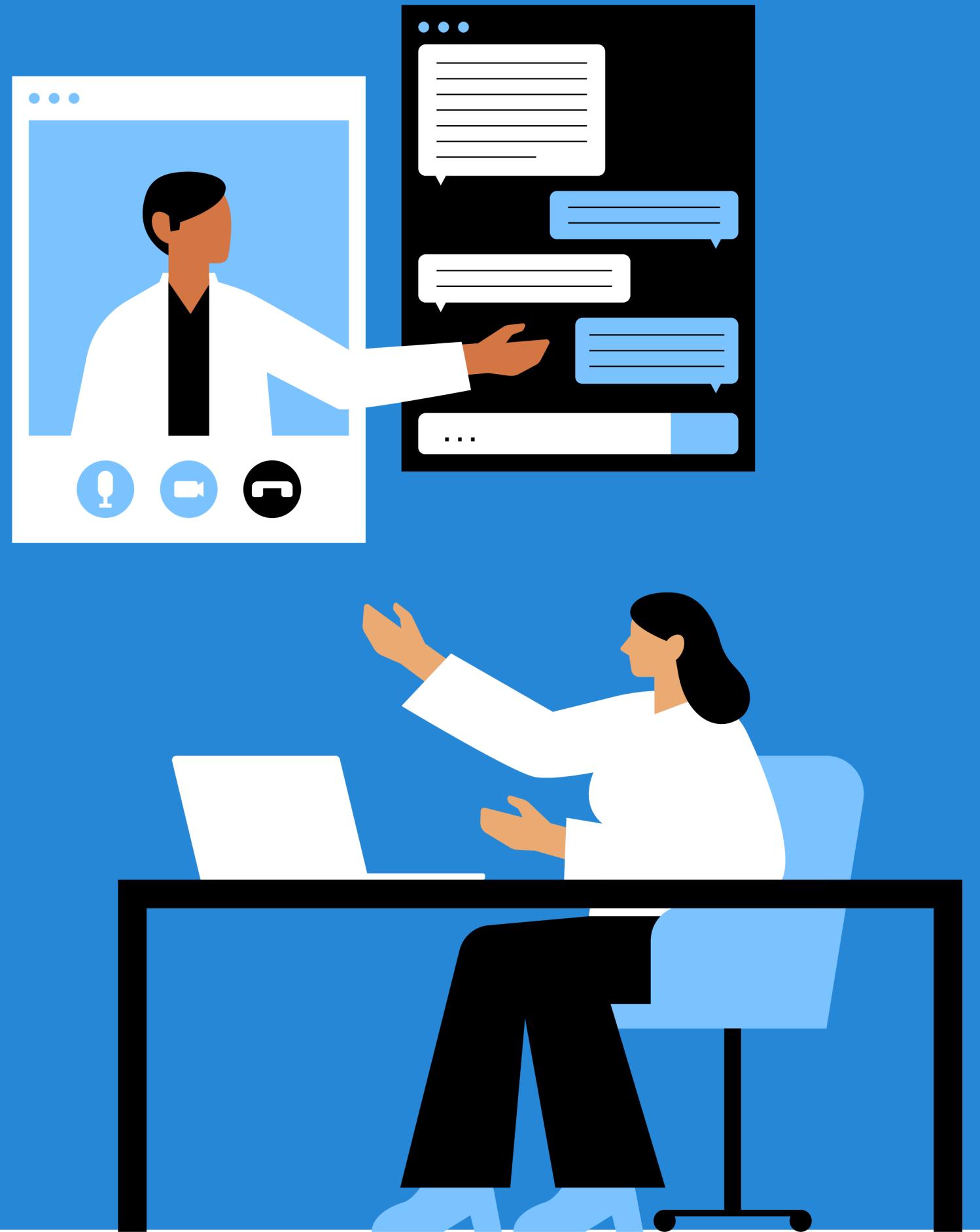


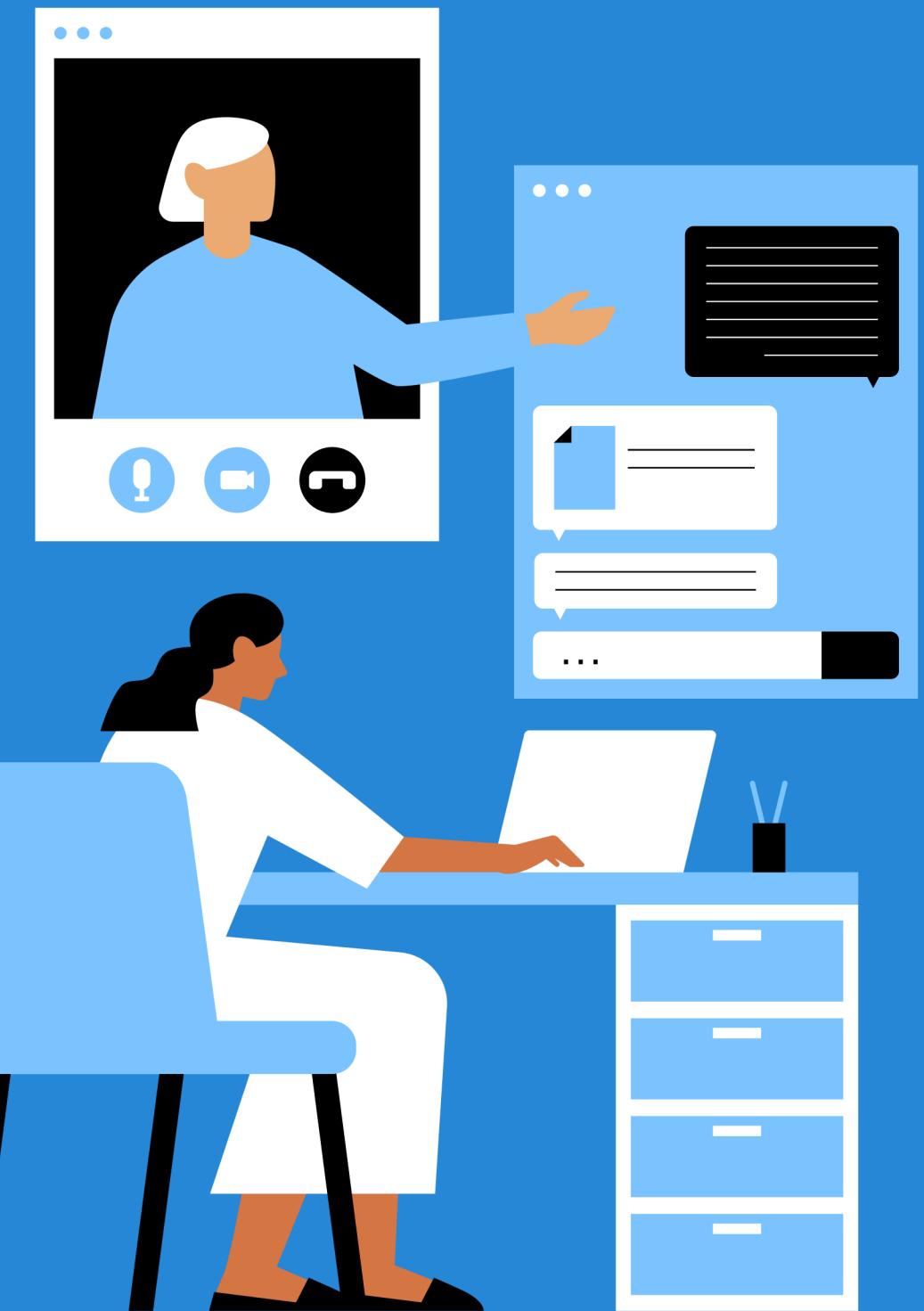
Atendimento-Fast

Grupo: Alex Alves, Almodá Luiz,
Claudheny Oliveira, Kevin Willams,
Maria Júlia,
Valdenor Curvelo



Introdução

- Desenvolvemos um sistema de agendamento online para serviços personalizados. O sistema permite o cadastro de clientes e prestadores, a configuração de horários disponíveis, agendamento de serviços, e o gerenciamento completo da agenda.



As funcionalidades principais incluem:

- Cadastro e login de usuários (clientes e prestadores);
- Cadastro e edição de serviços;
- Definição de horários disponíveis;
- Agendamento e cancelamento de compromissos;
- Visualização da agenda e histórico de agendamentos.
- Busca e visualização de perfil de clientes.

As tecnologias utilizadas incluem:

- Front-end: HTML, CSS, JavaScript;
- Back-end: Python, PHP e Django;
- Outros: JWT para autenticação, bcrypt para criptografia de senhas.



Relação com os Parâmetros da ISO/IEC 25010

- Nesta seção, o sistema de agendamento desenvolvido será analisado com base nas principais características propostas pelo modelo, buscando identificar como ele se comporta em relação a cada uma delas.
- Serão abordados critérios como funcionalidade, confiabilidade, usabilidade, eficiência de desempenho, segurança, compatibilidade, manutenibilidade e portabilidade. Para cada um, serão apresentados exemplos concretos de como o sistema atende (ou ainda não atende) a esses requisitos de qualidade, com base nas funcionalidades implementadas e nos testes realizados até o momento



Funcionalidade: Atende aos requisitos principais (cadastro, agendamento, agenda).

Melhoria: Filtros avançados na busca de clientes.

Confiabilidade: Sistema estável com tratamento de erros.

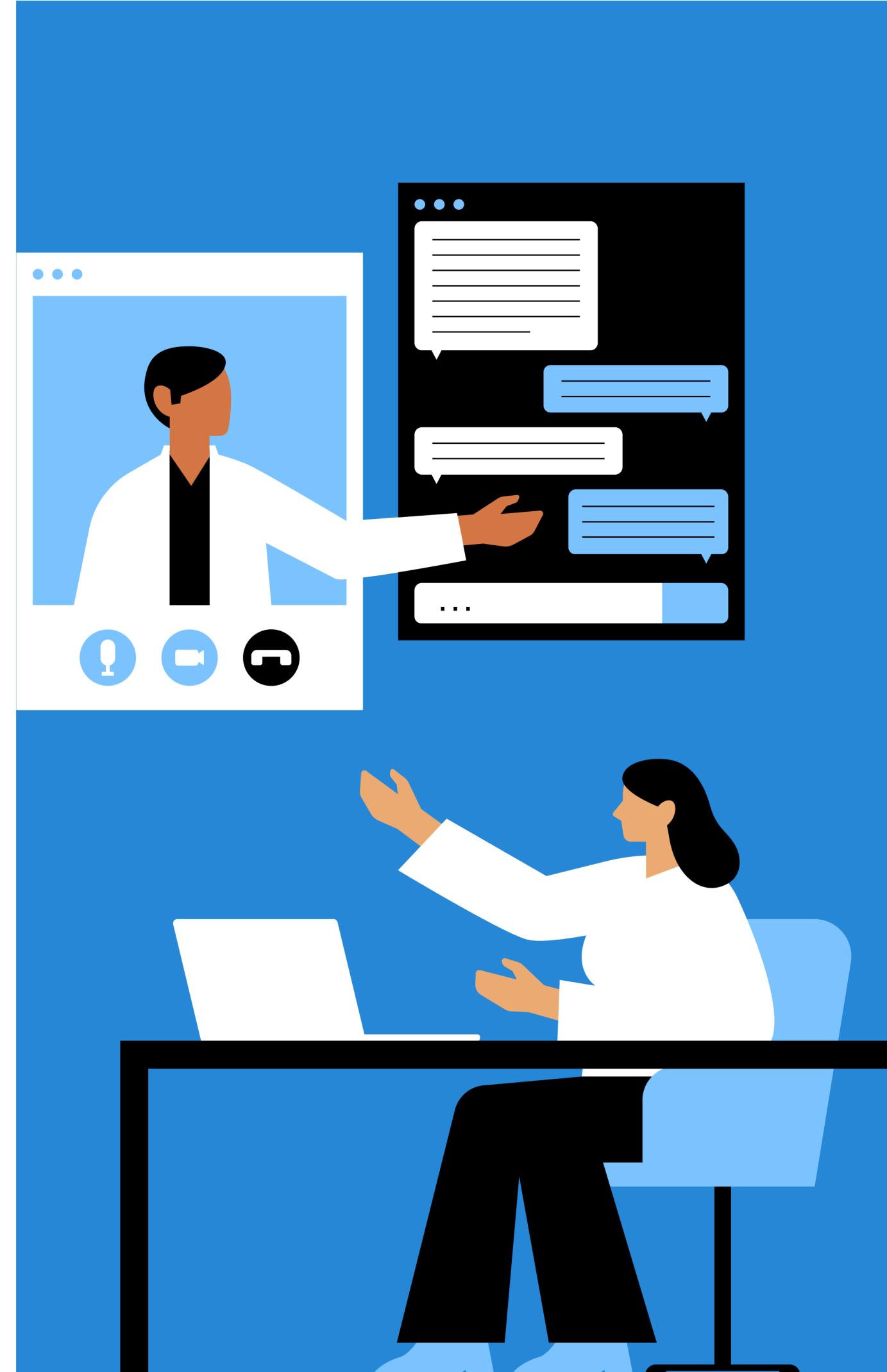
Melhoria: Falta teste de estresse para acessos simultâneos.

Usabilidade: Interface simples e intuitiva.

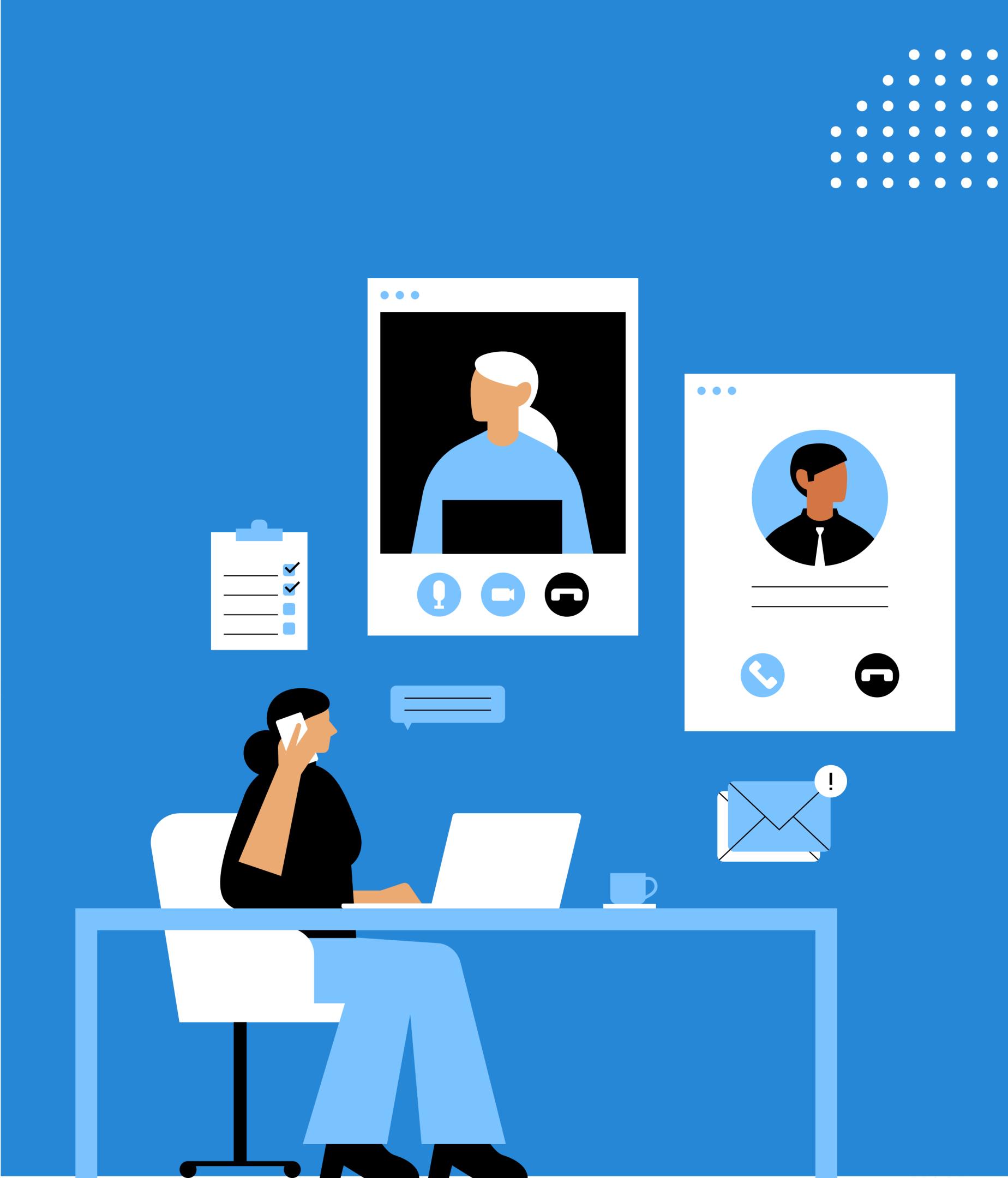
Melhoria: Planeja-se modo escuro e acessibilidade.

Eficiência: Boa performance em conexões comuns; respostas rápidas.

Melhoria: Faltam testes em redes lentas e dispositivos fracos.



- **Segurança:** Usa JWT e senhas criptografadas.
Melhoria: Falta HTTPS e autenticação multifator.
- **Compatibilidade:** Funciona bem em navegadores e dispositivos diversos.
Exemplo: Layout responsivo em smartphones.
- **Manutenibilidade:** Código modular e bem organizado.
Melhoria: Ausência de testes unitários automatizados
- **Portabilidade:** Ambiente com Docker facilita instalação e uso em outros sistemas.



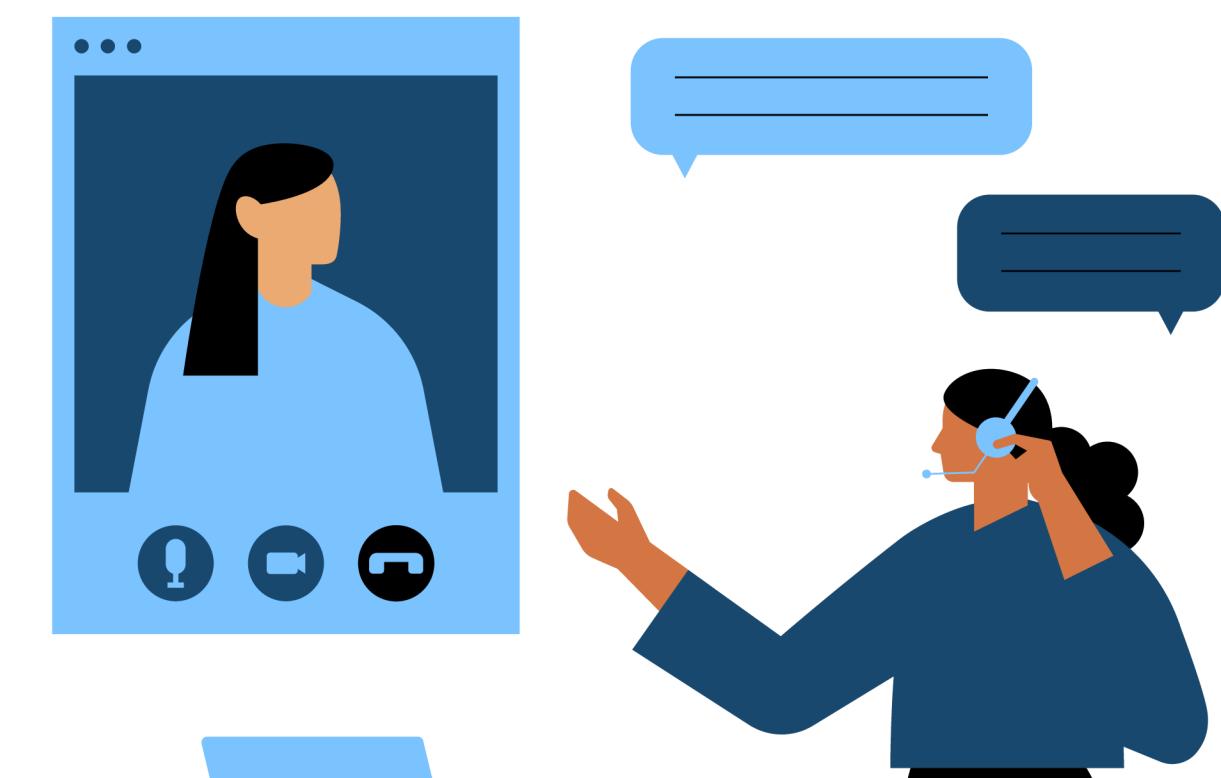
Objetivo:

Este sistema tem como objetivo facilitar o agendamento de serviços por parte de clientes, oferecendo uma plataforma simples e eficiente para marcar horários com prestadores de serviço.

A proposta é eliminar problemas comuns como agendamentos duplicados, falhas de comunicação e desorganização na gestão de horários.

Público-alvo:

O público-alvo inclui prestadores de serviço autônomos e pequenos negócios que precisam de uma solução prática para organizar suas agendas e atender melhor seus clientes.



Requisitos Funcionais

Cadastro e Autenticação

- RF01: Cadastro de usuários (clientes e prestadores) com nome, e-mail, senha, telefone e tipo de conta.
- RF02: Login com validação de e-mail e senha.

Prioridade: Essencial

Gerenciamento de Serviços

- RF03: Cadastro de serviços (nome, duração, valor, descrição).
- RF04: Edição e exclusão de serviços.

Gerenciamento de Horários

- RF05: Definição de dias e horários disponíveis.
- RF06: Bloqueio automático de horários ocupados ou fora do expediente.
- RF07: Geração de horários disponíveis com base na duração dos serviços.

Agendamentos

- RF08: Agendamento de serviços por clientes com base na agenda do prestador.
- RF09: Validação de conflitos de horário (evita agendamentos duplicados).
- RF10: Envio de confirmação de agendamento ao cliente.
- RF11: Cancelamento ou remarcação por clientes dentro de um prazo definido.

Gestão da Agenda

- RF12: Visualização da agenda por dia, semana e mês.
- RF13: Cancelamento ou remarcação de agendamentos pelo administrador.
- RF14: Exibição do histórico de agendamentos por cliente.

Gestão de Clientes

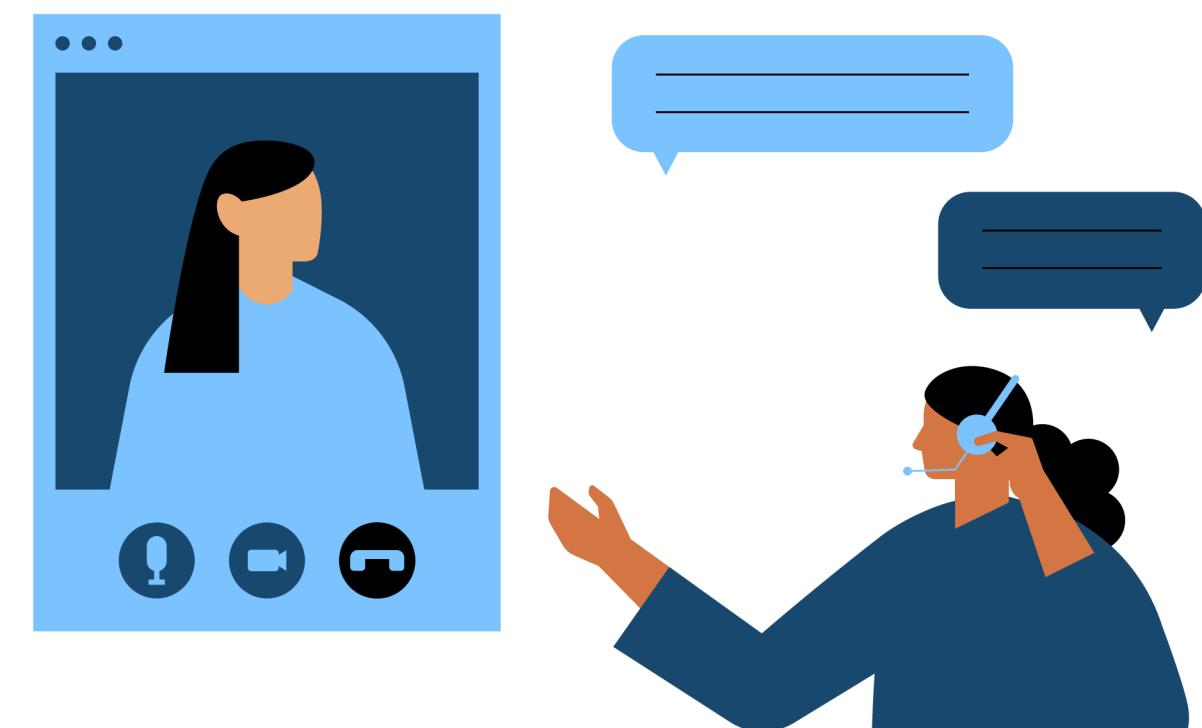
- RF15: Visualização de perfil de clientes com dados e histórico, busca por nome ou telefone.

Classificação por Prioridade

- **Essencial:** RF01–RF03, RF05–RF10
- **Importante:** RF04, RF11–RF13, RF15
- **Desejável:** RF14

Requisitos Não Funcionais:

- **RNF01 - Acesso via Navegador:** Sistema acessível por navegadores modernos, com layout responsivo para desktop e mobile. (Essencial)
- **RNF02 - Banco de Dados Relacional:** Uso de banco de dados relacional para armazenar informações. (Essencial)
- **RNF03 - Tempo de Resposta:** Funcionalidades devem responder em até 2 segundos em uso normal. (Importante)
- **RNF04 - Segurança de Acesso:** Autenticação segura com senhas criptografadas. (Essencial)



Metodologia

- A equipe utilizará a metodologia Incremental + Kanban, com o sistema sendo desenvolvido em módulos independentes e entregues em versões incrementais. As tarefas serão gerenciadas em um quadro Kanban digital, priorizando funcionalidades essenciais com base no valor para o usuário e viabilidade técnica.

Desafios

- Sincronização em tempo real dos horários disponíveis;
- Suporte a diferentes fusos horários;
- Interface amigável para desktop e mobile;
- Gestão eficiente de cancelamentos e remarcações.

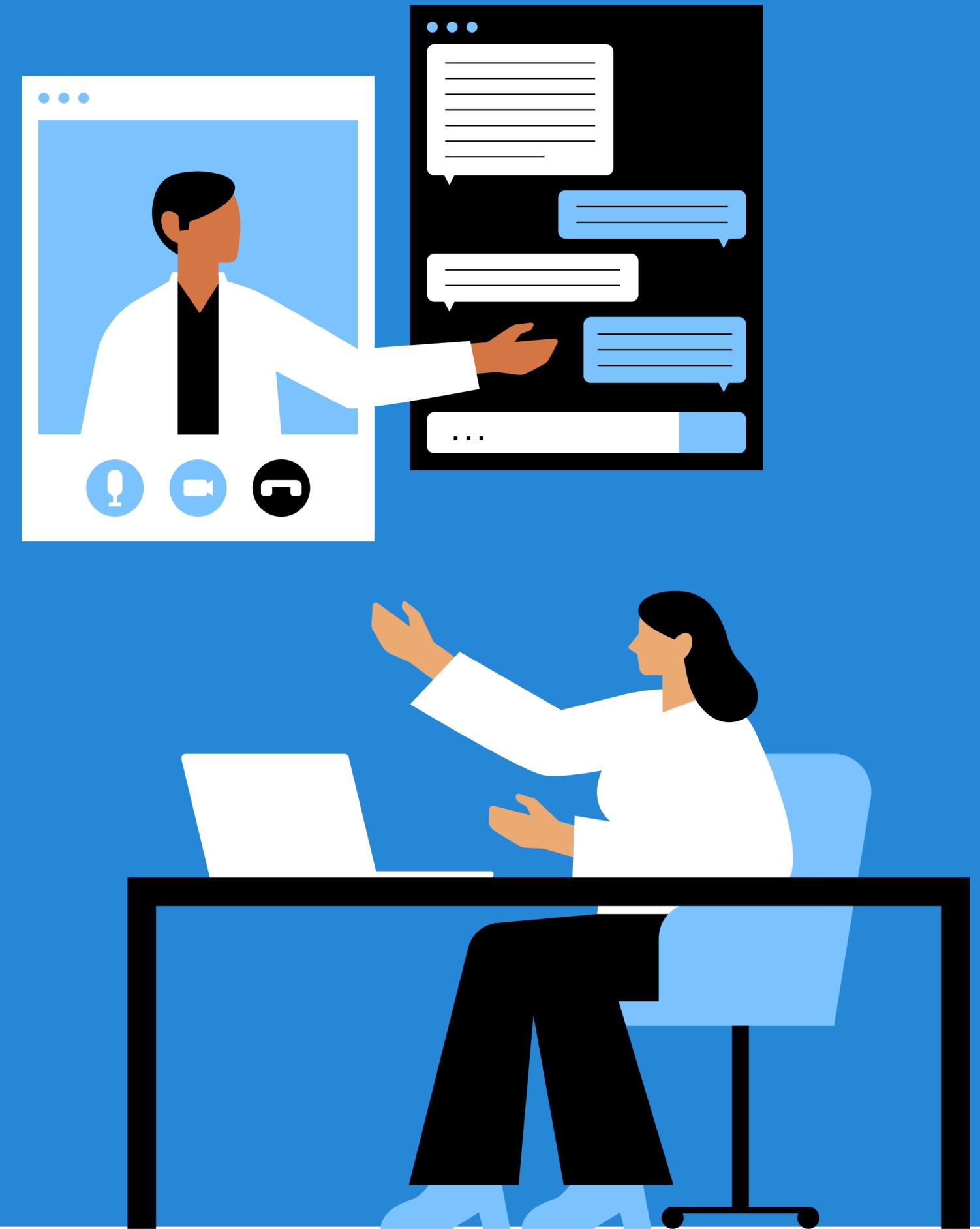


Relatório de testes:

O relatório apresenta os primeiros casos de teste do sistema Atendimento-Fast, focando nas funcionalidades principais e escritos em Gherkin para facilitar a comunicação. A estratégia busca validar requisitos essenciais, encontrar falhas iniciais e servirá de base para futuros testes no projeto.

Suítes de testes:

- Suíte 1: Cadastro e login de usuários – Sanity, Smoke
- Suíte 2: Gerenciamento de serviços – Sanity, Regression
- Suíte 3: Criação de agendas – Regression
- Suíte 4: Agendamento e cancelamento – Smoke, Regression
- Suíte 5: Visualização de agenda e histórico – Sanity, Regression
- Suíte 6: Busca e visualização de clientes – Sanity, Regression



Casos de Teste em Gherkin:

- **Suíte 1: Cadastro e Autenticação**
 - Cadastro de cliente com dados válidos
 - Login com e-mail e senha corretos
- **Suíte 2: Gerenciamento de Serviços**
 - Cadastro de novo serviço com nome, duração e descrição
- **Suíte 3: Agenda**
 - Definir horários disponíveis pelo prestador
 - Gerar horários automaticamente com base na duração do serviço

- **Suíte 4: Agendamento e Cancelamento**
 - Cliente agenda serviço com sucesso
 - Cancelamento dentro do prazo permitido
- **Suíte 5: Visualização de Agenda e Histórico**
 - Administrador visualiza agenda semanal
 - Histórico de agendamentos por cliente
- **Suíte 6: Busca e Visualização de Clientes**
 - Busca por nome ou telefone
 - Visualização do perfil e histórico do cliente

Considerações Finais:

Os testes cobrem a primeira versão do sistema. Serão incluídos casos para múltiplos atendentes, segurança avançada e desempenho. A automação está sendo preparada para integração contínua.

Tecnologias do Back-End

- **Django:** Framework Python para lógica do sistema e gerenciamento do banco.
- **ORM:** Comunicação com o banco de dados usando modelos Python.
- **Views:** Funções que controlam o que cada página ou ação faz no sistema.
- **get_object_or_404:** Método para buscar dados no banco com tratamento automático de erro.

Tecnologias do Front-End

- **HTML:** Estrutura das telas (formulários, listas e campos de busca)
- **CSS:** Aparência e responsividade (cores, fontes e espaçamento)
- **JavaScript:** Interatividade(formato de dados e modal)

