

Sistema de Gestão de Clientes e Funcionários

1st Eduardo Brício Diniz
Belo Horizonte, Brasil
eduardobricio10@gmail.com

2nd Gabriel Ramos Fontes
Belo Horizonte, Brasil
gabrielramosfontes@gmail.com

3rd João Vitor Batista Camargos
Belo Horizonte, Brasil
joaovitorbatista12337@gmail.com

Abstract—Este documento descreve as funcionalidades de um sistema de gestão de clientes e funcionários, incluindo a geração de UUIDs, uso de bancos de dados Excel, envio de e-mails e interface de menu para interação com usuários. Este sistema é projetado para melhorar a eficiência operacional e facilitar a comunicação entre diferentes partes interessadas dentro da organização.

Index Terms—gestão de clientes, funcionários, UUID, banco de dados, envio de e-mails, comunicação organizacional, eficiência operacional

I. INTRODUÇÃO

O sistema de gestão de clientes e funcionários apresentado aqui é um sistema abrangente projetado para facilitar a administração de clientes e funcionários em uma organização. Este sistema inclui funcionalidades para gerenciar registros de clientes e funcionários, enviar faturas e recibos, atualizar status de clientes para VIP, manter bancos de dados globais e pessoais, e permitir a interação através de menus intuitivos. A implementação deste sistema pode transformar a maneira como as organizações lidam com as informações dos clientes e funcionários, resultando em maior eficiência e melhor experiência do usuário.

A gestão de informações de clientes e funcionários é um desafio constante para muitas organizações. A eficiência no tratamento desses dados é crucial para garantir a qualidade do atendimento ao cliente e a gestão eficaz dos recursos humanos. Neste contexto, um sistema integrado que automatiza e centraliza essas funções é extremamente vantajoso. Este sistema visa resolver esses desafios por meio da automação de processos e da centralização de dados, proporcionando uma solução robusta e eficaz para a gestão de clientes e funcionários.

A utilização de tecnologias como UUIDs para identificação única e bancos de dados baseados em Excel oferece uma abordagem prática e acessível para a gestão de dados. Esses componentes permitem que o sistema seja facilmente implementado e mantido, mesmo em organizações que não possuem uma infraestrutura de TI avançada. Além disso, a capacidade de enviar e-mails automatizados melhora significativamente a comunicação com os clientes, garantindo que as notificações importantes sejam entregues de maneira oportuna e precisa.

Por fim, o sistema inclui uma interface de menu intuitiva, que permite aos usuários navegar e acessar as diversas funcionalidades de maneira simples e eficiente. Esta usabilidade é essencial para garantir que todos os usuários, independentemente de seu nível de habilidade técnica, possam interagir com o sistema sem dificuldades. A combinação dessas

funcionalidades torna este sistema uma ferramenta poderosa para qualquer organização que busca melhorar sua gestão de clientes e funcionários.

II. METODOLOGIA

A. Diagrama UML

O diagrama UML do sistema ilustra a estrutura das classes e suas interações, proporcionando uma visão clara da arquitetura do sistema. Este diagrama é essencial para entender a organização do código e as relações entre as diferentes classes, facilitando a manutenção e a expansão futura do sistema.

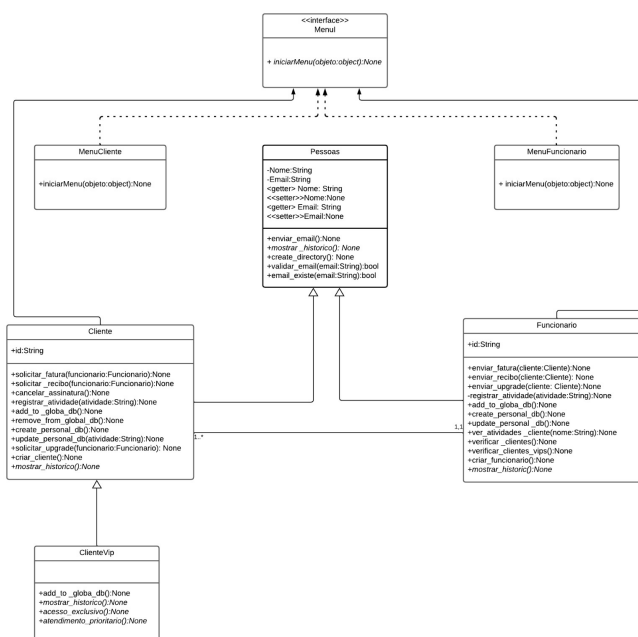


Fig. 1. Diagrama UML do Sistema

B. Uso da Biblioteca pandas

A biblioteca *pandas* foi fundamental para a manipulação e gestão dos dados de clientes e funcionários neste sistema. Ela fornece estruturas de dados e funções de análise de dados de alto desempenho para Python, facilitando a leitura, escrita e manipulação de arquivos Excel, que foram utilizados como banco de dados no projeto.

A leitura de arquivos Excel foi realizada utilizando a função `read_excel()`, que permitiu importar os dados diretamente

para DataFrames. Isso garantiu que os dados fossem facilmente acessíveis e manipuláveis.

Para salvar as atualizações nos dados, utilizamos a função `to_excel()`, que exporta DataFrames para arquivos Excel. Esta abordagem assegura que todas as alterações sejam persistidas de maneira organizada. Além disso, funções como `concat()` e `query()` foram utilizadas para manipulação de dados e realização de consultas, permitindo adicionar novas atividades e buscar registros específicos com eficiência.

A capacidade de criar, ler e atualizar DataFrames com *pandas* proporcionou uma base sólida para a gestão de dados no sistema, facilitando a implementação de funcionalidades como registro de atividades, envio de e-mails e gerenciamento de banco de dados pessoais e globais. A flexibilidade e a robustez do *pandas* tornaram possível a criação de um sistema eficiente e escalável, adequado para atender às necessidades de pequenas e médias empresas.

C. Funcionalidades do Cliente

As funcionalidades do cliente no sistema incluem:

- **Solicitar Fatura:** O cliente pode solicitar uma fatura a um funcionário, o que registra a atividade e aciona o envio da fatura por e-mail. Este processo garante que todas as solicitações de fatura sejam documentadas e tratadas de maneira eficiente.
- **Solicitar Recibo:** O cliente pode solicitar um recibo a um funcionário, registrando a atividade e acionando o envio do recibo por e-mail. A funcionalidade de recibo é vital para manter registros financeiros precisos e transparentes.
- **Cancelar Assinatura:** O cliente pode solicitar o cancelamento de sua assinatura, registrando esta atividade. Isso permite que o cliente tenha controle total sobre seus serviços e possa interromper a assinatura quando desejar.
- **Solicitar Upgrade:** O cliente pode solicitar um upgrade de status para VIP. Este recurso incentiva a fidelização dos clientes, oferecendo benefícios adicionais para aqueles que optam por um upgrade.
- **Mostrar Histórico:** O cliente pode visualizar o histórico de suas atividades. Ter acesso ao histórico completo de interações e transações aumenta a transparência e permite que o cliente acompanhe todas as suas ações dentro do sistema.

D. Funcionalidades do Funcionário

As funcionalidades do funcionário no sistema incluem:

- **Criar Funcionário:** Possibilidade de criar novos funcionários. Esta funcionalidade facilita a expansão da equipe e a gestão de novos membros dentro do sistema.
- **Enviar Fatura:** Enviar faturas solicitadas por clientes. A capacidade de enviar faturas rapidamente melhora a eficiência operacional e a satisfação do cliente.
- **Enviar Recibo:** Enviar recibos solicitados por clientes. Manter um registro preciso de todas as transações financeiras é essencial para a integridade financeira da organização.

- **Enviar Upgrade:** Promover um cliente ao status de VIP. Este recurso ajuda a reconhecer e recompensar clientes valiosos, promovendo a lealdade do cliente.
- **Ver Atividades do Cliente:** Ver o histórico de atividades de um cliente específico. A capacidade de visualizar as atividades do cliente permite um atendimento mais personalizado e eficiente.
- **Verificar Clientes:** Visualizar o banco de dados de todos os clientes. Manter um banco de dados atualizado e acessível é crucial para a gestão eficaz dos clientes.
- **Verificar Clientes VIPs:** Visualizar o banco de dados de todos os clientes VIPs. Identificar e gerenciar clientes VIPs permite oferecer serviços exclusivos e diferenciados.
- **Mostrar Histórico:** Visualizar o próprio histórico de atividades. Permitir que os funcionários visualizem seu próprio histórico de atividades promove a responsabilidade e o auto-gerenciamento.

E. Sistema de IDs

O sistema de IDs é baseado na geração de UUIDs (Universally Unique Identifiers) para garantir que cada instância de Cliente e Funcionário tenha um identificador único. O UUID é gerado na criação de cada objeto, assegurando que não haja duplicidade de IDs e facilitando o rastreamento e a gestão de registros no sistema. Este método é altamente eficaz para garantir a integridade dos dados e evitar conflitos de identificação.

```
python Copiar código

cliente1 = Cliente(str(uuid.uuid4()), nome1, email1)
funcionario1 = Funcionario(str(uuid.uuid4()), nome1, email1)
```

Fig. 2. Diagrama do Sistema de IDs

F. Banco de Dados Global

O banco de dados global é implementado usando arquivos Excel para armazenar informações de clientes e funcionários. A classe Cliente e Funcionario têm métodos para adicionar, remover e atualizar registros no banco de dados global. Esta abordagem permite uma gestão centralizada e acessível de todos os dados importantes da organização.

```
python Copiar código

df = pd.read_excel('clientes.xlsx')
new_row = pd.DataFrame({'ID': [self.id], 'Nome': [self.nome], 'Email': [self.email]})
df = pd.concat([df, new_row], ignore_index=True)
df.to_excel('clientes.xlsx', index=False)
```

Fig. 3. Diagrama do Banco de Dados Global

O uso de arquivos Excel permite uma abordagem simples e eficaz para gerenciar grandes quantidades de dados sem a necessidade de um banco de dados complexo. No entanto, para empresas com volumes de dados extremamente altos, pode ser necessário considerar a migração para um sistema de banco de

dados mais robusto no futuro. A flexibilidade e a familiaridade dos arquivos Excel tornam esta solução ideal para pequenas e médias empresas. A simplicidade no uso de Excel também significa que a equipe precisa de menos treinamento para começar a usar o sistema, o que pode reduzir os custos de implementação.

G. Banco de Dados Pessoal

Cada cliente e funcionário tem seu próprio banco de dados pessoal para registrar atividades. Este banco de dados é um arquivo Excel nomeado com o nome do cliente ou funcionário. A manutenção de bancos de dados pessoais permite um rastreamento detalhado das atividades de cada indivíduo, facilitando a auditoria e a análise de desempenho. Este nível de detalhe é fundamental para garantir que todas as interações e transações sejam registradas com precisão e possam ser revisadas conforme necessário.

```
python Copiar código

df = pd.DataFrame({"Atividade": []})
df.to_excel(f'{self.nome}_atividades.xlsx', index=False)
```

Fig. 4. Diagrama do Banco de Dados Pessoal

O banco de dados pessoal oferece uma visão individualizada das atividades de cada cliente e funcionário, permitindo uma gestão mais personalizada. Além disso, a separação dos dados pessoais e globais ajuda a evitar a sobrecarga do banco de dados principal, garantindo que o sistema permaneça rápido e responsivo. A facilidade de acesso e atualização desses dados também facilita a manutenção e a integridade das informações.

H. Menu

O sistema de menu oferece uma interface para interação com os clientes e funcionários. Dependendo da escolha do usuário, diferentes funcionalidades são ativadas. O menu é implementado nas classes *MenuCliente* e *MenuFuncionario*, proporcionando uma navegação intuitiva e fácil acesso às funcionalidades principais. Um menu bem projetado é essencial para a usabilidade do sistema, permitindo que os usuários encontrem e utilizem as funções de forma rápida e eficiente.

```
python Copiar código

print('Digite 1 para solicitar uma fatura')
print('Digite 2 para solicitar um recibo')
# E assim por diante
```

Fig. 5. Diagrama do Menu do Sistema

O menu do sistema foi projetado para ser simples e direto, permitindo que os usuários, independentemente de seu nível de habilidade técnica, possam navegar e usar o sistema eficientemente. Esta simplicidade é crucial para garantir que todos os funcionários, desde os mais técnicos até os menos experientes, possam interagir com o sistema sem dificuldades. A inclusão de opções de ajuda e guias dentro do menu também pode melhorar a experiência do usuário, proporcionando suporte adicional quando necessário.

I. Envio de Emails

O envio de e-mails é realizado usando o módulo *smtp*. Os métodos 'enviar fatura', 'enviar recibo' e 'enviar upgrade' utilizam o método 'enviar email' da classe *Pessoa* para enviar e-mails específicos. Esta funcionalidade automatiza a comunicação com os clientes, garantindo que eles recebam notificações importantes de forma oportuna e eficiente.

```
python Copiar código

msg = email.message.Message()
msg['Subject'] = assunto
msg['From'] = 'jjoapoo@gmail.com'
msg['To'] = email_destino
msg.set_payload(corpo_email)

s = smtplib.SMTP('smtp.gmail.com: 587')
s.starttls()
s.login(msg['From'], password)
s.sendmail(msg['From'], [msg['To']], msg.as_string().encode('utf-8'))
```

Fig. 6. Diagrama do Sistema de Envio de Emails

A capacidade de enviar e-mails diretamente do sistema automatiza a comunicação com os clientes, garantindo que eles recebam notificações importantes de forma oportuna e eficiente. Isso não só melhora a experiência do cliente, mas também reduz a carga de trabalho manual para os funcionários. O envio automatizado de e-mails ajuda a manter uma comunicação consistente e profissional com os clientes.

Automatizar o envio de e-mails também permite que a empresa mantenha um registro de todas as comunicações enviadas, facilitando a auditoria e o monitoramento das interações com os clientes. Além disso, a personalização dos e-mails enviados pode aumentar a satisfação do cliente e a eficácia da comunicação.

J. Interface Menu

A interface *Menu* define um contrato abstrato para os menus de interação no sistema, assegurando que todas as classes derivadas implementem o método *iniciarMenu()*. Essa interface é fundamental para garantir a consistência e a padronização das operações de menu tanto para clientes quanto para funcionários.

A implementação da interface *Menu* pelas classes *MenuCliente* e *MenuFuncionario* permite a criação de menus específicos para cada tipo de usuário. O método *iniciarMenu()* na classe *MenuCliente* apresenta opções como solicitar faturas, recibos, verificar histórico e sair do menu, facilitando a interação do cliente com o sistema. Similarmente, *MenuFuncionario* oferece funcionalidades como ver atividades de clientes, acessar bancos de dados de clientes e solicitar upgrades, proporcionando um ambiente de trabalho eficiente para os funcionários.

Essa abordagem modular e padronizada assegura que a interação com o sistema seja intuitiva e eficiente. Utilizando a interface *Menu*, o sistema pode ser facilmente mantido e expandido, garantindo uma base sólida para futuras adaptações e melhorias.

III. DEMONSTRAÇÃO DE RESULTADO

Nesta seção, apresentamos os resultados obtidos com a implementação do sistema, incluindo exemplos de banco de dados e e-mails enviados. A demonstração de resultados é fundamental para ilustrar a eficácia e a funcionalidade do sistema em um ambiente real. Mostramos como o sistema pode ser usado para gerenciar efetivamente as informações dos clientes e funcionários, melhorando a eficiência e a comunicação.

A. Print do Banco de Dados

A seguir, um exemplo de como o banco de dados de clientes é armazenado e gerenciado no sistema.

ID	Nome	Email
32469a0b-0da2-46e0-afde-eaabba43c17a	João	joao@email.com
5c8daf7c-70bf-4c21-be82-7c56d25995e5	Ronaldinho	joaovitorbatista12337@gmail.com

Fig. 7. Print do Banco de Dados de Clientes

O banco de dados de clientes é organizado de maneira que permite fácil acesso e manipulação dos dados. Este print ilustra como as informações dos clientes são armazenadas e gerenciadas, proporcionando uma visão clara da estrutura do banco de dados e da eficiência do sistema. Manter um banco de dados bem organizado é crucial para a eficiência operacional e a satisfação do cliente, pois permite uma resposta rápida às solicitações e consultas.

B. Print de Email Enviado

Exemplo de um e-mail enviado pelo sistema para um cliente que solicitou uma fatura.

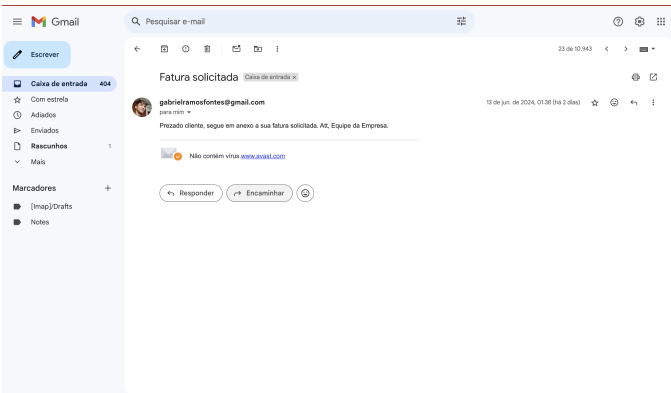


Fig. 8. Print de um Email Enviado

O envio de e-mails é uma funcionalidade crucial para a comunicação com os clientes. Este print mostra um exemplo de e-mail enviado, destacando a profissionalidade e a clareza das comunicações geradas pelo sistema. E-mails bem formatados e personalizados aumentam a satisfação do cliente e a percepção de profissionalismo da empresa. A capacidade de personalizar e automatizar e-mails melhora significativamente a eficiência do atendimento ao cliente e a consistência das comunicações.

C. Inputs e Outputs

Nesta seção, detalhamos os inputs fornecidos pelos usuários e os outputs gerados pelo sistema.

Input:

- Nome do Cliente: "Marcos"
- Email do Cliente: "marcos@example.com"
- Solicitação: "Solicitar Fatura"

Output:

- Registro da atividade "Solicitou Fatura" no banco de dados pessoal do cliente.
- Envio de e-mail de fatura para "marcos@example.com".

Estes exemplos de input e output demonstram como o sistema processa solicitações e gera respostas automáticas, garantindo que as ações dos clientes sejam tratadas de forma eficiente e documentada. A capacidade do sistema de registrar e responder a interações dos clientes em tempo real é um dos principais benefícios desta implementação. Isso permite que a empresa mantenha um alto nível de serviço ao cliente, melhorando a satisfação e a fidelidade.

IV. CONCLUSÃO

Este sistema de gestão de clientes e funcionários oferece uma solução integrada para a administração de registros, comunicação e rastreamento de atividades. A utilização de UUIDs para IDs únicos, a manutenção de bancos de dados pessoais e globais, e a implementação de menus intuitivos e envio de e-mails automatizados são algumas das funcionalidades que tornam este sistema eficaz e eficiente.

Implementar este sistema em uma organização pode melhorar significativamente a gestão de dados e a comunicação interna, resultando em operações mais suaves e um melhor atendimento ao cliente. A capacidade de gerenciar informações de clientes e funcionários de forma centralizada e automatizada permite que as organizações respondam rapidamente às necessidades dos clientes e mantenham um alto nível de serviço. Além disso, a capacidade de manter registros detalhados de atividades e interações garante que a empresa possa realizar auditorias e análises de desempenho com facilidade, promovendo uma cultura de transparência e responsabilidade.

A utilização de tecnologias simples, como arquivos Excel para bancos de dados, facilita a implementação do sistema em empresas de diversos tamanhos, sem a necessidade de grandes investimentos em infraestrutura de TI. Isso torna o sistema acessível para pequenas e médias empresas que desejam melhorar sua gestão de clientes e funcionários sem comprometer o orçamento.

Além disso, a capacidade de enviar e-mails automatizados melhora a eficiência da comunicação e garante que os clientes recebam atualizações e notificações importantes de maneira oportuna. A personalização dos e-mails enviados também contribui para uma melhor experiência do cliente, aumentando a satisfação e a fidelidade.

Em resumo, este sistema de gestão de clientes e funcionários é uma ferramenta poderosa que pode transformar a maneira

como as organizações lidam com suas informações e interações diárias. A combinação de funcionalidades abrangentes, facilidade de uso e implementação acessível faz deste sistema uma escolha ideal para qualquer organização que busca melhorar sua eficiência operacional e qualidade do atendimento ao cliente.

ACKNOWLEDGMENT

Os autores agradecem a todos que contribuíram para a realização deste trabalho. Sem a dedicação e o trabalho árduo de toda a equipe, este projeto não teria sido possível. Agradecemos também a nossa orientadora Gabriela, cujo feedback valioso ajudou a moldar e aprimorar o sistema.

REFERENCES

- [1] Google. "Send email from a printer, scanner, or app;" [Online]. Available: <https://support.google.com/mail/answer/78799>. [Accessed: Jun. 10, 2024].
- [2] Hashtag Programação, "Como Enviar E-mails com Python e SMTP," YouTube, 10-Mar-2021. [Online]. [Accessed: Jun. 10, 2024].
- [3] The pandas development team, "pandas documentation," pandas.pydata.org, [Online]. [Accessed: Jun. 10, 2024].
- [4] Python Software Foundation, "smtplib — SMTP protocol client," [Online]. Available: <https://docs.python.org/3/library/smtplib.html>. [Accessed: Jun. 10, 2024].
- [5] Python Software Foundation, "uuid — UUID objects according to RFC 4122," [Online]. Available: <https://docs.python.org/3/library/uuid.html>. [Accessed: Jun. 10, 2024].