

Etec de Heliópolis Arq. Ruy Ohtake – Fundação lochep – Venturus

Desenvolvimento de Sistemas – AMS

Ana Luiza Bertarelli

Felipe Alves Fornazeiro

Michelly Moreira

Samyra Alves

Relatório de Desenvolvimento e Testes

Global Core - L.A.R

São Paulo

2025

## **Relatório de Desenvolvimento – Projeto L.A.R**

**Período:** 18/06 -02/07

**Desenvolvedor:** Felipe Fornazeiro

### **1. Sistema de Rotas**

As páginas anteriormente exigiam scroll para navegação.

Agora, o sistema foi atualizado para utilizar links que direcionam diretamente para cada seção desejada.

### **2. Aba de Notícias**

Layout reformulado para melhor usabilidade.

Implementada funcionalidade de busca por título através de integração com API.

Adicionado botão de refresh da página para recuperação em caso de erro.

Criada uma tela dedicada para exibição das notícias, substituindo o redirecionamento para a fonte original.

### **3. Guia**

O conteúdo do guia foi transformado em uma página independente.

A estrutura antiga com sections roláveis foi substituída por uma navegação direta via página dedicada.

#### **1. Sobre Nós**

Site da Global Core, criado.

Criada a página "Sobre Nós" da Global Core.

Realizado deploy da página com acesso via link externo.

## Relatório de Desenvolvimento – Projeto L.A.R

**Período:** 30/06 – 02/07

**Desenvolvedora:** Samyra Alves

### 1. Instalação de Dependências

Inicialização do projeto no diretório correto após correção de sintaxe no comando cd.

Instalação dos pacotes i18next e react-i18next.

Atualização das dependências gerais com o comando npm i.

### 2. Correção de Vulnerabilidades

Identificada uma vulnerabilidade de baixa severidade por meio do comando npm audit.

Correção automática aplicada com sucesso utilizando npm audit fix.

### 3. Execução da Aplicação

Aplicação iniciada localmente com npm run dev.

URL local gerada: <http://localhost:3000>

### 4. Configuração de Internacionalização (i18n)

Arquivo i18n.ts criado e configurado com os idiomas: pt, en, es, fr, ar e zh.

Arquivos JSON com as traduções foram salvos na pasta src/idioms.

### 5. Criação de Providers

Componente TranslationProvider.tsx desenvolvido para encapsular o contexto de tradução.

## 6. Integração no Layout

TranslationProvider implementado no arquivo layout.tsx, garantindo aplicação global da funcionalidade de tradução.

## 7. Componente de Idiomas

Arquivo CountrySelector.tsx renomeado para LanguageSelector.tsx.

Adicionada a funcionalidade de troca dinâmica de idioma utilizando o hook useTranslation.

## 8. Internacionalização no Header

Textos substituídos por chamadas de tradução utilizando t('...').

Componentes LanguageSelector e ThemeSwitcher integrados ao cabeçalho (header).

## 9. Problemas Encontrados e Soluções

Erro 1: Element type is invalid...

Causa: Import incorreto do arquivo i18n como .js.

Solução: Corrigido para extensão .ts.

Erro 2: Erro no Turbopack: invalid JSON...

Causa: Arquivo pt.json estava vazio.

Solução: Adição correta do conteúdo ao JSON.

## 10. Considerações Finais

A funcionalidade de internacionalização foi implementada com sucesso.

Todos os erros encontrados foram corrigidos.

O sistema encontra-se pronto para testes multilíngues.

