

## HISTÓRICO DE VERSÕES

Data	Versão	Descrição	Autor	Aprovado por
08/05/2025	1.0	Criação dos DVP	EVERTON HIAN DOS SANTOS PINHEIRO	
21/05/2025	1.1	Alterações no DVP	ALLAN KARDEC DE JESUS FELIZ NAVEGANTES	

	<p><b>Documento Visão do Projeto NexSay</b></p>	
---	---	---

## Responsáveis

### 1. Equipe NexSay

**EVERTON HIAN DOS SANTOS PINHEIRO**

**ALLAN KARDEC DE JESUS FELIZ NAVEGANTES**

## 1. Objetivo

O documento tem por finalidade coletar, analisar e definir necessidades de alto nível e características gerais do sistema Nexsay. Concentra-se nas potencialidades requeridas pelos afetados e usuários-alvo, como estes requisitos serão abordados no projeto de software.

A visão do projeto documenta o ambiente geral de processos a ser desenvolvido para o sistema durante o projeto, fornecendo a todos os envolvidos uma descrição compreensível deste e de suas macro-funcionalidades.

O Documento de Visão de Projeto documenta apenas as necessidades e funcionalidades do sistema que estarão sendo atendidas no projeto de software. Quando necessário este documento pode ser atualizado durante todo o ciclo de desenvolvimento da solução.

## 2. Descrição do Projeto

O projeto consiste na criação de um sistema de mensagens instantâneas. O sistema será desenvolvido com backend em Python e frontend em React.js, que se comunicarão exclusivamente por meio de API RESTful e WebSocket. O objetivo principal é permitir a troca de mensagens de texto entre usuários cadastrados, com autenticação segura via token, histórico de conversas e gerenciamento de contatos.

### 3. Justificativa e Escopo da solução

O sistema servirá como base acadêmica e prática para aplicações reais, demonstrando o uso de boas práticas de engenharia de software, segurança da informação, arquitetura RESTful, separação de responsabilidades e integração entre frontend e backend em tempo real.

#### Escopo incluído:

- **Escopo incluído:**
  - Cadastro e login de usuários com autenticação JWT e 2FA
  - Envio e recebimento de mensagens de texto
  - Comunicação em tempo real via WebSocket
  - Histórico completo de conversas com criptografia de ponta a ponta
  - Gerenciamento de contatos (adicionar, bloquear, excluir)
  - Sistema de logs para todas as ações sensíveis do sistema (auditoria, segurança, mensagens)

### 4. Envolvimento

#### 5.1. Abrangência

O projeto é de uso acadêmico e restrito a um ambiente de demonstração. Será utilizado em navegadores modernos, com foco em usuários simulados (alunos, avaliadores e desenvolvedores).

Apesar disso, o sistema é desenvolvido com estrutura modular e escalável, permitindo futuras expansões ou uso como base para projetos maiores.

As áreas diretamente envolvidas no ciclo de vida e operação do sistema são:

- Desenvolvimento de Software (mobile e backend);

- Infraestrutura e Segurança;
- Produto e UX;
- Marketing e Comunicação;
- Suporte e Atendimento ao Cliente;
- Equipe Executiva e Estratégica.

### 5.2. Equipe de Desenvolvimento Nexsay

<b>Descrição</b>	<ul style="list-style-type: none"> <li>Responsável pela identificação dos requisitos do software e pelo desenvolvimento dos modelos estáticos e dinâmicos do projeto</li> </ul>
<b>Papel no desenvolvimento</b>	<ul style="list-style-type: none"> <li>Identificar e descrever as necessidades do usuário, especificando as funcionalidades do software que irão atendê-las.</li> <li>Levantar os requisitos funcionais e não funcionais do projeto.</li> <li>Definir o que irá interagir com o sistema.</li> <li>Identificar dentro da visão lógica do sistema, a melhor forma de acomodar as necessidades do usuário, e o impacto da solução adotada sobre os requisitos do sistema.</li> <li>Em suma, responsável pela geração de um produto que atenda aos requisitos que foram identificados junto ao usuário.</li> </ul>
<b>Insumos ao projeto de software</b>	<ul style="list-style-type: none"> <li>Documento de visão e demais documentos de requisitos de software.</li> <li>Aplicação desenvolvida de acordo com o especificado (artefatos do projeto).</li> </ul>

### 5.3. Papel dos Atores do Sistema

<b>Ator</b>	<b>Administrador</b>
<b>Descrição</b>	Responsável pelo gerenciamento e monitoramento da plataforma
<b>Papel</b>	Controlar servidores, moderar conteúdo,

	bloquear contas suspeitas
<b>Insumos ao sistema</b>	Logs, relatórios, alertas de segurança
<b>Representante</b>	Equipe interna de infraestrutura e segurança

<b>Ator</b>	<b>Usuário Final</b>
<b>Descrição</b>	Pessoa física que utiliza o sistema para comunicação pessoal
<b>Papel</b>	Enviar/receber mensagens, imagens e documentos
<b>Insumos ao sistema</b>	Dados de contato, mensagens, arquivos
<b>Representante</b>	Qualquer pessoa que instale o app

## 6. Principais Necessidades e Funcionalidades

### 6.1. Requisitos Funcionais

**Tabela 1 – Gerenciamento de Conversas e Mensagens**

ID	Descrição da Funcionalidade	Atores Envolvidos
RF01	Iniciar nova conversa	Usuário
RF02	Enviar mensagem de texto	Usuário
RF03	Visualizar mensagens novas	Usuário
RF04	Visualizar mensagens antigas	Usuário
RF05	Apagar mensagem para si	Usuário

**Tabela 2 – Gerenciamento de Contatos e Perfil**

ID	Descrição da Funcionalidade	Envolvidos
RF06	Adicionar novo contato	Usuário
RF07	Bloquear contato	Usuário
RF08	Ver lista de contas	Usuário
RF09	Ver informações básicas de contatos	Usuário

**Tabela 3 – Controle de Usuários e Sessões**

ID	Descrição da Funcionalidade	Envolvidos
RF10	Cadastrar Usuários	Usuário
RF11	Autenticar Usuários	Usuário
RF12	Recuperação de senha e redefinição segura	Usuário
RF13	Excluir conta	Usuário

## 6.2 Requisitos Não Funcionais

ID	Descrição do Requisito Não Funcional
RNF01	Comunicação segura via HTTPS
RNF02	Criptografia em repouso (AES-256)
RNF03	Registro de logs de mensagens
RNF04	Monitoramento de falhas de autenticação
RNF05	Resposta em tempo real
RNF06	Logs de auditoria

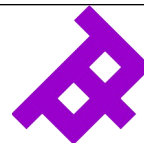
## 7. Restrições Tecnológicas

- **Frontend:** React.js
- **Backend:** API restful
- **Banco de dados:** PostgreSQL
- **Segurança:** HTTPS obrigatório, 2FA com token temporário via e-mail, hash de senhas com bcrypt
- **Logs:** Implementação de sistema de logs no banco
- **Infraestrutura de teste:** Localhost e simulação com usuários fictícios

## 8. Critérios de Aceitação do Sistema

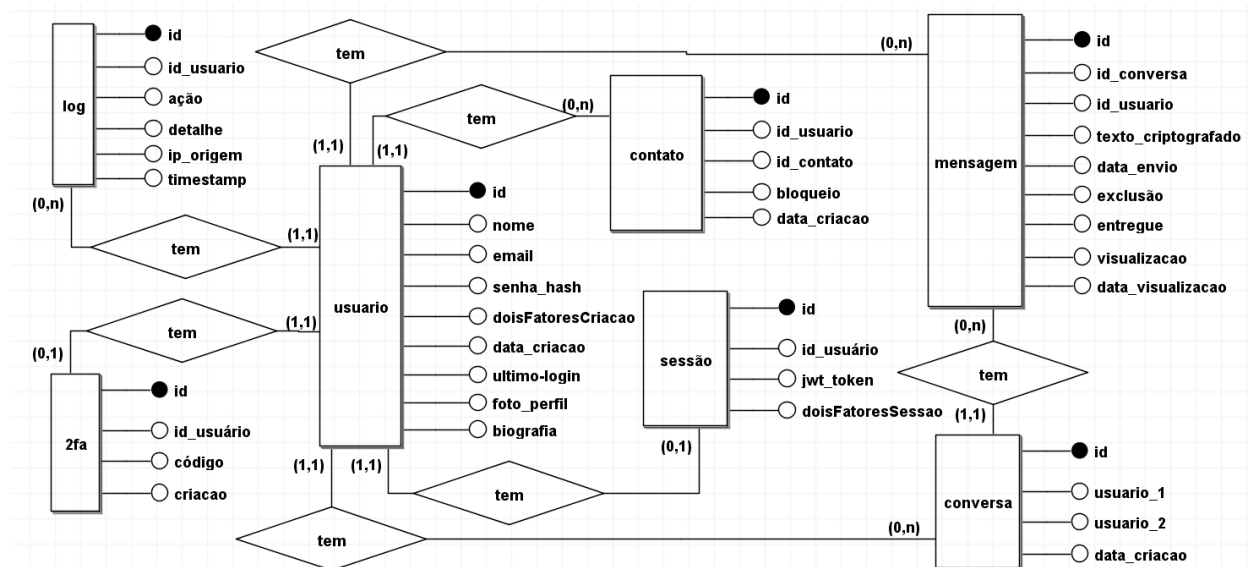
- O sistema deve permitir a criação e login de usuários com 2FA funcional
- A troca de mensagens deve ocorrer em tempo real com histórico criptografado armazenado corretamente
- Logs devem registrar toda ação importante com timestamp, IP e ID do usuário
- O sistema deve ser acessível em dispositivos móveis e navegadores modernos
- O sistema será aceito após demonstração funcional para o professor e colegas, com validação dos casos de uso e revisão de código
- A documentação do projeto (modelo de dados, estrutura de pastas, APIs) deverá ser entregue junto com o sistema





## 9. Banco de dados

### 9.1 API



#### TABELA: usuarios

##### Finalidade:

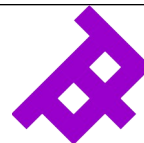
Armazena os dados essenciais dos usuários cadastrados na plataforma para permitir autenticação, identificação e funcionalidades como 2FA.

##### Atributos:

- **id** (UUID) – Identificador único de cada usuário. Chave primária.
- **nome** (TEXT) – Nome do usuário.
- **email** (TEXT) – Endereço de e-mail único para login. Obrigatório.
- **senha\_hash** (TEXT) – Senha do usuário armazenada de forma criptografada (hash seguro).
- **dois\_fatores\_ativo** (BOOLEAN) – Define se o 2FA está ativado para o usuário. Padrão: TRUE.
- **data\_criacao** (TIMESTAMP) – Data de criação da conta. Padrão: hora atual.
- **ultimo\_login** (TIMESTAMP) – Última vez que o usuário acessou a plataforma.
- **foto\_perfil** (text) – Caminho para a foto de perfil armazenada



## Documento Visão do Projeto NexSay



### TABELA: sessão

#### Finalidade:

Controla sessões autenticadas dos usuários, com tokens JWT e validação 2FA para ações sensíveis.

#### Atributos:

- **id (UUID)** – Identificador da sessão. Chave primária.
- **id\_usuario (UUID)** – Referência ao usuário logado.
- **jwt\_token (TEXT)** – Token JWT vinculado à sessão.
- **doisFatoresSessao (BOOLEAN)** – Marca se o segundo fator de autenticação foi completado.

### TABELA: contatos

#### Finalidade:

Relaciona usuários que se adicionaram como contatos entre si, permitindo bloqueio e controle de interações.

#### Atributos:

- **id (UUID)** – Identificador único da relação. Chave primária.
- **id\_usuario (UUID)** – Dono da lista de contatos.
- **id\_contato (UUID)** – Usuário que foi adicionado como contato.
- **bloqueio (BOOLEAN)** – Indica se o contato foi bloqueado. Padrão: FALSE.
- **data\_criacao (TIMESTAMP)** – Data de adição.
- **UNIQUE(usuario\_id, contato\_id)** – Não deve haver contato repetido.

### TABELA: conversas

#### Finalidade:

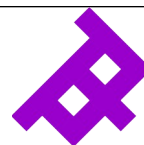
Representa uma conversa entre dois usuários (1:1), usada como agrupador de mensagens.

#### Atributos:

- **id (UUID)** – Identificador único da conversa.
- **id\_usuario1 (UUID)** – Um dos participantes.



## Documento Visão do Projeto NexSay



- **id\_usuario2** (UUID) – Outro participante.
- **data\_criacao** (TIMESTAMP) – Data/hora da criação da conversa.
- **UNIQUE**(usuario1\_id, usuario2\_id) – Garante que não existam conversas duplicadas (os ids são ordenados).

### TABELA: mensagens

#### Finalidade:

Armazena o histórico de mensagens trocadas entre usuários em conversas 1:1, com suporte a criptografia, exclusão lógica e status.

#### Atributos:

- **id** (UUID) – Identificador da mensagem.
- **id\_conversa** (UUID) – Conversa à qual a mensagem pertence.
- **id\_usuario** (UUID) – Usuário que enviou a mensagem.
- **texto\_criptografado** (TEXT) – Mensagem criptografada de ponta a ponta.
- **data\_envio** (TIMESTAMP) – Data/hora de envio.
- **visualizacao** (TIMESTAMP) – Quando a mensagem foi visualizada. Pode ser NULL.
- **exclusao** (BOOLEAN) – Se o remetente apagou a mensagem no próprio app.
- **entregue** (BOOLEAN) – Marca se o backend enviou via WebSocket.
- **data\_visualizacao** (TIMESTAMP) – Quando a mensagem foi lida

### TABELA: logs

#### Finalidade:

Auditoria das ações sensíveis do sistema, como login, envio de mensagem, bloqueio, alteração de senha, etc.

#### Atributos:

- **id** (UUID) – Identificador do log.
- **id\_usuario** (UUID) – Quem realizou a ação.
- **acao** (TEXT) – Tipo de ação registrada (ex: "login", "enviar\_mensagem", "bloquear").

- **detalhe** (TEXT) – Informações adicionais, como ID da mensagem ou usuário afetado.
- **ip\_origem** (INET) – IP de onde veio a ação.
- **timestamp** (TIMESTAMP) – Quando ocorreu a ação.

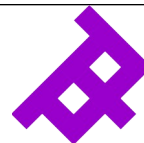
## TABELA: 2fa

### Finalidade:

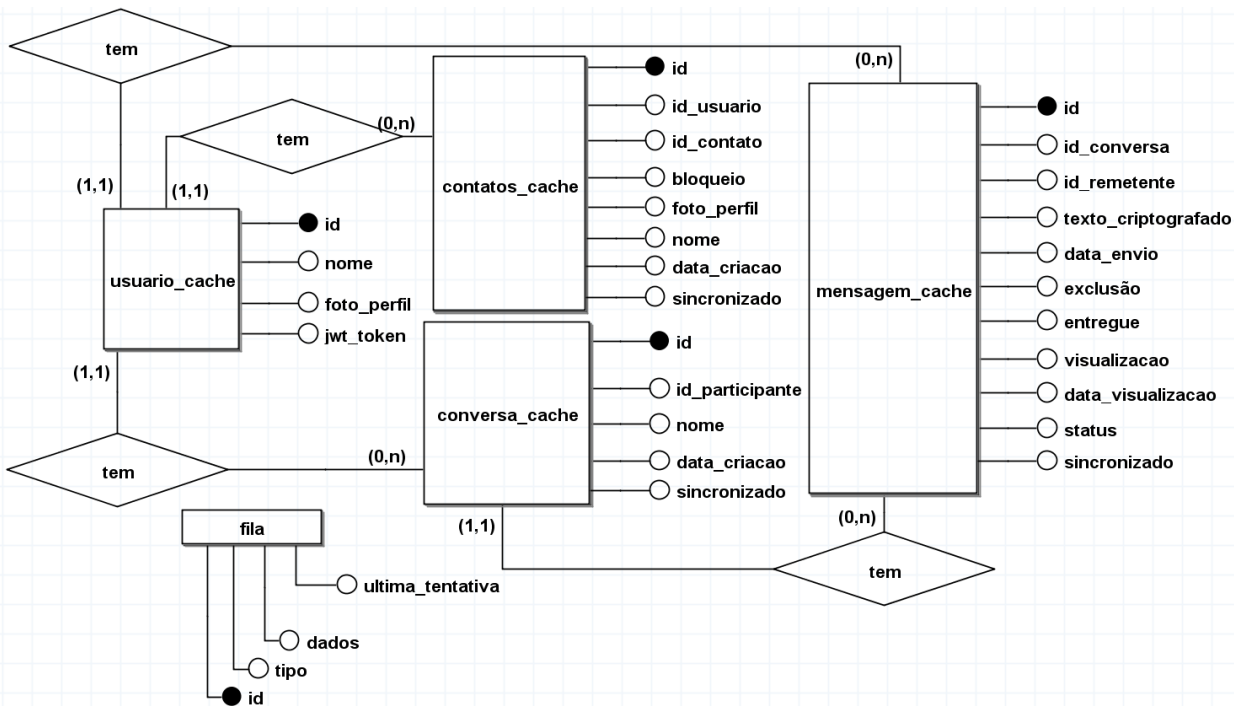
Tabela temporária para o armazenamento do código de verificação enviado no e-mail do usuário

### Atributos:

- **id** (UUID) – Identificador do log.
- **id\_usuario** (UUID) – Quem realizou a ação.
- **código** (hash) – código de verificação
- **timestamp** (TIMESTAMP) – Quando ocorreu a ação.



## 9.2 React



### TABELA: usuarios\_cache

#### Finalidade:

Armazena **dados mínimos** do usuário logado para identificação e exibição básica.

#### Atributos:

- **id** (UUID) – ID do usuário logado.
- **nome** (TEXT) – Nome do usuário.
- **foto\_perfil** (text) – foto de perfil do usuário
- **jwt\_token** (TEXT) – Token atual da sessão, armazenado **com criptografia** (ex: AES local).

### TABELA: contatos\_cache

#### Finalidade:

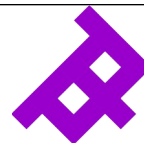
Permitir exibição rápida dos últimos contatos, mesmo offline.

#### Atributos:

- **id** (UUID) – ID da relação de contato.



## Documento Visão do Projeto NexSay



- **id\_contato** (UUID) – ID do contato.
- **nome\_contato** (TEXT) – Nome do contato.
- **foto\_perfil** (TEXT) – caminho da foto de perfil
- **bloqueio** (BOOLEAN) – Estado de bloqueio local.
- **sincronizado** (BOOLEAN) – FALSE se a alteração ainda **não foi enviada** ao servidor.

### TABELA: conversas\_cache

#### Finalidade:

Permite exibição de conversas recentes e navegação offline entre chats.

#### Atributos:

- **id** (UUID) – ID da conversa.
- **id\_participante** (UUID) – ID do outro usuário (não o logado).
- **nome\_participante** (TEXT) – Nome do outro usuário.
- **data\_criacao** (TIMESTAMP) – Data da última mensagem.
- **sincronizado** (BOOLEAN) – FALSE se for nova conversa criada offline.

### TABELA: mensagens\_cache

#### Finalidade:

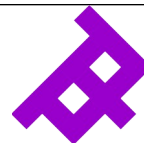
Armazena mensagens recentes criptografadas por conversa para leitura offline e envio pendente.

#### Atributos:

- **id (UUID)** – ID da mensagem (ou temporário offline).
- **id\_conversa** (UUID) – ID da conversa.
- **id\_remetente** (UUID) – Id do autor da mensagem
- **texto\_criptografado** (TEXT) – Conteúdo criptografado.
- **data\_envio** (TIMESTAMP) – Data em que a mensagem foi enviada
- **status** (TEXT) – Pode ser: enviada, pendente, erro
- **texto\_criptografado** (TEXT) – Mensagem criptografada de ponta a ponta.
- **data\_envio** (TIMESTAMP) – Data/hora de envio.



## Documento Visão do Projeto NexSay



- **visualizacao** (BOOLEAN) – Se a mensagem foi visualizada.
- **exclusao** (BOOLEAN) – Se o remetente apagou a mensagem no próprio app.
- **entregue** (BOOLEAN) – Marca se o backend enviou via WebSocket.
- **data\_visualizacao** (TIMESTAMP) – Quando a mensagem foi lida
- **sincronizado** (BOOLEAN) – FALSE se ainda não foi enviada ao servidor.

### TABELA: fila

#### Finalidade:

Fila de ações locais que precisam ser enviadas ao backend assim que a conexão for restabelecida.

#### Atributos:

- **id** (UUID) – Identificador único da tarefa.
- **tipo** (TEXT) – Ex: 'nova\_mensagem', 'bloquear\_contato', 'nova\_conversa'
- **dados** (JSON) – Dados a serem enviados.
- **ultima\_tentativa** (TIMESTAMP) – Última tentativa de envio.