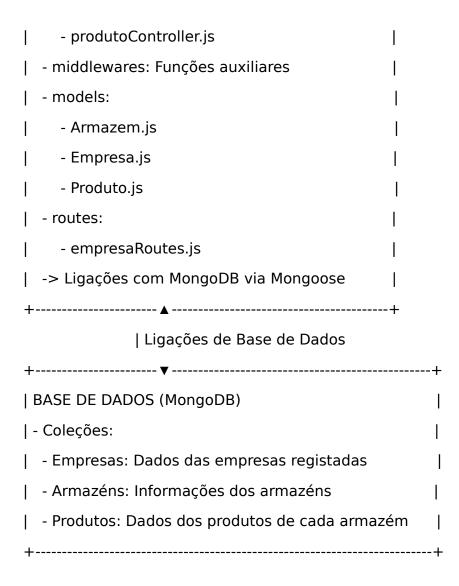
StockMaster	
Arquitetura: Cliente-Servidor	
+ CLIENTE	
- ReactJS: Interface de utilizador	
- Screens (Componentes):	
- AdicionarProdutoScreen.jsx	
- CriarArmazemScreen.jsx	
- EditarArmazemScreen.jsx	
- EditarProdutoScreen.jsx	
- HomeScreen.jsx	
- LoginScreen.jsx	
- MeusArmazensScreen.jsx	
- ProdutosScreen.jsx	
- SignupScreen.jsx	
-> Comunicação com o servidor via API (H	TTP/REST)
+	
Requisições e Respostas (JSC	ON)
+	+
SERVIDOR (Backend)	
- Node.js com ExpressJS: Servidor Web	I
- Estrutura de Pastas:	I
- app.js e server.js	I
- controllers:	I
- armazemController.js	1
- authController.js	
- empresaController.js	



Explicação:

1. Cliente (Frontend):

- Desenvolvido em ReactJS.
- Componentes no diretório Screens representam diferentes páginas da aplicação (ex.: registo, login, armazéns, produtos).
- Comunica-se com o servidor através de APIs REST, enviando e recebendo dados em formato JSON.

2. Servidor (Backend):

- o Construído com **Node.js** e **ExpressJS**.
- o Estrutura organizada em controllers, middlewares, models e routes.

o Usa **Mongoose** para interagir com a base de dados MongoDB.

3. Base de Dados (MongoDB):

- o Armazena dados estruturados em coleções (empresas, armazéns e produtos).
- o Relacionamentos definidos nos modelos (ex.: produtos pertencem a armazéns e empresas).