

Intellux News

Este projeto oferece uma plataforma web que integra autenticação, leitura de notícias, organização por favoritos e geração automática de resumos utilizando modelos de IA.

Yasmin Altino Araújo

Sumário

1	Implementação	1
1.1	Frontend	1
1.2	Backend	1
1.3	Infraestrutura	1
2	Arquitetura e Fluxos	2
2.1	Visão Geral (alto nível)	2
2.2	Fluxo de Autenticação (Google OAuth + JWT)	2
2.3	Fluxo de Favoritar	2
2.4	Fluxo de Gerar Resumo	2
3	Prints e Exemplos de Uso	3

Lista de Figuras

1	Arquitetura de alto nível do Intellux News	2
2	Login via Google	3
3	Aba Favoritos	3
4	Modal de Resumo	4
5	Aba Resumos	4
6	Responsividade do header	5
7	Responsividade do feed	5

1 Implementação

O **Intellux News** foi desenvolvido com uma arquitetura dividida entre frontend (React + Vite) e backend (NestJS), integrando autenticação via Google, gerenciamento de favoritos, geração de resumos por IA e banco de dados MySQL. O sistema foi projetado para oferecer navegação fluida, persistência completa e resumos automáticos de notícias.

1.1 Frontend

O frontend foi desenvolvido em React com TypeScript, estruturado em componentes funcionais e hooks especializados. O fluxo da aplicação é centralizado por hooks como:

- **useAuth** e **useUrlAuth**: controlam autenticação, leitura do token retornado pelo Google e armazenamento no **localStorage**.
- **useNews**, **useFavorites** e **useSummary**: realizam busca de notícias, gerenciamento de itens favoritos e solicitação/listagem de resumos.

A interface é composta por componentes reutilizáveis, incluindo cartões de notícia, cartões de resumo, barra lateral, cabeçalho e modal para geração de análises. A aplicação contém também uma página de login simples, cujo objetivo é redirecionar o usuário para o fluxo OAuth do Google.

A página principal (**FeedPage**) organiza as abas de navegação (todas, favoritos e resumos) e conecta interações do usuário aos serviços do backend. A comunicação ocorre através de um serviço Axios padronizado, responsável por enviar o JWT no cabeçalho e normalizar todos os payloads antes do envio.

1.2 Backend

O backend utiliza NestJS com o padrão Controller–Service–Module, garantindo organização e encapsulamento. O fluxo de autenticação segue OAuth do Google, seguido de geração de JWT para uso no frontend.

Os módulos principais incluem:

- **News**: registro e deduplicação de notícias.
- **Favorites**: gerenciamento dos itens favoritos pelo usuário.
- **Summaries**: envio de texto ao Ollama e persistência dos resumos.
- **User**: registro e busca de usuários.

Toda entrada passa por DTOs validados com **ValidationPipe**. A integração com IA utiliza o modelo **gemma2:2b**, acessado pelo endpoint configurado em **OLLAMA_URL**.

1.3 Infraestrutura

O sistema foi containerizado via Docker Compose, incluindo:

- **db** (MySQL)
- **ollama** (modelo Gemma)
- **backend_app**
- **frontend_app**

As variáveis de ambiente e credenciais ficam centralizadas em **backend/.env**. Essa separação facilita o desenvolvimento, a implantação e a replicação do ambiente.

2 Arquitetura e Fluxos

2.1 Visão Geral (alto nível)

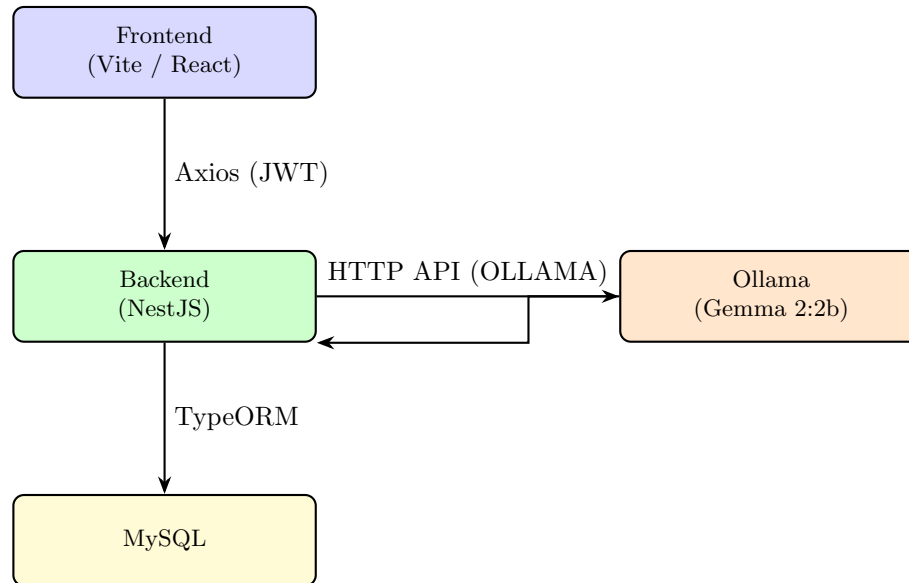


Figura 1: Arquitetura de alto nível do Intellux News

2.2 Fluxo de Autenticação (Google OAuth + JWT)

- Frontend → GET `/auth/google/login`
- Google Login → callback → Backend
- Backend gera JWT → redireciona para:
`http://localhost:5173/feed?token=...&user=...`
- Frontend (`useUrlAuth`) salva token/usuário → UI autenticada

2.3 Fluxo de Favoritar

- Usuário clica “coração” → Frontend chama POST `/favorites`.
- Backend persiste (`user`, `news`) e retorna sucesso.
- Frontend atualiza estado local → reflete em “Favoritos”.

2.4 Fluxo de Gerar Resumo

- Frontend chama POST `/summaries`.
- Backend: `findOrCreate` da notícia → chama Ollama → persiste resumo.
- Frontend exibe no modal e na aba “Resumos”.

3 Prints e Exemplos de Uso

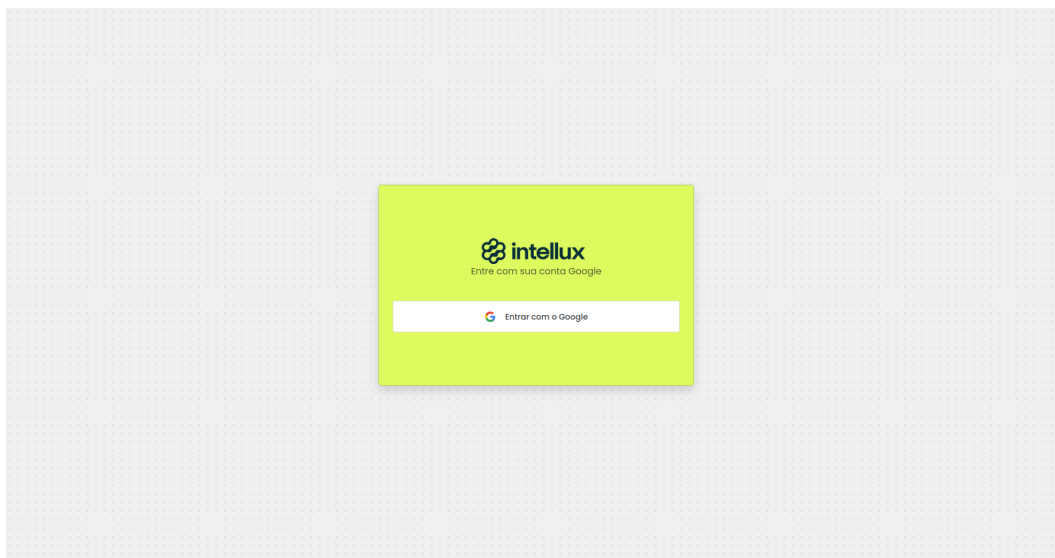


Figura 2: Login via Google

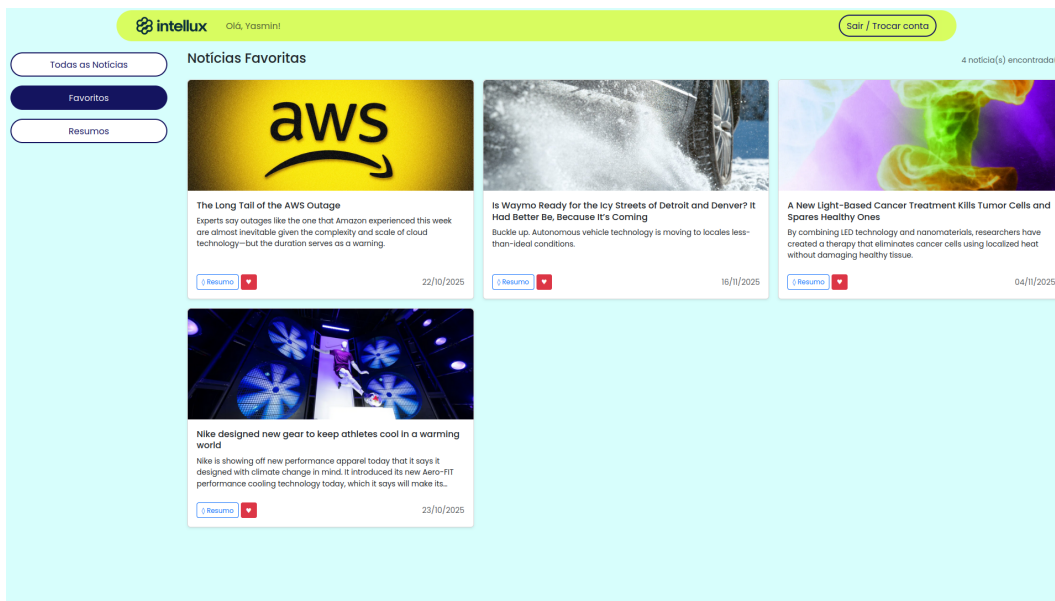


Figura 3: Aba Favoritos

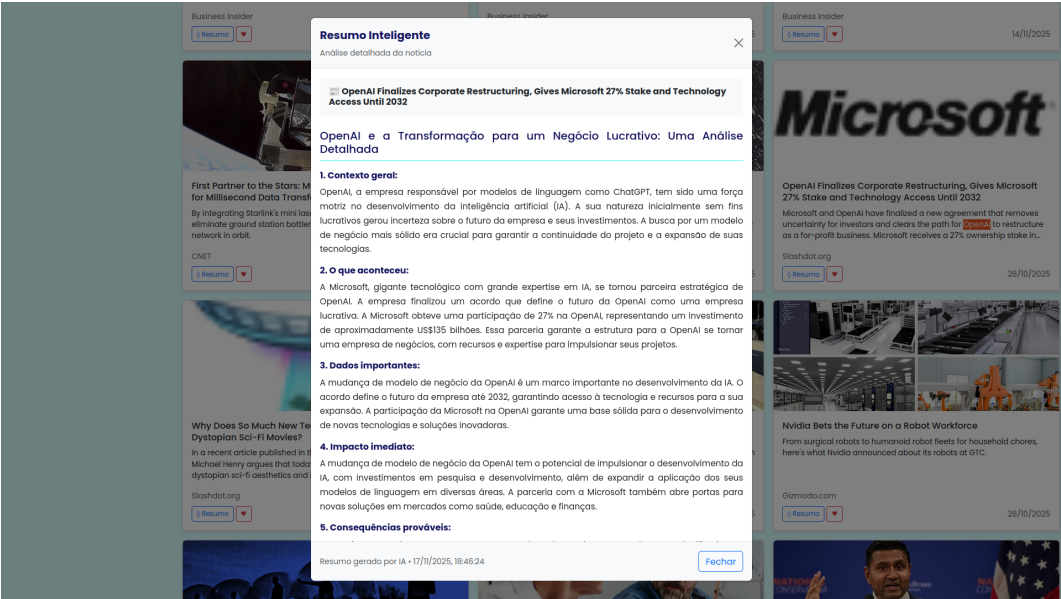


Figura 4: Modal de Resumo

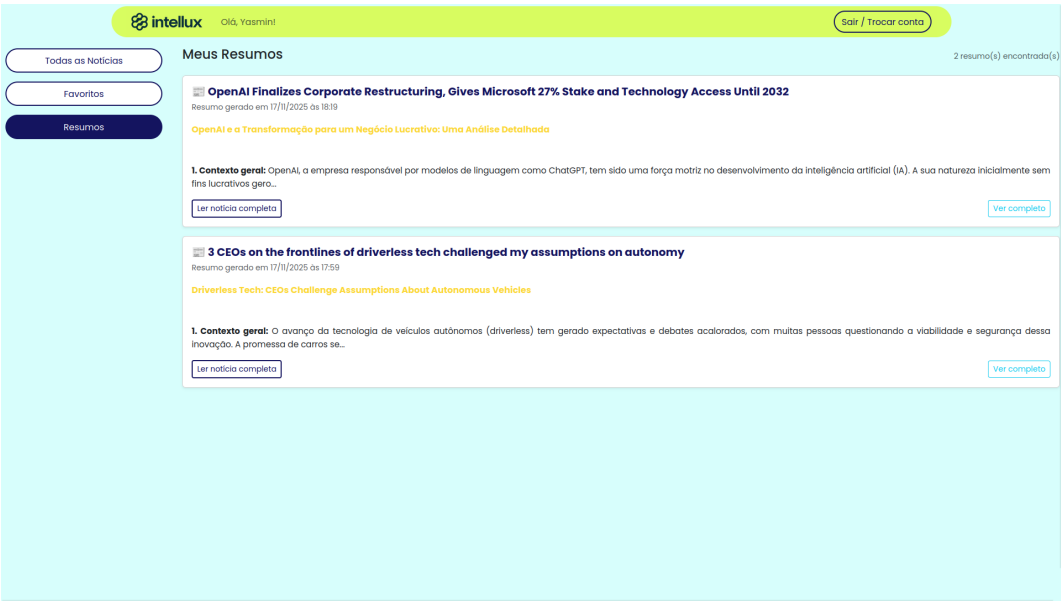


Figura 5: Aba Resumos

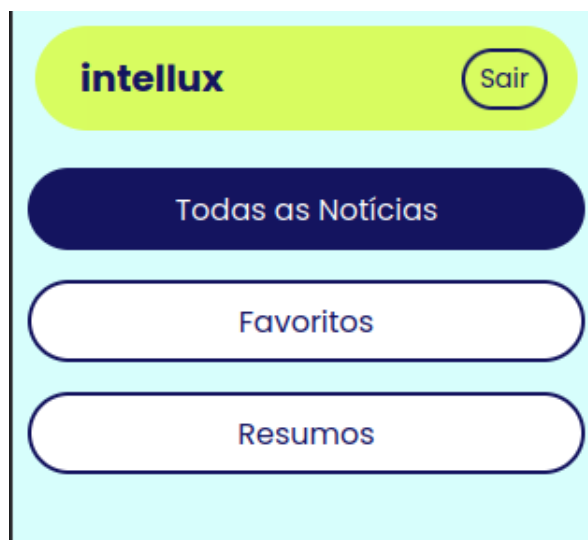


Figura 6: Responsividade do header

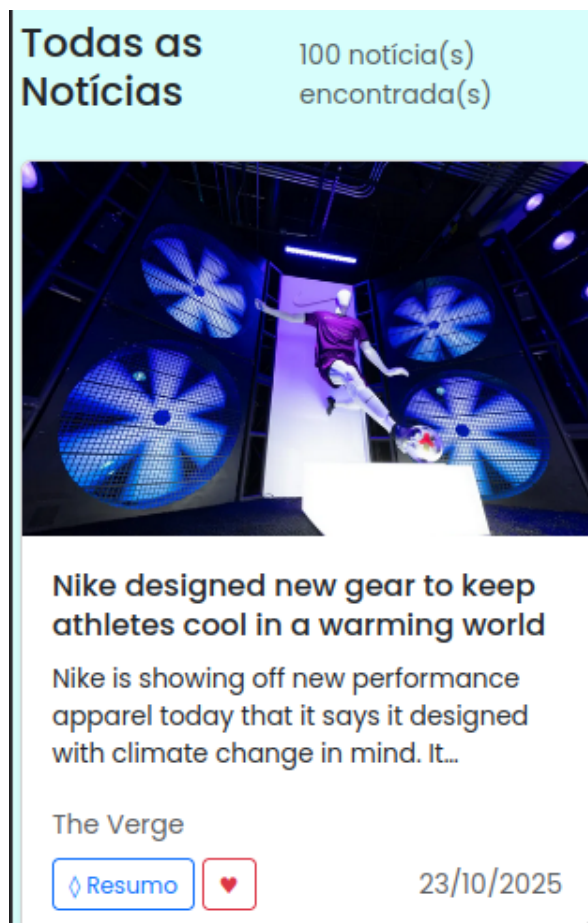


Figura 7: Responsividade do feed