**FIAP**

Luna Faustino Lima - 552473 - 2TDSPS

Larissa Araújo Gama Alvarenga - 96496 - 2TDSPS

Larissa Lopes Oliveira - 552628 - 2TDSPC

**JAVA ADVANCED – GLOBAL SOLUTION 01/2025**

São Paulo

2025

**SOBRE A SOLUÇÃO**

O **Abrigue-se** é um sistema web desenvolvido em Java com Spring Boot para gerenciamento eficiente de abrigos emergenciais. A aplicação oferece uma solução completa para coordenadores e gestores públicos monitorarem a ocupação de abrigos, o bem-estar das pessoas abrigadas e a disponibilidade de recursos essenciais.

**OBJETIVO**

Facilitar a gestão de abrigos em emergências, fornecendo:

* Controle de capacidade e ocupação
* Cadastro e acompanhamento de pessoas abrigadas
* Gestão de recursos e suprimentos
* Análises inteligentes via IA
* Sistema de alertas automatizados

**REPOSITÓRIO**

<https://github.com/LunaFaustino/JAVA-ADVANCED-GS-2025.git>

**LINK DEPLOY**

[abrigue-se-app.azurewebsites.net](https://abrigue-se-app.azurewebsites.net/)

**LINK APRESENTAÇÃO**

**REQUISITOS IMPLEMENTADOS**

**1. Aplicação Web Spring MVC + Thymeleaf**

* **Controllers**: AbrigoController, PessoaController, RecursoController, DashboardController, AssistenteController
* **Templates Thymeleaf**: Interface responsiva com Bootstrap
* **Arquitetura MVC** completa com separação clara de responsabilidades

**2. Templates Dinâmicos**

* **Layout Base**: Template reutilizável (layout/base.html)
* **Fragmentos**: Componentes reutilizáveis para navegação e alertas
* **Conteúdo Dinâmico**: Templates específicos para cada funcionalidade
* **Responsividade**: Interface adaptável para diferentes dispositivos

**3. OAuth 2 para Autenticação**

* **Provedor**: Autenticação via GitHub
* **Spring Security**: Configuração completa de segurança
* **Sessões**: Gerenciamento automático de sessões
* **Perfil do Usuário**: Integração com dados do GitHub

**4. CRUD Completo com Validação**

* **Entidades**: Abrigo, Pessoa, Recurso com relacionamentos JPA
* **Validações**: Bean Validation com mensagens personalizadas
* **Operações CRUD**: Create, Read, Update, Delete para todas as entidades
* **Tratamento de Erros**: Validações client-side e server-side

**5. Internacionalização (i18n)**

* **Idiomas Suportados**: Português (PT-BR) e Inglês (EN)
* **Troca Dinâmica**: Alternância de idioma sem necessidade de login
* **Mensagens**: Todas as mensagens da interface traduzidas
* **Configuração**: InternationalizationConfig com LocaleResolver

**6. Spring AI**

* **Integração**: Azure OpenAI GPT-4
* **Funcionalidades**:
  + Análise individual de abrigos
  + Análise panorâmica do sistema
  + Recomendações inteligentes
  + Identificação de problemas críticos
* **Service**: AssistenteAIService com prompts especializados

**7. RabbitMQ (Produtor e Consumidor)**

* **Configuração**: RabbitMQConfig com filas específicas
* **Produtor**: MessagePublisherService para envio de alertas
* **Consumidor**: MessageConsumerService com listeners
* **Filas Implementadas**:
  + abrigo.capacidade.baixa.queue
  + recurso.estoque.baixo.queue
  + alertas.criticos.queue

**8. Testes (Unitários e de Integração)**

* **Testes Unitários**: Services, Repositories, Entidades
* **Testes de Integração**: Fluxos completos da aplicação
* **Cobertura**: Validações, regras de negócio, repositories
* **Configuração**: Profile específico para testes com H2