

# Inn Tech

Ana Cristina Carvalho Silva  
Departamento de Engenharias e Tecnologias  
UFERSA  
Pau dos Ferros/RN, Brasil  
ana.silva54276@alunos.ufersa.edu.br

Matheus Vinícius Ferreira Pinheiro  
Departamento de Engenharias e Tecnologias  
Ufersa  
Pau dos Ferros/RN, Brasil  
matheus.pinheiro@alunos.ufersa.edu.br

Vinicius Silva  
Departamento de Engenharias e Tecnologias  
UFERSA  
Pau dos Ferros/RN, Brasil  
vinicius.silva94304@alunos.ufersa.edu.br

***Index Terms*—Pousada. Aplicação. Software.**

## I. INTRODUÇÃO

Para construção de uma solução que beneficie os principais interessados, sejam esses clientes e usuários, é fundamental que as necessidades e principais dificuldades enfrentadas sejam mapeadas e compreendidas.

Entretanto, a construção inicial do projeto, com etapas de refinamento e definição de requisitos não são os únicos fatores que garantem o sucesso de todas as etapas do ciclo de vida de um sistema, que vão desde o planejamento até as manutenções, sendo também insuficiente para satisfação total dos interessados.

Considerando a necessidade de garantir a qualidade da solução, existem técnicas da engenharia de software que auxiliam no processo, a citar: técnicas de Verificação e Validação (V&V), sendo empregadas com o propósito de assegurar a qualidade de um produto ou sistema. Elas permitem determinar se o projeto está sendo construído corretamente por meio da verificação e se ele atende às exigências e necessidades do cliente por meio da validação. Esse procedimento é abordado mediante o uso das técnicas de (V&V).

Portanto, por meio deste trabalho, busca-se aplicar os conhecimentos adquiridos em sala de aula sobre verificação e validação na prática, utilizando-os no projeto denominado *Inn Tech*, que tem como propósito revolucionar a gestão de pequenas pousadas, proporcionando aos proprietários um software que simplifica e agiliza o processo de gestão do estabelecimento. Com isso, o dono da empresa poderá concentrar os seus esforços no que mais importa, os seus hóspedes, com uma solução simples, prática e eficaz. O sistema oferece controle de hóspedes, reservas de quartos, fluxo de caixa e relatórios financeiros, mostrando-se indispensável para que uma gerência bem-sucedida seja alcançada.

## II. FUNDAMENTAÇÃO TEÓRICA

Teste caixa branca compreende o exercício da lógica interna e das interfaces de todos os componentes da aplicação.

Teste caixa preta exercita as entradas e saídas do software objetivando descobrir erros que podem estar presentes no funcionamento, comportamento e/ou desempenho do programa.

Verificação consiste em verificar a existência de falhas e problemas com o software antes que a aplicação seja entregue ao cliente ou ao público final. Essas falhas e problemas podem se manifestar, por exemplo, no código, nas funcionalidades, na interface e em outras partes do sistema (PRESSMAN; PENTEADO, 2006).

Validação tem como objetivo entender se o produto que está sendo analisado atende às expectativas do cliente (PRESSMAN; PENTEADO, 2006).

O teste de unidade compreende testar, individualmente, cada unidade presente no sistema (PRESSMAN; PENTEADO, 2006).

Prototipação consiste em criar um esboço inicial de uma ideia para que, posteriormente, o projeto seja elaborado (HOVEDA, 2021). Na prototipação das telas, os modelos apresentam níveis diferentes de fidelidade, sendo divididos em bixa, média e alta fidelidade (FRANCISCO, 2021). *Figma* é uma ferramenta que possibilita a construção de protótipos em diferentes níveis de fidelidade (HARADA, 2022).

De acordo com (REINEHR, 2020) existem diversas definições para requisitos, e o Glossário padrão de terminologia de engenharia de software do Institute of Electrical and Electronic Engineers (IEEE, 1990) define requisito como:

“1. Uma condição ou capacidade necessária para um usuário resolver um problema ou alcançar um objetivo. 2. Uma condição ou capacidade que deve ser atendida ou tida por um sistema ou componente do sistema para satisfazer a um contrato, padrão, especificação ou outro documento formalmente imposto. 3. Uma representação documentada de uma condição ou capacidade conforme estabelecido em 1 e 2”.

## III. TRABALHOS RELACIONADOS

O setor de hotelaria vem crescendo constantemente; dados disponibilizados pelo governo revelam um aumento de 74% nos primeiros meses de 2022, e os números continuam a crescer (TURISMO, 2022). Com esse crescimento no setor, a

informatização dos sistemas torna-se cada vez mais essencial para atender à demanda do mercado. Nesse contexto, nosso software se posiciona no mercado como uma solução simples e acessível para pousadas de pequeno porte, visando o gerenciamento e agilidade. Alguns elementos são essenciais para a escolha do melhor software para cada caso; alguns deles são: orçamento, tamanho do estabelecimento, número de franquias, nicho de serviço, entre outros. Após uma pesquisa de mercado, foi possível perceber que a maioria dos softwares mais bem avaliados no mercado é de origem estrangeira. Como resultado, suas assinaturas são cobradas em dólares, o que pode resultar em um aumento significativo dos custos, dependendo da moeda local. Para contextualizar em nossa realidade, o dólar tem um valor de mercado cinco vezes maior que o real, o que acaba tornando inviável a adoção desses sistemas por pequenos estabelecimentos. Nosso software entra no mercado para preencher essa lacuna, fornecendo um sistema completo e fácil de usar capaz de gerenciar todas as atividades essenciais de uma pousada. Ele é projetado para impulsionar o negócio, a um custo muito mais acessível em comparação aos grandes softwares disponíveis no mercado.

#### IV. PROPOSTA DE SOLUÇÃO

Conforme (PRESSMAN; PENTEADO, 2006), a engenharia de requisitos fornece um mecanismo adequado para compreender as necessidades e exigências do cliente, analisar a viabilidade, validar a especificação e administrar os requisitos à medida que eles são transformados em um sistema em operação.

Com apoio da engenharia de requisitos, por meio de uma entrevista com cliente, conseguimos identificar o domínio da aplicação, os processos envolvidos no dia a dia do proprietário da pousada, e as dores do fluxo atual. Atualmente é utilizado um método ultrapassado, demorado, e manual. Dado que as atividades são realizadas de forma manual e com intervenção humana, há também um alto índice de probabilidade de erros.

##### A. Requisitos

Nesta subseção são apresentados os requisitos do sistema, possibilitando a compreensão das funcionalidades e como o sistema deve se comportar, obedecendo as regras de negócio.

As prioridades dos requisitos foram estabelecidas considerando as principais necessidades para o cliente, propondo uma primeira entrega de um produto mínimo viável (MVP), sendo elas: alta, média e baixa, de forma que o nome sugere o grau de importância para os interessados.

Na tabela I serão apresentados os requisitos do projeto. A tabela está subdividida em colunas: nome do requisito, descrição e prioridade.

##### B. Regras de negócio

Nesta subseção serão apresentadas as regras de negócio, onde serão descritas algumas restrições do sistema.

A tabela II está subdividida em colunas: id, nome, descrição

Tabela I: Requisitos de sistema

Nome do requisito	Descrição	Prioridade
RF001 Login	O sistema deve permitir que o usuário realize o <i>login</i> . Para isso, ele deve informar os seguintes dados: login e senha.	Alta
RF002 Cadastrar usuário	O sistema deve permitir que o administrador realize o cadastro de um novo usuário. Para isso, deve ser informado os seguintes dados: usuário, senha e confirmação de senha.	Alta
RF003 Buscar usuário	O sistema deve permitir que o administrador busque usuários por: Usuário.	Média
RF004 Editar usuário	O sistema deve permitir que o administrador altere informações do usuário, permitindo que qualquer informação de cadastro seja atualizada.	Alta
RF005 Excluir usuário	O sistema deve permitir que administrador exclua o registro de um usuário.	Alta
RF006 Cadastrar hóspede	O sistema deve permitir que o funcionário cadastre um novo hóspede, para isso, ele deve informar os seguintes dados: CPF, nome, e telefone.	Alta
RF007 Editar hóspede	O sistema deve permitir que o funcionário altere informações do hóspede, permitindo que qualquer informação de cadastro seja atualizada.	Média
RF008 Excluir hóspede	O sistema deve permitir que o funcionário exclua o registro de hóspede.	Média
RF009 Buscar hóspede	O sistema deve permitir buscar hóspedes, para isso ele deve informar o nome. Como resultado do filtro devem ser exibidos os registros correspondentes.	Alta
RF010 Cadastrar quarto	O sistema deve permitir que o administrador cadastre um novo quarto, ele deve informar os seguintes dados: Número, capacidade, descrição (opcional), e ocupação.	Alta
RF011 Editar quarto	O sistema deve permitir que o administrador altere informações do quarto, permitindo que qualquer informação de cadastro seja atualizada.	Alta
RF012 Excluir quarto	O sistema deve permitir que administrador exclua o registro de quarto.	Alta
RF013 Cadastrar reserva	O sistema deve permitir que o funcionário cadastre uma nova reserva, ele deve informar os seguintes dados: data e horário da reserva, valor, informar se o pagamento está confirmado (sim, não), observações (opcional), selecionar o quarto e os hóspedes, e data de saída.	Alta
RF016 Excluir reserva	O sistema deve permitir que usuário exclua o registro da reserva.	Baixa
RF017 Listagem de pagamento	O sistema deve permitir que o usuário visualize uma listagem dos pagamentos das reservas. Devem ser listados todos os registros de reservas pagos.	Alta
RF018 Buscar pagamento	O sistema deve permitir buscar pagamentos pelo número da reserva e pelo status do pagamento. Como resultado do filtro, o sistema deve exibir as informações referentes aos dados inseridos.	Alta
RF019 Cadastrar saídas	O sistema deve permitir que o proprietário cadastre uma nova saída, ele deve informar os seguintes dados: valor, motivo, e observação (opcional). O sistema deve guardar a informação do criador da saída.	Alta
RF020 Editar saídas	O sistema deve permitir que o proprietário altere informações da saída, permitindo que o valor, data de saída e motivo sejam atualizados.	Alta
RF021 Buscar saídas	O sistema deve permitir que o proprietário busque saídas por: data de saída, e criador. Como resultado do filtro devem ser exibidos os registros correspondentes.	Média
RF022 Excluir saídas	O sistema deve permitir que usuário exclua o registro de uma saída.	Baixa

Fonte: Autoria Própria.

Tabela II: Regras de negócio

ID	Nome	Descrição
RN001	Tipo de usuário	No cadastro de usuários, o sistema deve oferecer a possibilidade de selecionar o tipo de usuário, seja como Administrador ou não

Fonte: Autoria Própria.

## V. ABORDAGEM

O processo de Verificação e Validação (V&V) é integrado ao longo de todo o ciclo de vida do projeto, e é subdividido em duas técnicas fundamentais:

**Inspecção de Software:** Esta técnica envolve a análise dos artefatos do projeto, incluindo documentos de requisitos, diagramas de projeto e código-fonte (PRESSMAN; PENTEADO, 2006).

**Testes de Software:** Esta abordagem implica a aplicação de técnicas de teste (PRESSMAN; PENTEADO, 2006).

Foram empregadas as técnicas de Inspecção de Software e Testes de Software. Mediante o acesso ao código e a documentação, foi usada a abordagem caixa branca. Como o intuito de testar as unidades da aplicação, testes unitários foram parte do roteiro empregado. Ademais, foi elaborado um protótipo baseado nos requisitos elicitados e, posteriormente fora realizada uma pesquisa de usabilidade por meio SUS (*System Usability Scale*).

### A. Inspecção de Software

(VETTORAZZO, 2018) argumenta que a inspecção de software tem como objetivo descobrir possíveis defeitos, e que essas podem ser feitas nos requisitos, projetos, nos dados de configuração e testes utilizados para descobrir erros.

Dado que a análise dos artefatos do projeto são cruciais para assegurar que o sistema que está sendo desenvolvido da maneira adequada, no presente trabalho, foi aplicada a técnica de inspecção no documento de requisito.

Conforme (VETTORAZZO, 2018), durante uma inspecção, é recomendado que um modelo de *checklist* seja utilizado para registrar os erros encontrados. Esse modelo específico é apresentado na página 81 do livro 'Engenharia de Software' e será empregado na execução do *checklist* de validação de requisitos. Na tabela III serão apresentados os resultados da aplicação desse modelo.

A autora defende, ainda, que o padrão Software **Engineering Body of Knowledge (SWEBOK)** apoia na validação dos requisitos, possuindo fases como: revisão de requisitos, prototipação; validação de modelos; testes de aceitação. Essas foram consideradas na construção deste trabalho.

### B. Testes de Software

O sistema possui três módulos: core, financeiro e reserva. No presente trabalho, foi utilizada a técnica de caixa branca denominada teste unitário. O teste unitário é uma prática de engenharia de software na qual unidades individuais de um programa, como funções, métodos ou classes, são testadas para garantir que funcionem como esperado. O objetivo principal dos testes unitários é isolar cada parte do código para verificar se ela produz o resultado desejado e se comporta corretamente, independentemente do restante do sistema.

No módulo core, a classe hospede.py foi testada. Um caso de teste foi desenvolvido para criar um hóspede e, em seguida, consultar esse hóspede no banco de dados com os dados fornecidos, a fim de garantir a consistência das informações e a manipulação adequada dos dados por parte do sistema.

Tabela III: Checklist de validação de requisitos

ITEM	ITEM PARA VERIFICAÇÃO	SIM	NÃO
Não ambíguo: É não ambíguo se, e somente se, cada requisito declarado seja suscetível a apenas uma interpretação			
1	Cada requisito está descrito com clareza, concisão e sem ambiguidade?	X	
Consistente: É consistente se, e somente se, nenhum dos requisitos do documento, tomado individualmente, está em conflito com qualquer outro requisito do mesmo documento.			
2	Existem requisitos conflitantes?		X
Completo: É completo se, e somente se, conter toda, e apenas a informação necessária para que o software correspondente seja produzido.			
3	Existem requisitos implícitos	X	
4	Os requisitos exibem a distinção clara entre funções, dados e restrições?	X	
5	As restrições e dependências foram claramente descritas?	X	
6	Existem requisitos que contém algum nível desnecessário de detalhe do projeto?		X
7	Os requisitos definem todas as informações a serem apresentadas aos usuários?		X
8	Os requisitos descrevem as respostas do sistema ao usuário devido às condições de erro?		X
9	Existem situações não tratadas pelos requisitos que precisam ser consideradas?		X
10	O documento possui realmente toda a informação prometida em sua introdução?	X	

Fonte: (VETTORAZZO, 2018)

Também foi realizado um teste na função str, que é responsável por retornar a representação em formato de *string* do objeto. Nesse caso, o teste utiliza o hóspede criado para verificar se o retorno dessa função está de acordo com os padrões estabelecidos.

No módulo financeiro, foram testadas as classes pagamento.py e saida.py. Foram desenvolvidos casos de teste para ambas as classes. No caso do pagamento, foi criada uma instância de pagamento, em seguida, o valor foi atualizado e a persistência no banco de dados foi verificada, tanto em relação ao valor quanto à alteração. Para o caso da saída, um usuário foi criado e associado a uma saída no sistema. Foi verificado no banco de dados se tanto o valor quanto o usuário que realizou a saída estão de acordo com as saídas esperadas.

No módulo de reserva, foi realizado o teste da classe quarto.py. Foi desenvolvido um caso de teste para a criação de um quarto e, em seguida, foram recuperados os dados. Verificamos se tanto os dados do quarto quanto o usuário que realizou o procedimento estão consistentes com o esperado.

Para realizar os casos de teste, utilizamos o módulo de testes integrado do Django, conhecido como "*Django Test Framework*", que oferece uma variedade de ferramentas para testes unitários, de integração e funcionais. No caso do nosso sistema, ele foi empregado para a realização de testes unitários. Além disso, para gerar análises sobre os testes realizados, utilizamos a biblioteca "*coverage*", que exibe informações relacionadas à cobertura dos testes. No nosso sistema, a cobertura atingiu 78% dos testes unitários.

### C. Usabilidade do sistema

Esta subseção detalha as informações referentes ao protótipo, a usabilidade do Sistema e os resultados obtidos por meio da pesquisa realizada.

Depois que os requisitos foram elicitados, foi elaborado um protótipo de alta fidelidade. Depois de finalizado, foi realizado um formulário utilizando o método SUS com o objetivo de medir o grau de usabilidade da aplicação. Esse é uma maneira que ajuda a avaliar a efetividade, eficiência e satisfação do que está sendo submetido ao método (BROOKE, 1996; TEIXEIRA, 2015). O protótipo da aplicação está disponível em: <https://encurtador.com.br/fiER1>.

A divulgação do formulário ao público, que foi realizada por meio das redes sociais, começou no dia 18 de outubro de 2023 e ficou aberto até o dia 19 de outubro do mesmo ano. No total, foram obtidas 43 respostas. É importante ressaltar que, só era possível responder a pesquisa se o respondente concordasse em disponibilizar os dados fornecidos. Não foram coletados nenhum dado como nome, e-mail, data de nascimento ou qualquer outra informação pessoal dos usuários, apenas as respostas inseridas no formulário referentes as perguntas.

É importante mencionar, também, que não é possível identificar nenhum dos indivíduos que se disponibilizaram em responder nem mesmo relacionar as respostas a eles. Vale ressaltar, ainda, que a possibilidade do efeito de bola de neve não é descartada, isto é, uma pessoa que mandou para outra, que mandou para outra e assim sucessivamente (GOODMAN, 1961).

A primeira seção do formulário é composta pela descrição da pesquisa e pelo consentimento necessário para prosseguir com ela. A segunda e última seção apresenta o link do protótipo e as perguntas abordadas pelo SUS. O formulário está disponível em: <https://forms.gle/DBbV1dD7MEkqpVmy5>.

A seguir, estão listadas as perguntas do formulário (TREND, 2023):

- 1) Eu acho que gostaria de usar esse sistema com frequência.
- 2) Eu acho o sistema desnecessariamente complexo.
- 3) Eu achei o sistema fácil de usar.
- 4) Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
- 5) Eu acho que as várias funções do sistema estão muito bem integradas.
- 6) Eu acho que o sistema apresenta muita inconsistência.
- 7) Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
- 8) Eu achei o sistema atrapalhado de usar.
- 9) Eu me senti confiante ao usar o sistema.
- 10) Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Depois que o formulário foi encerrado, o cálculo do *score* do SUS foi realizado para analisar os resultados. Para todas as perguntas de números ímpares, é subtraído 1 a partir da pontuação fornecida pela resposta do usuário. Já para calcular as perguntas de números pares, deve-se subtrair de 5 a resposta dada. Depois disso, próximo passo é somar os valores de cada respondente e multiplicar o valor total por 2,5. Na pontuação final, os valores variam em uma escala que vai de 0 a 100

(BROOKE, 1996; TEIXEIRA, 2015). O resultado obtido por meio da pesquisa será comparado a IV (TREND, 2023).

Tabela IV: Escala de pontuação do SUS

PONTUAÇÃO SUS	NOTA	CLASSIFICAÇÃO
Maior que 80,3	A	Excelente
68 - 80,3	B	Bom
68	C	Ok
51 - 68	D	Ruim
Menor que 51	E	Horrível

Fonte: Trend (2023).

Para que a usabilidade seja considerada como excelente e receba a nota A, sendo essa a melhor pontuação possível de ser alcançada, é necessário que os valores obtidos sejam maiores que 80,3. Para valores entre 68 e 80,3, a nota recebida será B e a pontuação é considerada como boa. Se a pontuação for 68, isso resultará na nota C e será listada como ok, ficando, assim, na média da escala. A nota D é atribuída quando a pontuação está entre 51 e 68. Caso seja esse o resultado, ele será considerado como ruim. Por fim, estando o *score* menor que 51, será compreendida como horrível, a pior pontuação possível a ser alcançada (TREND, 2023).

Após coletar as respostas, a pontuação obtida com o protótipo *Inn Tech* resultou em **81,74**, recebendo a nota **A** para essa pontuação, a maior possível de ser atingida. Sendo assim, a usabilidade da aplicação é considerada como excelente. As respostas dos respondentes estão elencadas na tabela V.

## VI. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O objetivo deste estudo foi analisar a aplicação das técnicas de verificação e validação em um projeto do mundo real, com a intenção de incorporar essas técnicas em todas as fases do ciclo de vida de um sistema. Para atingir esse objetivo, foram conduzidas inspeções e testes de software, uma pesquisa de usabilidade, testes de caixa branca e inspeções com validação do documento de requisitos e prototipagem.

Dentre os resultados obtidos, destacamos que o protótipo alcançou a nota A, sendo classificado como excelente. Nos testes unitários realizados, a cobertura atingida foi 78%.

Nesse contexto, é evidente a importância dessa abordagem, pois ela desempenha um papel crucial na garantia de que o processo ocorra de maneira adequada, assegurando assim uma melhor utilização dos recursos de manutenção e, consequentemente, uma melhoria na qualidade geral.

Para futuros trabalhos, planejamos disponibilizar uma versão piloto para uso, permitindo que a experiência do público alvo direcione as modificações e a implementação de novas versões, de acordo com as necessidades e expectativas dos clientes e as decisões da equipe responsável.

## REFERÊNCIAS

BROOKE, J. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, v. 189, 11 1996.

Tabela V: Respostas da pesquisa de usabilidade

SUJEITO	1	2	3	4	5	6	7	8	9	10	Total
1	4	1	5	2	5	1	5	1	4	1	92,5
2	4	1	5	1	3	1	4	1	5	1	90
3	5	1	4	2	5	1	5	1	4	2	90
4	5	5	4	5	4	4	5	1	4	4	57,5
5	2	1	5	1	4	4	5	1	5	1	82,5
6	3	3	3	3	3	3	3	3	3	3	50
7	5	3	5	3	4	1	5	1	5	1	87,5
8	4	1	4	1	4	2	4	1	4	1	85
9	4	2	5	5	3	3	4	2	4	3	62,5
10	5	1	5	1	5	1	5	1	4	1	97,5
11	5	1	5	5	5	1	5	1	5	1	90
12	4	2	5	1	4	2	5	1	4	2	85
13	3	1	5	2	2	2	4	2	4	1	75
14	3	4	3	5	4	2	2	3	3	4	42,5
15	4	1	4	2	4	1	5	1	4	1	87,5
16	4	1	4	4	5	2	5	1	5	2	82,5
17	5	1	1	1	4	1	5	1	5	1	87,5
18	5	1	4	1	4	1	5	1	5	1	95
19	5	1	5	1	5	1	5	1	5	1	100
20	5	2	4	2	4	2	4	2	4	2	77,5
21	4	2	4	2	5	1	5	2	5	2	85
22	1	2	1	1	3	1	5	4	1	5	45
23	3	2	4	2	3	1	4	1	3	1	75
24	3	1	5	1	2	1	5	1	4	1	85
25	4	1	4	1	4	1	4	1	4	1	87,5
26	5	1	5	1	5	1	5	1	5	1	100
27	4	3	5	2	4	3	5	2	4	1	77,5
28	3	1	5	1	4	3	5	1	4	1	85
29	5	1	5	1	5	1	5	1	5	1	100
30	5	2	4	4	4	2	5	1	4	1	80
31	4	2	4	4	3	3	4	2	4	1	67,5
32	5	1	5	1	5	1	5	1	5	1	100
33	5	1	5	1	5	1	5	1	5	1	100
34	5	5	3	4	4	1	4	1	4	3	65
35	5	1	5	1	5	1	5	1	5	1	100
36	3	2	4	3	3	2	4	2	3	2	65
37	5	1	5	1	5	1	5	1	5	1	100
38	1	1	4	2	4	2	4	2	4	1	72,5
39	3	3	4	2	3	2	3	2	4	2	65
40	1	1	5	1	5	1	5	1	5	1	90
41	3	1	5	2	3	3	5	2	3	1	75
42	5	1	5	1	5	1	1	1	5	1	90
43	4	1	4	2	4	1	5	1	4	1	87,5
Score médio das respostas											81,74

Fonte: Autoria Própria.

br/noticias/viagens-e-turismo/2022/10/taxa-de-ocupacao-hoteleira-cresce-74-nos-oito-primeiros-meses-do-ano: :text=A Acesso em 04 Set. 2023.

VETTORAZZO, A. d. S. **Engenharia de Software**. [S.l.]: SAGAH, 2018.

FRANCISCO, T. **Protótipos: baixa, média ou alta fidelidade?** 2021.

Disponível em: (<https://medium.com/ladies-that-ux-br/prot%C3%B3tipos-baixa-m%C3%A9dia-ou-alta-fidelidade-71d897559135>).

Acesso em 17 out. 2023.

GOODMAN, L. A. Snowball sampling. **The annals of mathematical statistics**, JSTOR, p. 148–170, 1961.

HARADA, E. **O que é Figma e como você pode usufruir dessa ferramenta de design**. 2022. Disponível em: (<https://www.tecmundo.com.br/software/236320-figma-voce-usufruir-dessa-ferramenta-design.htm>).

Acesso em 18 out. 2023.

HOVEDA, U. **O QUE É UM PROTÓTIPO, QUAIS OS TIPOS, POR QUE USAR E COMO FAZER?** [S.l.]: Manaus, 2021. Disponível em: (<https://kenzie.com.br/blog/prototipo/>). Acesso em 16 out. 2023.

PRESSMAN, R. S.; PENTEADO, R. D. T. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw-Hill, 2006.

REINEHR, M. A. P. R. T. **Engenharia de Requisitos**. ed. Porto Alegre: SAGAH, 2020.

TEIXEIRA, F. **O que é o SUS (System Usability Scale) e como usá-lo em seu site**. 2015. Disponível em: (<https://brasil.uxdesign.cc/o-que-%C3%A9-o-sus-system-usability-scale-e-como-us%C3%A1-lo-em-seu-site-6d63224481c8>). Acesso em 17 out. 2023.

TREND, U. **Escala de Usabilidade do Sistema de Medição e Interpretação (SUS)**. 2023. Disponível em: (<https://uiuxtrend.com/measuring-system-usability-scale-sus/#interpretation>). Acesso em 17 out. 2023.

TURISMO, M. do. **Taxa de ocupação hoteleira cresce 74 primeiros meses do ano**. [S.l.]: São Paulo, 2022. Disponível em: <https://www.gov.br/pt->