

Plano de Testes

Projeto Hotel Serrano

Autores do **Projeto:**
Alessandra Franco
Jussan Silva
Kemely Teixeira
Mariana Gonçalves
Maria Eduarda Silva
Pamela Cristina
Rian Bastos
Rodrigo Matos
Thiago Falcão
Vitor Gustavo da Silva
Vitor Gustavo de Oliveira

Gerente do Projeto: Lucas José Caetano

Histórico de Revisões

Data	Versão	Descrição	Autor
16/06/2023	1.0	Criação do plano de teste	Mariana Gonçalves
21/06/2023	1.1	Revisão do Plano de teste	Jussan

1.

2. Introdução

O objetivo deste plano de teste é documentar a abordagem e os detalhes dos testes para o sistema desenvolvido pela empresa Hotel Serrano. O sistema é um sistema de hotelaria que possui requisitos funcionais e não funcionais bem definidos. O plano de teste visa garantir a qualidade e o bom funcionamento do sistema antes de sua implantação.

2.1 Objetivos

- Verificar se todas as funcionalidades do sistema estão implementadas corretamente e funcionando conforme especificado nos requisitos.
- Identificar e corrigir eventuais erros, falhas e inconsistências no sistema.
- Validar a interface do usuário e garantir que seja intuitiva e responsiva.
- Avaliar a segurança do sistema e verificar se as medidas de proteção de dados estão implementadas corretamente.
- Avaliar a confiabilidade do sistema e garantir que as funcionalidades estejam disponíveis e funcionando corretamente.
- Validar a integração do sistema com o banco de dados MariaDB hospedado na AWS.
- Garantir que o sistema siga as boas práticas de desenvolvimento de software e o padrão MVC.
- Garantir que o sistema esteja em conformidade com a LGPD (Lei Geral de Proteção de Dados).
- Garantir que a identidade visual do hotel seja seguida, incluindo cores, fontes e logotipo.

2.2 Escopo

O escopo dos testes abrange todas as funcionalidades descritas nos requisitos funcionais e não funcionais. Os testes serão realizados em todas as telas e componentes do sistema, incluindo o registro de solicitações de reservas, consulta de solicitações, solicitação de alteração de reserva e cancelamento de reserva.

3. Abordagens de Testes

A abordagem de teste será baseada nos seguintes princípios:

- Teste de unidade: Serão realizados testes de unidade para verificar o funcionamento individual de cada funcionalidade do sistema.
- Teste de integração: Serão realizados testes de integração para garantir a interação correta entre os diferentes módulos e componentes do sistema.
- Teste de sistema: Serão realizados testes de sistema para verificar o sistema como um todo, incluindo todas as funcionalidades integradas.
- Teste de usabilidade: Serão realizados testes de usabilidade para validar a interface do usuário e garantir que seja intuitiva e responsiva.
- Teste de segurança: Serão realizados testes de segurança para avaliar a proteção dos dados dos hóspedes e verificar se as medidas de segurança estão implementadas corretamente.
- Teste de conformidade: Serão realizados testes de conformidade para verificar se o sistema está em conformidade com a LGPD e as boas práticas de desenvolvimento de software.

3.1 Requisitos

Aqui são definidos os requisitos e as abordagens que serão utilizadas para os requisitos.

Requisitos Funcionais	Requisitos Não Funcionais
RF001 – Reservas	NF001- Interface do Usuário
RF002 – Checkin e CheckOut	NF002 – Segurança
RF003 – Despesas	NF003 – Confiabilidade
RF004 – Administração	NF004 – Banco de Dados
RF005 – Segurança	NF005 – Boas práticas para o desenvolvimento
RF006 - Relatórios	NF006 – Cronograma
	NF007 - Identidade Visual do hotel

Requisito	RF001 - Reservas
Descrição:	<ul style="list-style-type: none"> • Os clientes devem poder solicitar reservas para quartos disponíveis via telefone. • O atendente deve poder procurar quartos pelo ID.

	<ul style="list-style-type: none"> • O atendente deve poder visualizar as reservas atuais e futuras dos clientes. • d. O atendente deve poder modificar ou cancelar as reservas de cada cliente, desde que solicitado pelo mesmo e dentro do prazo de 24 horas antes do check-in.
--	---

Requisito	RF002 - Checkin e CheckOut
Descrição:	<ul style="list-style-type: none"> • O atendente deve poder registrar o check-in e check-out dos hóspedes. • Ao realizar o check-out, o sistema deve agendar automaticamente a limpeza do quarto. • Os clientes devem poder solicitar seu histórico de check-ins e check-outs anteriores.

Requisito	RF003 - Despesas
Descrição:	<ul style="list-style-type: none"> • Os clientes devem poder adicionar produtos consumidos à sua conta durante a estadia. • O sistema deve registrar as despesas associadas a cada reserva. • Os clientes devem poder solicitar um resumo de suas despesas durante a estadia.

Requisito	RF004 - Administração
Descrição:	<p>O administrador deve ter acesso a um painel de controle para monitorar várias informações, incluindo:</p> <ul style="list-style-type: none"> • Despesas gerais do hotel. • Lista completa de reservas, incluindo detalhes dos hóspedes e datas. • Catálogo de produtos disponíveis para consumo. • Dados dos hóspedes, incluindo informações de contato e histórico de estadias. • Status das limpezas agendadas e histórico de limpezas concluídas. • Informações dos funcionários, incluindo detalhes de contato, salário e funções.

Requisito	RF005 - Segurança
-----------	-------------------

Descrição:	<ul style="list-style-type: none"> • O sistema deve exigir autenticação para acessar funcionalidades administrativas. • Os dados dos hóspedes, reservas, despesas e produtos devem ser armazenados de forma segura.
-------------------	---

Requisito	RF006 - Relatórios
Descrição:	<ul style="list-style-type: none"> • O sistema permite gerar relatórios sobre as despesas, reservas, produtos, hóspedes, limpeza, funcionários e disponibilidade de quartos. • Os relatórios devem poder ser filtrados por datas, quartos, hóspedes, etc..

Requisito	NF001 - Interface do usuário
Descrição:	<ul style="list-style-type: none"> • A interface do usuário deve ser intuitiva e responsiva. • A aplicação deve ser acessível em dispositivos desktop.

Requisito	NF002 - Segurança
Descrição:	<ul style="list-style-type: none"> • Medidas de segurança devem ser implementadas para proteger as informações confidenciais dos hóspedes. • O sistema deve ser desenvolvido em conformidade com a LGPD (Lei Geral de Proteção de Dados).

Requisito	NF003 - Confiabilidade
Descrição:	<ul style="list-style-type: none"> • O sistema deve ser confiável e garantir a disponibilidade das funcionalidades.

Requisito	NF004 - Banco de Dados
Descrição:	<ul style="list-style-type: none"> • Deve ser utilizado um banco de dados MariaDB hospedado na AWS para armazenar as informações.

Requisito	NF005 - Boas práticas de desenvolvimento
------------------	---

Descrição:	<ul style="list-style-type: none"> • O sistema deve ser desenvolvido seguindo boas práticas de desenvolvimento de software e Padrão MVC.
-------------------	---

Requisito	NF006 - Cronograma
Descrição:	<ul style="list-style-type: none"> • O projeto será dividido em fases, com prazos específicos para cada fase. • Reuniões regulares com a equipe de desenvolvimento e stakeholders devem ser realizadas para acompanhar o progresso e resolver problemas ou dúvidas. • O cronograma deve ser flexível para acomodar mudanças de requisitos ou imprevistos durante o desenvolvimento.

Requisito	NF007 - Identidade Visual do hotel
Descrição:	<ul style="list-style-type: none"> • O sistema deve seguir a identidade visual do hotel, incluindo cores, fontes e logotipo. <ul style="list-style-type: none"> • Style guide Sistema de Hotelaria • Todo desenvolvimento de UI se inicia com um Guia de Estilos a ser seguido. Ele servirá de direção para a utilização de componentes e cores, tendo como objetivo seguir um padrão para todo o sistema. Com isso, utilizaremos o Style guide abaixo, que nos guiará nos padrões de Interface para nosso sistema de Hotelaria, garantindo que tudo seja desenvolvido dentro de um padrão pré-estabelecido. • Fontes: O tipo de letra principal é o Roboto. <ul style="list-style-type: none"> • Roboto tem uma natureza dupla. Tem um esqueleto mecânico e as formas são em grande parte geométricas. Ao mesmo tempo, a fonte apresenta curvas amigáveis e abertas. Enquanto alguns grotescos distorcem suas formas de letra para forçar um ritmo rígido, Roboto não se compromete, permitindo que as letras sejam ajustadas em sua largura

natural. Isso cria um ritmo de leitura mais natural, mais comumente encontrado nos tipos humanista e serif.

- Cores: O esquema de cor visa apontar as cores que podem ser utilizadas no sistema, formando um padrão visual comum.
 - Principais >> As cores primárias são usadas para os elementos. Cada cor tem um tom mais escuro e mais claro.
 - Light blue: #05DDFB / Blue: #009FB7 / Dark Blue: #005E6B
 - Light purple: #BDAED0 / purple: #947EB0 / Dark purple: #503176
 - Light orange: #FF6C6C / orange: #FE4A49 / Dark orange: #BA0B0B
 - Light green: #748E83 / green: #3E5E50 / Dark green: #234436
 - Light yellow: #FFE18B / yellow: #FED766 / Dark yellow: #BF961D
 - Light brown: #9F6316 / brown: #7D4600 / Dark brown: #301B00
- Tons de Cinza >> As cores cinza são usadas para texto, planos de fundo, linhas e bordas.
 - Smoke e Snow têm dois tons mais escuros, usados para ícones e planos de fundo.
 - Smoke: #E0E6ED / Dark Smoke: #D3DCE6 / Extra Dark Smoke: #C0CCDA
 - snow: #F9FAFC / Dark snow: #EFF2F7 / Extra dark snow: #E5E9F2
- Layout:
 - Todas as telas deverão ter um espaçamento uniforme e poderá ser constituídas das seguintes disposições:
 - pasta: Imagens imagem1.jpg

3.2 Detalhamento da abordagem de teste

- Teste de unidade: Serão criados casos de teste para cada funcionalidade individual do sistema, verificando se o resultado esperado é alcançado..
- Teste de integração: Serão identificadas as interações entre os módulos e componentes do sistema. Serão criados casos de teste para verificar se as integrações estão ocorrendo corretamente.
- Teste de sistema: Será realizado um teste abrangente em todas as funcionalidades do sistema. Serão criados casos de teste para simular diferentes cenários de uso e verificar se o sistema se comporta corretamente em cada um deles.
- Teste de usabilidade: Serão criados casos de teste para avaliar a usabilidade do sistema, incluindo a navegação entre as telas, a facilidade de uso e a responsividade da interface.
- Teste de segurança: Serão realizados testes de penetração para identificar possíveis vulnerabilidades no sistema. Será verificado se as medidas de segurança, como autenticação e proteção de dados, estão implementadas corretamente.
- Teste de conformidade: Será verificado se o sistema está em conformidade com a LGPD e as boas práticas de desenvolvimento de software.

3.2.1 Cenário de Testes

Serão criados diferentes cenários de teste que abrangem os principais fluxos de uso do sistema, como:

- Registro de uma nova reserva e verificação de sua inclusão no sistema.
- Modificação ou cancelamento de uma reserva existente.
- Realização do check-in de um hóspede e verificação do agendamento de limpeza do quarto.
- Adição de produtos consumidos à conta do hóspede e verificação do registro correto das despesas.
- Acesso do administrador ao painel de controle e verificação das informações disponíveis.

- Geração de relatórios sobre diferentes aspectos do sistema, como despesas, reservas, produtos, hóspedes, limpeza, funcionários e disponibilidade de quartos.

3.2.2 Caso de Testes

Registro de uma nova reserva:

- O atendente seleciona a opção de registrar uma nova reserva.
- O atendente insere as informações necessárias do cliente e da reserva.
- O sistema exibe uma confirmação de que a reserva foi registrada com sucesso.
- O atendente verifica se a reserva é exibida corretamente no sistema.

Modificação de uma reserva existente:

- O atendente busca a reserva pelo ID do cliente.
- O atendente seleciona a opção de modificar a reserva.
- O atendente faz as alterações necessárias na reserva.
- O sistema exibe uma confirmação de que a reserva foi modificada com sucesso.
- O atendente verifica se as alterações são refletidas corretamente no sistema.

Realização do check-in de um hóspede:

- O atendente busca a reserva pelo ID do cliente.
- O atendente registra o check-in do hóspede.
- O sistema atualiza o status da reserva para "check-in realizado" e agenda a limpeza do quarto.
- O atendente verifica se o status da reserva e o agendamento da limpeza estão corretos.

Adição de produtos consumidos à conta do hóspede:

- O hóspede solicita a adição de um produto consumido à sua conta.
- O atendente registra o produto e associa-o à reserva do hóspede.
- O sistema registra a despesa na conta do hóspede.
- O hóspede verifica se a despesa está corretamente associada à sua conta.

Acesso do administrador ao painel de controle:

- O administrador faz login no sistema.
- O sistema verifica as credenciais do administrador e permite o acesso ao painel de controle.
- O administrador verifica se as informações gerais, reservas, produtos, hóspedes, limpeza e funcionários são exibidos corretamente.

4. Recursos

Essa seção apresenta os recursos recomendados para o projeto do SGM, suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

- recursos que serão utilizados (humanos, hardware, software, etc)
- quais softwares que devemos utilizar (automatização de testes, VScode)

4.1 Trabalhadores

Função	Nome	Responsabilidades Específicas ou Comentários
<i>PO</i>	<i>Lucas José Dias Caetano</i>	Especialista em levantamento de requisitos e venda de projetos
<i>Scrum Master</i>	<i>Jussan</i>	<u>Gestão estratégica de equipes e projetos com ênfase em organização e planejamento</u>
<i>Desenvolvedor</i>	<i>Maria Eduarda da Silva</i>	Tech Leader – Desenvolvimento Back-End
<i>Desenvolvedor</i>	<i>Rodrigo Moreira de Matos</i>	Desenvolvimento Back-End
<i>Desenvolvedor</i>	<i>Kemily Teixeira</i>	Desenvolvimento Back-End
<i>Desenvolvedor</i>	<i>Vitor Gustavo da Silva</i>	Desenvolvimento Back-End
<i>Desenvolvedor</i>	<i>Vitor Gustavo de Oliveira</i>	Tech Leader - Desenvolvimento Front-End
<i>Desenvolvedor</i>	<i>Alessandra</i>	Desenvolvimento Front-End
<i>Desenvolvedor</i>	<i>Pamela Cristina</i>	Desenvolvimento Front-End
<i>Desenvolvedor</i>	<i>Tiago Falcão</i>	Desenvolvimento Front-End
<i>Testes</i>	<i>Mariana Gonçalves</i>	<i>Tech Leader – QA</i>
<i>Testes</i>	<i>Rian</i>	QA

4.2 Sistema

A tabela seguinte expõe os recursos do sistema para o projeto de teste.

Recursos do Sistema
<i>Servidor de Banco de Dados MariaDb RDS AWS</i>
<i>Terminais Clientes ? PCs (conectados via LAN) quantidade de terminais a ser definida pelo cliente ? PC com tela sensível ao toque (conectado a uma LAN e à internet) quantidade de terminais a ser definida pelo cliente</i>
<i>Repositório de Testes – Git Hub 2 PCs de Desenvolvimento de Teste</i>

5. Cronograma

A tabela abaixo descreve o cronograma a ser seguido no projeto.

Milestone	Data de Início	Data de Término
<i>Planejar Teste</i>	<i>07/06/2023</i>	<i>18/06/2023</i>
<i>Executar Teste</i>	<i>21/06/2023</i>	<i>24/06/2023</i>
<i>Avaliar Teste</i>	<i>25/06/2023</i>	<i>27/06/2023</i>

6. Matriz de Risco

Requisito	Risco	Impacto	Probabilidade	Severidade	Ações de Mitigação
Reservas	Dificuldade no registro	Reservas incorretas ou perdas	Moderada	Alta	Testes intensivos de inserção de dados, revisões regulares
Check-in e Check-out	Falha no registro	Dados incorretos, agendamento inadequado	Baixa	Média	Testes rigorosos de check-in e check-out, monitoramento regular
Despesas	Falha no registro	Contas de despesas incorretas	Moderada	Alta	Testes abrangentes, reconciliações periódicas
Administração	Dificuldade de acesso, exibição de informações	Falta de acesso, dificuldade na tomada de decisões	Baixa	Média	Testes de autenticação, verificação das informações exibidas
Segurança	Violação de segurança	Roubo de informações, violação de privacidade	Baixa	Alta	Implementação de medidas de segurança, testes de penetração
Relatórios	Geração incorreta ou incompleta	Informações imprecisas ou ausentes	Moderada	Média	Testes abrangentes, revisões regulares
Interface do usuário	Interface confusa ou não responsiva	Dificuldade de uso, baixa satisfação do usuário	Moderada	Média	Testes de usabilidade, coleta de feedback

Segurança	Não conformidade com a LGPD	Multas legais, danos à reputação	Baixa	Alta	Implementação de medidas de segurança, auditorias regulares
Confiabilidade	Falha no funcionamento	Indisponibilidade das funcionalidades	Baixa	Média	Testes de estresse, monitoramento contínuo
Banco de Dados	Problemas de armazenamento	Perda de dados, indisponibilidade	Baixa	Alta	Backup regular, testes de recuperação
Boas práticas de desenvolvimento	Não adesão às melhores práticas	Baixa qualidade, dificuldade de manutenção	Moderada	Média	Adesão às melhores práticas, revisões de código
Cronograma	Atrasos no desenvolvimento	Dificuldade de entrega no prazo	Moderada	Alta	Reuniões regulares, cronograma flexível
Identidade Visual	Não seguir a identidade visual	Inconsistência visual	Baixa	Média	Adesão à identidade visual definida, revisão visual