

StartInfo - Letramento Digital para Idosos

Plano de Teste

Versão 1.0

Histórico de Revisões			
Versão	Data	Descrição	Autor
1.0	17/01/2025	Atualização das informações contidas no cabeçalho e rodapé.	Grupo
2.0	24/01/2025	Início de preenchimento dos tipos de teste.	Grupo
3.0	31/01/2025	Ajustes nos teste	Grupo
4.0	07/02/2025	Ajustes finais no documento	Grupo
5.0	14/02/2025	Entrega Final	Grupo

Sumário

1.	INTRODUÇÃO	4
2.	IDENTIFICAÇÃO DO PROJETO	5
3.	CENÁRIO OPERACIONAL	5
4.	ITENS DE TESTE	5
5.	PREMISSAS	5
6.	RESTRIÇÕES	5
7.	ABORDAGEM DOS TESTES	6
8.	ARTEFATOS DE ENTRADA	7
9.	ARTEFATOS DE SAÍDA	7
10.	FLUXOS DE TRABALHO DE TESTE	7
11.	INFRAESTRUTURA NECESSÁRIA	7
12.	RESPONSABILIDADES DA EQUIPE	8
13.	COMPONENTES DE TESTES DE INTEGRAÇÃO	8
14.	GERENCIAMENTO DE COMUNICAÇÃO	8
15.	MATRIZ DE COMUNICAÇÃO	9
16.	CRONOGRAMA	9
17.	MARCOS	9
18.	RISCOS	9
19.	ANEXOS	10
20.	REFERÊNCIAS	10
21.	APROVAÇÕES	10

Plano de Teste

1. Introdução

Este documento apresenta o plano de teste para o projeto StartInfo, cujo objetivo é promover a inclusão digital através de tutoriais simples e intuitivos.

1.1 Finalidade, Escopo e Objetivos do Projeto

O projeto visa integrar idosos ao universo digital, ensinando o uso de ferramentas como PIX, e-mail e redes sociais, por meio de um website educativo.

1.2 Missão de Avaliação

- Localizar o maior número possível de erros.
- Avaliar os riscos da qualidade percebida.
- Informar sobre a qualidade do produto.

1.3 Motivadores de Testes

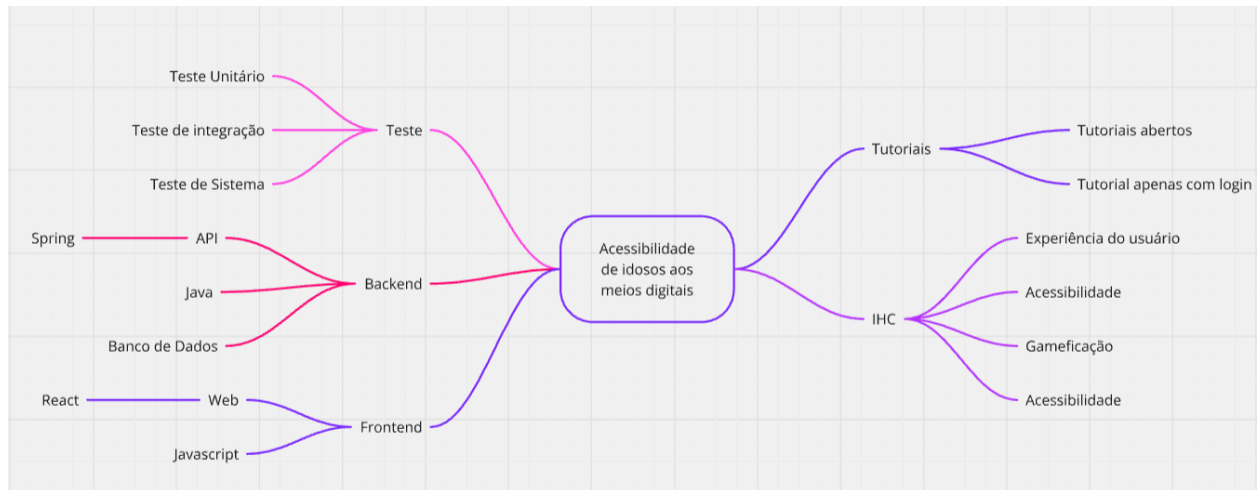
- **Riscos de qualidade:** garantir que a interface do usuário seja intuitiva e responsiva.
- **Riscos técnicos:** verificar a integração entre frontend e backend.
- **Requisitos funcionais:** assegurar que funcionalidades essenciais, como cadastro e login, funcionem corretamente.
- **Requisitos não funcionais:** validar desempenho e segurança.
- **Casos de uso:** verificar a experiência do usuário final.

2. Identificação do Projeto

Projeto	Star Info
Requisitante	Pessoas acima de 60 anos
Gerente de Projetos	Ricardo

3. Cenário Operacional

O website educativo será testado em diferentes dispositivos (desktop, tablet e smartphone) e navegadores (Chrome, Firefox e Edge) para garantir compatibilidade. Um fluxograma detalhado será criado utilizando Visio para mapear o fluxo do sistema.



4. Itens de Teste

- **Website Educativo (Tutoriais)**
- **Funcionalidades:** Cadastro, Login e Tutoriais.

5. Premissas

- A comunicação entre backend e frontend deve ser eficiente.
- O sistema deve garantir uma experiência intuitiva para idosos.

6. Restrições

O prazo para entrega do projeto é 14/02/2025..

7. Abordagem dos Testes

Teste de Unidade

- **Ferramentas Utilizadas:** Maven, JUnit, Mockito.

Teste de Integração

- **Ferramentas Utilizadas:** Vscode, Docker, Debaver, Java Spring Boots, Maven, JUnit e Mock.

Teste de Sistema

- **Ferramentas Utilizadas:** Teste manuais.

7.1 Tipo de Teste

Para garantir a qualidade do StartInfo, escolhemos três tipos principais de teste: Teste de Unidade, Teste de Integração e Teste de Sistema. Cada um desses testes desempenha um papel crucial na validação do software e na identificação de problemas em diferentes níveis.

1. Teste de Unidade

Descrição: O teste de unidade foca na verificação de componentes individuais do software, assegurando que cada parte funcione corretamente isoladamente.

Objetivo da Técnica: Identificar e corrigir erros em estágios iniciais do desenvolvimento, garantindo que cada unidade de código atenda aos requisitos especificados.

Técnica: Utilização de frameworks de teste como JUnit (para Java) para automatizar a execução dos testes.

Ferramentas Utilizadas: JUnit, Mockito.

Critérios de Êxito/Conclusão: Todos os testes unitários devem ser executados com sucesso, sem falhas, e todas as funcionalidades devem se comportar conforme o esperado.

Titulo: Teste de Unidade cadastro de Cliente

Status: **Passou**

- Criamos um teste para o formulário de cadastro, verificando se os campos eram preenchidos corretamente e se a função de envio era chamada com os dados esperados.
- Utilizamos mocks para simular a resposta do servidor sem realmente fazer uma requisição HTTP.
- O teste validou que o feedback correto (mensagem de sucesso ou erro) era exibido ao usuário.

```
49
50
51 @Test
52 void deveSalvarClienteComSucesso() {
53     when(usuarioService.save(any(type:Usuario.class))).thenReturn(usuarioLogado);
54     when(clienteRepository.save(any(type:Cliente.class))).thenReturn(cliente);
55
56     Cliente clienteSalvo = clienteService.save(cliente, usuarioLogado);
57
58     assertNotNull(clienteSalvo);
59     assertEquals(expected:"João da Silva", clienteSalvo.getNomeCompleto());
60     assertEquals(expected:"123.456.789-00", clienteSalvo.getCpf());
61
62     verify(usuarioService, times(wantedNumberOfInvocations:1)).save(cliente.getUsuario());
63     verify(clienteRepository, times(wantedNumberOfInvocations:1)).save(cliente);
64     verify(emailService, times(wantedNumberOfInvocations:1)).enviarEmailConfirmacaoCadastroCliente(cliente);
65 }
66
```

PROBLEMAS 2 SAÍDA CONSOLE DE DEPURACÃO RESULTADOS DE TESTE TERMINAL PORTAS

```
%TSTREE3,deveSalvarClienteComSucesso(br.com.ifpe.startinfo.service.ClienteServiceTest),false,1,false,2,deveSalvarClienteComSucesso(),,[engine:junit-jupiter]/[class:br.com.ifpe.startinfo.service.ClienteServiceTest]/[method:deveSalvarClienteComSucesso()]
%TESTS 3,deveSalvarClienteComSucesso(br.com.ifpe.startinfo.service.ClienteServiceTest)
%TESTE 3,deveSalvarClienteComSucesso(br.com.ifpe.startinfo.service.ClienteServiceTest)
%RUNTIME1820
```

Test Runner for Java

```
deveSalvarClienteComSucesso()
> 32 resultados mais antigos
```

main* 0.2 Spring Boot-StartInfoApiApplication-startinfo-api- (startinfo-api-ihc) Java: Ready Rebeka Sena (há 19 horas) Ln 45, Col 43 Espaços: 4 UTF-8 CRLF () Java Go Live

Titulo: Teste de Unidade Login de Cliente

Status: **Passou**

- Criamos um teste para o formulário de login, simulando a digitação do e-mail e senha e clicando no botão "Entrar".
- O teste verificou se a função de autenticação era chamada corretamente com os dados fornecidos.
- Utilizamos **Axios Mock Adapter** para simular a resposta do backend e validar se a navegação acontecia corretamente após o login bem-sucedido.

```
54
55
56 @Test
57 void deveAutenticarUsuarioComSucesso() {
58     when(usuarioRepository.findByUsername(username:"teste@teste.com")).thenReturn(Optional.of(usuario));
59     when(passwordEncoder.matches("senha123", "senha123Criptografada")).thenReturn(value:true);
60
61     UsernamePasswordAuthenticationToken authenticationToken =
62         new UsernamePasswordAuthenticationToken("teste@teste.com", "senha123");
63     when(authenticationManager.authenticate(any(type:UsernamePasswordAuthenticationToken.class)))
64         .thenReturn(authenticationToken);
65
66     Usuario usuarioAutenticado = usuarioService.authenticate(username:"teste@teste.com", password:"senha123");
67
68     assertNotNull(usuarioAutenticado);
69     assertEquals(expected:"teste@teste.com", usuarioAutenticado.getUsername());
70
71     verify(authenticationManager, times(wantedNumberOfInvocations:1)).authenticate(any(type:UsernamePasswordAuthenticationToken.class));
72     verify(usuarioRepository, times(wantedNumberOfInvocations:1)).findByUsername(username:"teste@teste.com");
73 }
74
```

PROBLEMAS 2 SAÍDA CONSOLE DE DEPURACÃO RESULTADOS DE TESTE TERMINAL PORTAS

```
%TSTREE3,deveAutenticarUsuarioComSucesso(br.com.ifpe.startinfo.service.UsuarioServiceTest),false,1,false,2,deveAutenticarUsuarioComSucesso(),,[engine:junit-jupiter]/[class:br.com.ifpe.startinfo.service.UsuarioServiceTest]/[method:deveAutenticarUsuarioComSucesso()]
%TESTS 3,deveAutenticarUsuarioComSucesso(br.com.ifpe.startinfo.service.UsuarioServiceTest)
%TESTE 3,deveAutenticarUsuarioComSucesso(br.com.ifpe.startinfo.service.UsuarioServiceTest)
%RUNTIME1889
```

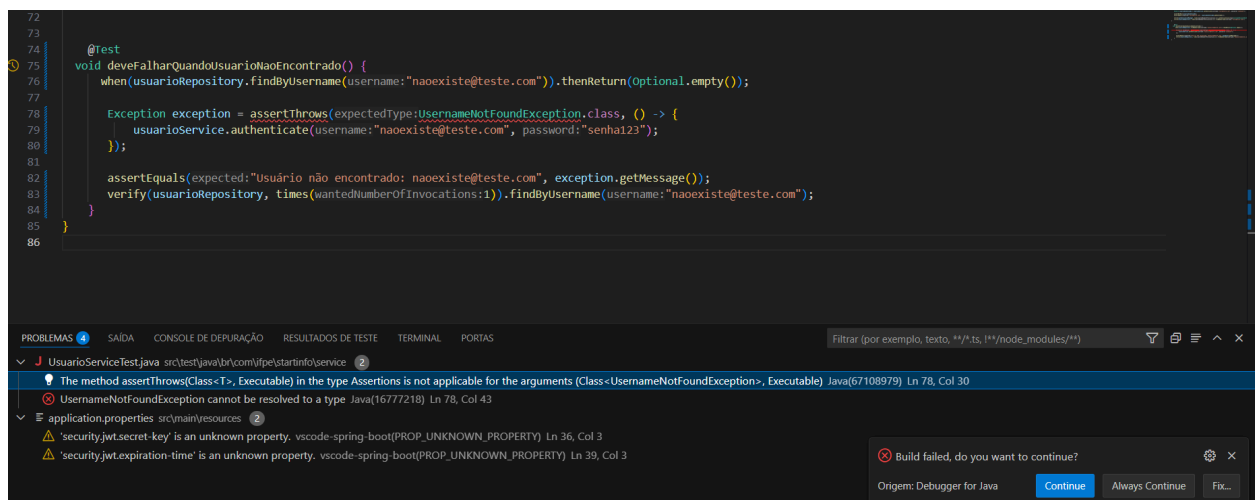
Test Runner for Java

```
deveAutenticarUsuarioComSucesso()
> 30 resultados mais antigos
```

Titulo: Teste de Unidade Login de Cliente

Status: **Não Passou**

- O teste foi similar ao anterior, mas simulamos um login com credenciais inválidas.
- A API retornou um erro 401 (não autorizado), e verificamos se a mensagem de erro apropriada era exibida ao usuário.
- O teste falhou porque a mensagem de erro não estava sendo renderizada corretamente, indicando um problema na exibição do feedback de erro no componente.



```
72
73
74 @test
75 void deveFalharQuandoUsuarioNaoEncontrado() {
76     when(usuarioRepository.findByUsername(username:"naoexiste@teste.com")).thenReturn(Optional.empty());
77
78     Exception exception = assertThrows(expectedType:UsernameNotFoundException.class, () -> {
79         usuarioService.authenticate(username:"naoexiste@teste.com", password:"senha123");
80     });
81
82     assertEquals(expected:"Usuário não encontrado: naoexiste@teste.com", exception.getMessage());
83     verify(usuarioRepository, times(wantedNumberOfInvocations:1)).findByUsername(username:"naoexiste@teste.com");
84 }
85
86
```

PROBLEMAS 4 SAÍDA CONSOLE DE DEPURACÃO RESULTADOS DE TESTE TERMINAL PORTAS

Filtrar (por exemplo, texto, **/*.ts, !**/node_modules/**)

- ✖ UsuarioServiceTest.java src/test/java/br/com/ipe/startinfo/service 2
- ⚠ The method assertThrows(Class<T>, Executable) in the type Assertions is not applicable for the arguments (Class<UsernameNotFoundException>, Executable) Java(67108979) Ln 78, Col 30
- ✖ UsernameNotFoundException cannot be resolved to a type Java(16777218) Ln 78, Col 43
- ⚠ application.properties src/main/resources 2
- ⚠ 'security.jwt.secret.key' is an unknown property. vscode-spring-boot(PROP_UNKNOWN_PROPERTY) Ln 36, Col 3
- ⚠ 'security.jwt.expiration-time' is an unknown property. vscode-spring-boot(PROP_UNKNOWN_PROPERTY) Ln 39, Col 3

Build failed, do you want to continue?

Origem: Debugger for Java Continue Always Continue Fix...

2. Teste de Integração

Descrição: O teste de integração tem como objetivo verificar a interação entre diferentes componentes de um sistema, garantindo que as partes individuais, como banco de dados, APIs e serviços, funcionem de maneira integrada e sem erros. O foco do teste é validar o fluxo de dados e o comportamento do sistema quando as diversas partes estão em comunicação.

Objetivo da Técnica: De assegurar que o teste de integração entre os módulos e/ou componentes do sistema funcione corretamente quando integrados. Isso envolve validar a comunicação entre diferentes serviços, componentes de software e bancos de dados, garantindo que as dependências externas e internas sejam atendidas e que o sistema como um todo opere conforme esperado.

Técnica: Testes baseados em cenários que simulam a interação entre diferentes partes do sistema.

Ferramentas Utilizadas: Vscod, Docker, Dbeaver, Java Spring Boots, Maven, JUnit e Mockito.

Conclusão: Após a implantação dos diretórios test.../startinfo/api/acesso/, com as respectivas Classes de Testes: UsuarioControllerIntegrationTest e ClienteControllerIntegrationTest. Além do /model/acesso/UsuarioServiceIntegrationTest inclui as dependências do Mockito e Junit no

pom.xml. Foi criado um segundo application-test.properties com a configuração default H2 para simular testes no Banco de Dados sem ter que interferir no projeto real.

Ao executar o comando de testes - **mvn test -X** (teste completo e detalhado), constatamos os seguintes erros:

Início do Teste:

```
PS C:\Users\Ricardo Santos\Desktop\yapoatam\startinfo-api-ihc> mvn validate
[INFO] Scanning for projects...
[INFO] -----< br.com.ifpe:startinfo-api >-----
[INFO] Building startinfo-api 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.757 s
[INFO] Finished at: 2025-02-15T10:22:19-03:00
[INFO] -----
PS C:\Users\Ricardo Santos\Desktop\yapoatam\startinfo-api-ihc> mvn test
[INFO] Scanning for projects...
[INFO] -----< br.com.ifpe:startinfo-api >-----
[INFO] Building startinfo-api 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] -----
[INFO] --- resources:3.3.1:resources (default-resources) @ startinfo-api ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO] Copying 4 resources from src\main\resources to target\classes
[INFO] -----
[INFO] --- compiler:3.13.0:compile (default-compile) @ startinfo-api ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 31 source files with javac [debug parameters release 17] to target\classes
[INFO] -----
[INFO] --- resources:3.3.1:testResources (default-testResources) @ startinfo-api ---
[INFO] skip non existing resourceDirectory C:\Users\Ricardo Santos\Desktop\yapoatam\startinfo-api-ihc\src\test\resources
[INFO] -----
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ startinfo-api ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 6 source files with javac [debug parameters release 17] to target\test-classes
[INFO] -----
[INFO] --- surefire:3.5.2:test (default-test) @ startinfo-api ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running br.com.ifpe.startinfo.api.acesso.UsuarioControllerIntegrationTest
```

Final do Teste:

```
[ERROR] Failures:
[ERROR] UsuarioControllerIntegrationTest.testCreateUsuario:44 Status expected:<201> but was:<404>
[ERROR] ClienteControllerIntegrationTest.testCreateCliente:44 Status expected:<201> but was:<400>
[INFO] -----
[ERROR] Tests run: 6, Failures: 2, Errors: 0, Skipped: 0
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 27.841 s
```

3. Teste de Sistema

Descrição: O teste de sistema avalia o sistema completo em um ambiente que simula o uso real, verificando se ele atende aos requisitos funcionais e não funcionais.

Objetivo da Técnica: Validar o comportamento geral do sistema e garantir que ele funcione conforme as expectativas dos usuários finais.

Técnica: Testes manuais e automatizados em um ambiente controlado, cobrindo todos os aspectos do sistema.

Ferramentas Utilizadas: Junit

CrITÉrios de Êxito/Conclusão: O sistema deve passar em todos os testes funcionais e não funcionais, sem apresentar falhas críticas ou erros.

3.1 Testes Exploratórios

Descrição: É uma abordagem de teste em que os responsáveis pelos testes, ou testers, embarcam num processo de investigação e descoberta para testar um produto de forma eficaz.

Título: Teste Exploratório

Erro no envio de Email devido a falta do login e senha do email “spring.mail.”

No arquivo application.properties, o usuário tentava realizar o cadastro porém éramos informado de erro em usuário e senha e descobrimos que estava faltando o email do projeto e senha de app para que o arquivo pudesse realizar o envio de email de confirmação de cadastro.

Título: Teste Exploratório

Erro na validação do backend “foneCelular”

Na página de cadastro, o usuário tentava realizar o cadastro com o número de celular porém ocorria erro pois no back-end estava solicitando que fosse recebido exatos 15 caracteres onde o número poderia ser até 10 caracteres no mínimo “81955552222” porém também poderia ser 15 caracteres no máximo “(81)9 8888-2222” realizando então a alteração da quantidade solicitada no back-end, após a alteração o cadastro conseguiu ser realizado com êxito.

8. Artefatos de Entrada

Gerador de CPF e massa de dados fictícios para cadastro.

9. Artefatos de Saída

- Relatórios de testes e evidências de execução.

10. Fluxos de Trabalho de Teste

- Execução de testes funcionais e não funcionais.
- Registro e rastreamento de defeitos.
- Validação contínua de correções aplicadas.
- Sessões de feedback com usuários finais.

11. Infraestrutura Necessária

Recursos		
Hardware	Quantidade	Descrição
Computadores pessoal	3	Full stack
Computadores do Laboratórios	3	Full stack
Software	Quantidade	Descrição
JavaScript	3	Frontend
React	3	Frontend
Spring	3	Backend
Postman	3	Backend
Docker	3	Backend

Postgres	3	Backend
Dbeaver	3	Backend

12. Responsabilidades da Equipe

Recursos Humanos		
Papel	Recursos Mínimos	Responsabilidades Específicas
Rebeka	Dev/Qa	Planejamento e execução de testes
Ricardo	Dev/Qa	Planejamento e execução de testes
Wendel	Dev/Qa	Planejamento e execução de testes

13. Componentes de Testes de Integração

- Comunicação entre frontend e backend via API REST.
- Integração com banco de dados PostgreSQL.
- Testes de autenticação e autorização.
- Validação do fluxo de dados entre camadas do sistema.

14. Gerenciamento de Comunicação

14.1 Resolução de Conflitos

A resolução de conflitos no projeto observa a hierarquia abaixo, sendo que deve-se procurar sempre a resolução entre as partes diretamente envolvidas no conflito (1ª instância). Quando não solucionado, o conflito é escalado para as instâncias superiores.

Conflito	1ª Instância	2ª Instância	Última Instância
Definição de escopo	Resolvido entre as partes		
Alteração em tutoriais		Resolvido pelo gerente.	

14.2 Relatórios

Tipo de Relatório	Objetivo	Destinatário	Periodicidade
Relatório de Acompanhamento de Serviços	Relato do desempenho de prazo, riscos e das pendências do projeto de teste/serviços.	Gerente de Projeto	Quinzenal, entregue até os dias: 10 e 25 de cada mês.

Sumário de Teste	Avaliação dos resultados da fase de teste.	Gerente de Projeto	de Mensal, entregue até o dia 14 de cada mês.
------------------	--	--------------------	---

14.3 Reuniões

- Reuniões assíncronas, conversas via WhatsApp e encontros presenciais

15. Matriz de Comunicação

Nome	Rebeka Sena	Papel	Dev/QA
Telefone		E-mail	rcncs@discente.ifpe.edu.br
Responsabilidades	Dev/Qa		
Área	QA	Responsável pela Área	Equipe

Nome	Ricardo Santos	Papel	Dev/QA
Telefone		E-mail	rjss4@discente.ifpe.edu.br
Responsabilidades	Gerente/Dev/Qa		
Área	Gerência	Responsável pela Área	Equipe

Nome	Wendel Lira	Papel	Dev/QA
Telefone		E-mail	wel@discente.ifpe.edu.br
Responsabilidades	Dev/Qa		
Área	Desenvolvimento	Responsável pela Área	Equipe

16. Cronograma

17. **Planejamento:** Janeiro 2025.

18. **Execução dos testes:** Fevereiro 2025.

19. Marcos

- Início do planejamento do projeto (28/11/2024).
- desenvolvimento do projeto (05/12/2024).

- desenvolvimento do projeto (12/12/2024).
- desenvolvimento do projeto (26/12/2024).
- Cronograma e Definição de testes (17/01/2025).
- Mapeamento de teste (20/01/2025).
- Execução dos testes (24/01 a 14/02).
- entrega final do projeto e apresentação IHC (13/02/2025).
- entrega final do projeto e apresentação Teste (14/02/2025).

20. Riscos

- 21. Problemas de usabilidade para idosos.
- 22. Falhas na integração backend/frontend.

23. Anexos

Teste - CRONOGRAMA 2025			
DATA(S)	ATIVIDADES	Responsável(eis)	Status
1/17/2025	Cronograma e Definição dos Tipos de Testes do Projeto.	Grupo	
1/24/2025	Testes Unitarios.	Grupo	
1/31/2025	Testes de Integração.	Grupo	
2/7/2025	Testes de Sistema.	Grupo	
2/14/2025	Entrega do Relatório de Teste E Apresentação Final.	Grupo	
FERIADO(S)			
1/15/2025	PADROEIRO DA CIDADE DE JABOATÃO DOS GUARARAPES		
2/7/2025	Apresentação Parcial		
2/14/2025	Apresentação Final		
	Segunda Chamada		
	AVALIAÇÃO FINAL		

PROPOSTA:	GRUPO:	DATA:
Website dedicado à Inclusão Digital dos Idosos, com Foco em Educação Digital.	REBEKA, RICARDO E WENDEL	28 / NOVEMBRO / 2024

<p>OBJETIVO(S):</p> <p>Promover a inclusão digital dos idosos por meio de tutoriais simples e intuitivos, visando integrar aqueles que ainda não têm familiaridade com a tecnologia ao novo universo digital.</p> <p>Além de explicar o que são, também ensinaremos como utilizar essas ferramentas. Por exemplo: PIX, E-mail e Redes Sociais, Etc.</p>	<p>OBJETO(S):</p> <p>Website Educativo.</p>
<p>BARREIRA(S):</p> <p>Tempo e disponibilidade para desenvolver.</p> <p>DESAFIO(S):</p> <p>Dificuldade Técnica; Organização do Projeto; Comunicação entre o Grupo; Aprendizagem Externa de Conhecimentos Técnicos.</p>	<div><p>TEMA:</p><p>Letramento e Inclusão Digital de Idosos.</p><p>ASSUNTO:</p><p>Inclusão digital e acessibilidade para idosos no uso de tecnologias.</p></div> <div><p>TECNOLOGIA(S):</p><p>Cronograma, Figma, React, PostgreSQL, Java e Spring Boot, Postman e Dbeaver.</p><p>DISCIPLINAS RELACIONADAS:</p><p>Ética e Cidadania, Desenvolvimento Web 2 e 3, Banco de Dados e Interação Humano Computador.</p></div>

24. Referências

<https://stackoverflow.com/questions>

<https://www.objective.com.br/insights/teste-de-integracao/>

<https://www.zaptest.com/pt-br/o-que-sao-testes-de-sistema-um-mergulho-profundo-nas-abordagens-tipos-ferramentas-dicas-e-truques-e-muito-mais>

<https://www.devmedia.com.br/testes-exploratorios-teoria-e-pratica/22791>

<https://dayvsonlima.medium.com/entenda-de-uma-vez-por-todas-o-que-s%C3%A3o-testes-unit%C3%A1rios-para-que-servem-e-como-faz%C3%AA-los-2a6f645bab3>

25. Aprovações

Aprovações		
Participante	Assinatura	Data
WENDEL	Wendel Lira	14/02/2025
RICARDO	Ricardo Santos	14/02/2025
REBEKA	Rebeka Sena	14/02/2025

5. Cronograma de atividades

Contém uma descrição de marcos importantes (*milestones*) das atividades (incluindo as datas de início e fim da atividade). Apenas marcos relevantes devem ser listados, ou seja, aqueles que contribuirão nas atividades de testes. Por exemplo: projeto de testes, execução de testes ou avaliação de testes.

6. Documentação complementar

Apresenta-se uma relação dos documentos pertinentes ao projeto.

Itens de um Plano de Teste***Conteúdo******1. Introdução***

Contém uma identificação do projeto, descrição dos objetivos do documento, o público ao qual ele se destina e escopo do projeto a ser desenvolvido. Pode adicionalmente conter termos e abreviações usadas, além de informar como o plano deve evoluir.

2. Requisitos a serem testados

Esta seção descreve em linhas gerais o conjunto de requisitos a serem testados no projeto a ser desenvolvido, comunicando o que deve ser verificado. Exemplos de requisitos a serem testados são: desempenho, segurança, interface de usuário, controle de acesso, funcionalidades.

3. Estratégias e ferramentas de teste

Apresenta um conjunto de tipos de testes a serem realizados, respectivas técnicas empregadas e critério de finalização de teste. Além disso, é listado o conjunto de ferramentas utilizadas.

4. Equipe e infra-estrutura

Contém descrição da equipe e da infra-estrutura utilizada para o desenvolvimento das atividades de testes, incluindo: pessoal, equipamentos, software de apoio, materiais, dentre outros. Isto visa garantir uma estrutura adequada para a execução das atividades de testes previstas no plano.