

Documentação e Testes

(SISAR) - Sistema Aprova Rápido

Turma: CQS-CCP1AN-BUE1

Nome dos Alunos:

Gabriel Carvalho dos Santos: 821159957
gabrielspfc1109@gmail.com - **Ciência da computação**

Gustavo Mendes de Lima - 821139542
gustavo08.mendes@gmail.com - **Ciência da computação**

Lucas Rodrigues Santos - 823124699
lucasrodrigues2003santos@gmail.com - **Análise e
Desenvolvimento de Sistemas**

Matheus dos Santos de Souza - 821133576
Matheussantos20024@gmail.compra - **Ciência da computação**

Sumário

1. Planejamento de testes de software:.....	3
Alocação de Recursos.....	4
Marcos do Projeto.....	5
2. Documentos de desenvolvimento de software:.....	6
Automação de Testes:.....	8
CI/CD (Integração Contínua e Entrega Contínua):.....	8
Infraestrutura de Desenvolvimento e Hospedagem:.....	8
Ferramentas de Colaboração e Gerenciamento:.....	8
Documento de requisitos;.....	9
Introdução.....	9
Visão Geral do Sistema.....	9
Requisitos Funcionais.....	10
Requisitos Não Funcionais.....	13
Restrições.....	14
Critérios de Aceitação.....	14
Planejamento de testes:.....	14
Tipos de Testes.....	18
Recursos a Serem Empregados.....	21
Cronograma das Atividades.....	22
Roteiro de Testes.....	22
3. Gestão de Configuração de Software.....	23
4. Repositório de Gestão de Configuração de Software.....	23

1. Planejamento de testes de software:

1.1. Cronograma de Atividades

Nome	Data Início	Data Final	Semana 1					Semana 2					Semana 3				
			S	T	Q	Q	S	S	T	Q	Q	S	S	T	Q	Q	S
Gustavo	04/11/2024	05/11/2024	Análise dos Requisitos														
Gabriel	06/11/2024	08/11/2024			Planejamento dos Testes												
Matheus	08/11/2024	09/11/2024				Criação dos Casos de Teste											
Lucas	10/11/2024	12/11/2024						Configuração do Ambiente									
Gabriel	13/11/2024	15/11/2024										Execução dos Testes Unitários					
Gustavo	16/11/2024	18/11/2024											Testes de Segurança				
Matheus	19/11/2024	21/11/2024												Testes de Usabilidade			
Gustavo	22/11/2024	23/11/2024													Testes de Performance		
Gabriel	24/11/2024	30/11/2024														Relatório de Testes	

1.2. Alocação de Recursos.

Essa seção especifica os recursos humanos, de hardware e de softwares necessários para a realização dos testes.

- **Recursos Humanos**
 - **Tester Principal:** Responsável pela criação de casos de teste e pela execução de testes de aceitação.
 - **Desenvolvedor:** Encarregado de resolver problemas identificados nos testes.
 - **Gerente de Qualidade:** Monitora o processo de testes e verifica o cumprimento dos requisitos.
 - **Analista de Requisitos:** Disponível para esclarecer dúvidas sobre funcionalidades do sistema.
- **Recursos de Hardware**
 - **Servidores de Teste:** Para a execução de testes de carga e desempenho.
 - **Dispositivos Móveis e Computadores:** Para realizar testes em diferentes plataformas.
- **Recursos de Software**
 - **Sistema de Gerenciamento de Testes:** Ferramenta para organização e acompanhamento dos casos de teste.

- **Ferramentas de Automação de Testes:** Softwares como Jest ou Cypress para testes automatizados.
- **Banco de Dados:** Instâncias de banco de dados dedicadas para a execução dos testes.

1.3. Marcos do Projeto.

Atividade	Descrição	Início
Conclusão da Análise de Requisitos	Finalização da análise de requisitos e alinhamento entre equipes de desenvolvimento e testes	03/11/2024
Finalização dos Casos de Teste	Término da criação dos casos de teste detalhados	08/11/2024
Configuração Completa do Ambiente	Preparação do ambiente de testes concluída	09/11/2024
Conclusão dos Testes Unitários	Conclusão dos testes em unidades isoladas do sistema	12/11/2024
Conclusão dos Testes de Integração	Verificação das interações entre os módulos concluída	15/11/2024
Conclusão dos Testes de Aceitação	Avaliação da funcionalidade completa do sistema terminada	18/11/2024
Testes de Regressão Concluídos	Execução de todos os testes após correções realizadas	21/11/2024
Entrega do Relatório Final	Entrega do relatório completo com resultados dos testes	23/11/2024

2. Documentos de desenvolvimento de software:

2.1. Plano de projeto:

2.1.1. Planejamento do projeto

1. Planejamento Inicial

- Reunião de levantamento de requisitos com os stakeholders.
- Definição do escopo detalhado e das entregas.
- Criação do cronograma do projeto.

2. Design e Prototipação

- Criação de wireframes e protótipos da interface.
- Validação dos protótipos com os stakeholders.

3. Desenvolvimento

- Backend: Implementação com Nest.js, MySQL para banco de dados e utilização do PM2 para gestão.
- Frontend: Desenvolvimento responsivo com Next.js.
- Integração com sistemas existentes da prefeitura.

4. Testes e Validação

- Testes unitários e integração.
- Testes de usabilidade com engenheiros, arquitetos e agentes públicos.

5. Treinamento e Implementação

- Treinamento para os agentes públicos e stakeholders principais.
- Implantação inicial e monitoramento da adoção.

6. Manutenção e Suporte

- Correção de erros e ajustes com base no feedback dos usuários.
- Atualizações para melhoria de performance e funcionalidades futuras.

2.1.2. Escopo

Arquitetura do sistema.

- Frontend:

- Interface do usuário, gerenciamento de dados com (Tabelas e Formulários) e Dashboard.
- **Backend:**
 - Servidor de aplicação e banco de dados.

Principais Funcionalidades.

- **Armazenamento e Gerenciamento de Processos.**

Cadastro e armazenamento digital de processos relacionados a construções urbanas. Classificação e organização dos processos por categorias, status e datas.

- Avaliação e Aprovação/Reprovação de Processos.
- **Interface para análise dos processos pelos técnicos responsáveis.**
 - Funcionalidade para aprovar ou reprovar processos com base em critérios definidos.
 - Histórico de mudanças e status dos processos, e Agendamento de Reuniões.
- **Sistema para gerar e agendar reuniões baseadas em datas fornecidas pelos técnicos responsáveis.**
 - Integração com calendário para o envio automático de convites e lembretes.
 - Funcionalidade para registrar datas e decisões das reuniões.
 - Geração de Relatórios Mensais.
- **Criação de relatórios mensais detalhados sobre processos aprovados e reprovados.**
 - Exportação dos relatórios em formatos (PDF, Excel, etc.).

Plano de implementação:

- **Fases do projeto:**

- **Definição de Cronograma:** Estabelecer um cronograma para a implementação das diferentes etapas do sistema, alinhando com as metas da diretiva de tempo mínimo para análise de processos.
- **Levantamento de requisitos:** Identificar todas as funcionalidades e necessidades do sistema junto às partes interessadas, como técnicos, gestores e administradores urbanos.
- **Testes de integração:** Verificar se os diferentes módulos do sistema interagem corretamente, como o fluxo de aprovação de processos, marcação de reunião e geração de relatórios.

2.1.3. Recursos;

Automação de Testes:

- **Selenium:** permite automatizar testes de aplicações web.
- **Cypress:** ferramenta para automação de testes front-end de aplicações web.
- **Postman:** ferramenta para automação de testes de APIs.

CI/CD (Integração Contínua e Entrega Contínua):

- **GitHub:** Para orquestração de pipelines de CI/CD.
- **Docker:** Para containerização das aplicações e suas dependências.

Infraestrutura de Desenvolvimento e Hospedagem:

- **AWS / Azure:** Para hospedagem da aplicação em nuvem.
- **Nginx / Apache:** Para servidores web e gerenciamento de tráfego HTTP.
- **Prometheus / Grafana:** Para monitoramento de desempenho e alertas.

Ferramentas de Colaboração e Gerenciamento:

- **Trello:** Ferramenta de colaboração visual baseada em quadros e cartões, ideal para gerenciamento de tarefas e projetos de maneira simples e intuitiva.

- **GitHub:** plataforma que equipes de desenvolvimento podem trabalhar de forma conjunta em um repositório, enviando suas contribuições por meio de "commits" e revisões de código.

2.1.4. Estimativas de projeto.

- **Testes de Integração:** 3 semanas de testes e ajustes.
- **Treinamento de Usuários:** 2 semanas para garantir que os stakeholders estejam totalmente preparados para usar o sistema.
- **Lançamento:** 1 semana para o lançamento oficial e monitoramento pós-lançamento.

2.2. Documento de requisitos;

Introdução

- **Objetivo do Documento:** Este documento descreve os requisitos para o desenvolvimento de um sistema de gestão e avaliação de processos relacionados a construções em áreas urbanas da cidade de São Paulo. O sistema visa substituir o gerenciamento feito manualmente em planilhas, proporcionando uma solução mais eficiente e otimizada.
- **Escopo do Sistema:** O sistema será responsável por:
 - Registrar, acompanhar e avaliar processos de construção.
 - Organizar as informações de forma estruturada.
 - Facilitar a consulta e o monitoramento do status dos processos de licenciamento e aprovações.
 - Gerenciar os dados relacionados aos processos de maneira eficiente, com controle e visibilidade em tempo real.
- **Definições e Abreviações:**
 - **Processo de Construção:** Um conjunto de etapas relacionadas ao licenciamento e aprovação de projetos de construção.
 - **Stakeholder:** Qualquer parte interessada no processo, como órgãos públicos, engenheiros, arquitetos, proprietários, etc.

Visão Geral do Sistema

- **Descrição Geral:** O sistema será uma plataforma web que permitirá o registro, acompanhamento e a avaliação de processos de construção na cidade de São Paulo. A plataforma substituirá as planilhas utilizadas atualmente, proporcionando automação, organização e controle dos dados.

- **Principais Stakeholders:**

- **Secretaria Municipal de Urbanismo e Licenciamento (SMUL):** Responsável pelo gerenciamento do sistema.
- **Engenheiros e Arquitetos:** Profissionais que irão submeter projetos e acompanhar os processos de aprovação.
- **Proprietários de Imóveis:** Interessados no status dos seus processos de licenciamento.
- **Agentes Públicos:** Funcionários responsáveis pela avaliação e aprovação dos projetos de construção.

- **Objetivos:**

- Reduzir o tempo de análise de processos.
- Melhorar a organização e visibilidade dos processos.
- Eliminar erros manuais causados pelo uso de planilhas.
- Garantir a conformidade com as regulamentações municipais.

Requisitos Funcionais

RF001 - Recepção de Processos

- **Descrição:** O sistema deve permitir a recepção manual dos processos de construção já criados em outro sistema. Esses processos podem ser inseridos no sistema manualmente por um usuário autorizado.
- **CrITÉrios de Aceitação:**
 - O processo deve ser registrado no sistema por um usuário autorizado.
 - O sistema deve permitir o cadastro manual do processo, associando-o a um número de identificação único.
 - O sistema deve registrar a data e hora da recepção do processo.

RF002 - Verificação de Admissibilidade

- **Descrição:** O sistema deve verificar se o processo é admissível ou inadmissível. O processo será considerado admissível se estiver completo e conforme as normas pré-definidas. Se inadmissível, o proprietário será notificado para reenvio de documentos.

- **Critérios de Aceitação:**
 - Caso o processo seja **inadmissível**, o sistema deve gerar uma notificação para o proprietário ou responsável e permitir a reabertura do processo para correções dentro de um prazo determinado (ex: 15 dias).
 - Caso o processo seja **admissível**, ele será movido para a fase de análise.

RF003 - Análise do Processo

- **Descrição:** Após ser considerado admissível, o processo deve ser enviado para análise. Nessa fase, o sistema deve permitir que um técnico responsável possa visualizar os documentos e realizar as avaliações necessárias para tomar a decisão sobre o processo.
- **Critérios de Aceitação:**
 - O processo deve ser atribuído automaticamente a um técnico ou responsável pela análise.
 - O sistema deve permitir a marcação de status do processo como "Em Análise", "Admissibilidade", "inadmitido" etc.
 - O sistema deve permitir que o analista registre o parecer sobre o processo (deferido ou indeferido).

RF004 - Finalização do Processo

- **Descrição:** O processo deve ser finalizado com uma decisão de "Deferido" ou "Indeferido", com base na análise realizada. O sistema deve permitir que essa decisão seja registrada e notificada ao proprietário.
- **Critérios de Aceitação:**
 - O analista deve ser capaz de registrar a decisão final sobre o processo (Deferido ou Indeferido).
 - Caso o processo seja **indeferido**, o sistema deve permitir a abertura de até 3 instâncias de recurso.
 - O sistema deve gerar notificações automáticas para os envolvidos, informando sobre a decisão final ou sobre a necessidade de enviar documentos adicionais (se for o caso).

RF005 - Instâncias de Recurso (Indeferimento)

- **Descrição:** Quando o processo for **indeferido**, o proprietário poderá recorrer da decisão até três vezes. O sistema deve gerenciar essas instâncias de recurso, registrando o histórico de cada instância e permitindo que o processo seja reavaliado em cada uma delas.
- **Critérios de Aceitação:**
 - O sistema deve permitir que até três instâncias de recurso sejam registradas, se aplicável.
 - Para cada instância de recurso, o sistema deve permitir que os documentos de defesa sejam enviados pelo proprietário e analisados novamente.

- o O status do processo deve ser atualizado conforme o progresso de cada instância de recurso, e o sistema deve registrar as decisões de cada instância (Se deferido ou indeferido novamente).
- o Após a terceira instância, o processo será considerado finalizado (com ou sem deferimento).

RF006 - Relatórios de Acompanhamento de Processos

- **Descrição:** O sistema deve permitir a geração de relatórios que ajudem os responsáveis a acompanhar o status dos processos de construção, incluindo a admissibilidade, análise, decisões de deferimento ou indeferimento, e status das instâncias de recurso.
- **Critérios de Aceitação:**
 - o O usuário deve poder gerar relatórios filtrados por status, tipo de análise, data de entrada, entre outros parâmetros.
 - o O relatório deve incluir informações sobre o histórico de decisões e os prazos de cada processo, facilitando o acompanhamento do progresso dos processos.

RF006 - Cadastro de Subprefeituras

- **Descrição:** O sistema deve permitir o cadastro e gerenciamento das **subprefeituras** da cidade de São Paulo. Cada subprefeitura será vinculada aos processos de construção, para indicar qual é a subprefeitura responsável pela análise do processo.
- **Critérios de Aceitação:**
 - o O sistema deve permitir o cadastro de subprefeituras, incluindo informações como nome, código, localização e outras características relevantes.
 - o O cadastro de subprefeituras deve ser feito por um usuário autorizado, como um administrador do sistema.
 - o O sistema deve permitir que, ao cadastrar um processo, a subprefeitura responsável seja selecionada e vinculada ao processo, embora não seja um campo obrigatório.

RF007 - Cadastro de Unidades

- **Descrição:** O sistema deve permitir o cadastro de **unidades**, que são as divisões administrativas dentro de cada subprefeitura (ex: regiões ou bairros) que podem ser relacionadas a processos específicos.
- **Critérios de Aceitação:**
 - o O sistema deve permitir o cadastro de unidades, com informações como nome, código, e a subprefeitura à qual pertence.
 - o O cadastro das unidades deve ser feito por usuários autorizados.
 - o O sistema deve permitir que um processo de construção seja vinculado a uma unidade específica, embora este vínculo não seja obrigatório.

RF008 - Cadastro de Tipos de Alvará

- **Descrição:** O sistema deve permitir o cadastro de **tipos de alvará** (ex: alvará de construção, alvará de licença, etc.), que serão usados para categorizar e definir os prazos e as regras de análise para os processos de construção. O tipo de alvará será obrigatório no registro do processo.
- **CrITÉrios de Aceitação:**
 - O sistema deve permitir o cadastro de tipos de alvará, incluindo descrição, categoria (ex: tipo de alvará para obras residenciais, comerciais, etc.) e as regras associadas a cada tipo (ex: prazos, documentos obrigatórios).
 - O sistema deve garantir que, ao registrar um processo, o tipo de alvará seja sempre escolhido e associado ao processo.
 - O sistema deve associar prazos automáticos de análise e de inadmissibilidade com base no tipo de alvará escolhido (ex: se o tipo de alvará for de construção residencial, o prazo para inadmissibilidade será de 15 dias).

RF009 - Vinculação de Subprefeituras, Unidades e Tipos de Alvará ao Processo

- **Descrição:** Quando o processo for registrado, ele pode ser vinculado a uma subprefeitura, unidade e tipo de alvará. O tipo de alvará é obrigatório, enquanto a subprefeitura e a unidade são opcionais, mas devem estar disponíveis para seleção.
- **CrITÉrios de Aceitação:**
 - O sistema deve permitir a seleção do tipo de alvará de forma obrigatória no momento do registro do processo.
 - O sistema deve permitir que o tipo de alvará selecionado determine automaticamente o prazo de inadmissibilidade (ex: 15 dias para reenvio de documentos) e outros prazos e critérios relacionados.
 - O sistema deve permitir que, se necessário, o processo seja vinculado a uma subprefeitura e/ou unidade, mas sem tornar essas informações obrigatórias no cadastro inicial do processo.
 - O sistema deve garantir que, ao registrar ou atualizar o processo, as informações relacionadas à subprefeitura, unidade e tipo de alvará sejam corretamente armazenadas.

Requisitos Não Funcionais

RNF001 - Desempenho

- O sistema deve ser capaz de processar até 1000 processos simultaneamente sem impacto significativo no desempenho.

RNF002 - Usabilidade

- A interface deve ser simples e intuitiva, permitindo que os agentes públicos e os usuários responsáveis pela gestão dos processos realizem tarefas rapidamente.

RNF003 - Segurança

- O sistema deve garantir que apenas usuários autenticados e autorizados possam acessar e modificar os processos.
- Todos os dados e documentos devem ser armazenados de forma criptografada.

RNF004 - Escalabilidade

- O sistema deve ser escalável para suportar um aumento no número de processos e usuários conforme a demanda aumenta.

RNF005 - Compatibilidade

- O sistema deve ser acessível por navegadores modernos (Chrome, Firefox, Edge, Safari) e em dispositivos móveis.

Restrições**Tecnológicas**

- O sistema será desenvolvido utilizando tecnologias web modernas, como Next.js para o frontend e Node.js para o backend. O banco de dados será relacional (exemplo: MySQL).

Regulatórias

- O sistema deve estar em conformidade com as normas municipais de licenciamento e construção de São Paulo, incluindo as políticas de armazenamento e processamento de dados pessoais (como a Lei Geral de Proteção de Dados - LGPD).

Critérios de Aceitação

- O sistema será validado com base nos requisitos acima, com testes de usabilidade, desempenho e segurança.
- Todos os fluxos de processo (admissibilidade, análise, deferimento/indeferimento e instâncias de recurso) serão testados em um ambiente de homologação com dados de processos reais.

2.3. Planejamento de testes:

2.3.1 Plano de Testes

2.3.1.1 Introdução

Este plano de testes visa garantir que o sistema de gestão e avaliação de processos de construção atenda aos requisitos estabelecidos e funcione conforme esperado. O sistema tem como objetivo substituir o processo manual de gerenciamento de construções, melhorando a eficiência, organização e controle dos dados relacionados ao licenciamento e aprovação de projetos de construção em áreas urbanas da cidade de São Paulo. O plano de testes cobrirá todas as funcionalidades do sistema, incluindo o cadastro de processos de construção, verificação de admissibilidade, análise e finalização de processos, gestão de recursos e geração de relatórios.

2.3.1.2 Escopo

O escopo deste plano de testes abrange os seguintes componentes do sistema:

- o Cadastro de processos de construção, subprefeituras, unidades e tipos de alvará.
- o Funcionalidades de verificação de admissibilidade e notificação de inadmissibilidade.
- o Atribuição e gestão de processos durante a fase de análise, incluindo o registro de decisões de deferimento ou indeferimento.
- o Gestão de instâncias de recurso em casos de indeferimento de processos.
- o Geração de relatórios de acompanhamento de processos.
- o Funcionalidades de controle de acesso e segurança do sistema (usuários autorizados).
- o Este plano de testes não incluirá testes de integração com outros sistemas externos ou infraestrutura de rede, que são abordados em planos de testes separados.

2.3.1.3 Objetivos

Os objetivos dos testes são:

- o Validar que o sistema permita o registro de processos, subprefeituras, unidades e tipos de alvará de acordo com as especificações.
- o Verificar se a verificação de admissibilidade e as notificações de inadmissibilidade funcionam conforme esperado.
- o Garantir que o sistema de análise e decisão dos processos (deferido ou indeferido) funcione de maneira adequada.
- o Validar o fluxo de instâncias de recurso, incluindo a limitação a três instâncias e o correto armazenamento do histórico de decisões.
- o Testar a geração de relatórios detalhados sobre os processos, incluindo filtros e informações de status.
- o Certificar-se de que o controle de acesso e as permissões para usuários estão implementados corretamente, permitindo que somente usuários autorizados possam executar ações específicas.

2.3.1.4 Requisitos a serem testados

Requisitos funcionais que serão validados durante os testes:

- o **RF001** - Recepção de Processos: Testar a recepção manual de processos e garantir que todos os dados sejam registrados corretamente.
Critérios de aceitação: O processo é registrado, atribuído a um número de identificação único, e a data e hora da recepção são registradas.'020000
- o **RF002** - Verificação de Admissibilidade: Validar que o sistema verifica corretamente se o processo é admissível ou inadmissível, gerando notificações quando necessário.
Critérios de aceitação: O sistema deve gerar notificação se o processo for inadmissível e permitir reabertura para correções.
- o **RF003** - Análise do Processo: Verificar se o processo é enviado corretamente para análise e se a avaliação é registrada corretamente.
Critérios de aceitação: O status do processo deve ser atualizado corretamente para "Em Análise", e o parecer do analista deve ser registrado.
- o **RF004** - Finalização do Processo: Testar a funcionalidade de finalização do processo, incluindo as decisões de deferimento ou indeferimento.
Critérios de aceitação: O sistema deve permitir o registro da decisão final e gerar notificações para os envolvidos.
- o **RF005** - Instâncias de Recurso (Indeferimento): Validar a gestão das instâncias de recurso, com até três reavaliações do processo.
Critérios de aceitação: O sistema deve permitir até três instâncias de recurso e atualizar o status de cada uma, registrando as decisões.
- o **RF006** - Relatórios de Acompanhamento de Processos: Validar a geração de relatórios detalhados sobre os processos.

Critérios de aceitação: O usuário deve ser capaz de gerar relatórios com filtros como status, data e tipo de análise.

- **RF007** - Cadastro de Subprefeituras: Verificar o cadastro e gerenciamento das subprefeituras.
Critérios de aceitação: O sistema deve permitir o cadastro de subprefeituras e vincular processos a elas.
- **RF008** - Cadastro de Unidades: Validar o cadastro de unidades administrativas e seu vínculo com processos de construção.
Critérios de aceitação: O sistema deve permitir o cadastro de unidades e vinculá-las aos processos de construção, quando necessário.
- **RF009** - Cadastro de Tipos de Alvará: Testar o cadastro de tipos de alvará e a associação obrigatória com os processos de construção.
Critérios de aceitação: O sistema deve garantir que o tipo de alvará seja selecionado obrigatoriamente e associado ao processo.
- **RF010** - Vinculação de Subprefeituras, Unidades e Tipos de Alvará ao Processo: Validar o correto vínculo de subprefeituras, unidades e tipos de alvará aos processos.
Critérios de aceitação: O processo deve ser vinculado ao tipo de alvará (obrigatório), e, quando necessário, a uma subprefeitura e unidade.

2.3.1.5 Estratégias, Tipos de Testes e Ferramentas a Serem Utilizadas.

Estratégias de Testes

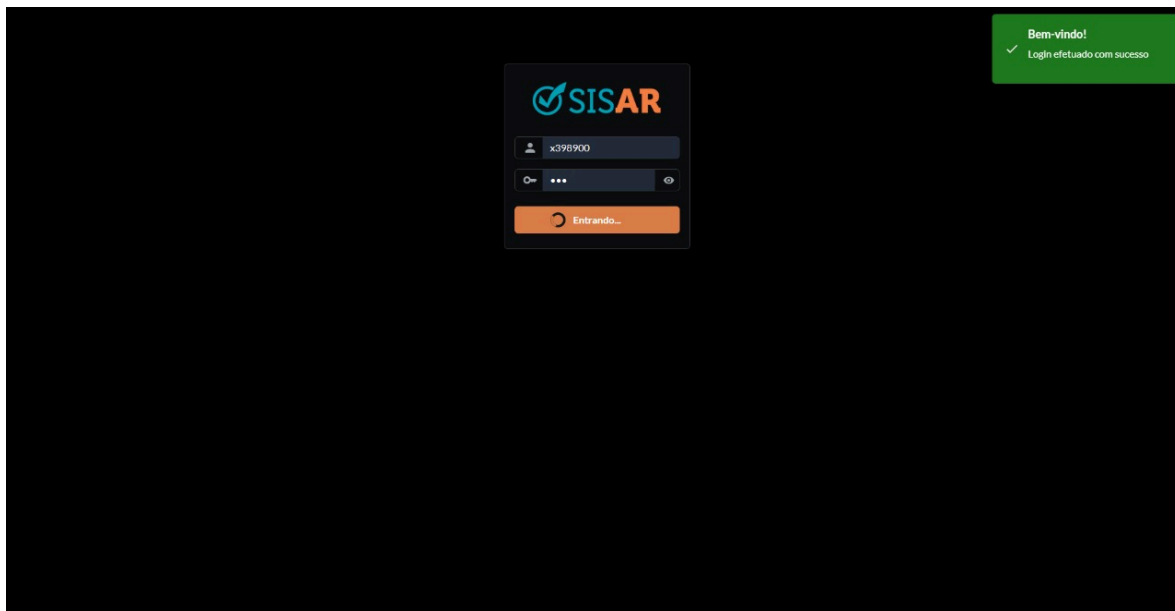
- Ciclo Iterativo de Testes: Realizar testes a cada incremento de desenvolvimento para validar funcionalidades e identificar falhas.
- Automação de Testes: Automatizar os testes de funcionalidades principais para maior eficiência.
- Teste de Aceitação com Stakeholders: Envolver os usuários-chave no processo de validação final para garantir aderência aos requisitos.

Ferramentas:

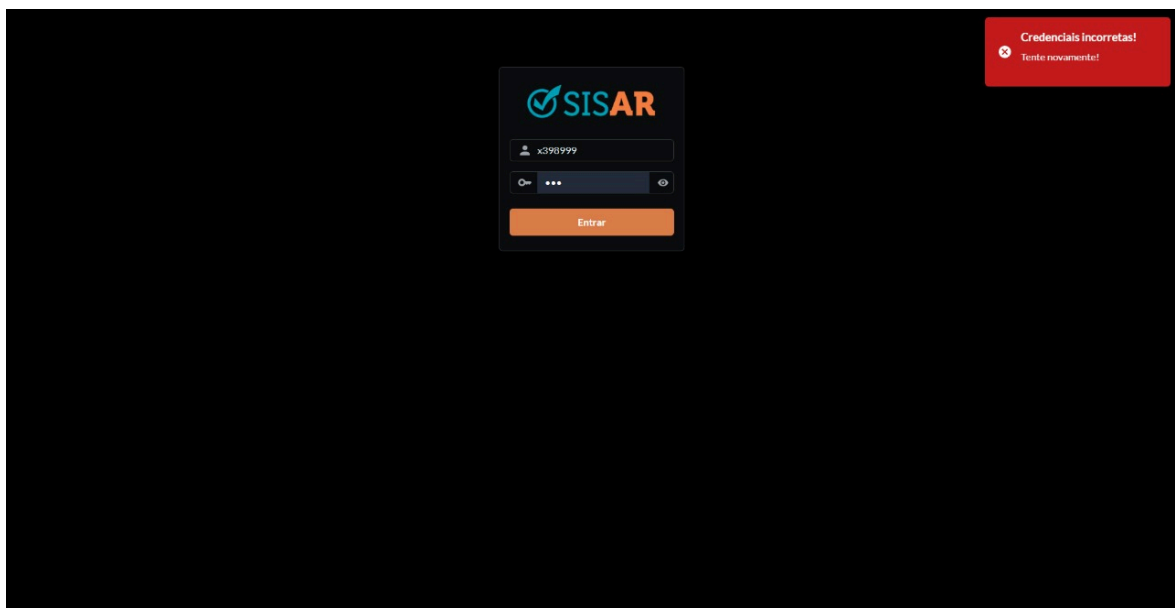
- **Selenium:** permite automatizar testes de aplicações web.
- **Cypress:** ferramenta para automação de testes front-end de aplicações web.
- **Postman:** ferramenta para automação de testes de APIs.

Tipos de Testes

1. **Testes de Segurança:** Avaliar a proteção do sistema contra acessos não autorizados, vulnerabilidade e ataques cibernéticos.



- Usuário cadastrado, acesso autorizado.



- Usuário não cadastrado, acesso negado.

Página Inicial

dezembro 2024

S M T W T F S

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

Reuniões 0

10/12/2024

#	Status	SEI	Tipo Req.	Requerimento	Protocolo	Tipo de Alvará	Tipo de Processo
1	Admissibilidade	1111.1111/1111111-9	1	awd	09/12/2024	wadaw	Múltiplas Interfaces

Registros por página 10 1-1 de 1

Gustavo Mendes
gmendes@prefeitura.sp.gov.br
Desenvolvedor

DevTools is now available in Portuguese!
Always match Chrome's language
Switch DevTools to Portuguese
Don't show again

89
Accessibility

These checks highlight opportunities to **improve the accessibility** of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so **manual testing** is also encouraged.

NAMES AND LABELS

Buttons do not have an accessible name

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

Background and foreground colors do not have a sufficient contrast ratio.

These are opportunities to improve the legibility of your content.

TABLES AND LISTS

Lists do not contain only `` elements and script supporting elements (`<script>` and `<script type="module">`)

These are opportunities to improve the experience of reading tabular or list data using assistive technology, like a screen reader.

3. **Testes de Performance:** Verificar se a plataforma suporta simultaneamente um grande número de acessos, especialmente em horários de pico.

Admissibilidade

Buscar:

Todos

Geral

Inicial ID	Sei	Data Envio	Data Criação	Status	Prazo
1	1111.1111/1111111-9	09/12/2024	10/12/2024	Admissível	Admitido

Registros por página 10 1-1 de 1

Gustavo Mendes
gmendes@prefeitura.sp.gov.br
Desenvolvedor

http://localhost:3001/inicial

DevTools is now available in Portuguese!
Always match Chrome's language
Switch DevTools to Portuguese
Don't show again

82
Performance

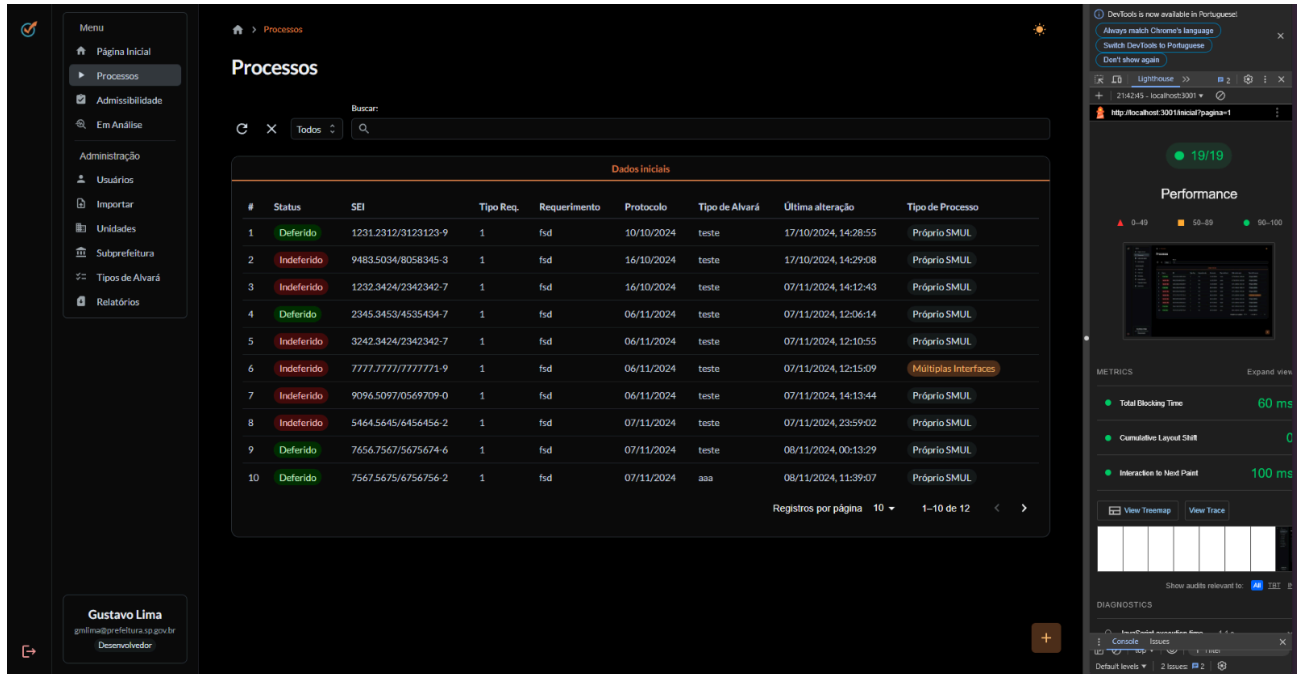
Values are estimated and may vary. The **performance score is calculated** directly from these metrics. [See calculator.](#)

0-49 50-49 90-100

METRICS

First Contentful Paint 0.5 s

Largest Contentful Paint 1.5 s



2.3.1.6 Recursos a Serem Empregados

- **Recursos Humanos:**
 - Desenvolvedores Front-end e Back-end.
 - Especialista em DevOps para gestão da infraestrutura.
 - Analistas de Negócios para levantamento de requisitos.
 - Testadores para validação funcional.
 -
- **Infraestrutura:**
 - Servidores em nuvem (AWS ou Azure) para hospedagem e escalabilidade.
 - Banco de dados relacional (PostgreSQL) para armazenamento de dados estruturados.
 - Ferramentas de versionamento (GitHub/GitLab).
- **Software e Licenças:**
 - IDEs como Visual Studio Code para desenvolvimento.
 - Jira para gerenciamento de tarefas e monitoramento de progresso.
 - Licença de ferramentas de testes automatizados, como Selenium e Postman.

2.3.1.7 Cronograma das Atividades

Nome	Data Inicio	Data Final	Semana 1					Semana 2					Semana 3				
			S	T	Q	Q	S	S	T	Q	Q	S	S	T	Q	Q	S
Gustavo	28/10/2024	01/11/2024	Teste de Interface														
Gabriel	01/11/2024	06/11/2024					Testes Unitários										
Matheus	06/11/2024	08/11/2024								Teste Funcionais							
Lucas	11/11/2024	13/11/2024										Teste de Integração					
Gabriel	13/11/2024	15/11/2024											Testes de Performance				
Gustavo	18/11/2024	20/11/2024												Teste de Segurança			
Matheus	20/11/2024	21/11/2024													Teste de Usabilidade		
Lucas	22/11/2024	22/11/2024														Relatório de Testes	

2.3.1.8 Definição dos Marcos do Projeto

1. Teste de Interface: 01/11/2024
2. Teste Unitários : 06/11/2024.
3. Teste Funcionais : 08/11/2025.
4. Conclusão dos Testes de Integração: 29/03/2025.
5. Lançamento Oficial do Sistema: 19/04/2025.

2.3.2 Casos de Testes

- Caso 1:** Registro de Processos.
 Descrição: Validar o cadastro correto de um novo processo com todos os campos obrigatórios preenchidos.
 Critério de Aceitação: Processo registrado com sucesso e disponível para consulta.
- Caso 2:** Acompanhamento de Processos.
 Descrição: Testar a funcionalidade de busca e filtro de processos por critérios, como status e responsável.
 Critério de Aceitação: Resultados precisos exibidos em tempo real.
- Caso 3:** Avaliação de Projetos.
 Descrição: Verificar a possibilidade de agentes públicos adicionarem observações e aprovarem projetos.
 Critério de Aceitação: Observações salvas e status do processo atualizado.

2.3.3 Roteiro de Testes

1. Configuração do Ambiente: Preparar o servidor, banco de dados e endpoints da API para testes.
2. Execução dos Testes Funcionais: Garantir que todas as funcionalidades principais estão operacionais.
3. Testes de Integração: Validar a comunicação entre os módulos, como cadastro e acompanhamento.
4. Validação de Performance: Realizar testes de carga com JMeter.
5. Relatório de falhas: Documentar problemas encontrados e encaminhar ao desenvolvimento.
6. Re-execução de Testes: Confirmar as correções realizadas.

3. Gestão de Configuração de Software

- Controle de Versão: O sistema usará Git para gerenciamento de código, com branches dedicadas para desenvolvimento, homologação e produção.
- Gestão de Configuração: Ferramentas como Jenkins ou GitHub Actions para integração contínua e entrega contínua (CI/CD).
- Documentação: Registro das mudanças, incluindo notas de versão e históricos de commits.

4. Repositório de Gestão de Configuração de Software

O código será armazenado no GitHub em um repositório privado, com a seguinte estrutura:

- Branch Main: Código pronto para produção.
- Branch Develop: Código em desenvolvimento e integração.
- Branches Feature: Recursos específicos sendo desenvolvidos.
- Branches Hotfix: Correções emergenciais.

Conteúdo do Repositório:

- Código-fonte organizado por módulos (backend, frontend).
- Scripts de banco de dados e configuração de infraestrutura.
- Documentação do projeto (manuais, requisitos, cronogramas).
- Testes automatizados.