

Entrega 2 - Primeiros Módulos e Comunicação

Disciplina: Sistemas Distribuídos

Data de Entrega: 26/09/2025

Alunos: Danilo Carvalho De Oliveira, Guilherme Faria da Silva, Vinicius Henrique Domingues

1. Sobre o Projeto

Sistema de lista de compras compartilhada com **atualizações em tempo real**. Utiliza arquitetura de **microserviços** e **WebSockets** para sincronização instantânea entre usuários.

Funcionalidades

- ☒ Criar/deletar listas de compras
 - ☒ Adicionar/remover/marcar itens
 - ☒ **Sincronização em tempo real** entre todos os usuários
 - ☒ Interface moderna com tema claro/escuro
-

2. Arquitetura

Stack Tecnológica

- **Frontend:** React + Socket.IO Client
- **Backend:** 2 microserviços Node.js/Express
- **Comunicação:** REST + WebSockets (Socket.IO)
- **Banco:** MongoDB
- **Infraestrutura:** Docker + Nginx + AWS EC2

Microserviços

1. **listas-service (3001):** CRUD de listas
 2. **itens-service (3002):** CRUD de itens + Socket.IO para tempo real
-

3. Como Executar

Pré-requisitos

- Git, Docker, Docker Compose

Execução Local

```
git clone https://github.com/danilocarv/Sistemas_Distribuidos.git
```

```
cd Sistemas_Distribuidos
```

```
docker-compose up --build -d
```

Acessar: `http://localhost`

Deploy AWS

1. Criar instância EC2 AWS Linux
 2. Instalar Docker + Docker Compose
 3. Clonar repo e executar `docker-compose up --build -d`
-

4. Configuração dos Serviços

Docker Compose Structure

services:

frontend: # React app (porta 80)

listas-service: # API listas (porta 3001)

itens-service: # API itens + Socket.IO (porta 3002)

mongodb: # Banco de dados

nginx: # Gateway/Proxy

Endpoints Principais

- **Frontend:** `http://localhost`
- **API Listas:** `/api/lists`

- **API Itens:** [/api/items](#)
 - **WebSocket:** Socket.IO automático via frontend
-

5. Requisitos Atendidos

- **Arquitetura:** Microserviços distribuídos
- **Comunicação:** REST + WebSockets
- **Coordenação:** Sincronização tempo real via Socket.IO
- **Persistência:** MongoDB
- **Containerização:** Docker
- **Deploy:** AWS EC2