



# Sistema de envío de SMS

---

Luigi Matteo Girke

# Objetivos

- Fornecer uma plataforma económica e eficiente
- Acessível a partir de qualquer dispositivo devido ao facto de estar na Web
- Mais funcionalidades: mensagens um-para-muitos, modelos de mensagens, definições de personalização do utilizador.



# Agenda

01

## **Viagem do utilizador**

Tudo que um  
utilizador pode fazer

02

## **Aprofundamento técnico**

Como funciona por  
dentro

03

## **Provas & sucessos**

Transformar  
problemas em  
soluções

# Modelo Cliente-Servidor

## Cliente (Browser):

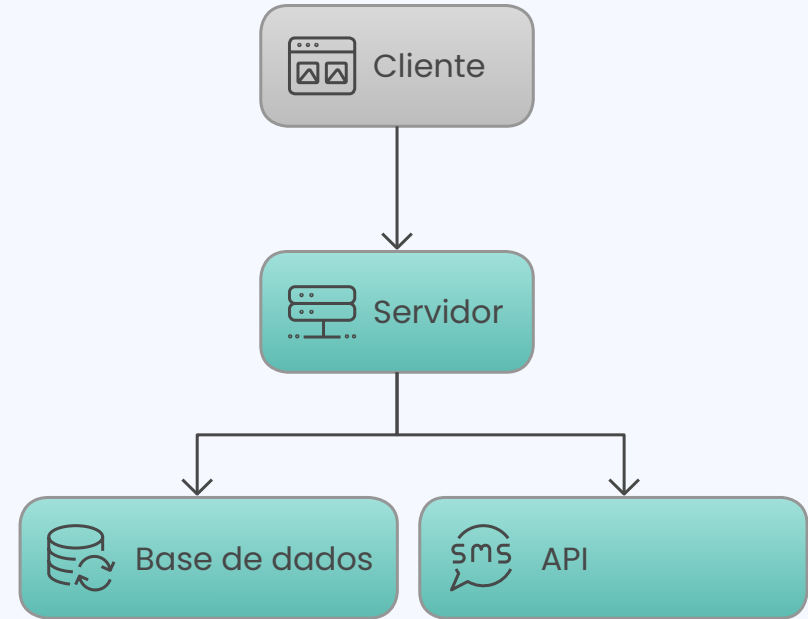
- Renderiza a IU
- Envia pedidos ao utilizador

## Servidor:

- Trata da lógica e do encaminhamento
- Liga-se aos serviços externos

## Serviços externos:

- Base de dados: Armazena utilizadores, mensagens, contactos
- API: Envia e cancela SMS





# 01

# Viagem do utilizador




---

Tudo que um utilizador pode fazer

# Iniciar sessão

- Integração do Active Directory (AD) para um início de sessão seguro
- Contas estão todas controladas no servidor AD
- Permissão baseada na existência da conta e nos grupos corretos do AD



**SMS**  
**etpzp**

## Iniciar sessão

para começar a enviar mensagens SMS

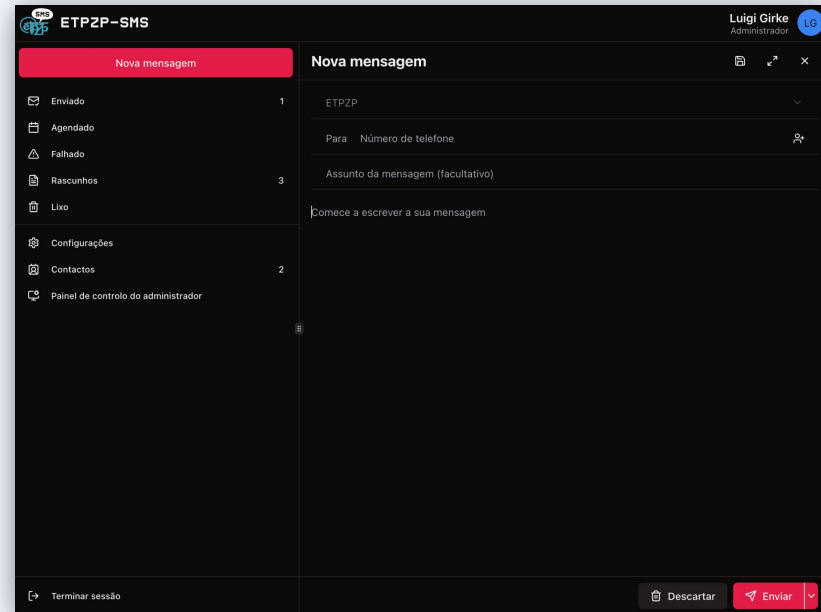
E-mail  
2812@etpzp.pt

Palavra-passe  
..... 

Iniciar sessão

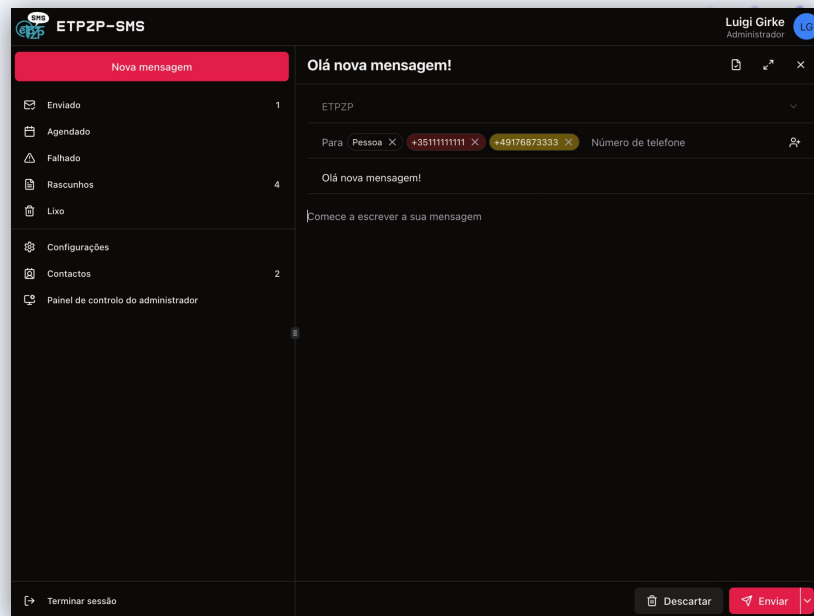
# Escrever e enviar

- Página dedicada a novas mensagens
- Campos: Remetente (fixo), Destinatários, Assunto, Conteúdo
- Enviar imediatamente ou programar
- Guardar rascunho manual + automático



# Entrada de destinatário inteligente

- Adicionar vários destinatários com o teclado ou o rato
- Sugestões em tempo real de contactos e destinatários anteriores
- Tratamento de erros para números inválidos ou duplicados



# Programação de mensagens

- Escolher data e hora futuras
- Pré-visualizar o estado da mensagem antes de a enviar
- Cancelar em qualquer altura antes da hora marcada

Selecione data e hora

Escolha quando enviar a sua mensagem.

May 2025						
Su	Mo	Tu	We	Th	Fr	Sa
				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

Hora

14

Minuto

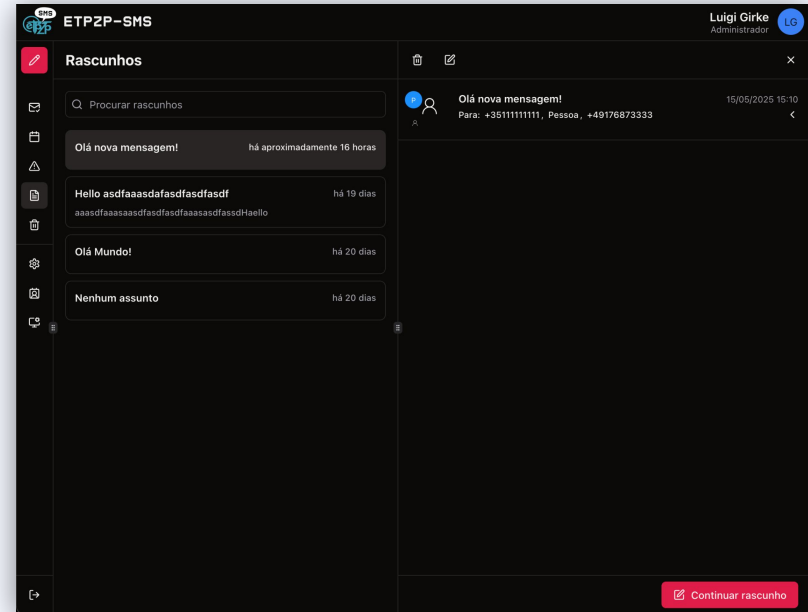
06

Cancelar

Enviar agora

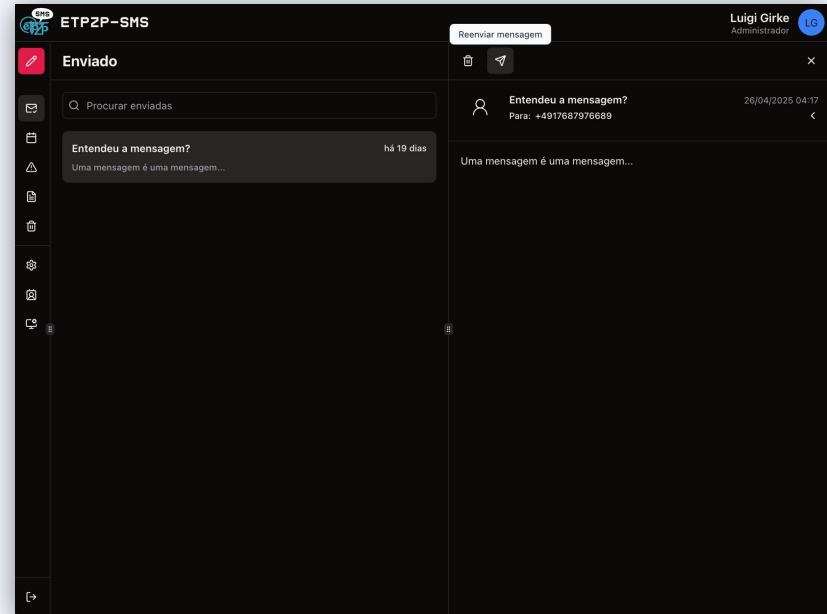
# Gestão de mensagens

- Ver: Enviadas, Rascunhos, Programadas, Falhadas, Lixo
- Cada mensagem mostra o estado de entrega e os destinatários
- Capacidade de filtrar por assunto e conteúdo



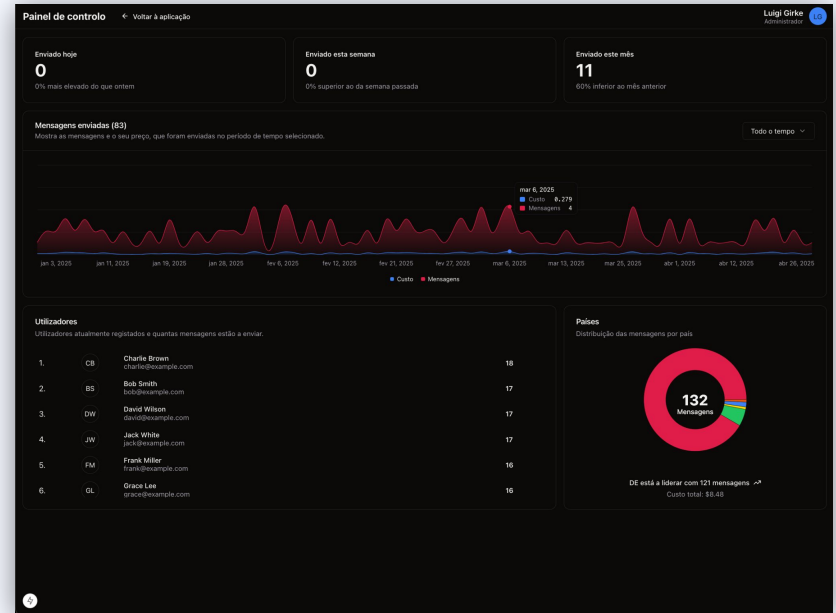
# Acções de mensagens

- Reenviar ou editar mensagens
- Mover para o lixo ou recuperar
- Cancelar mensagens programadas
- Fichas de destinatário detalhadas que apresentam mais informações quando clicadas



# Painel de controlo do administrador

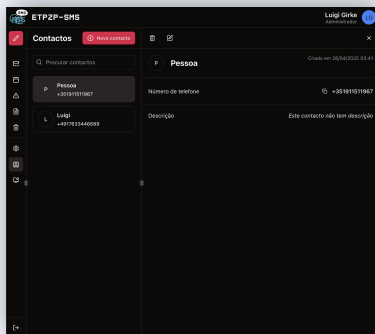
- Estatísticas por utilizador/data/país
- Gráficos de área e de pizza
- Alternar entre intervalos de datas



# Outras páginas

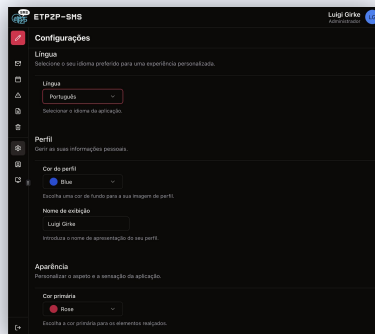
## Contactos

- Adicionar, editar, eliminar contactos
- Ver detalhes do contacto



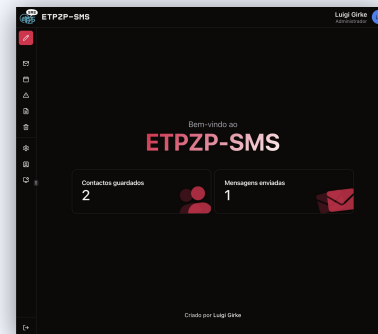
## Configurações

- Alterar temas e idioma (3 línguas)
- Atualizar as preferências do utilizador



## Início

- Hiperligações para páginas importantes



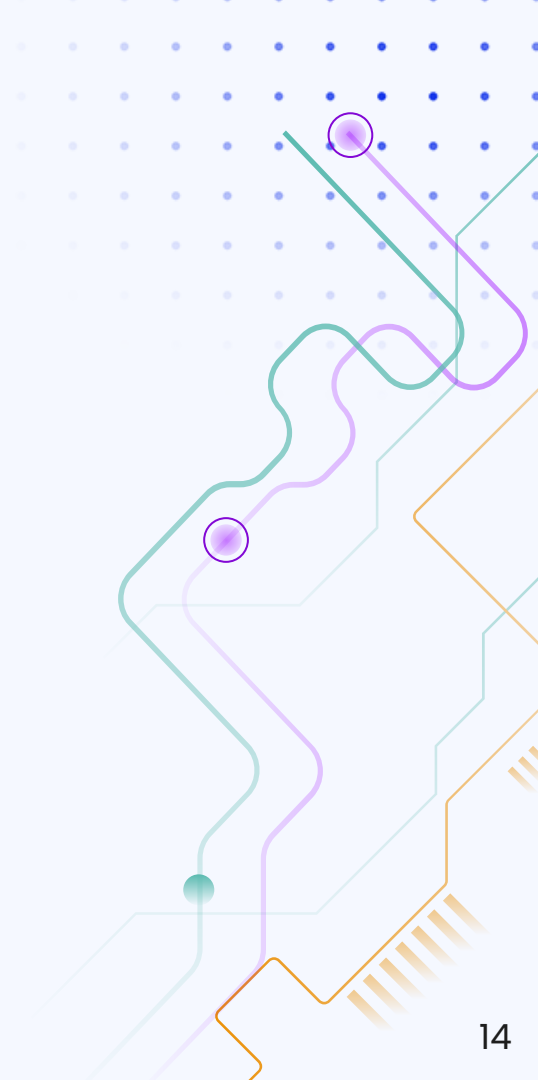


# 02

# Aprofundamento técnico

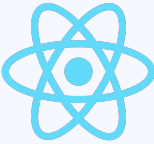
---

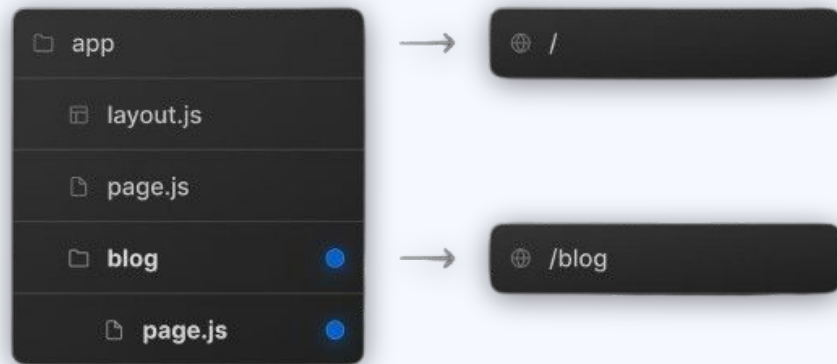
Como funciona por dentro



# Noções básicas da estrutura de Web

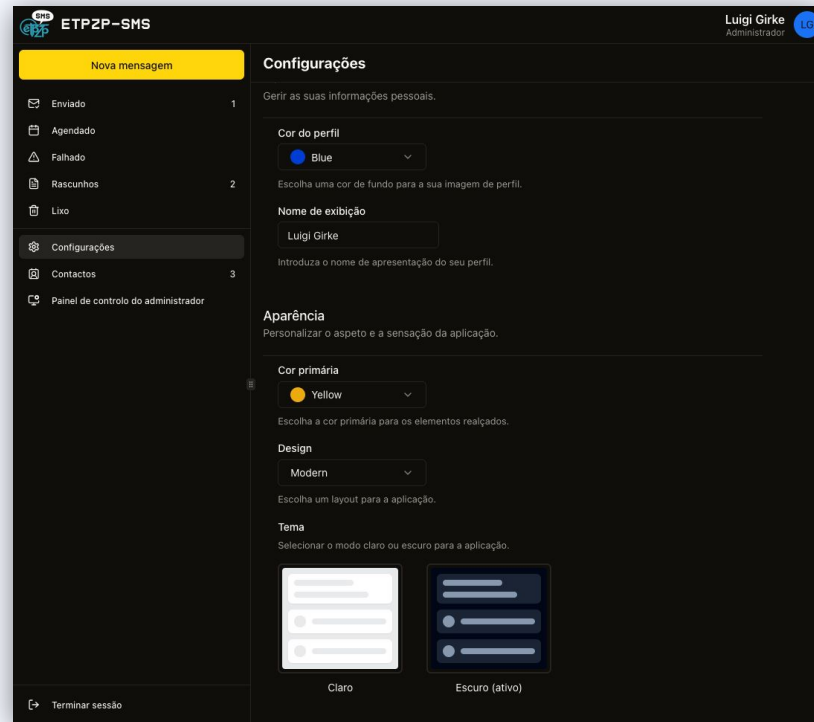
- Construído com React + Next.js
- Encaminhamento baseado em ficheiros
- Componentes de layout e servidor

NEXT.js +   
React



# IU & Thema

- ShadCN + TailwindCSS
- Painéis redimensionáveis
- Temas dinâmicos de claro/escuro + cor



# Base de dados

- PostgreSQL
- Tabelas de: Utilizadores, contactos, mensagens, destinatários

user	
id	serial
name	varchar(255) NN
email	varchar(255) NN
role	varchar(20) NN
created_at	timestamp NN
updated_at	timestamp NN
first_name	varchar(50) NN
last_name	varchar(50) NN
lang	varchar(2) NN
profile_color_id	smallint NN
display_name	varchar(50) NN
primary_color_id	smallint NN
appearance_layout	varchar(20) NN
dark_mode	boolean NN

contact	
id	serial
user_id	integer NN
name	varchar(255) NN
phone	varchar(50) NN
description	varchar(255)
created_at	timestamp NN
updated_at	timestamp NN

message	
id	serial
user_id	integer NN
sender	varchar(255)
subject	varchar(255)
body	text NN
created_at	timestamp NN
send_time	timestamp NN
sms_reference_id	bigint
status	varchar(20) NN
in_trash	boolean NN
api_error_code	smallint
api_error_details_json	text
cost	numeric(6,4)
cost_currency	varchar(10)

recipient	
id	serial
message_id	integer
phone	varchar(50) NN
index	smallint NN

# Consulta á base de dados

- `pg` para a ligação
- Função auxiliar para consultas seguras baseadas em pool

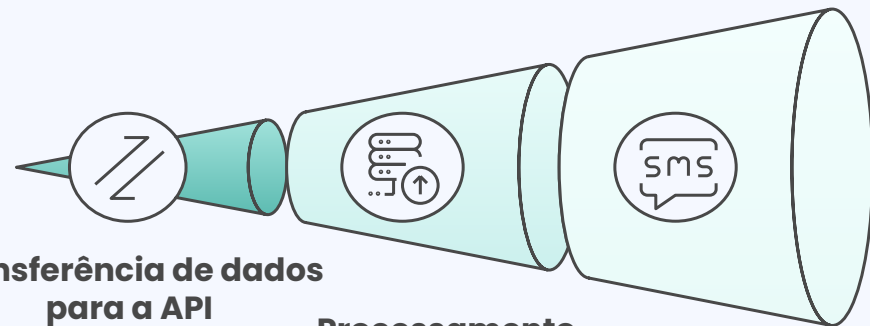


PostgreSQL

```
lib > db > ts index.ts > ...
1  import { Pool, QueryResult } from "pg";
2
3  const pool = new Pool({
4    host: process.env.POSTGRES_HOST,
5    port: Number(process.env.POSTGRES_PORT),
6    user: process.env.POSTGRES_USER,
7    password: process.env.POSTGRES_PASSWORD,
8    database: process.env.POSTGRES_DB,
9  });
10
11 async function db(query: string, params?: any[]): Promise<QueryResult> {
12   const client = await pool.connect();
13   try {
14     const res = await client.query(query, params);
15     return res;
16   } catch (err) {
17     console.error("Database query error", err);
18     throw err; // Rethrow the error for handling in the calling function
19   } finally {
20     client.release(); // Always release the client back to the pool
21   }
22 }
23
24 export default db;
```

# Integração da API SMS

- GatewayAPI
- Chamadas REST com feedback de erro
- Cancelar via API para reembolso
- Utilizou primeiro a SMSAPI



## Transferência de dados para a API

Os dados são enviados da aplicação para a API

## Processamento da API

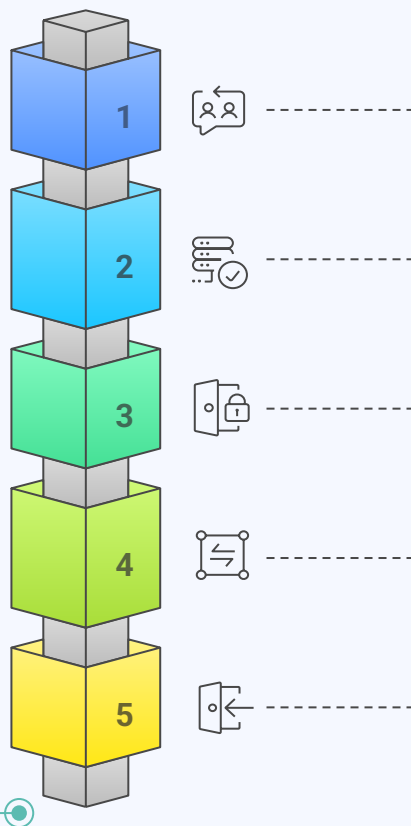
A API processa e prepara os dados

## Envio de SMS para o telefone

O SMS é entregue para o telemóvel do destinatário telemóvel

# Autenticação

- Active Directory (AD)
- Iron-session para cookies
- Detecção de administradores a partir de grupos do AD



## Utilizador submete formulário

O utilizador preenche e envia o formulário de início de sessão.

## O servidor verifica as credenciais

O servidor verifica as credenciais do utilizador em relação ao servidor AD.

## Criação de cookies

Se as credenciais forem válidas, é criado um cookie seguro.

## Verificação de cookies no middleware

Nos pedidos subsequentes, o cookie é validado.

## Acesso do utilizador concedido

Se o cookie for válido, o utilizador ganha acesso.

# Auto-hospedagem



## Docker

- Executa o servidor Node.js e o Postgres em contentores isolados
- Garante um ambiente consistente e reinícios fáceis



## Nginx

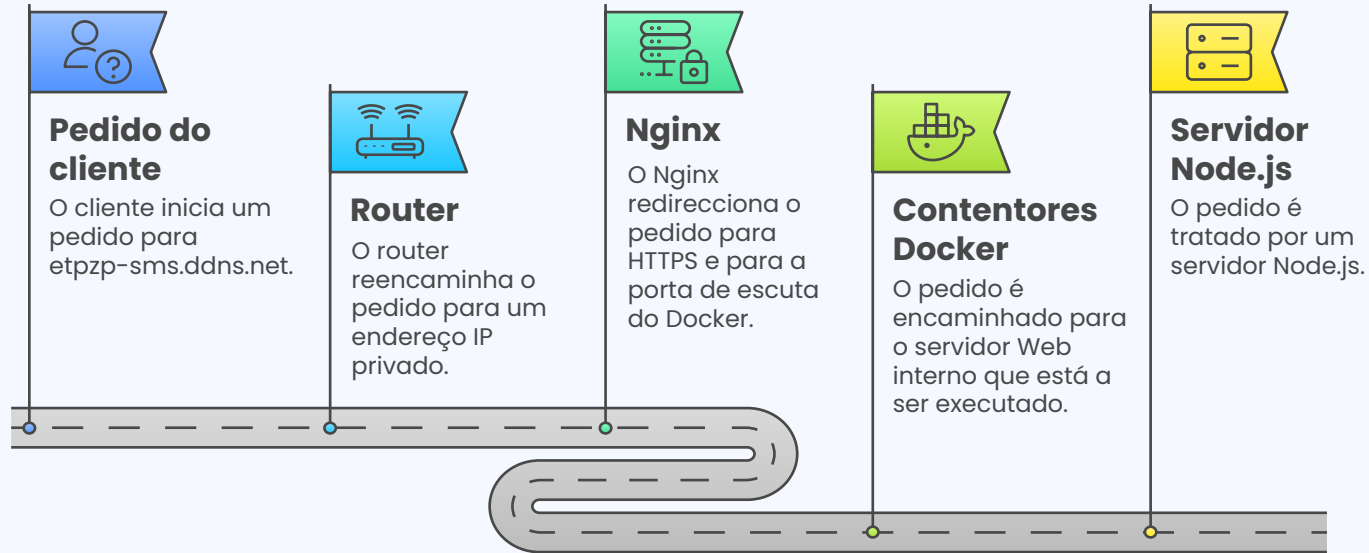
- Proxy inverso para encaminhar o tráfego para contentores Docker
- Configura o SSL e redirecciona o HTTP para HTTPS



## No-IP

- Mapeia o IP público para a rede local
- Permite o acesso remoto sem um IP estático

# Fluxo de pedidos de implantação





# 03

# Provas & sucessos

---

Transformar problemas em soluções

# Desafio para poupar rascunhos



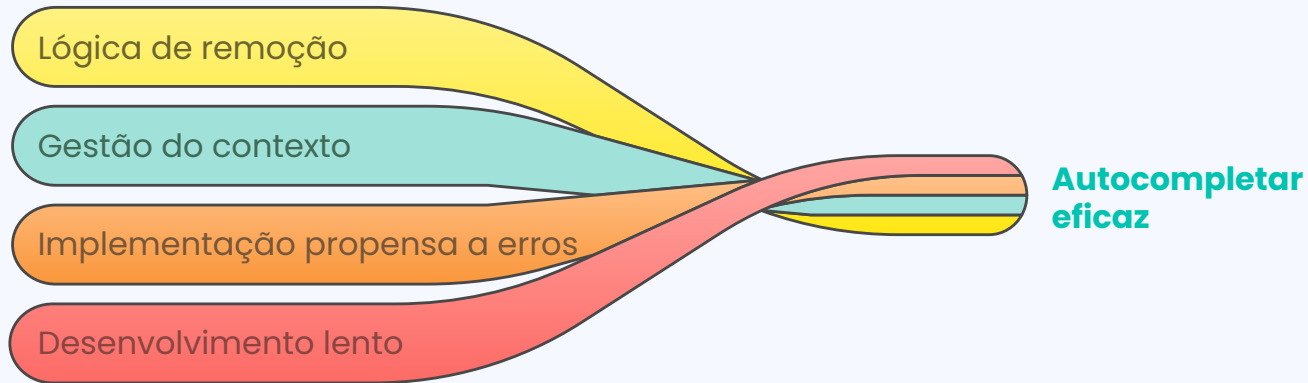
As actualizações de URL desencadearam novas renderizações, limpavam valores de formulários e fecharam popups.

A introdução de um contexto para manter o estado do rascunho nas renderizações resolveu o problema.



# Complexidade da entrada do destinatário

- A criação do menu suspenso de autocompletar/sugestões era demorada e propensa a erros
- Garantia de valores actualizados através da gestão cuidadosa do contexto e da lógica de remoção



# Desafio de Arquitetura de Configurações

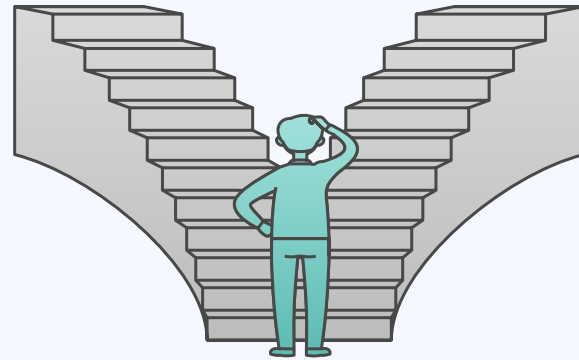
- As gravações no armazenamento local não accionam renderizações do React
- O contexto das configurações permite actualizações automáticas por item

## Um formulário para todos configurações (rejeitado)

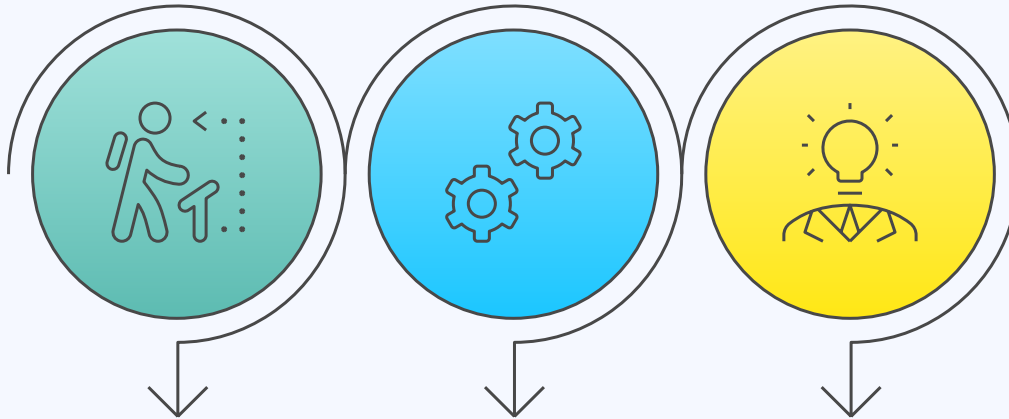
os registos de carga útil completa tornavam a validação/tratamento de erros complicada

## Formulários múltiplos por cada configuração (escolhido)

as gravações por item fornecem persistência instantânea, eliminam erros e mantêm a IU reactiva



# Resumo



## Viagem do utilizador

Esboço da experiência do cliente experiência

## Aprofundamento técnico

Tecnologias utilizadas no projeto

## Provas & sucessos

Problemas enfrentados e resoluções



# Obrigado

---

Tenho todo o gosto em responder às vossas perguntas