UNIVERSIDADE ESTADUAL DO TOCANTINS – UNITINS
Curso de Sistemas de Informação – Disciplina: Sistemas Distribuídos

Período Letivo: 2025/2 – 7º Período – Câmpus Palmas

Alunos: João Victor Póvoa França e Hemilly Christinne Silva Pereira

Documentação de Planejamento de Arquitetura para Sistema Distribuído com Nginx, AWS e Node.js

Relatório Semana 2

PALMAS-TO

2025

Relatório de Progresso – 2ª Semana

Projeto: Node Balancer
Repositório: https://github.com/JoaoIto/node-balancer
Período: Segunda Semana de Desenvolvimento

1. Breve resumo do projeto

O **Node Balancer** é uma API escalável desenvolvida em **Node.js (Express)**, com **MongoDB** configurado em *replica set* para alta disponibilidade e **Nginx** atuando como balanceador de carga.
A arquitetura prioriza **resiliência**, **escalabilidade** e **tolerância a falhas**.

O repositório centraliza:

- A documentação do sistema;
- Instruções de configuração do *replica set*;
- Estrutura inicial do código em **TypeScript**.

2. Descrição das atividades realizadas (2ª semana)

2.1 Criação e organização do repositório no GitHub

Foi provisionado o repositório público **node-balancer**, contendo um arquivo **README** que descreve:

- A visão geral do sistema;
- As tecnologias utilizadas;
- Um diagrama de arquitetura.

Também foi habilitada a navegação básica de *Issues* e *Pull Requests*, ainda sem itens cadastrados.

2.2 Documentação técnica inicial

O **README** foi atualizado com:

- Visão do projeto;
- Lista de tecnologias (**Node.js/Express**, **MongoDB Replica Set**, **Nginx**, **Docker**, **monitoramento**);
- Seção "Configuração do Banco de Dados", com exemplos de URLs de conexão para *replica set* e comandos de inicialização e verificação (*mongod --replSet*, *rs.initiate*, *rs.status*);
- Inclusão de um diagrama ilustrativo (*img.png*).

2.3 Estrutura base do código (TypeScript)

Foi criada a estrutura inicial com:

- Diretório **/src** para o código-fonte;
- Arquivos de configuração (**tsconfig.json**, **package.json**, **package-lock.json**);
- Arquivos de controle (**.gitignore**, **.env.local** com variáveis sensíveis, incluindo *MONGODB_URI* conforme instruções do README).

Essa base estabelece o backend em **TypeScript**, com versionamento padronizado e estrutura pronta para evolução.

2.4 Padronização e versionamento

Foram consolidados **sete commits iniciais**, estabelecendo a base do projeto e da documentação.

Arquivos e diretórios adicionados ou destacados

- **README.md** – visão geral, tecnologias, configuração do *replica set* e diagrama;
- **/docs** – pasta dedicada à documentação (materiais de operação e infraestrutura);
- **/src** – código-fonte principal da API;
- **.env.local** – exemplo de configuração de ambiente com *MONGODB_URI*;
- **tsconfig.json**, **package.json**, **package-lock.json**, **.gitignore** – arquivos de configuração e controle.

Observação: o *README* faz referência explícita à string de conexão do *replica set* e aos comandos para inicialização e validação do cluster, utilizados nas validações locais desta semana.

3. Dificuldades encontradas

- **Consistência do Replica Set em ambiente local:** necessidade de garantir que todas as instâncias (**localhost:27017/18/19**) estejam ativas antes da inicialização da API, evitando *timeouts* de conexão. O *README* foi essencial para padronizar esse processo.
- **Organização do ambiente:** separação adequada das variáveis sensíveis no arquivo **.env.local**, assegurando segurança e reprodutibilidade.
- **Base de infraestrutura e documentação:** consolidação das decisões sobre **Nginx**, **Docker** e **monitoramento**, alinhando testes prévios em Windows com a visão de produção em ambiente Linux e containers.

4. Próximas etapas planejadas

1. **Configuração do Nginx:**
    - Publicar no repositório exemplos de *nginx.conf* para balanceamento entre múltiplas instâncias Node.js;
    - Definir *headers* e *health-check* básico;
    - Criar guias diferenciados para ambiente de **desenvolvimento (Windows)** e **produção (Linux)**.
2. **Versionamento de scripts NPM e automação (CI):**
    - Adicionar scripts no *package.json* (*build*, *start*, *dev*, *lint/test*);
    - Implementar *pipeline* de build com **GitHub Actions** para validação contínua.
3. **Modelagem e endpoints iniciais da API:**
    - Desenvolvimento de CRUD básico com conexão resiliente ao *replica set* (retries, timeouts, logs).
4. **Orquestração com Docker:**
    - Criação de um **docker-compose.yml** para execução simultânea de múltiplas instâncias da API e do *replica set*, facilitando testes de balanceamento.
5. **Monitoramento e observabilidade:**
    - Estruturar logs, métricas e monitoramento conforme diretrizes já citadas na documentação.

5. Conclusão

A segunda semana consolidou:

- O repositório GitHub do projeto;
- A documentação técnica;
- A base de código em **TypeScript**;
- A configuração funcional do **MongoDB em *replica set***.

Com a fundação de código e documentação estabelecida, as próximas etapas focarão na implementação de **endpoints**, **automação contínua (CI)**, **conteinerização** e na **documentação detalhada do Nginx** para ambientes de desenvolvimento e produção.