
Documentação de Sistema da Samarino Hotels

Versão Alpha

Autores:

Anne Carolina Borges Gontijo Gomes, 14/0016546;

Artur Renan Alves Barbosa, 16/0002656;

Lucas da Silva Souza, 16/0013020.

Brasília, Distrito Federal, Brasil.

21 de Novembro de 2018.

Índice

1.	DESCRIÇÃO DO PROJETO DEFINIDO PARA O PROJETO.....	5
2.	PLANO DE PROJETO.....	5
2.1.	Divisão de Trabalho.....	5
2.2.	Mensurações.....	5
2.2.1.	Project backlog (Relatório de Progresso do Projeto).....	5
2.2.2.	Iteration backlog (Relatório de Progresso da Iteração).....	5
2.3.	Objetivos do Projeto.....	6
3.	LISTA DE ITENS DE TRABALHO.....	6
3.1.	Introdução.....	6
3.2.	Gerenciamento de Itens de Trabalho.....	6
3.3.	Estimativa Ágil.....	6
4.	LISTA DE RISCOS.....	6
5.	PLANO DAS ITERAÇÕES REALIZADAS.....	7
5.1.	Introdução.....	7
5.2.	Marcos importantes.....	7
5.3.	Objetivos de Alto Nível.....	7
5.4.	Atribuição de Itens de Trabalho.....	7
5.5.	Problemas.....	8
5.6.	Critérios de Avaliação.....	8
6.	DOCUMENTO DE VISÃO.....	8
6.1.	Introdução.....	8
6.2.	Posicionamento.....	8
6.2.1.	Declaração do problema.....	9
6.2.2.	Declaração de posição do produto.....	9
6.3.	Descrição do Stakeholder.....	9
6.3.1.	Sumário do Stakeholder.....	10
6.3.2.	Ambiente do usuário.....	10
6.4.	Visão global do produto.....	11
6.4.1.	Necessidades e recursos.....	11
7.	GLOSSÁRIO.....	11
8.	MODELOS DE CASOS DE USO.....	11
8.1.	Introdução.....	12
8.2.	Visão geral.....	12
8.3.	Diagrama de Casos de Uso.....	12
8.4.	Diagrama de Atores.....	12
8.4.1.	Visitante.....	13

8.4.2.	Usuário.....	13
8.4.3.	Administrador.....	13
9.	DESCRIÇÕES TEXTUAIS DE CASOS DE USO.....	14
9.1.	Breve descrição.....	14
9.2.	Breve descrição dos atores.....	14
9.3.	Casos de Uso.....	15
9.3.1.	Obter dados de acomodação.....	15
9.3.1.1.	Visão Geral.....	15
9.3.1.2.	Ator Principal.....	15
9.3.1.3.	Interessados e Interesses.....	15
9.3.1.4.	Pré-condição.....	15
9.3.1.5.	Pós-condição.....	15
9.3.1.6.	Cenários de Sucesso.....	15
9.3.1.7.	Fluxo Alternativo.....	15
9.3.2.	Cadastrar no sistema.....	15
9.3.2.1.	Visão Geral.....	15
9.3.2.2.	Ator Principal.....	16
9.3.2.3.	Interessados e Interesses.....	16
9.3.2.4.	Pré-condição.....	16
9.3.2.5.	Pós-condição.....	16
9.3.2.6.	Cenários de Sucesso.....	16
9.3.2.7.	Fluxo Alternativo.....	16
9.3.3.	Efetuar reservas.....	16
9.3.3.1.	Visão Geral.....	16
9.3.3.2.	Ator Principal.....	16
9.3.3.3.	Interessados e Interesses.....	16
9.3.3.4.	Pré-condição.....	16
9.3.3.5.	Pós-condição.....	16
9.3.3.6.	Cenários de Sucesso.....	16
9.3.3.7.	Fluxo Alternativo.....	16
9.3.4.	Gerenciamento de pagamentos e liberação de reservas.....	17
9.3.4.1.	Visão Geral.....	17
9.3.4.2.	Ator Principal.....	17
9.3.4.3.	Interessados e Interesses.....	17
9.3.4.4.	Pré-condição.....	17
9.3.4.5.	Pós-condição.....	17
9.3.4.6.	Cenários de Sucesso.....	17
9.3.4.7.	Fluxo Alternativo.....	17
9.3.5.	Dados sobre todos os quartos próprios e de hotéis associados.....	17
9.3.5.1.	Visão Geral.....	17
9.3.5.2.	Ator Principal.....	17
9.3.5.3.	Interessados e Interesses.....	17
9.3.5.4.	Pré-condição.....	17
9.3.5.5.	Pós-condição.....	17
9.3.5.6.	Cenários de Sucesso.....	17

9.3.5.7.	Fluxo Alternativo.....	18
9.4.	Subfluxos.....	18
9.4.1.	Caso de Uso “Cadastrar”.....	18
10.	LISTA DE FERRAMENTAS.....	18
11.	DESCRIÇÃO DE ARQUITETURA DE SOFTWARE.....	18
11.1.	Propósito.....	18
11.2.	Objetivo da arquitetura e filosofia.....	18
11.3.	Suposições e dependências.....	19
11.4.	Requisitos significativos na arquitetura.....	19
11.5.	Decisões, contrastes e justificativas.....	19
11.6.	Mecanismos de arquitetura.....	19
12.	DESCRIÇÃO DE INFRAESTRUTURA DE IMPLANTAÇÃO.....	20
12.1.	Arquivos de código-fonte.....	20
12.2.	Arquivo de dados.....	20
13.	IMPLEMENTAÇÃO DE CASOS DE USO RELATIVOS A UM SUBSISTEMA.....	21
14.	CASOS DE TESTE DA IMPLEMENTAÇÃO.....	22
14.1.	Entradas.....	22
14.2.	Script.....	23
14.3.	Resultados Esperados.....	24
14.4.	Critérios de Avaliação.....	25

Documentação de Sistema da Samarino Hotels

Versão Alpha

1. Descrição do projeto definido para o projeto

O projeto de implementação da rede hoteleira Samarino se baseia na criação de um sistema de hotelaria, onde o usuário poderá fazer ou desfazer reservas em hotéis em quartos disponíveis pelos hotéis associados. Para todos estes serviços, o usuário deverá realizar o seu cadastro no sistema e realizar sua autenticação.

2. Plano de Projeto

2.1. Divisão de Trabalho

A organização de tarefas dentro deste documento está divididas pelas áreas de conhecimento (knowledge areas). Estas áreas de conhecimentos possuem objetivos e responsabilidades de exercerem o seu papel de encontrar bugs, resolvê-los e contribuir para projetos futuros dentro do código-fonte.

A todos estes, englobam a elicitação e análise de requisitos, implementação, estruturação de arquitetura, teste de software e validação.

Segue abaixo a divisão de trabalho para este projeto:

- Gerenciamento de Projeto e de Requisitos: Lucas Souza;
- Gerenciamento de Mudanças: Lucas Souza;
- Implementação: Artur Renan e Anne Caroline;
- Arquitetura de Software: Anne Caroline;
- Teste de Software: Artur Renan.

2.2. Mensurações

Estamos seguindo a estrutura do OpenUP, baseado no fato de desenvolver toda a topologia por trás do desenvolvimento, e não somente a codificação propriamente dita. Estes artefatos incluem este Plano de Projeto (feito pelo Google Docs), Lista de Itens de Trabalho (Google Sheets) e Avaliação de Status (help desk, cuja plataforma ainda não decidimos).

Seguiremos duas medidas primárias usando um ponto de sistema, sendo um ponto representando duas horas de trabalho:

- 2.2.1. Project backlog (Relatório de Progresso do Projeto): Principal ferramenta para verificar o status do projeto, dando informações exatas sobre o progresso de partes do projeto e de quantas partes ainda precisam ser concluídas. Este relatório deverá ser atualizado após a conclusão de cada iteração (descrita no item 2.2.2.).
- 2.2.2. Iteration backlog (Relatório de Progresso da Iteração): Ferramenta utilizada para verificar o status de uma iteração atual, obtido através de uma média de tempo para a conclusão do trabalho, analisando os prazos de entrega feitos pelas equipes no desenvolvimento da iteração. Este relatório deve ser atualizado com frequência. Há alguns critérios para que esta avaliação da equipe ocorra:

- Término de partes do projeto;
- Atualização da avaliação do esforço necessário para concluir o trabalho, parte esta realizada pelo desenvolvedor responsável pelo Item de Trabalho.

Este relatório permite reação às mudanças por parte da equipe de desenvolvimento, que podem trabalhar novos métodos para otimizar o tempo e, portanto, concluir partes do projeto com antecedência.

2.3. Objetivos do Projeto

Seguindo o método Eclipse Process Framework (EPF), está anexado junto a este documento o arquivo “Objetivo do Projeto”, com informações detalhadas.

3. Lista de itens de trabalho

3.1. Introdução

Neste item, trataremos sobre o agendamento dos itens necessários para a conclusão do projeto, escalando itens prioritários, estabelecendo prazos de entrega, analisar o impacto que este item pode proporcionar ao software e acompanhar o progresso de cada um dos itens. Neste item, são escalados as prioridades sobre:

- Casos de Uso (descritas no item 8.);
- Divisão de Trabalho (descrita no item 2.1.);
- Solicitações de mudanças ou report de defeitos (descrita no item 2.1.);
- Tarefas de Desenvolvimento.

3.2. Gerenciamento de Itens de Trabalho

Processo de observação de todo o trabalho que será executado no projeto e como o mesmo pode afetar o produto. Alguns dos Itens de Trabalho podem ser implementados neste projeto, podem ser priorizados, implementados em uma versão futura ou nunca ser implementado. Todos estes casos podem ser gerenciados por planilhas ou por ferramentas de Help Desk.

Os status do Gerenciamento de Itens de Trabalho são:

- Novo: Item de Trabalho criado, entretanto, ainda não atribuído para nenhum colaborador da equipe;
- Atribuído: Um membro da equipe foi escalado como o responsável pelo Item de Trabalho;
- Resolvido: O desenvolvedor realizou a implementação e testou o software, tendo esta tarefa parcialmente concluída;
- Verificado: O Item de Trabalho foi testado por uma equipe independente da mesma empresa, em busca de possíveis inconsistências;
- Terminado: Não está mais ativo.

3.3. Estimativa Ágil

O processo de Estimativa Ágil estipula um prazo para a resolução total do projeto total ou parcial através da Estimativa de Tamanho (extensão dos requisitos, restrições, dentre outros), Velocidade (tempo que a equipe de desenvolvedores podem entregar) e a Estimativa de Esforço (cálculo de tempo necessário para que a equipe desenvolva todas as tarefas).

4. Lista de riscos

Processo de catalogação de todas as iterações, requisitos, serviços ou restrições que podem entrar em contradição ou conflito dentro do software, afetando seu desempenho. A equipe de desenvolvimento deve informar os stakeholders acerca de possíveis bugs no software, categorizá-los, especificá-los e informar a complexidade deste risco. Pode ser feito por planilhas ou por templates fornecidos pela plataforma OpenUP.

Junto a este documento, siga o arquivo “Lista de Riscos do Projeto”, uma tabela referente aos possíveis riscos deste projeto, bem como a descrição de cada um dos temas.

5. Planos das iterações realizadas

5.1. Introdução

Iterações são definições de tempo dentro do projeto, composto por uma versão estável e executável do código-fonte, documentação, instaladores e ferramentas auxiliares, esta última podendo ser utilizada caso necessário. Com a execução do programa, podemos colher feedback dos stakeholders acerca de suas necessidades, críticas e elogios. É estipulado datas de início e término em cada iteração; cujo prazo são de, no máximo, quatro semanas. O Plano das Iterações usa como referência a Lista de Itens de Trabalho, descrita no item 3 deste documento.

5.2. Marcos importantes

Abaixo, estão as datas principais dentro deste processo de iterações:

Marco histórico:	Data:
Início do processo de Iterações	01/11/2018
Testes com Stakeholders	18/01/2019
Término do processo de Iterações	02/02/2019

5.3. Objetivos de Alto Nível

Baseado nos conceitos da Linguagem de Alto Nível, este subtema tem como objetivo:

- Utilização a linguagem natural;
- Liberdade para que os utilizadores mais assíduos no projeto possam ter acesso à novas versões sem a exigência de aval por parte da equipe de suporte;
- Definições sobre funcionalidades e informações (para todos os atores), bem como restrições ao sistema, sendo este, sob poder de acesso somente aos stakeholders.

5.4. Atribuição de Itens de Trabalho

Conforme descrito no item 2.1., referente à Divisão de Trabalho, segue abaixo as atribuições para cada desenvolvedor da nossa equipe:

Nome ou Descrição	Grau de Prioridade	Tamanho estimado (pontos)	Associado para	Esforço estimado (em horas)
Controle de acomodações disponíveis	5	8	Artur Renan	03
Realização de design	5	5	Anne Gontijo	30
Implementação e teste de parte de servidor	5	6	Artur Renan, Anne Gontijo	18
Implementação e teste da parte de cliente	5	10	Artur Renan, Anne Gontijo, Lucas Souza	30
Atualização da documentação final ao cliente	5	7	Lucas Souza	12
Edição final da documentação ao cliente	2	4	Lucas Souza	36
Conferência com os stakeholders	5	5	Artur Renan, Anne Gontijo, Lucas Souza	22

5.5. Problemas

- Podemos confiar completamente nas informações sobre acomodações disponíveis, dadas pelas redes associadas?
- Como será feita as formas de pagamento?
- Iremos integrar informações com companhias de viagem?

5.6. Critérios de Avaliação

- Garantia de 95% do sistema funcionando;
- Documentação final está favorável para ambas as partes;
- Versão executável com a presença dos stakeholders.

6. Documento de visão

6.1. Introdução

Este item tem como objetivo formalizar um contrato junto aos stakeholders, detalhando serviços, informações, manipulação de dados, bem como restrições.

Também é fundamental para a aprovação do projeto, para validações e para manutenções póstumas.

6.2. Posicionamento

6.2.1. Declaração do problema

O problema do	software da Samarino Hotels
afeta	o comportamento dos stakeholders
sobre o impacto da	integração de quartos disponíveis com a rede associada da Samarino Hotels
um solução bem sucedida seria	integração com os sistemas internos de cada rede associada, aliada com a validação feita pelos gerentes desta rede.

6.2.2. Declaração de posição do produto

Para	a gerência administrativa da Samarino
Que	necessita expandir seu negócio com um endereço na internet para acomodação
A Samarino Hotels	é uma agência de hotelaria
Isso	pode criar um leque de oportunidade para que turistas possam visualizar quartos de hotéis disponíveis, bem como seu preço, suas avaliações e seu local, tudo em seu computador ou em seu telefone; desde que tenha usuário criado e que esteja autenticado
Ao contrário	do método tradicional, em que o turista chega em uma cidade e procura alguma acomodação disponível, o que se torna impossível em datas comemorativas, por exemplo Ano Novo e Carnaval
Nosso produto	busca facilitar a vida de quem está viajando a férias ou a trabalho, proporcionando maior comodidade e segurança ao

	cliente.
--	----------

6.3. Descrição do Stakeholder

6.3.1. Sumário do Stakeholder

Nome:	Descrição:	Responsabilidades:
Primários	Gerência administrativa da Samarino Hotels	<ul style="list-style-type: none"> ➤ Conferir diariamente quantos quartos estão disponíveis; ➤ Atentar-se a feedbacks dos clientes; ➤ Proporcionar experiências únicas para o cliente em suas viagens, causando posteriormente a fidelização; ➤ Preparar a rede própria e associados para datas comemorativas com meses de antecedência.
Secundários	Gerentes das redes associadas da Samarino Hotels	<ul style="list-style-type: none"> ➤ Repassar dados totais sobre acomodações (disponíveis, ocupadas e ociosos) para a Samarino Hotels; ➤ Disponibilizar relatórios diários para a equipe da Samarino Hotels, seja em documentos ou em dados integrados; ➤ Fazer o controle sobre todos os quartos (entre ocupados, disponíveis ou ociosos) para contabilização; ➤ Se preparar com antecedência para datas comemorativas.

6.3.2. Ambiente do usuário

Abaixo, há o detalhamento do ambiente:

- O número de pessoas por quarto irá depender da capacidade total do mesmo, informação esta que a Samarino Hotels e redes associadas deverá informar;
- As informações sobre o quarto (preço, local, acomodação máxima, dentre outros) será de acesso público, entretanto,

caso o usuário queira reservar o quarto, o mesmo deverá ter um usuário no sistema e estar autenticado;

- Não haverá log off automático;
- O usuário poderá ver as avaliações do quarto;
- Para todos estes serviços, acima de tudo, o usuário deverá estar conectado com a internet.

6.4. Visão global do produto

6.4.1. Necessidades e recursos

Necessidade	Grau de prioridade	Características	Lançamento Planejado
Servidores para execução deste sistema.	4	Aluguel ou compra de equipamentos para gerenciar o sistema.	03/01/2019.
Autenticação do usuário.	5	Entrada do usuário no sistema através de um usuário e senha.	24/01/2019
Listagem de quartos disponíveis.	5	Verificar junto com os hotéis associados sobre as informações dos quartos.	24/01/2019
Divulgação da Samarino Hotels na mídia e redes sociais	4	Anúncios e preços promocionais para usuários que utilizam redes sociais, televisão ou outras plataformas de anúncio. Será feito externamente.	15/07/2019

7. Glossário

Acomodações: quartos disponíveis nos hotéis.

Autenticação: login do usuário ou administrador no sistema da Samarino Hotels.

Cientes: pessoas que buscam se hospedar junto a Samarino Hotels.

Hotéis: refere-se a rede hoteleira da Samarino Hotels ou de redes associadas.

Reservas: ato de reservar um quarto de hotel.

Stakeholders: partes interessadas no projeto.

8. Modelos de Casos de Uso

8.1. Introdução

Modelos de Casos de Uso é um método dentro do projeto de software, cujo objetivo é facilitar o entendimento de stakeholders, desenvolvedores e equipe de analistas.

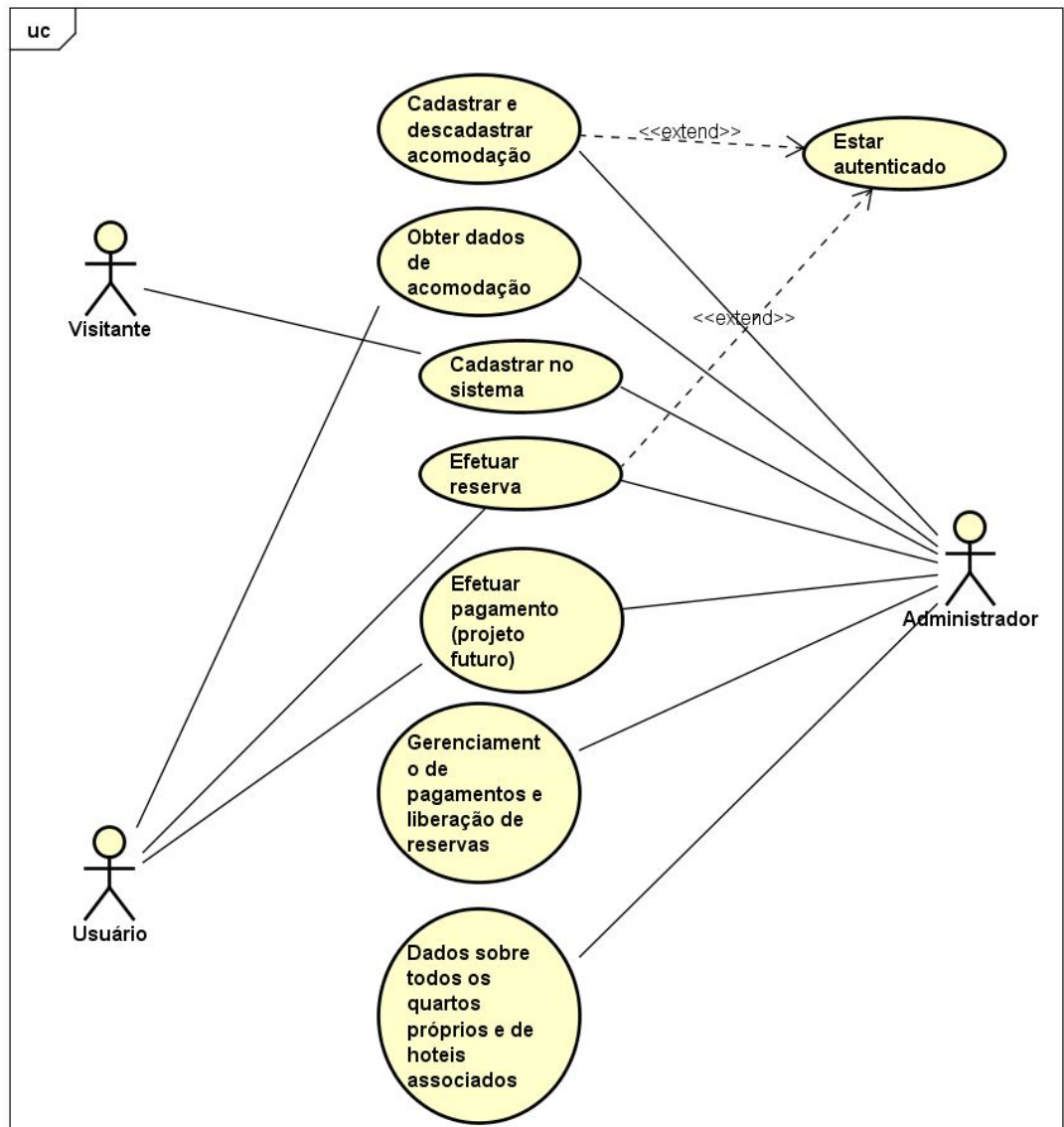
8.2. Visão geral

Neste sistema, há três atores, divididos por **usuário**, **visitante** ou **administrador**, em que todos estes deverão ter acesso à interface; com exceções:

- Visitante, cujo papel é somente se cadastrar no sistema da Samarino Hotels;
- Usuário, que não possui acesso à informações acerca das informações de todos os quartos da Samarino Hotels e de redes hoteleiras associadas.

8.3. Diagrama de Casos de Uso

Abaixo, está o Diagrama de Casos de Uso deste sistema.

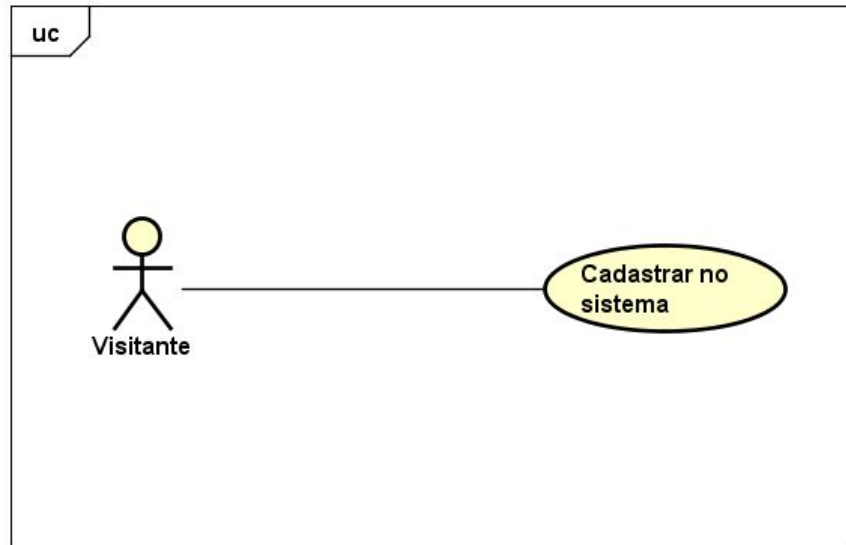


powered by Astah

8.4. Diagramas de Atores

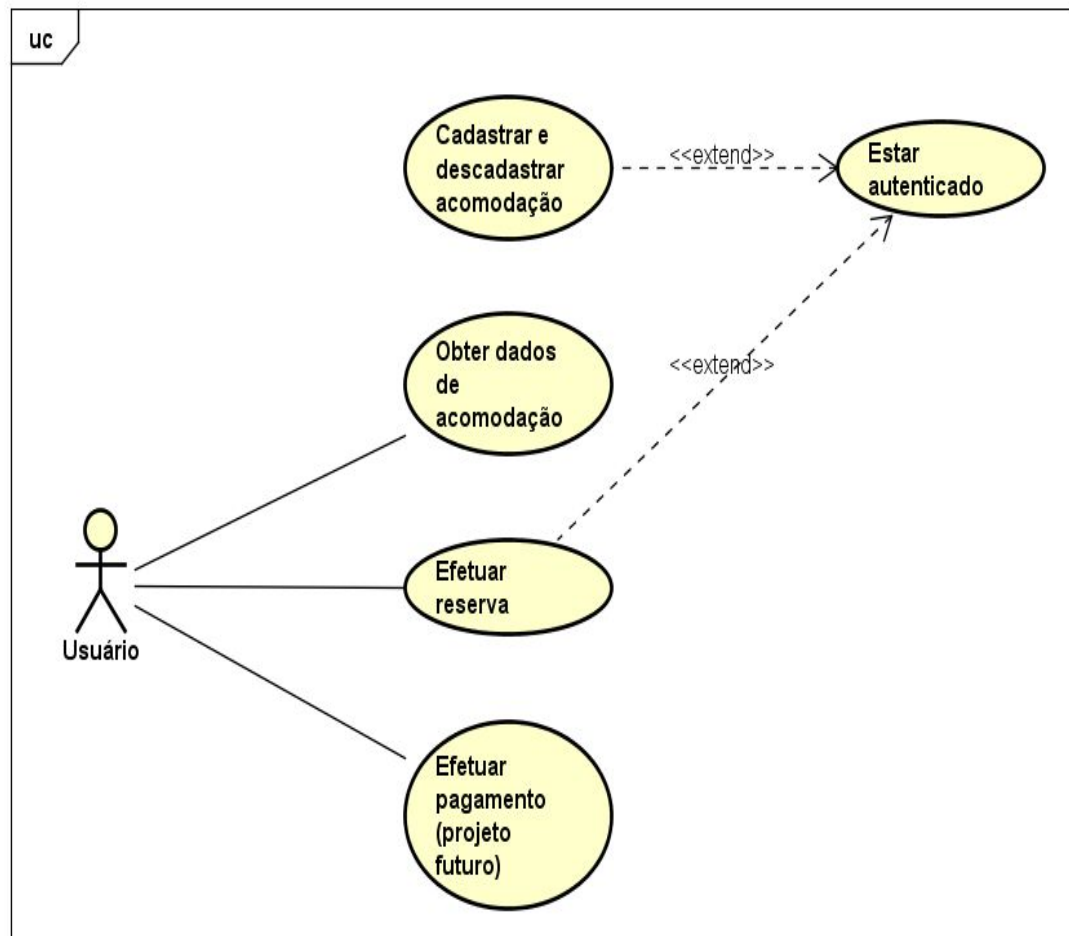
Abaixo, estão os Diagramas para cada Ator deste sistema.

8.4.1. Visitante



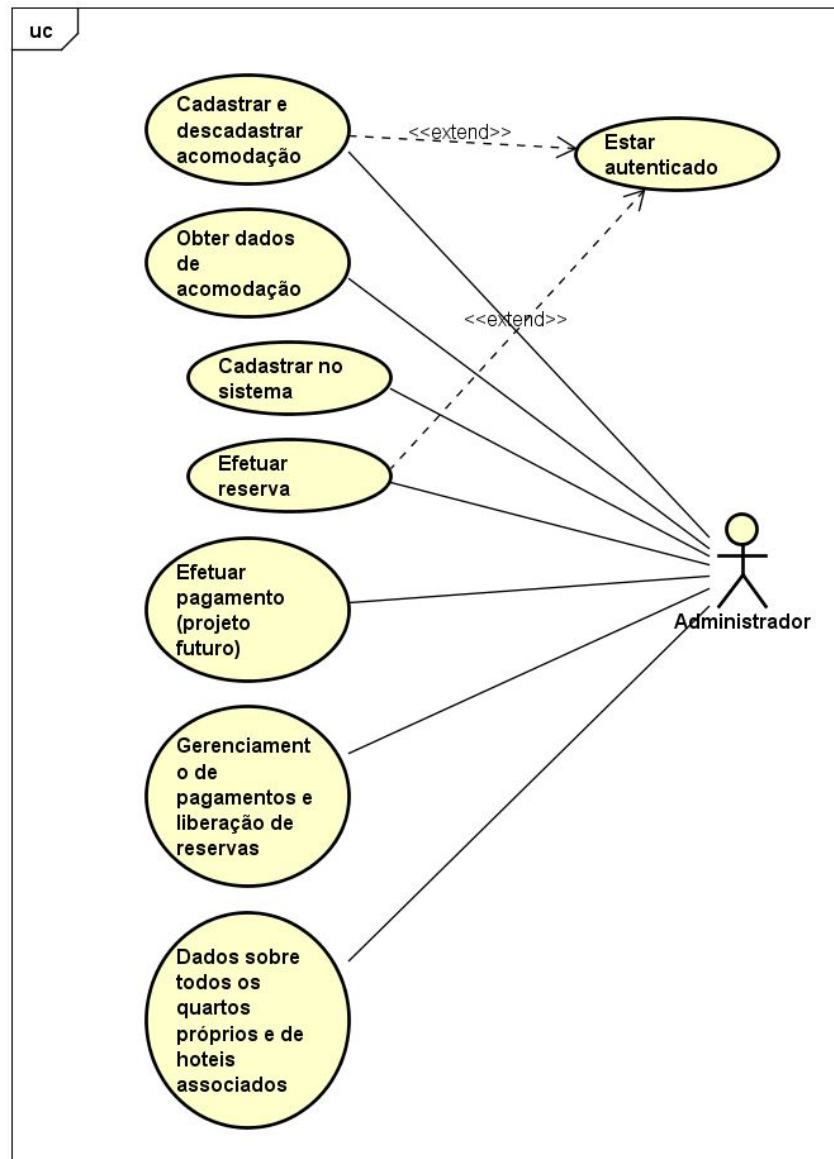
powered by Astah

8.4.2. Usuário



powered by Astah

8.4.3. Administrador



powered by Astah

9. Descrições textuais de Casos de Uso

9.1. Breve descrição

Neste tópico, faremos pontuações acerca de atores, casos de uso e subtemas dos Casos de Uso, descritas em todo o Item 8.

9.2. Breve descrição do atores

- **Visitante:** É o um usuário do sistema que tem acesso restrito as funcionalidades do mesmo. Enquanto ele não estiver autenticado, ou seja, enquanto não possuir login no sistema, ele só poderá realizar seu cadastro no sistema;
- **Usuário:** É o usuário que já possui um cadastro no sistema. Ao estar autenticado ele pode obter dados das acomodações, efetuar reversa em hotéis, e posteriormente efetuar pagamento da sua reserva via pay-pal;
- **Administrador:** É um ator do sistema ao qual gerencia o mesmo. Logo, esse ator poderá cadastrar e descadastrar as acomodações, obter dados das

acomodações, poderá cadastrar outros usuários no sistema, efetuar reservas, gerenciar a liberação das reservas dos usuários de acordo com a verificação do pagamento das reservas e poderá ter acesso aos dados sobre todos os quartos próprios e de hotéis associados.

9.3. Casos de Uso

9.3.1. Obter dados de acomodação

9.3.1.1. Visão Geral

Função de acesso para o Administrador e Usuário (desde que ambos estejam autenticados), em que todos estes visualizam em uma interface todas as informações acerca de um quarto, como lotação máxima, mobiliário (ar-condicionado, frigobar e fogão), quantidade de quartos, banheiro e localização.

9.3.1.2. Ator Principal

Usuário e Administrador.

9.3.1.3. Interessados e Interesses

Obter dados completos sobre o quarto

9.3.1.4. Pré-condição

Estar autenticado.

9.3.1.5. Pós-condição

Retorno de quartos de hotéis disponíveis para tal data ou período, com preço e todas as informações do quarto.

9.3.1.6. Cenários de sucesso

1. O visitante ou o Usuário ou o Administrador visita o sistema e pesquisa informando o período e a cidade a qual busca por acomodações;
2. O sistema retorna as acomodações disponíveis para aquele período e local;
3. O ator seleciona a opção de fazer reserva na acomodação desejada;
4. O sistema valida a reserva.

9.3.1.7. Fluxo Alternativo

3-5 O ator pode cancelar a qualquer momento. Cancelar toda a operação.

4 - O ator visitante não possui cadastro no sistema, logo precisa se cadastrar no sistema. O ator usuário e o Administrador devem estar logado no sistema.

9.3.2. Cadastrar no sistema

9.3.2.1. Visão Geral

O visitante deseja criar um novo cadastro no sistema. Ele abrirá a opção de login. O sistema abrirá uma tela a ser preenchida com dados do visitante, nome do usuário e senha desejada.

O Administrador poderá criar um cadastro para outro visitante que deseja se cadastrar no sistema mas que esteja com algum problema para fazer seu cadastramento.

- 9.3.2.2. Ator Principal
Visitante e Administrador.
- 9.3.2.3. Interessados e Interesses
Criar cadastro no sistema para que possa se autenticar e fazer uso do mesmo.
- 9.3.2.4. Pré-condição
Não possuir nenhum cadastro para o e-mail desejado.
- 9.3.2.5. Pós-condição
Após logado pode navegar livremente dentro do sistema. Pode cadastrar hotéis, fazer reservas.
- 9.3.2.6. Cenários de Sucesso
 - 1. O ator abre a tela de cadastro
 - 2. O sistema apresenta uma tela para que seja preenchida com os dados do novo usuário, como nome, endereço, CPF, RG, e-mail.
 - 3. O ator envia o cadastro
 - 4. O sistema valida o cadastro
 - 5. O sistema apresenta mensagem de “cadastro efetuado com sucesso”.
- 9.3.2.7. Fluxo Alternativo
 - 3- O e-mail pode não ser validado pelo sistema e o ator não ter seu cadastro finalizado.
 - 2-5- O ator pode desistir do cadastro a qualquer momento. Logo, cancelar a solicitação.
- 9.3.3. Efetuar reservas
 - 9.3.3.1. Visão Geral
Usuário e Administrador tem acesso a este caso de uso. É aberto. Com as informações dadas pelo item 9.3.1. o usuário ou administrador poderá realizar a reserva de um quarto conforme disponibilidade.
 - 9.3.3.2. Ator Principal
Usuário e Administrador.
 - 9.3.3.3. Interessados e Interesses
Realizar reserva de quarto.
 - 9.3.3.4. Pré-condição
Estar autenticado no sistema.
 - 9.3.3.5. Pós-condição
Confirmação da reserva via site.
 - 9.3.3.6. Cenários de Sucesso
 - 1. O ator entra na aba “Reservar”;
 - 2. Com os dados informados no item 9.3.1 (sobre Dados da Acomodação), o sistema pedirá para confirmar a reserva;
 - 3. O sistema enviará o aviso: “Reserva efetuada com sucesso!”.
 - 9.3.3.7. Fluxo Alternativo

1-3: O ator poderá cancelar a opção de reserva.

9.3.4. Gerenciamento de pagamentos e liberação de reservas

9.3.4.1. Visão Geral

Função exclusiva do Administrador, em que o ator checa os pagamentos e libera as reservas, caso o hóspede tenha pago. Também checa as vagas disponíveis.

9.3.4.2. Ator Principal

Administrador.

9.3.4.3. Interessados e Interesses

Validar o pagamento e realizar a liberação do quarto reservado, bem como checar se a acomodação está disponível.

9.3.4.4. Pré-condição

Estar logado e a reserva ter sido efetuada.

9.3.4.5. Pós-condição

Aprovar ou reprovar a reserva.

9.3.4.6. Cenários de Sucesso

1. O ator entra na aba de “Pagamentos”;
2. É aberto uma planilha com as reservas feitas e os pagamentos efetuados;
3. Caso o pagamento tenha sido aprovado, a reserva é confirmada.

9.3.4.7. Fluxo Alternativo

2: Caso não tenha vagas disponíveis, ele pode realocar ou negar a reserva.

3: Caso o hóspede não tenha pago a hospedagem, esta reserva estará anulada.

9.3.5. Dados sobre todos os quartos próprios e de hotéis associados

9.3.5.1. Visão Geral

O ator tem acesso a todas as informações dos quartos e hotéis, sejam eles próprios ou dos associados ao sistema.

9.3.5.2. Ator Principal

Administrador e Usuário (quando o usuário for “fornecedor” de hotéis).

9.3.5.3. Interessados e Interesses

Mostrar dados e informações referentes aos hotéis e quartos como status de reserva, quantidade de quartos disponíveis naquele hotel, quantidade de hotéis disponíveis.

9.3.5.4. Pré-condição

Estar logado no sistema, ou seja, estar autenticado.

9.3.5.5. Pós-condição

Retornar as informações desejadas ao solicitar uma busca.

9.3.5.6. Cenários de Sucesso

1. O Administrador clica para obter dados dos quartos e hotéis ;

2. O sistema retorna a lista de todos os quartos e hotéis cadastrados no sistema, seja ele com status de reserva ou não.;
3. O administrador seleciona o quarto ou hotel desejado dessa lista;
4. O sistema traz a escolha com os dados detalhados

9.3.5.7. Fluxo Alternativo

1-4 O ator pode cancelar a qualquer momento a busca.

9.4. Subfluxos

9.4.1. Caso de Uso “Cadastrar”

Aguardar a confirmação do e-mail para validar o cadastro.

10. Lista de Ferramentas

Dentre os principais, podemos destacar os componentes, templates, diretivas, roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas, como a de executar os testes unitários de uma aplicação.

Alguns outros pontos dessa plataforma que merecem destaque são o fato de que ela é open source, possui uma grande comunidade, existem várias empresas utilizando e tem muito material de estudo para quem deseja se aperfeiçoar.

Abaixo, estão as ferramentas utilizadas para a implementação do software e arquitetura do banco de dados:

- **Java** é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferente das linguagens de programação convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é interpretado por uma máquina virtual (Java Virtual Machine, mais conhecida pela sua abreviação JVM). A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não é a sua única linguagem;
- **PostgreSQL** é um sistema gerenciador de banco de dados objeto relacional (SGBDOR), desenvolvido como projeto de código aberto;
- **Astah** é uma ferramenta para elaboração de diagramas na linguagem UML, utilizadas como base no item 8 deste documento;
- **Angular** é uma plataforma e framework para construção da interface de aplicações usando HTML, CSS e, principalmente, JavaScript, criada pelos desenvolvedores da Google. Ele possui alguns elementos básicos que tornam essa construção interessante. Angular nos ajuda a criar Single-Page Applications com uma qualidade e produtividade surpreendente.

11. Descrição de arquitetura de software

11.1. Propósito

Este documento descreve todo o processo de filosofia, decisões, justificativas, elementos significativos e todos os outros aspectos acerca do sistema.

11.2. Objetivo da arquitetura e filosofia

A arquitetura de software de um sistema serve para definir a organização das funcionalidades desse sistema e propriedades ou requisitos não funcionais suportados pelo mesmo. O projeto de arquitetura é de suma importância para estruturar a engenharia do software, com seus projetos de manutenção, análise de requisitos, bem como suas funcionalidades, restrições e análise do hardware (desempenho, manutenção do dispositivo, dentre outros).

11.3. Suposições e dependências

Para que o sistema da Samarino Hotels possa ser executado 24 horas por dia, 7 dias por semana, dependemos de:

- Servidores de alto desempenho (armazenamento e processamento);
- Bons provedores de internet;
- Hospedagem de site;
- Sala de Servidores equipada com ar-condicionado (para que os servidores funcionem 24 horas por dia).

11.4. Requisitos significativos na arquitetura

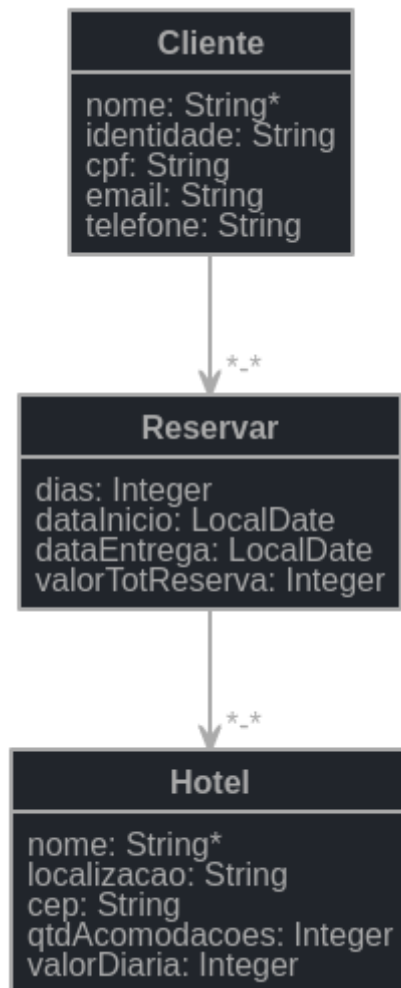
Não utilizamos referências externas para a execução deste projeto.

11.5. Decisões, contrastes e justificativas

Abaixo, registramos as intervenções feitas em decorrência do curto espaço de tempo para que seja realizado mais requisições de validações:

- Decidimos nos basear no princípio de Desenvolvimento em Camadas, baseado na divisão de tarefas em subcamadas para que os Produtos de Trabalho possam ser distribuídos na equipe;
- Contrastando os requisitos do usuário acerca do visitante, nós alteramos o requisito do mesmo somente para se cadastrar ao sistema da Samarino Hotels, em decorrência de estarmos focados em ambiente de usuário.

11.6. Mecanismos de arquitetura



12. Descrição de infraestrutura de implantação

12.1. Arquivos de código-fonte

Os arquivos de código fonte segue no seguinte link no GitHub:

<https://github.com/arturrenann/samarino>

12.2. Arquivos de dados

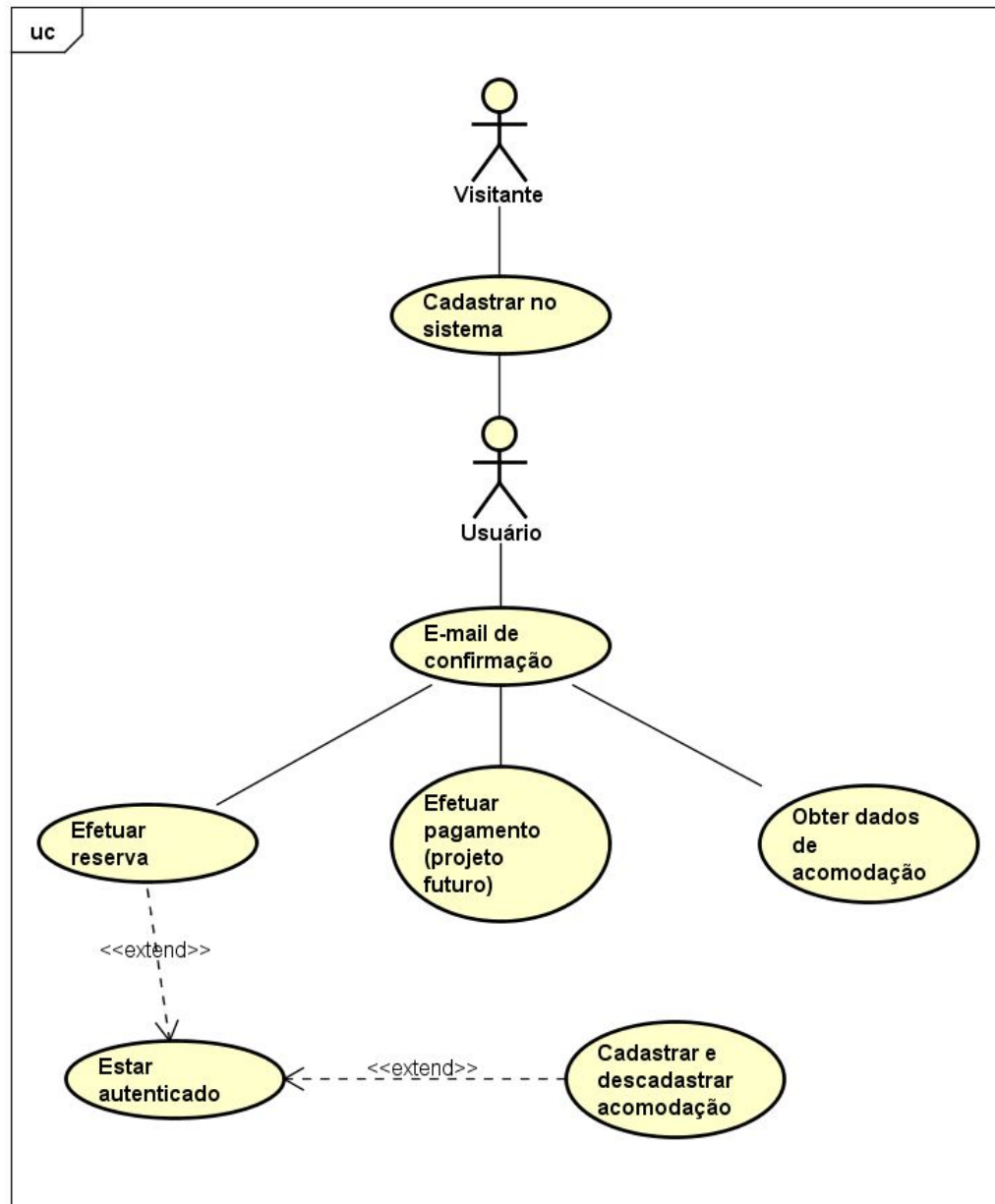
Os dados seguiram vazios para que o usuário e o admin possam criar os hotéis disponíveis e os clientes que podem fazer determinadas reservas, segue em anexo a imagem de como foi arquitetado as tabelas de nosso

banco de dados, utilizando o PostgreSQL.

Table	Owner
cliente	TrabalhoES
cliente_reserva	TrabalhoES
databasechangelog	TrabalhoES
databasechangelog...	TrabalhoES
hotel	TrabalhoES
jhi_authority	TrabalhoES
jhi_persistent_audi...	TrabalhoES
jhi_persistent_audi...	TrabalhoES
jhi_user	TrabalhoES
jhi_user_authority	TrabalhoES
reservar	TrabalhoES
reservar_hoteis	TrabalhoES

13. Implementação de casos de uso relativos a um subsistema

Conforme descrito no item 8 acerca da autenticação no sistema para usuários, o subsistema de autenticação está funcionando, enviando ao usuário cadastrado um e-mail para confirmação para que seu usuário seja validado no sistema. Abaixo, está o caso de uso acerca do subsistema:



powered by Astah

14. Casos de teste da implementação.

14.1. Entradas

Como o software foi feito para uma versão Alpha, todos os sistemas de cadastros, podem ser feitos pelos diferentes usuários, no qual, a única requisição obrigatória seria a de Nome, logo com todas as telas produzidas, para a utilização em meio a um ambiente de desenvolvimento, no qual os acessos só podem ser feitos por quem está logado no sistema, por fato de ambiente não está completamente desenvolvido.

Seguem os casos de implementação para o teste unitário.

```
private static final String DEFAULT_NOME = "AAAAAAAAAA";
private static final String UPDATED_NOME = "BBBBBBBBBB";
```

```

private static final String DEFAULT_IDENTIDADE = "AAAAAAAAAA";
private static final String UPDATED_IDENTIDADE = "BBBBBBBBBB";

private static final String DEFAULT_CPF = "AAAAAAAAAA";
private static final String UPDATED_CPF = "BBBBBBBBBB";

private static final String DEFAULT_EMAIL = "AAAAAAAAAA";
private static final String UPDATED_EMAIL = "BBBBBBBBBB";

private static final String DEFAULT_TELEFONE = "AAAAAAAAAA";
private static final String UPDATED_TELEFONE = "BBBBBBBBBB";

```

14.2. Script

Nome do Teste	Teste de criação de cliente
Caso de Uso Testado	Criar Teste
Descrição do Teste	Foi feito o teste unitário para a criação de um cliente
Pré-condições	-
Pós-condições	-
Observações:	Segue o código de teste unitário feito em Java.
Resultado: (Aprovado/Reprovado/Aviso/Incompleto)	Simulação da criação, esperasse um resultado TRUE.

	Passo do Teste	Expectativa de Resultado do Teste	Aprovado	Reprovado
1	createCliente()	TRUE	x	
2	updateCliente()	TRUE	x	

```

@Test
@Transactional
public void createCliente() throws Exception {
    int databaseSizeBeforeCreate =
        clienteRepository.findAll().size();

    // Create the Cliente
    restClienteMockMvc.perform(post("/api/clientes")

```



```

        .contentType(TestUtil.APPLICATION_JSON_UTF8)

        .content(TestUtil.convertObjectToJsonBytes(cliente)))
        .andExpect(status().isCreated());

        // Validate the Cliente in the database
        List<Cliente> clienteList = clienteRepository.findAll();

        assertThat(clienteList).hasSize(databaseSizeBeforeCreate +
1);
        Cliente testCliente = clienteList.get(clienteList.size() -
1);

        assertThat(testCliente.getNome()).isEqualTo(DEFAULT_NO
ME);

        assertThat(testCliente.getIdentidade()).isEqualTo(DEFAULT
_IDENTIDADE);

        assertThat(testCliente.getCpf()).isEqualTo(DEFAULT_CPF);

        assertThat(testCliente.getEmail()).isEqualTo(DEFAULT_EM
AIL);

        assertThat(testCliente.getTelefone()).isEqualTo(DEFAULT_
TELEFONE);
    }

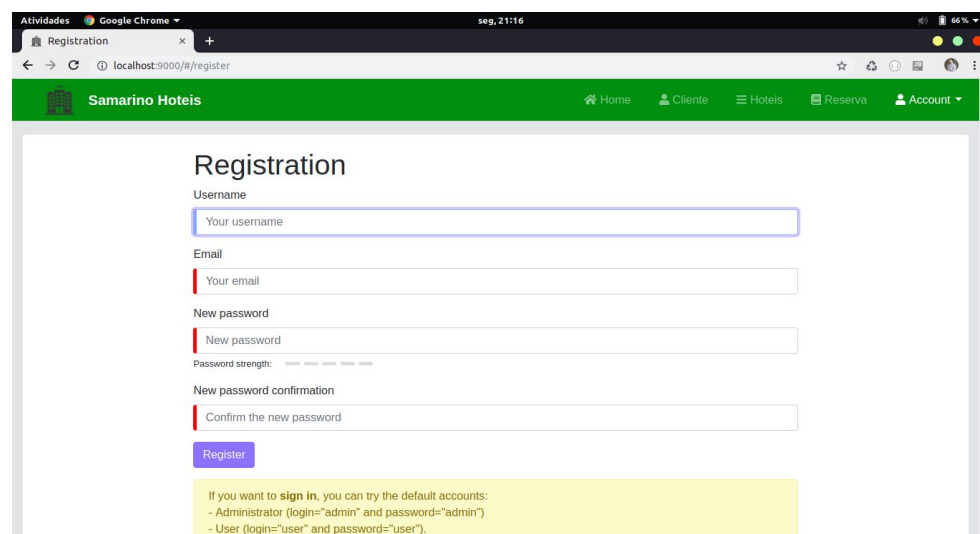
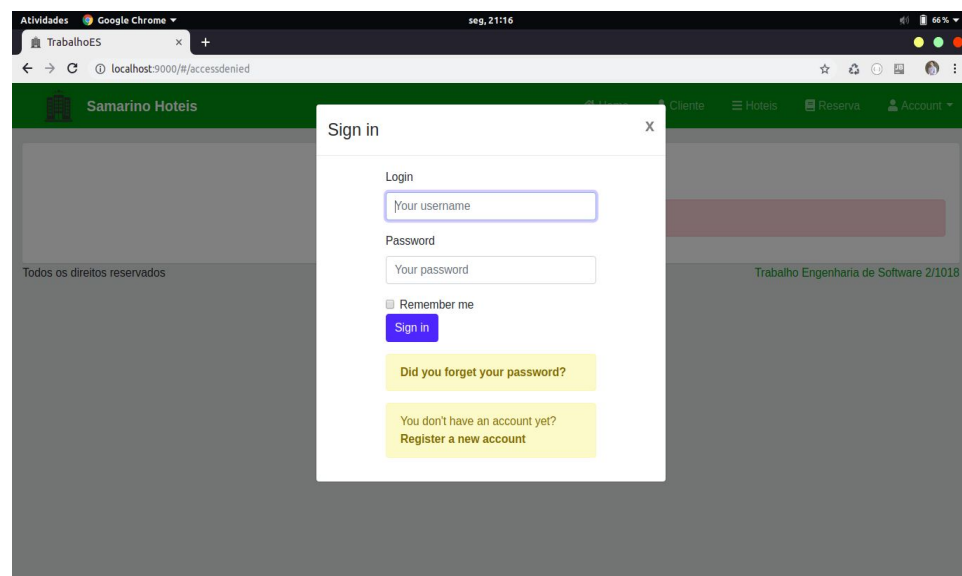
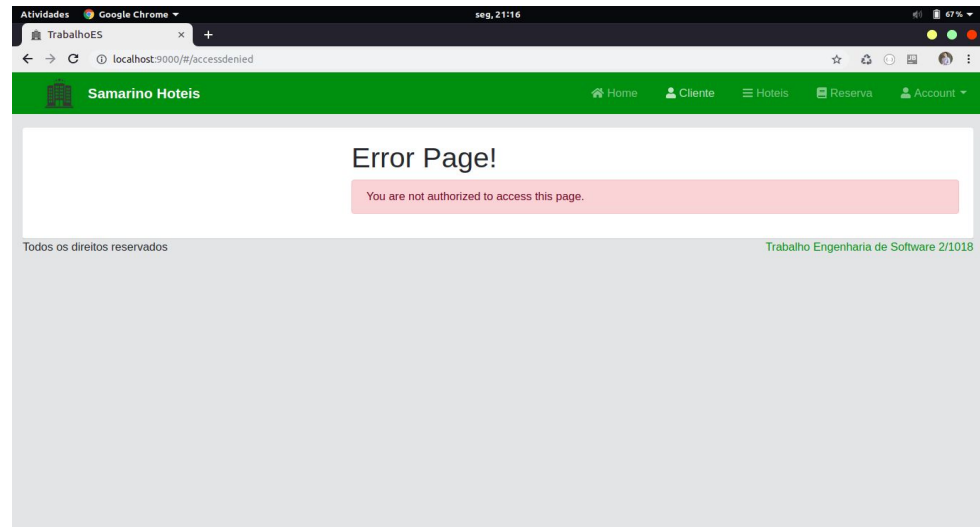
```

14.3. Resultados Esperados

Em um caso de login no sistema:

Seriam esperados para a versão Alpha do projeto, onde seria ser possível a implementação de CRUD e sistema de Login e registro no sistema o qual todos estão em conformidade com o que foi proposto pelos requisitos.

Testador: Artur Renan Alves Barbosa



Todas essas em conformidade com os requisitos.

14.4. Critérios de Avaliação

Todos os critérios foram feitos com testes de execução no qual são testados todos os possíveis casos para casos de uso no sistema para a liberação de uma versão Alpha com o mínimo de funcionalidades necessárias para o funcionamento do sistema Samarino, para um futuro sistema completo de gerenciamento de hotéis. Nos casos mostrados, cadastro de cliente é feito o teste unitário e no teste de funcionamento é mostrado o funcionamento de usuário não logado no sistema.
