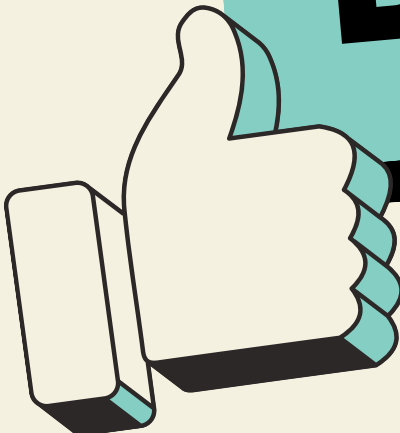
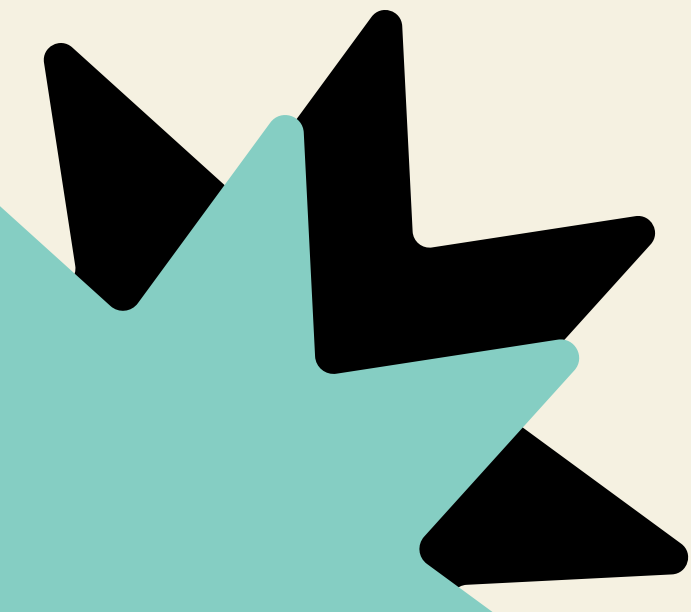
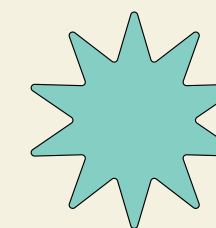
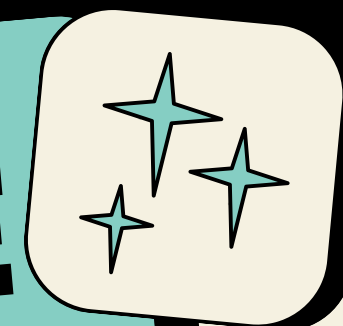
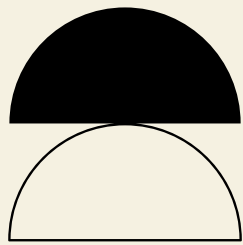
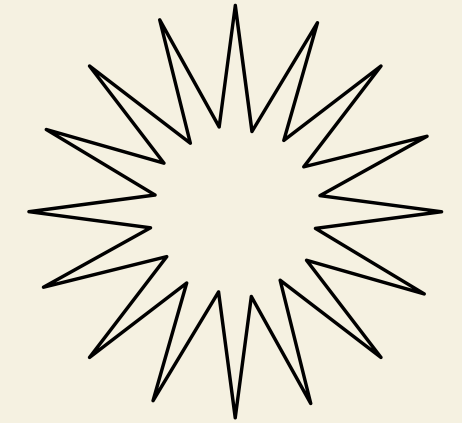
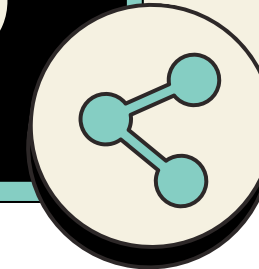


# GESTÃO E QUALIDADE DE SOFTWARE

A 3D-style thumbs up icon with a white hand and teal wristband.



# INTRODUÇÃO



- Desenvolvimento de um sistema de controle de acesso para empresas.
- Registro de entradas e saídas com uso do CPF como identificador único.
- Garante segurança, rastreabilidade e eficiência na gestão de acessos.
- Funcionalidades: cadastro de usuários e geração de relatórios.
- Desenvolvido no NetBeans IDE 8.2 com foco em qualidade, segurança e usabilidade.



# REQUISITOS

## FUNCIONAIS

**PONTOS  
FORTES**

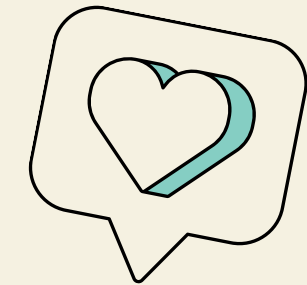
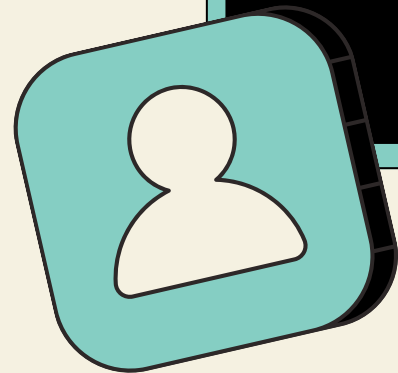
**PONTOS  
FRACOS**

**OPORTU-  
NIDADES**

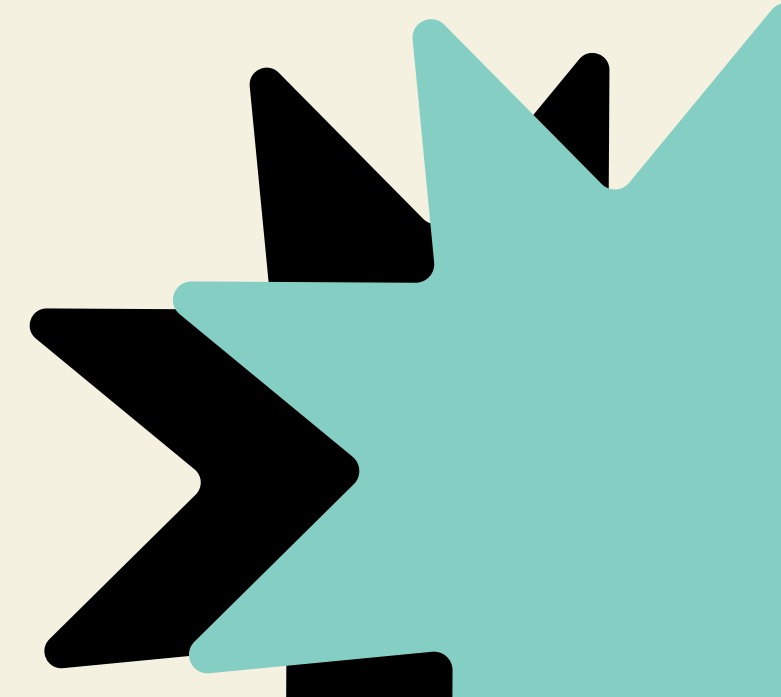
**AMEAÇAS**

- Registro de acessos de usuários.
- Cadastro e gerenciamento de clientes e funcionários.
- Geração de relatórios detalhados.
- Login com CPF e chave de acesso.

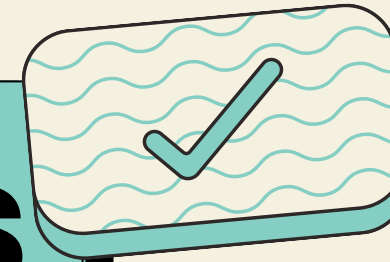
# REQUISITOS NÃO FUNCIONAIS



Além das funções principais, o sistema precisa atender a requisitos não funcionais, que garantem sua qualidade. Ele deverá ter alto desempenho, processando os registros em tempo real sem atrasos. A segurança também será prioridade, com criptografia e proteção de dados. O sistema será escalável, com capacidade de crescer conforme o número de usuários aumentar. A usabilidade será um foco importante, com uma interface amigável e intuitiva. Por fim, o sistema deve ser confiável e resistente a falhas.



# TIPOS DE TESTES



## TESTES FUNCIONAIS

Incluem testes de unidade (para funções individuais), integração (para comunicação entre módulos) e sistema (para avaliar o funcionamento geral).

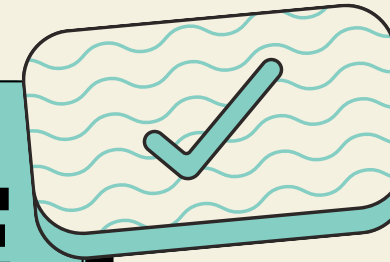
## TESTE DE INTERFACE

Foca na experiência do usuário, garantindo uma navegação fluida.

## TESTES DE QUALIDADE E SEGURANÇA

Incluem teste de regressão (para evitar impactos em funcionalidades antigas) e teste de segurança (para garantir proteção contra acessos indevidos).

# CASOS DE TESTE



## CADASTRO, ATUALIZAÇÃO E EXCLUSÃO DE USUÁRIOS

**Verifica o cadastro de novos usuários, atualização de dados e exclusão com bloqueio de acesso.**

## REGISTROS DE ALTERAÇÕES

**Avalia se as alterações no sistema são registradas nos logs de auditoria.**

## VALIDAÇÃO DE CPF ÚNICO

**Garante que o sistema não permita cadastro de dois usuários com o mesmo CPF.**

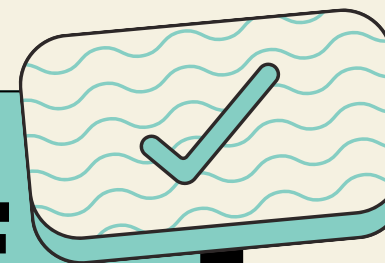
# INFRAESTRUTURA



A infraestrutura necessária para o sistema inclui servidores para aplicação e banco de dados, além de computadores com a IDE NetBeans para o desenvolvimento. O sistema também contará com totens de identificação, catracas eletrônicas, câmeras de reconhecimento facial e leitores biométricos, todos conectados por uma rede segura com conexão LAN, repetidores e switch para garantir a estabilidade na comunicação entre os dispositivos.



# QUALIDADE DE SOFTWARE

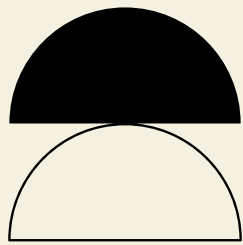
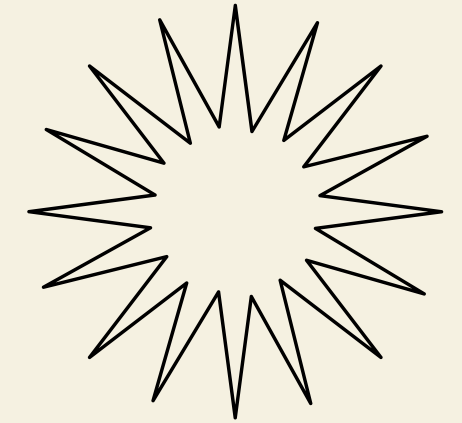
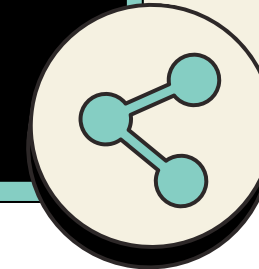


A qualidade do sistema será medida por métricas como a taxa de defeitos encontrados, tempo de resposta, disponibilidade contínua e facilidade de uso. Esses critérios garantem que o sistema seja eficiente, estável e útil para o usuário final, com foco na segurança e controle de acesso, essenciais para seu sucesso.





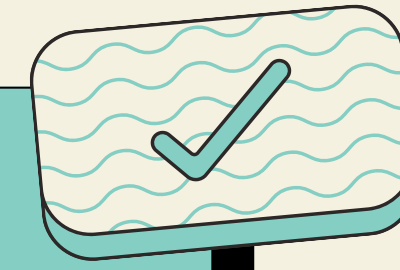
# GESTÃO DE CONFIGURAÇÃO



- Controle de Versão: Utilização do Git com GitHub como repositório remoto.
- Gestão de Desenvolvimento: Organização em branches, definição de baselines e uso do Jira para controle de tarefas.
- Documentação e Aprovação: Mudanças documentadas, discutidas e aprovadas antes de serem aplicadas, garantindo organização e segurança.



# MANUTENÇÃO

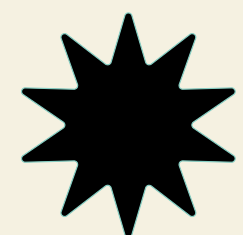
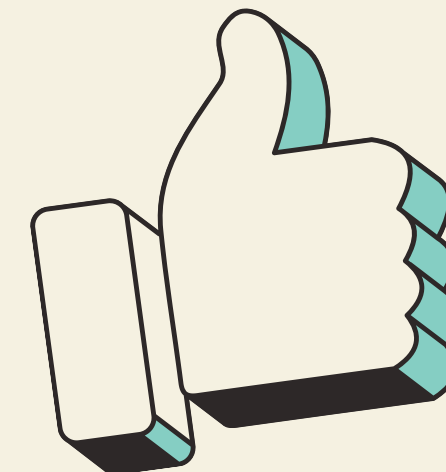
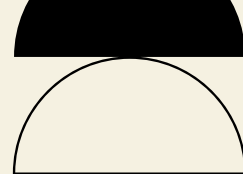


**Manutenção Preventiva** A manutenção preventiva visa garantir o funcionamento contínuo do sistema. Envolve a realização de testes periódicos, aplicação de atualizações e execução de backups regulares. Essas ações ajudam a evitar falhas antes que elas ocorram.

1

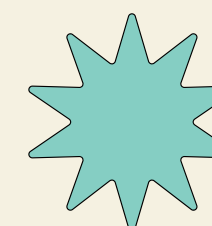
**Manutenção Corretiva** Essa manutenção é aplicada sempre que ocorrem falhas inesperadas no sistema. O objetivo é identificar, corrigir e registrar os erros por meio de relatórios de incidentes, minimizando o impacto para os usuários.

2



## RISCOS E SOLUÇÕES

Foram identificados riscos como atrasos, falta de conhecimento técnico, problemas de integração, incompatibilidades e perda de dados. Para mitigá-los, foram definidas estratégias como treinamentos, acompanhamento do cronograma, testes de integração desde o início e uso do GitHub para backups, tornando o projeto mais seguro e controlável.



# GRUPO



**ADRIANO DE OLIVEIRA SILVA – RA: 823126902**

**BEATRIZ SILVA DE JESUS - RA: 824219590**

**CAROLINA SANTOS DA CRUZ - RA: 824140092**

**CHRISTIAN BATISTA DE LIMA - RA: 824126605**

**FABIO NOGUEIRA ASSEIRO - RA: 823130657**

**MARIANA HILDEBRAND DANTAS - RA: 824118462**

**MARINNA PEREIRA CARNEIRO DA SILVA - RA: 824142121**

**NICOLAS CAUÊ RIBEIRO DOS SANTOS - RA: 823152263**



