



# **Documentação de um Produto de Software**

## **Projeto Tea**

**Nome dos Alunos:**

**Bruno Carvalho dos Santos – 722410014**

**Maria Clara de Santana Santos – 7225116041**

**Micael Silva Vieira Pimentel - 722414878**

**Paolla Vitória Oliveira Santos – 722414427**

### Sumário

1.	Introdução .....	2
1.1	Tema .....	2
1.2	Objetivos a serem alcançados.....	3
1.3	Escopo principal .....	3
2.	Definição do Modelo de Processo.....	4
3.	Requisitos do Sistema de Software .....	4
3.1	Requisitos Funcionais .....	4
3.2	Requisitos Não-Funcionais.....	6
4.	Projeto .....	7
4.1	Arquitetura Lógica .....	7
4.2	Arquitetura Física .....	9
5.	Protótipo de Interface.....	10
6.	Critérios de Qualidade de Software .....	4
7.	Testes.....	5

## 1. Introdução

### 1.1 Tema

Desenvolvimento de uma aplicação web capaz de auxiliar portadores do Transtorno do Espectro Autista (TEA), utilizando um assistente de IA (inteligência artificial regenerativa baseada em llms) oferecido por meio de um chatbot com atendimento humanizado, dispondo-se outras funções tais como: lista de tarefas, parte do cuidador, que gerar capaz de monitorar, receber notificações, acompanhar e receber relatórios relacionado ao seu dependente.

Para o funcionamento da aplicação será utilizado a api da google que fornece o acesso ao modelo Gemini de inteligência artificial que fica responsável por algumas sugestões e interação humanizada do chatbot, não sendo indispensável um sistema backend para realizar o processamento de informações realizando requisições a api Gemini, integrar a api as requisições advindas do frontend, gerar relatórios com base nos dados salvos nos bancos de dados relacional e não relacionais, e ficando responsável por gerar demais funções.

A solução oferece pode permitir maior adaptabilidade e auxiliar os portadores do tea e seus responsáveis, auxiliando desde do diagnostico, convivência, e situações rotineiras.

---

## 1.2 Objetivos a serem alcançados

O público-alvo deste documento inclui os desenvolvedores e testadores envolvidos no projeto, bem como os futuros usuários finais: autistas jovens e adultos, seus cuidadores, e profissionais da área relacionada (como educadores e psicólogos).

Os objetivos da solução desenvolvida são:

- **Fornecer suporte personalizado:** Oferecer um chatbot inteligente e recursos dedicados para auxiliar pessoas com autismo e seus cuidadores.
- **Facilitar a comunicação e informação:** Ajudar desde o diagnóstico, rotina diária, tratamento, alimentação e demais dúvidas com o uso de IA generativa.
- **Organização e acompanhamento da rotina:** Permitir a criação de rotinas personalizadas e o acompanhamento do progresso.
- **Identificação de padrões comportamentais:** Identificar padrões de comportamento a partir das interações e fornecer relatórios úteis para responsáveis legais.
- **Promoção da inclusão digital e acessibilidade:** Garantir que a aplicação seja acessível e intuitiva para o público neurodivergente, com opções de personalização de interface, como paleta de baixo contraste.
- **Disseminar conhecimento:** Oferecer conteúdo informativo sobre o Transtorno do Espectro Autista para o público-alvo e qualquer pessoa interessada no tema.

Os objetivos sofreram adaptações durante o percurso do projeto. Inicialmente, a equipe tinha pouco conhecimento sobre o TEA, o que dificultou a proposta inicial. Após pesquisas e entrevistas, a ideia foi reformulada para ser mais sensível e alinhada às necessidades reais. Além disso, a escolha inicial de cores vibrantes foi adaptada para uma paleta de baixo contraste para garantir maior acessibilidade visual a usuários com hipersensibilidade sensorial.

## 1.3 Escopo principal

O Projeto TEA possui um expressivo potencial de contribuição social, oferecendo uma solução acessível que demanda apenas um dispositivo com acesso à internet. O projeto visa auxiliar pessoas com autismo e seus cuidadores por meio de um chatbot inteligente e outras ferramentas, como listas de rotinas e exercícios.

Os benefícios esperados para os usuários englobam desde o apoio no momento do diagnóstico, a orientação sobre como reagir e agir durante o tratamento, o esclarecimento de dúvidas eventuais por meio do chatbot, e a contribuição significativa para a qualidade de vida de pessoas com TEA.

O escopo principal da aplicação inclui:

- **Chatbot Inteligente:** Resposta a dúvidas e orientação sobre temas relacionados ao autismo.

- 
- **Reação ao Diagnóstico:** Informações e orientações acolhedoras para lidar com o diagnóstico de autismo de forma empática e clara.
  - **Tratamento e Cuidados:** Dicas e sugestões de acompanhamento e práticas úteis para facilitar o tratamento contínuo.
  - **Alimentação:** Recomendações sobre nutrição e dicas práticas para lidar com seletividade alimentar.
  - **Gestão de Rotinas e Exercícios:** Criação, edição e exclusão de listas de rotinas e exercícios personalizados.
  - **Painel do Responsável:** Ferramentas para monitoramento do uso, estatísticas, gerenciamento de rotinas e alertas automáticos.

## 2. Definição do Modelo de Processo

O processo de desenvolvimento do software "Projeto TEA" segue uma metodologia ágil, especificamente o **SCRUM**. O desenvolvimento é dividido em sprints de duas semanas, com entregas incrementais de funcionalidades. Reuniões diárias são realizadas para discutir o progresso, identificar problemas e obstáculos. Ao final de cada sprint, são realizadas reuniões de revisão para discutir os resultados alcançados e planejar o próximo sprint.

As etapas principais do processo de desenvolvimento são:

- **Requisitos:** Definição dos requisitos do software, identificando funcionalidades, protocolos de comunicação e a interface do usuário. Esta fase inicial foi crucial para o entendimento da demanda e a reformulação da ideia original do projeto.
- **Projeto:** Definição da arquitetura lógica e física do software, incluindo os componentes do sistema e as tecnologias a serem utilizadas.
- **Implementação:** Desenvolvimento dos módulos do sistema, integração das funcionalidades e implementação das bases de dados e da API.
- **Testes:** Realização de testes de unidade, integração e sistema para garantir que o software funcione de acordo com os requisitos e especificações.
- **Manutenção:** Manutenção contínua do software, incluindo correção de bugs, adição de novas funcionalidades e melhoria de desempenho.

## 3. Requisitos do Sistema de Software

### 3.1 Requisitos Funcionais

Os requisitos funcionais detalham as funções que o sistema deve desempenhar. Eles estão alinhados com o objetivo central de fornecer suporte personalizado a pessoas com TEA através de um chatbot inteligente.

CÓDIGO	DESCRIÇÃO	PRIORIDADE
RF01	CHATBOT PERSONALIZADO POR CADA USUÁRIO	ALTA
RF02	PERMITIR TROCAS DE MENSAGENS ENTRE USUARIOS E CHATBOT	ALTA
RF03	RESPONDER DUVIDAS COM BASE EM CONHECIMENTO DISPONIVEL NA WEB	ALTA
RF04	IDENTIFICAR PADRÕES DE COMPORTAMENTO A PARTIR DAS INTERAÇÕES	ALTA
RF05	MANTER HISTORICO DE MENSAGENS E INTERAÇÕES	MÉDIA
RF06	LISTA DE EXERCICIOS PERSONALIZAVEIS	BAIXA
RF07	LISTA DE ROTINA PERSONALIZAVEIS	BAIXA
RF08	CADASTRO E PERFIS DIFERENCIADOS (USUARIO RESPONSÁVEL E USUARIO PORTADOR DE TEA)	ALTA
RF09	ENVIO AUTOMÁTICO DE RELATORIOS DE USO A RESPONSÁVEIS	ALTA
RF10	ALERTAS AUTOMATICOS PARA RESPONSÁVEIS EM CASO DE COMPORTAMENTO FORA DO PADRÃO	ALTO
RF11	AGENDA COM LEMBRETES E NOTIFICAÇÃO	MEDIA
RF12	EXPORTAÇÕES DE RELATÓRIOS EM PDF	MEDIA

RF14	CANAL DE SUPORTE TECNICO E AJUDA DENTRO DA APLICAÇÃO	MEDIA
RF15	IDENTIFICADOR ÚNICO PARA CADA USUARIO	ALTA
RF16	CAPACIDADE DE UM RESPONSÁVEL VINCULAR E GERENCIAR MÚLTIPLOS USUARIOS COM TEA	ALTA

### 3.2 Requisitos Não-Funcionais

Os requisitos não-funcionais descrevem as qualidades do sistema, como desempenho, segurança e usabilidade.

CÓDIGO	DESCRIÇÃO	PRIORIDADE
RNF01	DISONIBILIDADE DE TEMPO INTEGRAL	ALTA
RNF02	INTERFACE FOCADA EM ACESSIBILIDADE E INTUITIVIDADE PARA PORTADORES DE TEA	ALTA
RNF03	TEMPO DE RESPOSTA INFERIOR A 3SEGUNDOS	ALTA
RNF04	SUGESTÕES AUTOMÁTICAS DE ROTINAS E EXERCICIOS PELO CHATBOT	MÉDIA
RNF05	PROTEÇÃO DE DADOS CONFORME LGPD	ALTA
RNF06	COMPATIBILIDADE COM DISPOSITIVOS DIFERENTES	ALTA
RNF07	ESCALABILIDADE	MÉDIA
RNF08	BACKUP AUTOMÁTICO DAS INFORMAÇÕES	ALTA

RNF09	SUPORTE A PELO MENOS DOIS IDIOMAS (INGLÊS E PORTUGUÊS)	ALTA
RNF10	AUTENTICAÇÃO DE MÚLTIPLOS FATORES	MÉDIA
RNF11	BAIXO CONSUMO DE DADOS MÓVEIS	ALTA

## 4. Projeto

### 4.1 Arquitetura Lógica

A arquitetura lógica do Projeto TEA é dividida em camadas, com foco na comunicação e processamento eficiente de dados.

- **Front-End:** A interface do usuário é desenvolvida como uma aplicação web utilizando HTML5, CSS3 e Javascript (com possibilidade futura de React). Esta camada é responsável pela interação direta com o usuário, exibindo as informações do chatbot, listas de rotinas e exercícios, e o painel do responsável. A aplicação permite a visualização de dados e a interação com as funcionalidades.
- **Back-End:** O backend é construído com Node.js e Express.js, e utiliza Axios.js para comunicação. É responsável por processar as requisições do frontend, interagir com a API Gemini para o chatbot, e gerenciar a lógica de negócios e o acesso aos bancos de dados.
  - IA (ChatbotService.js): Lida com as requisições para a API Gemini, processando as interações do chatbot e fornecendo respostas humanizadas.
  - Autenticação e Rotas: Gerenciamento de usuários, perfis (TEA e Responsável), autenticação (JWT e Bcrypt) e autorização para diferentes funcionalidades.
- Bancos de Dados:
  - PostgreSQL: Banco de dados relacional utilizado para armazenamento de dados estruturados como login, listas de rotinas (to-do list) e listas de exercícios.

TABELA: USUARIO		
CAMPO	TIPO	DESCRIÇÃO
ID	SERIAL	IDENTIFICADOR DO USUÁRIO
NOME	TEXT	NOME DO USUÁRIO

EMAIL	TEXT	E-MAIL DO USUARIO
SENHA_HASH	TEXT	SENHA CRIPITOGRAFADA
TIPO-PERFIL	TEXT	TIPO DE PERFIL: TEA OU RESPONSIBLE
RESPONSAVEL_ID	INTERGER	ID DO RESPONSAVEL SE APLICAVEL

TABELA:TO-DO LIST		
CAMPO	TIPO	DESCRIÇÃO
ID	SERIAL	IDENTIFICADOR DA TAREFA
USER-ID	INTERGER	ID DO USUÁRIO DONO DA TAREFA
DESCRIÇÃO	TEXT	DESCRIÇÃO DA TAREFA
CONCLUÍDO	BOOLEAN	INDICA SE A TAREFA FOI CONCLUÍDA
DATA	DATE	DATA DE CRIAÇÃO OU EXECUÇÃO PLANEJADA

TABELA: EXERCICIOS		
CAMPO	TIPO	DESCRIÇÃO
ID	SERIAL	IDENTIFICADOR DO EXERCICIO
USER-ID	INTERGER	ID DO USUÁRIO DONO DO EXERCICIO
TITULO	TEXT	NOME OU TITULO DO EXERCICIO
DESCRIÇÃO	TEXT	DESCRIÇÃO DO EXERCICIO
SUGERIDO-BOT	BOOLEAN	INDICA SE FOI SUGERIDO PELO CHATBOT
DATA-CRIAÇÃO	TIMESTAMP	DATA DA CRIAÇÃO

- MongoDB: Banco de dados não relacional utilizado para armazenar as interações do chatbot, permitindo maior flexibilidade e escalabilidade para dados não estruturados.

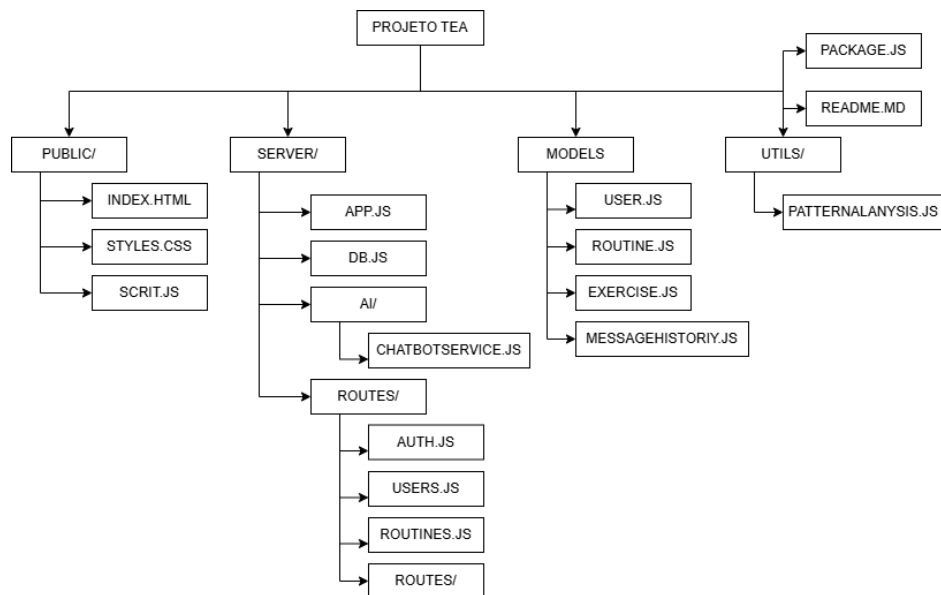
COLEÇÃO: MENSAGENS DO CHATBOT		
CAMPO	TIPO	DESCRIÇÃO
_ID	OBJECTID	ID AUTOMÁTICO DO MONGODB



USER_ID	INTERGER	RELACIONA COM A TABELA USERS DO POSTGERSQL
MESSAGENS	ARRAY	LISTA DE MENSAGENS COM ORIGEM, TEXTO E TIMESTAMP

- **Fluxo de Comunicação:** O usuário acessa a interface pelo navegador. O JavaScript do frontend envia requisições para a API back-end. A API processa os dados, interage com a IA (API Gemini) se necessário, e retorna uma resposta. As ações do usuário e as interações do chatbot são armazenadas no banco de dados. O responsável pode acessar os relatórios e alertas gerados.

Arquitetura do Projeto (Diagrama Geral):



## 4.2 Arquitetura Física

A arquitetura física descreve os componentes de hardware e rede, e como eles estão dispostos. Para o "Projeto TEA", a implantação física focaria na disponibilidade e segurança da aplicação web.

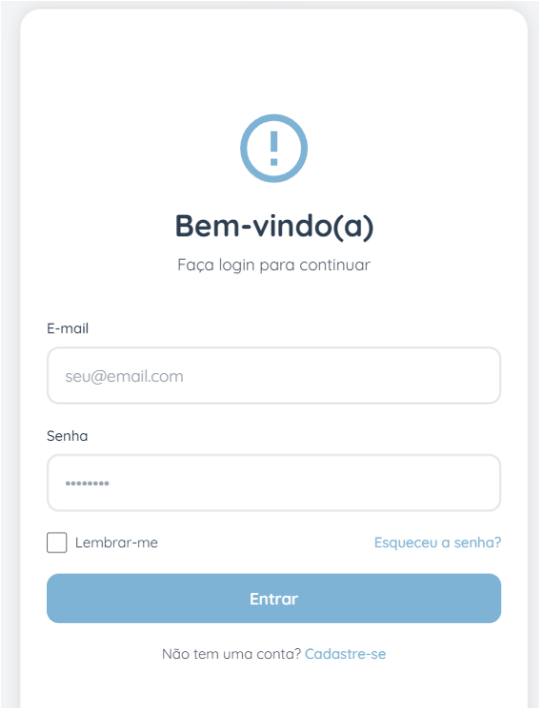
- **Hospedagem:**
  - **Front-End:** Vercel ou Netlify para distribuição de conteúdo estático.
  - **Back-End:** Render ou Railway para o servidor da aplicação.
- **Conexão com Bancos de Dados:**
  - **PostgreSQL TCP/IP:** Utilizado para autenticação, rotinas (to-do list) e exercícios.
  - **MONGODB TCP:** Utilizado para armazenar interações do chatbot.
- **Protocolos de Rede:**

- **HTTPS:** Comunicação segura com o site, protocolo seguro para envio de dados via web.
- **TLS/SSL:** Camada de segurança para o HTTPS, garante a confidencialidade e integridade dos dados durante a transmissão.
- **REST (HTTP):** Estrutura da API entre o front e back, define os métodos (GET, POST, PUT, DELETE) usados para manipular recursos via HTTP.
- **WEBSOCKET:** Comunicação em tempo real, permite troca de mensagens contínua e bidirecional entre cliente servidor.
- **JWT (JSON WEB TOKENS):** Token seguro em formato JSON, usado para manter sessões autenticadas via HTTPS.
- **Segurança:** Implementação de autenticação JWT e criptografia com Bcrypt. Proteção de dados conforme a LGPD. Backups automáticos das informações para evitar perda de dados.
- **Disponibilidade:** Utilização de serviços de hospedagem que oferecem alta disponibilidade e escalabilidade para garantir que o software esteja sempre online e performático, mesmo com o aumento de usuários.

## 5. Protótipo de Interface

As interfaces foram projetadas para serem intuitivas e focadas na acessibilidade para portadores de TEA e seus cuidadores.

- **Página de Login:** Interface inicial para acesso ao sistema, com campos de e-mail e senha, e opções de "Lembrar-me" e "Esqueceu a senha?".

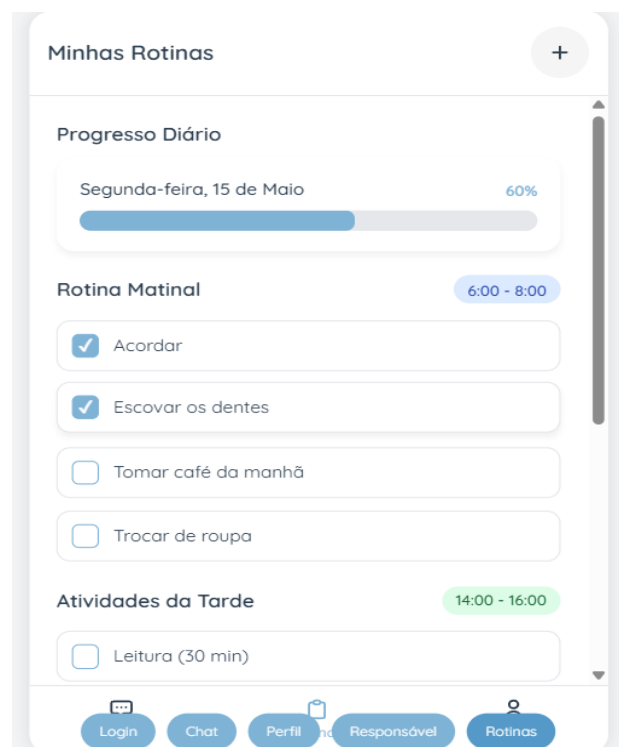


O protótipo de interface de login apresenta um design limpo e moderno. No topo, há um ícone de alerta (exclamação dentro de um círculo) em azul. Abaixo dele, o texto "Bem-vindo(a)" é exibido em uma fonte sans-serif em negrito, seguido por "Faça login para continuar" em uma fonte menor. O formulário contém dois campos de entrada: "E-mail" com o placeholder "seu@email.com" e "Senha" com caracteres ocultos por pontos. Abaixo dos campos, há uma caixa de seleção para "Lembrar-me" e um link "Esqueceu a senha?". Um botão azul com o texto "Entrar" está centralizado. Na base, há o link "Não tem uma conta? Cadastre-se".

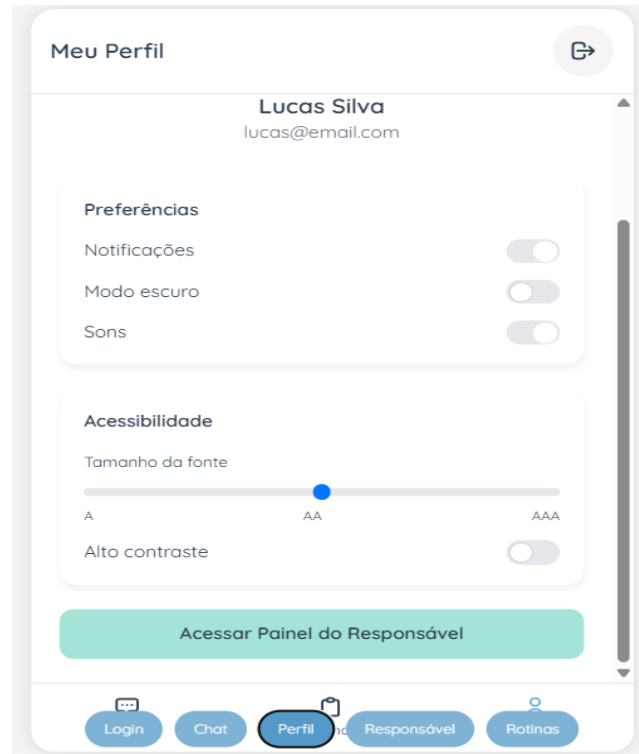
- **Chatbot (Amigo Virtual):** Tela de conversação com o chatbot, exibindo balões de mensagens do usuário e do "Amigo Virtual". Permite a interação em tempo real com o assistente de IA.



- **Minhas Rotinas:** Tela onde o usuário pode visualizar e gerenciar suas rotinas diárias e atividades. Permite marcar tarefas como concluídas e acompanhar o progresso.



- **Meu Perfil:** Tela de configurações do usuário, onde é possível ajustar preferências como notificações, modo escuro e sons. Também inclui configurações de acessibilidade, como tamanho da fonte e alto contraste, e acesso ao Painel do Responsável.



- **Painel do Responsável:** Dashboard completo para cuidadores, apresentando estatísticas de uso (rotinas concluídas, interações com chat, tempo de uso semanal), progresso semanal (rotinas diárias, exercícios de comunicação, atividades cognitivas) e opções de gerenciamento (adicionar nova rotina, personalizar atividades, gerar relatório, configurações avançadas). Exibe também alertas recentes, como "Rotina matinal não concluída" e "Baixo engajamento nas atividades".



## 6. Critérios de Qualidade de Software

Para o Projeto TEA, os seguintes critérios de qualidade são fundamentais para garantir a eficácia e a aceitação da aplicação:

- **Confiabilidade:** O sistema deve garantir que os dados de interação do usuário e os registros de rotinas sejam precisos e persistentes, sem perdas ou falhas, especialmente no histórico do chatbot e nos relatórios para os responsáveis.
- **Segurança:** É crucial que o sistema seja seguro e protegido contra ataques externos e vazamentos de dados, pois lida com informações sensíveis e pessoais dos usuários, em conformidade com a LGPD. Isso inclui a segurança na transmissão (HTTPS/TLS/SSL), armazenamento de senhas (criptografia com Bcrypt) e autenticação (JWT).
- **Desempenho:** O sistema deve ser capaz de processar as interações do chatbot e gerar relatórios em tempo hábil. O tempo de resposta do chatbot deve ser inferior a 3 segundos para garantir uma experiência fluida e satisfatória ao usuário.
- **Manutenibilidade:** O sistema, composto por diferentes componentes (frontend, backend, bancos de dados, integração com IA), deve ser fácil de manter e atualizar. Isso inclui a correção de bugs, a adição de novas funcionalidades e a melhoria contínua do desempenho, evitando problemas de compatibilidade.
- **Usabilidade:** O sistema deve ser fácil de usar para todos os usuários, incluindo autistas e seus cuidadores. A interface deve ser intuitiva, clara e acessível, permitindo que as informações e funcionalidades

---

sejam facilmente compreendidas e utilizadas. O design deve contemplar acessibilidade visual, como opções de contraste e tamanho da fonte.

## 7. Testes

### 7.1. Plano de Testes

O plano de testes para o Projeto TEA incluirá diversas abordagens para garantir a qualidade e a funcionalidade do software:

- **Teste de Integração:** Verificar se todos os módulos do sistema (frontend, backend, bancos de dados, API Gemini) estão se comunicando e funcionando corretamente juntos.
- **Teste de Unidade:** Testar individualmente cada componente ou função do código para garantir que funcionem conforme o esperado.
- **Teste de Sistema:** Avaliar o sistema completo para verificar se atende a todos os requisitos funcionais e não-funcionais especificados.
- **Teste de Usabilidade:** Avaliar a facilidade de uso e a experiência do usuário, especialmente considerando a interface focada em acessibilidade e intuitividade para portadores de TEA.
- **Teste de Desempenho:** Verificar a velocidade, escalabilidade e estabilidade do sistema sob diferentes cargas de uso, como o tempo de resposta do chatbot e o processamento de dados.
- **Teste de Segurança:** Avaliar a vulnerabilidade do sistema a ataques, garantindo a proteção de dados sensíveis conforme a LGPD e a robustez da autenticação e criptografia.
- **Teste de Regressão:** Realizar testes para garantir que novas alterações no código não introduzam bugs ou causem falhas em funcionalidades existentes.
- **Teste de Aceitação:** Envolver usuários finais (autistas, cuidadores, psicólogos) para verificar se o sistema atende às suas necessidades e expectativas do mundo real.

### 7.2. Roteiro de Testes

Este roteiro detalha etapas específicas de teste para garantir a funcionalidade do Projeto TEA.

- **Teste de Configuração:**
  - Verificar se o sistema de backend está devidamente configurado para integrar com a API Gemini.
  - Verificar se a API do sistema de backend está devidamente configurada e acessível.
  - Verificar se o front-end está devidamente configurado e se conecta corretamente ao backend.
  - Verificar a conexão com os bancos de dados PostgreSQL e MongoDB.
- **Teste de Coleta de Dados (Interações do Chatbot e Rotinas/Exercícios):**
  - Verificar se o chatbot está coletando e armazenando as interações corretamente no MongoDB.

- 
- Verificar se as rotinas e exercícios criados/editados pelos usuários e responsáveis são armazenados corretamente no PostgreSQL.
  - **Teste de Processamento de Dados:**
    - Verificar se o backend está recebendo e processando corretamente as requisições do frontend.
    - Verificar se o backend está interagindo corretamente com a API Gemini e processando as respostas da IA.
    - Verificar se os dados (interações, rotinas, exercícios) são armazenados corretamente nos respectivos bancos de dados.
    - Verificar se os dados para relatórios são processados e agregados corretamente.
  - **Teste de Geração de Relatórios e Alertas:**
    - Verificar se o front-end está recebendo os dados corretamente do sistema de backend para gerar relatórios.
    - Verificar se os relatórios gerados (em PDF) apresentam as informações corretas e completas sobre o uso e progresso do usuário.
    - Verificar se os alertas automáticos para responsáveis em caso de comportamento fora do padrão são gerados e enviados corretamente.
  - **Teste de Integração:**
    - Verificar se todos os sistemas estão integrados corretamente.
    - Verificar se a autenticação e autorização funcionam perfeitamente entre frontend, backend e bancos de dados (ex: acesso diferenciado para usuários TEA e responsáveis).
    - Verificar se os dados estão sendo transmitidos corretamente entre o frontend, backend e os serviços da API Gemini.
    - Verificar se o fluxo de dados entre os sistemas está correto.
    - Verificar se as interfaces entre os sistemas (APIs REST, WebSockets) estão funcionando corretamente.
  - **Teste de Desempenho:**
    - Verificar se o sistema está funcionando corretamente em diferentes condições de uso, como com diferentes volumes de dados ou múltiplos usuários simultâneos no chatbot.
    - Verificar se o sistema está respondendo rapidamente aos comandos do usuário.
    - Verificar se o sistema está executando as tarefas de forma eficiente e sem erros.
    - Verificar se o sistema está usando os recursos de hardware de forma eficiente.