os principais usuários seriam administradores de hotéis ou responsáveis pela gestão operacional.

2.3. **Interesse:**

- Automação de tarefas administrativas e operacionais.
- Acesso a relatórios e indicadores de desempenho.

3. Clientes Finais (Hóspedes)

3.1. **Descrição:** Hóspedes que podem interagir com o sistema para realizar reservas, consultar informações ou acompanhar o status de suas estadias.

3.2. **Interesse:** Interface amigável, acesso rápido a informações e confiabilidade.

4. Stakeholders Potenciais

4.1. **Investidores ou Parceiros Comerciais**

4.2. **Descrição:** Pessoas ou empresas interessadas em transformar o projeto em um produto comercial.

4.3. **Interesse:**

- Obter retorno financeiro ou participação em um produto inovador.
- Colaborar no desenvolvimento de funcionalidades adicionais.

5. Empresas de Tecnologia e Startups

5.1. **Descrição:** Organizações que podem contratar o autor ou incorporar a solução em seus portfólios.

5.2. **Interesse:** Aproveitar o sistema como uma base ou inspiração para novos produtos.

1.5 Prazos e Recursos

- 1** Prazos: Por ser um projeto voltado para a fixação e aprendizado do processo de desenvolvimento de software, não há um prazo final rígido ou data limite estabelecida. No entanto, para garantir um progresso constante, planejo realizar atualizações semanais do projeto, sempre aos sábados, de acordo com o meu cronograma de estudos.
- 2** **Recursos:** Para o desenvolvimento deste projeto, utilizarei todos os recursos didáticos disponíveis, como livros, vídeos, cursos e outras fontes de aprendizado. Além disso, farei uso de ferramentas baseadas em Inteligência Artificial para consultas e esclarecimento de dúvidas durante o processo de estudo e desenvolvimento.

Nota: Como o objetivo principal deste projeto é o aprendizado, comprometo-me a não utilizar ferramentas de IA para gerar código ou resolver problemas técnicos de maneira automatizada. Elas serão empregadas apenas para consultas e pesquisas, a fim de aprofundar meu entendimento e otimizar a busca por soluções.

Requisitos

1. Requisitos Funcionais

- RF01: O sistema deve permitir o cadastro e edição de quartos, incluindo tipo, preço e disponibilidade.
- RF02: O sistema deve permitir a reserva de quartos, de acordo com a disponibilidade e os critérios do hóspede.
- RF03: O administrador deve poder alterar, cancelar ou confirmar reservas feitas.
- RF04: O sistema deve registrar e atualizar automaticamente as datas de check-in e check-out.
- RF05: O sistema deve gerar alertas de estoque baixo e permitir a atualização de itens de consumo.
- RF06: O sistema deve gerar relatórios financeiros, de ocupação e desempenho do hotel.
- RF07: O sistema deve permitir o cadastro de hóspedes com informações detalhadas (nome, CPF, histórico de estadias, etc.).
- RF08: O sistema deve permitir a gestão de status de quartos, como "Disponível", "Ocupado", "Reservado", entre outros.

RF09: O sistema deve gerar relatórios detalhados sobre a taxa de ocupação do hotel em períodos específicos.

RF10: O sistema deve permitir reservas realizadas por diferentes canais (web, telefone, presencial).

Requisitos Não Funcionais

RNF01: A interface do sistema deve ser responsiva, funcionando bem em dispositivos móveis, tablets e desktops.

RNF02: O tempo de resposta do sistema deve ser inferior a 2 segundos para ações principais (reservas, consultas).

RNF03: O sistema deve suportar até 500 reservas simultâneas e 200 quartos cadastrados.

RNF04: O sistema deve garantir a segurança de dados com autenticação de usuário (administradores, funcionários e hóspedes).

RNF05: O sistema deve assegurar integridade de dados nas transações de reserva, check-in e check-out.

Casos de Uso

CU01: Como administrador, quero cadastrar novos quartos no sistema para manter as informações do hotel sempre atualizadas.

CU02: Como administrador, quero gerenciar as reservas realizadas para confirmar, alterar ou cancelar quando necessário.

CU03: Como administrador, quero registrar o check-in e check-out dos hóspedes para manter o controle de ocupação do hotel.

CU04: Como administrador, quero gerenciar o cadastro de hóspedes, incluindo suas informações e histórico de estadias.

CU05: Como administrador, quero receber alertas automáticos de estoque baixo para reabastecimento rápido.

CU06: Como administrador, quero gerar relatórios detalhados sobre a taxa de ocupação do hotel em datas específicas.

Design

1. Arquitetura do Sistema

A arquitetura será dividida em três camadas principais: Frontend, Backend e Banco de Dados. Usaremos uma arquitetura moderna baseada em RESTful APIs para comunicação entre o frontend e o backend, e utilizaremos JWT para autenticação e autorização.

1. Camada 1: Frontend (Angular)

1.1. Em Desenvolvimento

2. Camada 2: Backend (Spring Boot)

2.1. Spring Boot com Spring MVC

- **Controllers:** Controladores para cada recurso (Ex: UserController, ProductController) que receberão as requisições HTTP e chamarão os serviços.
- **Services:** Serviços para a lógica de negócio (Ex: UserService, ProductService), que interagem com o repositório e retornam resultados para os controladores.
- **Repositories:** Camada de persistência usando Spring Data JPA ou JDBC para interagir com o banco de dados.
- **DTOs (Data Transfer Objects):** Objetos que são utilizados para transferir dados entre as camadas (Controller -> Service -> Repository).

3. Spring Security com JWT

3.1. Autenticação e Autorização: Usar Spring Security para proteger as APIs, e JWT (JSON Web Tokens) para autenticação sem estado. O token será enviado pelo frontend em cada requisição.

3.2. Filtragem de Requisições: Implementar um filtro de segurança para validar o JWT em todas as requisições para endpoints protegidos.

3.3. CORS: Configurar CORS (Cross-Origin Resource Sharing) para permitir que o Angular (frontend) acesse as APIs Spring (backend).

4. Spring Data JPA

4.1. Entidades: As entidades Java que representam tabelas no banco de dados (Ex: User, Product).

4.2. Relacionamentos: Definir os relacionamentos entre as entidades usando anotações JPA, como @OneToMany, @ManyToOne, etc.

4.3. Repositórios: Usar Spring Data JPA para criar repositórios e abstrair a interação com o banco de dados.

5. Banco de Dados SQL Server

5.1. Estrutura de Tabelas: Utilizar um banco de dados relacional, como SQL Server, com tabelas para armazenar os dados essenciais, como usuários, produtos, pedidos, etc.

5.2. Schema: O banco de dados terá um schema normalizado, com tabelas separadas para diferentes recursos e chaves estrangeiras para manter a integridade referencial.

5.3. SQL Server como Backend: A comunicação com o SQL Server será feita através de JPA/Hibernate.

Fluxo de Processos

Cadastro de Quartos: O administrador registra novos quartos com informações detalhadas.

Reserva de quarto: o administrador/recepcionista realiza as reservas conforme disponibilidade de quartos

Check-in/Check-out: Processo de registro de check-in e check-out que impacta na disponibilidade do quarto.

Gestão de Estoque: Alertas automáticos para reabastecimento de itens de consumo.

Relatórios e Análises: O administrador pode gerar relatórios financeiros e de ocupação.

Documentação Técnica

Configuração do Ambiente Java 17 e Spring Boot: Instalar o JDK 17 e configurar o Spring framework

Banco de Dados SQL Server: Instalar o SQL Server e configurar a conexão no Spring Boot (application.properties).

Testes

Plano de Testes

Testes Unitários: Verificação das funcionalidades isoladas (ex.: cálculos de preço).

Testes de Integração: Validação da comunicação com o banco de dados.

Testes de Carga: Garantir que o sistema lida bem com múltiplas reservas simultâneas.

Casos de Teste CT01: Verificar o processo de reserva de um quarto.

CT02: Testar o processo de check-in e check-out.

CT03: Validar a geração de relatórios financeiros.

Manual do Usuário

Desenvolver junto do FAQ

Funcionalidades Futuras

Em aberto

Limitações

1. A versão inicial será apenas web, sem suporte a aplicativos móveis
2. Ainda será necessário adquirir conhecimento para desenvolver o frontend e realizar os testes
3. Apenas os administradores e os recepcionistas conseguirão realizar reservas nas primeiras versões

Conclusão

6. Conclusão

Este projeto tem o objetivo de transformar a gestão hoteleira em uma tarefa mais prática e eficiente, com a centralização de informações e a automação de processos essenciais para o sucesso do negócio