

# Recanto do Guerreiro

**Gabriel Alejandro Figueiro Galindo<sup>1</sup>, João Gabriel Perez Monteiro<sup>2</sup>,  
Pedro Henrique Moreira Caixeta Ferreira<sup>3</sup>**

<sup>1</sup>Instituto de Ciências Exatas e Informática  
Pontifícia Universidade de Minas Gerais (PUC Minas)  
Belo Horizonte – MG – Brasil

{gabriel.fgalindo@gmail.com<sup>1</sup>, jgpgmgin31@gmail.com<sup>2</sup>}@sga.pucminas.br

{pedro.caixeta@sga.pucminas.br<sup>3</sup>}@sga.pucminas.br

**Resumo.** *O objetivo do projeto é desenvolver um software que gerencia um local que é alugado para a realização de cerimônias e outros tipos de eventos na forma de uma agenda virtual. Por meio desse aplicativo, o administrador será capaz de gerenciar o local, junto com as suas reservas e os seus custos.*

## 1. Introdução

A engenharia de *software* desempenha um papel fundamental na modernização e otimização de processos em diversos setores da sociedade. No contexto específico do Recanto Guerreiro, um local frequentemente alugado para eventos e cerimônias, a demanda crescente por seus serviços destacou a necessidade de uma gestão mais eficiente e simplificada. Atualmente, a administração depende de uma agenda em papel, cujas limitações incluem falta de controle, susceptibilidade à perda de informações e dificuldade de acesso.

Diante dessa realidade, o projeto proposto busca desenvolver um *software* que ofereça uma solução mais eficaz e conveniente do que a tradicional agenda em papel. Através dessa ferramenta digital, os usuários serão capazes de realizar todo o processo administrativo de forma mais organizada e formalizada. O aplicativo será especialmente projetado para dispositivos móveis, visando atender às necessidades do usuário a qualquer momento a partir do seu celular. Além disso, sua *interface* será intuitiva, levando em consideração a provável falta de familiaridade tecnológica de seus usuários.

As justificativas para o desenvolvimento deste trabalho são fundamentadas na importância de fornecer soluções tecnológicas acessíveis para todos os membros da sociedade, independentemente de sua idade ou conhecimento técnico. A implementação desta aplicação não apenas resolverá os desafios enfrentados pelo Recanto Guerreiro, mas também poderá ser adaptada e reaproveitada no futuro para ajudar outras pessoas que enfrentam problemas semelhantes, demonstrando assim o potencial transformador da engenharia de *software* na vida cotidiana.

## 2. Referencial Teórico

O desenvolvimento do sistema de gestão de eventos do Recanto Guerreiro foi embasado em diversas teorias e estudos que forneceram a base conceitual necessária para a sua implementação. Inicialmente, ferramentas de pesquisa foram utilizadas para buscar os estudos necessários com o objetivo de achar princípios de *design* úteis para a criação de uma *interface* intuitiva.

Em seguida, foram explorados os aspectos técnicos essenciais, a importância da Extensão Universitária, e a aplicação dos princípios da Engenharia de *Software* para compreender melhor os aspectos sociais e éticos do projeto. Por fim, trabalhos relacionados foram analisados e selecionados como fontes de referência e inspiração no desenvolvimento do projeto.

### **2.1. Princípios de Design e Ferramentas de Pesquisa**

Para a implementação da aplicação, foram escolhidos aplicativos bem estabelecidos no mercado, como o *Sympla* e o *booking.com*, que ilustram boas práticas e fornecem diretrizes valiosas, para serem usados como exemplos durante o processo de desenvolvimento do projeto. Esses aplicativos são reconhecidos por suas *interfaces* simples e intuitivas, além de focarem no gerenciamento eficiente de reservas.

Além disso, foram utilizadas ferramentas como o *Google Acadêmico* para acessar artigos que detalham os desafios enfrentados por usuários idosos ao utilizarem *softwares*, que contribuem como referências no processo de desenvolvimento. Esses artigos abordam as dificuldades que pessoas mais velhas encontram ao interagir com tecnologias digitais, fornecendo um *framework* para um bom *design* de *interface* adaptado a esse público [Zajicek 2004], além de identificar deficiências comuns em *interfaces* eletrônicas para essa faixa etária [Wirtz et al. 2009]. Também foi investigado um estudo que explora o uso de símbolos e palavras em *interfaces* e como esses elementos podem auxiliar tanto usuários idosos quanto aqueles com baixo conhecimento prévio na utilização de aplicações [Reddy et al. 2020].

### **2.2. Aspectos Técnicos**

A construção de uma *interface* acessível e intuitiva para o Recanto Guerreiro envolve a aplicação de princípios de *design* centrado no usuário. Visando atender às necessidades dos usuários inexperientes, especialmente os idosos, a *interface* do aplicativo deve ser clara, com cores de baixa luminância e contraste e fontes de fácil leitura.

A navegação deve ser intuitiva, com *feedback* imediato para as ações dos usuários e mecanismos de prevenção de erros. O objetivo é proporcionar uma experiência de usuário eficiente e satisfatória, reduzindo a curva de aprendizado e aumentando a acessibilidade e a usabilidade do sistema.

### **2.3. Extensão Universitária**

A Extensão Universitária visa ligar a universidade aos outros setores da sociedade por meio de atividades de prática extensionista, com o objetivo de facilitar o compartilhamento do conhecimento produzido no ambiente acadêmico. A Extensão é uma parte fundamental da experiência acadêmica, encorajando o protagonismo estudantil e incentivando a abordagem de desafios socioambientais importantes na atualidade [PUC-Minas 2023].

Em virtude da importância da Extensão Universitária, o projeto desenvolvido não apenas responde às necessidades operacionais da PUC, mas também visa contribuir para o resto da sociedade e para a inclusão social de pessoas inexperientes e com falta de conhecimento tecnológico. O projeto visa reafirmar a importância da Extensão no ensino superior, por meio de sua inclusão no Proex, demonstrando sua capacidade de promover a construção de uma sociedade mais justa e inclusiva.

## 2.4. A Engenharia de Software

Atualmente, a Engenharia de *Software* tem um papel vital no desenvolvimento, implementação e manutenção de sistemas essenciais na sociedade moderna. Dada a dependência cada vez maior em sistemas de *software*, é essencial aplicar com precisão os princípios da Engenharia de *Software* e as práticas de desenvolvimento ágil para criar sistemas robustos e intuitivos.

Segundo um estudo pesquisado, micro e pequenas empresas de *software* enfrentam desafios significativos durante o processo de desenvolvimento [Fernandes et al. 2012]. Portanto, a adoção de uma metodologia ágil é crucial, dado o tamanho reduzido da equipe e a necessidade de flexibilidade frente às possíveis mudanças nos requisitos. Com o intuito de abordar essas questões, foram consultados dois outros artigos que oferecem modelos e arquiteturas de referência direcionados às pequenas empresas brasileiras, facilitando o desenvolvimento do projeto [Colenci and Walimir 2011] [Varoto 2002].

## 2.5. Trabalhos relacionados

Uma revisão da literatura revelou trabalhos relevantes que abordam questões semelhantes de gestão de eventos e desenvolvimento de aplicativos com esse propósito. Um desses exemplos é um artigo que aborda o desenvolvimento de um sistema *web* voltado à gestão de eventos de um cerimonial, apresentando em sua documentação a análise de requisitos, a metodologia utilizada e seus resultados, servindo como um ótimo exemplo de como o projeto deve ser desenvolvido [Silva 2020].

Outro exemplo, inclui uma pesquisa que examinou a usabilidade de aplicativos móveis destinados ao agendamento de eventos, o que pode fornecer orientação adicional para o desenvolvimento da *interface* do usuário no projeto [Souza et al. 2018]. Esses trabalhos destacam a importância de abordar os desafios específicos enfrentados por pequenas empresas no contexto do gerenciamento de eventos e fornecem um ponto de partida sólido para o trabalho.

## 3. Metodologia

O desenvolvimento do projeto seguiu a metodologia ágil *Scrum*, que proporciona flexibilidade e adaptabilidade ao longo do processo de desenvolvimento de aplicações [Schwaber and Sutherland 2017]. A metodologia *Scrum* foi aplicada por meio da organização do trabalho em *sprints*, períodos de tempo fixos e curtos no qual o time trabalha para completar uma quantidade específica de tarefas, com a ajuda do painel de projetos do *GitHub* para a organização e acompanhamento das atividades, com a divisão de tarefas sendo realizada com base nas habilidades e preferências individuais de cada membro da equipe.

Para o desenvolvimento do *front-end*, foram selecionadas as tecnologias *web* HTML, CSS e JavaScript, enquanto o Node.js foi escolhido para o desenvolvimento do *back-end*, devido à sua alta performance e escalabilidade. O gerenciamento do banco de dados foi realizado utilizando o MySQL e a plataforma *Amazon Web Services*, também conhecida como AWS, da *Amazon* foi escolhida para hospedar o banco de dados externo.

### 3.1. Sprint 1

Na primeira *sprint*, o projeto foi iniciado com uma contextualização de seus objetivos, junto com o levantamento dos requisitos e a assinatura dos termos. Também foi realizada uma reunião entre os membros da equipe e o cliente, para adquirir uma visão mais profunda e precisa do projeto e seus objetivos.

Além disso, a documentação do *software* começou a ser desenvolvida, por meio do editor *LaTeX Overleaf*, junto com a apresentação do trabalho no *powerpoint*. O repositório *GitHub* também foi criado para armazenar todo o código e os artefatos do projeto e dar início ao planejamento da *sprint 2* por meio do *dashboard* do *GitHub*.

### 3.2. Sprint 2

Continuando o desenvolvimento do projeto, na *sprint 2* os diagramas de casos de uso, o diagrama ER e os protótipos de tela do *software* foram realizados, assim dando uma imagem visual de como as funcionalidades do aplicativo devem ser implementadas. Correções na documentação e no diagrama ER foram realizadas posteriormente, tornando-as mais precisas e mais alinhadas com o que foi solicitado.

O *front-end* e o *back-end* do aplicativo tiveram o seu desenvolvimento iniciado, com a implementação dos seguintes requisitos: a gestão das reservas, a gestão dos clientes, o estabelecimento da conexão com o banco de dados local e o processo de verificação das datas disponíveis.

### 3.3. Sprint 3

Na *sprint 3*, a documentação, o diagrama ER e os protótipos de tela foram atualizados para incluir os requisitos da gestão do local. O *back-end* e o *front-end* para a gestão do local também foram implementadas, junto com correções na integração dos outros requisitos do sistema.

Enquanto isso, como nas *sprints* anteriores, foi realizado o planejamento da *sprint* seguinte e a reorganização do *GitHub*, garantindo a continuidade do desenvolvimento do projeto. Por fim, destaca-se a criação do diagrama lógico da aplicação.

### 3.4. Sprint 4

Continuando o desenvolvimento do *front-end*, alterações na identidade visual e no *design* da aplicação foram realizadas para tornar a *interface* mais atraente. Isso também inclui mudanças na formatação do texto e na paleta de cores utilizada pelo sistema.

Ajustes também foram realizados para garantir uma experiência de usuário otimizada em dispositivos móveis, independentemente do tamanho da tela. Por meio da adaptação automática da tela para as dimensões dos aparelhos móveis, os usuários do sistema terão acesso a um sistema com melhor visualização e usabilidade.

A respeito dos testes da aplicação, foram realizados testes de integração entre as partes do *front-end*, *back-end* e banco de dados. Os testes de integração revelaram que todas as partes estão devidamente conectadas e se comunicam sem problemas. Além dos testes de integração, também foram realizados testes para verificar se cada botão executa o comando ao qual foi designado e se o sistema exibe ao usuário corretamente as informações salvas no banco de dados.

### 3.5. Sprint 5

Concluindo o desenvolvimento do trabalho, o *software* foi conectado ao banco de dados externo, correções no *front-end* para criar uma uniformidade entre as páginas foram realizadas, o relatório da Proex foi preenchido e a documentação do sistema foi atualizada junto com o *powerpoint* da apresentação do trabalho.

O *GitHub* foi reorganizado, com os arquivos *README* e *CITATION* sendo preenchidos e a pasta "Artefatos" sendo reorganizadas em subpastas. Além disso, o vídeo de demonstração do *software* foi gravado e postado e os resultados finais do projeto foram registrados na documentação.

## 4. Resultados

A aplicação implementada forneceu uma série de benefícios e melhorias em relação ao sistema anterior. Com a implementação do *software*, a necessidade de folhear manualmente a agenda foi eliminada, e as operações passaram a ser realizadas digitalmente, reduzindo a possibilidade de erros humanos e a necessidade de escrever ou apagar dados manualmente.

Além disso, a digitalização do sistema facilitou o acesso às informações e possibilitou uma melhor organização e planejamento, uma vez que os dados podem ser rapidamente consultados e atualizados. Outro benefício significativo foi o aumento na segurança dos dados, já que o armazenamento digital reduz o risco de perda ou dano de informações importantes que poderiam ocorrer com a agenda física. Como resultado dessas mudanças, os processos de agendamento e administração do local se tornaram mais ágeis e simples, resultando em uma necessidade menor de tempo e esforço por parte do usuário, junto com um aumento geral na sua produtividade.

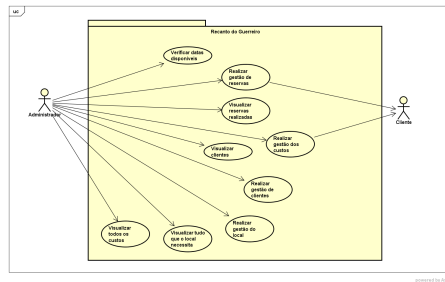
### 4.1. Modelagem

#### 4.1.1. Diagramas

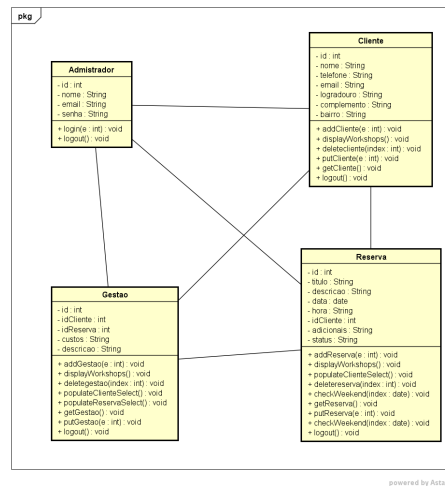
Durante o projeto, foram criados um total de 4 diagramas: o diagrama de Caso de Uso, o diagrama de Classe, o diagrama de Entidade e Relacionamento e o diagrama Lógico. Eles foram criados para demonstrar as entidades e as funcionalidades da aplicação antes da realização do processo de implementação, servindo um papel fundamental para a implementação e a realização do *back-end* da aplicação.

O diagrama ER, o diagrama de classe e o diagrama lógico demonstram as classes que foram implementados no sistema, junto com os seus atributos, suas chaves primárias e como elas se relacionam com as outras classes da aplicação. Enquanto isso, o diagrama de caso de uso foca mais nas operações que o administrador pode realizar no *software*, como verificar as datas disponíveis, visualizar a lista de clientes e outros processos administrativos. As imagens abaixo (section 4.1.1) mostram esses diagramas, ilustrando os conceitos discutidos.

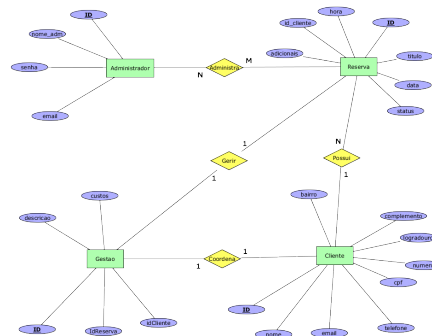
[Figura 1. Essa figura apresenta o Diagrama Caso de Uso]



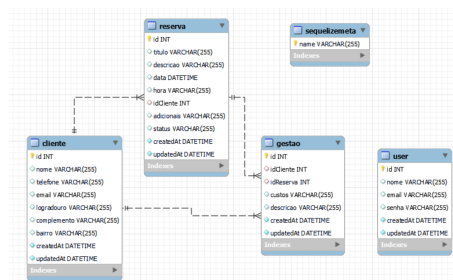
[Figura 2. Essa figura apresenta o Diagrama de Classes]



[Figura 3. Essa figura apresenta o Diagrama ER]



[Figura 4. Essa figura apresenta o Diagrama Lógico]



### 4.1.2. Requisitos

Depois de entrevistar o cliente, foram elaboradas listas de requisitos para definir o que será incluído no *software* e como ele será desenvolvido. Foram feitas duas listas: uma para os requisitos funcionais (1(a)) e outra para os requisitos não funcionais do sistema (1(b)). A lista de requisitos funcionais especifica as funcionalidades da aplicação e as ações que o administrador poderá realizar. Já a lista de requisitos não funcionais define critérios de segurança e usabilidade da aplicação. Após uma avaliação final do *software*, por meio de testes, concluiu-se que a aplicação atendeu a todos os requisitos propostos no início do projeto, possuindo todas as funcionalidades requisitadas e uma *interface* adequada.

Número de Ordem	Requisito	Descrição	Prioridade
[Número sequencial identificador do requisito.]	[Nome do requisito.]	[Descrição resumida do requisito.]	[Alta, média ou baixa.]
RF001	O administrador poderá verificar as datas disponíveis.	O administrador pode ver um calendário onde ele pode ver quais datas estão disponíveis.	Alta
RF002	O administrador poderá realizar a gestão das reservas.	O administrador pode gerenciar as reservas solicitadas - podendo marcá-las ou removê- las do calendário.	Alta
RF003	O administrador poderá visualizar as reservas realizadas.	O administrador poderá acessar um relatório com todas as reservas realizadas (incluindo a data e o nome de quem reservou).	Média
RF004	O administrador poderá realizar a gestão do local.	O administrador pode gerenciar o local reservado (podendo ver as suas condições e se ele precisa de manutenção).	Alta
RF005	O administrador poderá visualizar tudo que o local necessita.	O administrador poderá ver relatório de materiais necessários para realizar o evento para o qual ele foi reservado.	Média
RF006	O administrador poderá realizar a gestão dos custos.	O administrador poderá gerenciar os custos das reservas - podendo definir os custos com base nos materiais necessários e período de tempo da reserva.	Alta
RF007	O administrador poderá visualizar todos os custos.	O administrador terá acesso a um relatório onde poderá ver todos os custos de cada reserva.	Média
RF008	O administrador poderá realizar a gestão dos clientes.	O administrador poderá cadastrar os clientes.	Alta
RF009	O administrador poderá visualizar todos os clientes que já passaram pela empresa.	O administrador terá acesso a um relatório com todos os clientes, junto com as suas informações pessoais e reservas.	Média

(a) Requisitos Funcionais

Número de Ordem	Requisito	Descrição	Prioridade
[Número sequencial identificador do requisito.]	[Nome do requisito.]	[Descrição resumida do requisito.]	[Alta, média ou baixa.]
RFNF001	O sistema deve conter botões grandes para fácil visualização.	Requisito que irá garantir que os botões sejam facilmente visíveis pelo usuário. Melhorando a usabilidade e acessibilidade.	Baixa
RFNF002	O sistema deve conter letras de fácil leitura com fontes acima de 16.	Requisito que irá melhorar o tamanho das letras para melhor entendimento e visibilidade.	Baixa
RFNF003	O sistema deve ser responsivo em site em formato web e mobile para aparelhos a partir de 2018.	Requisito que trará melhor usabilidade e acessibilidade no sistema.	Alta
RFNF004	O sistema deverá ter uma autenticação para controle e segurança para o administrador.	Requisito com o intuito de realizar a segurança do sistema.	Alta

(b) Requisitos Não Funcionais

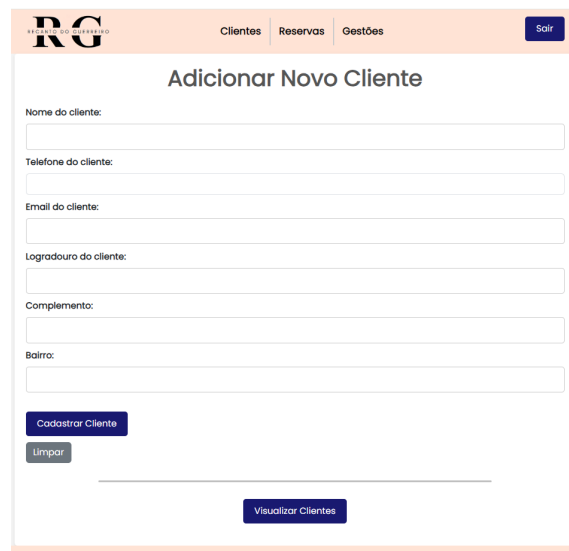
## 4.2. Telas do sistema

O sistema desenvolvido apresentou um total de 7 páginas, sendo elas: index.html, crudReserva.html, crudGestao.html, crudCliente.html, attReserva.html, attGestao.html e att-Cliente.html. A página index é responsável pelo *login* do usuário, enquanto que as outras são responsáveis pela gestão das entidades do sistema e a realização das operações *CRUD*.



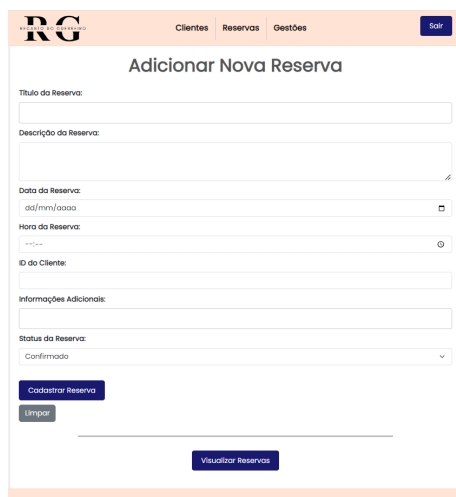
A tela de login apresenta o cabeçalho "Recanto do Guerreiro" e "Fazer Login". Abaixo, há campos para "Email" e "Senha" com ícones correspondentes. Um botão "Entrar" em azul está na base.

(c) index



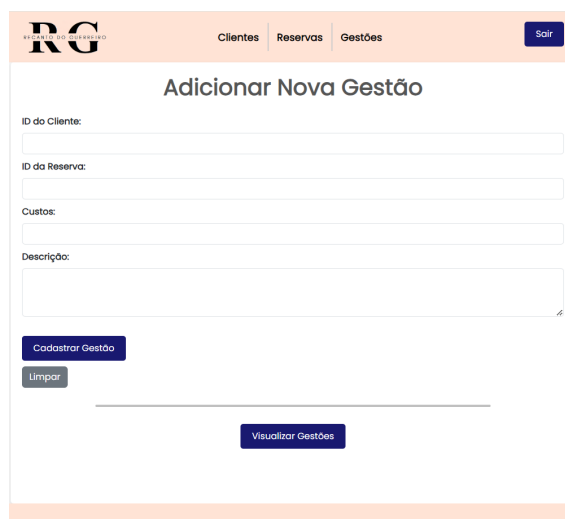
A tela "Adicionar Novo Cliente" possui campos para: Nome do cliente, Telefone do cliente, Email do cliente, Logradouro do cliente, Complemento e Bairro. Botões "Cadastrar Cliente" (azul) e "Limpar" (cinza) estão à esquerda. Um botão "Visualizar Clientes" (azul) está na base.

(d) crudCliente



A tela "Adicionar Nova Reserva" contém campos para: Título da Reserva, Descrição da Reserva, Data da Reserva (com máscara dd/mm/aaaa), Hora da Reserva (com relógio), ID do Cliente e Informações Adicionais. Um menu suspenso para "Status da Reserva" (com "Confirmado" selecionado) está na base. Botões "Cadastrar Reserva" (azul) e "Limpar" (cinza) estão à esquerda. Um botão "Visualizar Reservas" (azul) está na base.

(e) crudReserva



A tela "Adicionar Nova Gestão" possui campos para: ID do Cliente, ID da Reserva, Custos e Descrição. Botões "Cadastrar Gestão" (azul) e "Limpar" (cinza) estão à esquerda. Um botão "Visualizar Gestões" (azul) está na base.

(f) crudGestao

Devido ao fato que a aplicação foi desenvolvida com a intenção de ser utilizada por apenas um usuário, a página de cadastro não foi implementada, já que sua existência foi considerada desnecessária. Por causa disso, o usuário acessa o sistema a partir da tela de *login* utilizando uma conta criada previamente pelos desenvolvedores. Após realizar o *login* com sucesso, o usuário é redirecionado para a página de gerenciamento de clientes (crudCliente), onde poderá acessar as demais funcionalidades do sistema. Acima,



são apresentadas as imagens das principais telas da aplicação (1(c), 1(d), 1(e) e 1(f)), ilustrando o fluxo de navegação do usuário.

### 4.3. Protótipos


Antes do desenvolvimento do *software*, protótipos das telas foram realizados para ilustrar o resultado final desejado da aplicação e guiar o seu processo de desenvolvimento, principalmente com relação ao *front-end* e ao *design* do *software*.


Página Nova Reserva

**Voltar**

Título \_\_\_\_\_

Descrição \_\_\_\_\_

Data: DD/MM/AAAA 

Hora: 00:00 até 00:00 

[Selecionar cliente](#)

[Cadastrar cliente](#)

Informações Adicionais

**Salvar**

(g) Protótipo 1

**Reservas**


Lista de reservas


Título	Descrição	Data	Hora	Cliente	Adicionais	Status	Ações
--------	-----------	------	------	---------	------------	--------	-------

Adicionar Nova Reserva

Título da Reserva: \_\_\_\_\_


Descrição da Reserva: \_\_\_\_\_

Data da Reserva: da/mm/aaaa 

Hora da Reserva: --:-- 

ID do Cliente: \_\_\_\_\_

Informações Adicionais: \_\_\_\_\_

Status da Reserva: Ativo 

**Cadastrar Reserva**

**Limpar**


(h) Protótipo 3


**RG** **Reservas** **Gestões** **Sair**

**Adicionar Nova Reserva**

Título da Reserva: \_\_\_\_\_


Descrição da Reserva: \_\_\_\_\_

Data da Reserva: da/mm/aaaa 

Hora da Reserva: --:-- 

ID do Cliente: \_\_\_\_\_

Informações Adicionais: \_\_\_\_\_

Status da Reserva: Confirmado 

**Cadastrar Reserva**

**Limpar**

**Visualizar Reservas**

(i) Resultado Final - Tela de Reserva

Como pode ser visto nas imagens (1(g), 1(h) e 1(i)), apesar de existir uma diferença significativa entre o primeiro protótipo e a aplicação desenvolvida, essa diferença é quase inexistente entre o último protótipo e o resultado final do projeto, demonstrando a evolução do *front-end* e do sistema.

#### 4.4. Testes

O processo de teste do *software* foi realizado por todos os integrantes da equipe, em um total de 5 máquinas, por meio da ferramenta *Postman* e utilizando o *localhost* e o *AWS* da PUC-Minas. Todas as funcionalidades do sistema foram testadas, incluindo o processo de *login*, *logout* e o *CRUD* de todos os objetos da aplicação.

##### 4.4.1. Login e Logout

O botão da página de *login* funciona sem nenhuma complicação, sendo testada por meio de uma conta de usuário inserida previamente no sistema. Em conjunto, o botão de *logout* do sistema foi testado e validado, funcionando corretamente em todas as páginas em que ele está disponível.

##### 4.4.2. Create

Depois do processo de *login* e *logout*, a operação *create* foi testada para verificar se o usuário é capaz de criar novas entidades na aplicação. Caso a operação seja realizada com sucesso, uma mensagem de confirmação é exibida na tela do usuário. A imagem 4.4.2 abaixo demonstra a mensagem de confirmação sendo exibida, confirmando que a operação *create* está funcionando corretamente.

A imagem mostra a interface de usuário para adicionar um novo cliente. No topo, há uma barra de navegação com o logotipo 'RG' e links para 'Clientes', 'Reservas' e 'Gestões', além de um botão 'Sair'. O título principal da seção é 'Adicionar Novo Cliente'. Abaixo, há campos de formulário para: 'Nome do cliente:' (preenchido com 'Alberto'), 'Telefone do cliente:' (preenchido com '3198765432'), 'Email do cliente:' (preenchido com 'alberto@mail.com'), 'Logradouro do cliente:' (preenchido com 'rua Sergipe'), 'Complemento:' (preenchido com 'apto 502') e 'Bairro:' (preenchido com 'funcionarios'). Há dois botões na base dos campos: 'Cadastrar Cliente' e 'Limpar'. Uma mensagem de confirmação branca com o texto 'Cliente cadastrado com sucesso!' e um ícone de 'X' está sobreposta no centro da tela. Na parte inferior, há um botão 'Visualizar Clientes'.

### 4.4.3. Read

Após realizar o processo de criação de entidades do *software*, foi verificado se os objetos criados aparecem na lista da página como requisitado. Como pode ser visto nas imagens abaixo (section 4.4.3), os objetos adicionados são exibidos corretamente.

Minimizar Lista

Nome	Telefone	Email	Bairro	Ações
1	1	1	1	 
2	2	2	2	 
3	3	3	3	 
Alberto	3198765432	alberto@gmail.com	funcionarios	 

Visualizar Reservas

Título	Data	Hora	Status	Ações
aniversario	18/05/2024	14:30	Ativo	 
b	01/06/2024	15:02	Ativo	 
c	26/05/2024	16:45	Ativo	 
d	01/06/2024	16:47	Ativo	 
e	08/06/2024	17:17	Ativo	 
aniversarioa	01/06/2024	17:56	Ativo	 
f	15/06/2024	17:45	l	 
g	09/06/2024	17:48	0	 

Minimizar Lista

Cliente	ID da Reserva	Custos	Ações
1	aniversario	1	 
1	b	1	 
Alberto	aniversario	1700	 
1	b	3	 

### 4.4.4. Update

Além de criar entidades, o usuário também deve ser capaz de editá-las. Por causa disso, páginas de edição das entidades criadas foram adicionadas ao sistema e utilizadas para testar e validar o processo de *update* dos objetos. A figura 4.4.4 abaixo demonstra o teste sendo realizado, com uma entidade de gestão tendo os seus dados alterados.

Atualizar Gestão

ID da Gestão:

53

ID do Cliente:

145

ID da Reserva:

1

Custos:

1450

Descrição:

asdasdasdasd

Atualizar Gestão

#### 4.4.5. Delete

Por último, foi testado se o usuário é capaz de deletar entidades da aplicação, removendo-as também de suas respectivas listas no sistema. Como em todas as funcionalidades anteriores, a operação *delete* foi testada e validada com sucesso.

**Link do vídeo:**<https://youtu.be/fW2hQt6PCeU?si=p-WrZBpe9Xiz0wf8>

**Link do repositório:**<https://github.com/ICEI-PUC-Minas-PPLES-TI/plf-es-2024-1-ti4-0648100-recanto-do-guerreiro>

**Link da apresentação:**<https://github.com/ICEI-PUC-Minas-PPLES-TI/plf-es-2024-1-ti4-0648100-recanto-do-guerreiro/blob/master/Divulgacao/Apresentacao/ApresentacaoTIS4-Final.pdf>

### 5. Conclusões e trabalhos futuros

O desenvolvimento e a implementação do *software* de gestão de eventos para o Recanto Guerreiro evidenciaram a eficácia da tecnologia na modernização e otimização dos processos administrativos, que antes dependiam de métodos manuais. A transição de uma agenda em papel para um sistema digital resultou em uma organização mais eficiente e em uma gestão aprimorada dos eventos, reduzindo significativamente os erros e o tempo gasto em tarefas operacionais.

A aplicação também contribuiu para a inclusão digital de usuários com pouca familiaridade com a tecnologia, ao oferecer uma ferramenta acessível e intuitiva que os incentivou a utilizar e se adaptar a novos recursos tecnológicos. O *software* também destacou a importância da Extensão Universitária como uma ponte entre o conhecimento acadêmico e a solução de problemas reais na sociedade, exemplificando como a colaboração entre a academia e a comunidade pode gerar soluções eficazes e promover a inclusão tecnológica.

Embora o projeto tenha sido desenvolvido com sucesso, foram identificadas várias áreas que necessitam de melhorias futuras. A aplicação ainda precisa de um sistema de notificações, a inability de criar múltiplas reservas no mesmo dia e a geração de relatórios avançados, funcionalidades que enriqueceriam a experiência do usuário.

Para aprimorar o sistema, é essencial coletar e analisar o *feedback* dos usuários, a fim de identificar possíveis pontos de melhoria que podem orientar futuras atualizações. Além disso, realizar um estudo abrangente para avaliar a eficácia do *software* na gestão de eventos, na satisfação do usuário e na inclusão digital contribuirá para o aprofundamento do conhecimento e das habilidades da equipe, resultando em produtos futuros de maior qualidade.

## Referências

- [Colenci and Walmir 2011] Colenci, A. and Walmir, E. (2011). A referencial model for small companies of development software. *IEEE Latin America Transactions*, 9(1):823–829.
- [Fernandes et al. 2012] Fernandes, D. B., Tait, T. F. C., and Bruzaroosco, D. C. (2012). Uma contribuição para o processo de desenvolvimento de software em micro e pequenas empresas desenvolvedoras de software. In *Anais do VIII Simpósio Brasileiro de Sistemas de Informação*, pages 145–156. SBC.
- [PUC-Minas 2023] PUC-Minas, E. (2023). Vídeo institucional 2023. [https://youtu.be/aONirAMbuws?si=CnYPz\\_ufxi0C9Y6z](https://youtu.be/aONirAMbuws?si=CnYPz_ufxi0C9Y6z).
- [Reddy et al. 2020] Reddy, G. R., Blackler, A., Popovic, V., Thompson, M. H., and Mahar, D. (2020). The effects of redundancy in user-interface design on older users. *International Journal of Human-Computer Studies*, 137:102385.
- [Schwaber and Sutherland 2017] Schwaber, K. and Sutherland, J. (2017). Guia do scrum: Um guia definitivo para o scrum: As regras do jogo. *Available (in.)*, page 39.
- [Silva 2020] Silva, L. E. D. (2020). Desenvolvimento de um sistema web para a gestão de eventos de um cerimonial.
- [Souza et al. 2018] Souza, R. P., Silva, T. R. B., Oliveira, J. M., and de Oliveira, G. A. A. (2018). Avaliação conjunta de facilidade de uso em aplicativos móveis para apoio à participação em eventos.
- [Varoto 2002] Varoto, A. C. (2002). *Visões em arquitetura de software*. PhD thesis, Universidade de São Paulo.
- [Wirtz et al. 2009] Wirtz, S., Jakobs, E.-M., and Ziefle, M. (2009). Age-specific usability issues of software interfaces. In *Proceedings of the IEA*, volume 17.
- [Zajicek 2004] Zajicek, M. (2004). Successful and available: interface design exemplars for older users. *Interacting with computers*, 16(3):411–430.