



Gabriel Yamaoka Bernardes

João Lucas Melo de Oliveira

Laura Jane Antunes

Mariana Aiko Kakimoto Hirata

Mateus Ferreira Rosa

**Projeto Integrador: EcosRev**

Votorantim

2024

Gabriel Yamaoka Bernardes

João Lucas Melo de Oliveira

Laura Jane Antunes

Mariana Aiko Kakimoto Hirata

Mateus Ferreira Rosa

## **Projeto Integrador: EcosRev**

Trabalho desenvolvido guiado pela integração das disciplinas de Laboratório de Desenvolvimento Web, Integração e Entrega Contínua e Experiência do Usuário como requisito para obtenção de nota para a aprovação do 4º semestre do curso de Desenvolvimento de Software Multiplataforma.

Profº Orientador: Tiago Vanderlei de Arruda.

Votorantim

2024

## Sumário

1. Introdução.....	4
2. Objeto de Desenvolvimento .....	4
3. Acesso aos repositórios .....	5
4. Código Utilizado para Criação do Docker .....	6

## **1. Introdução**

O desenvolvimento de projetos integradores é fundamental para a formação acadêmica, proporcionando uma aplicação prática dos conhecimentos adquiridos ao longo do curso. Este trabalho de conclusão de semestre reflete a evolução técnica e conceitual dos integrantes, com foco na implementação de tecnologias modernas e no aperfeiçoamento das habilidades necessárias para o mercado de desenvolvimento de software.

Neste semestre, o projeto teve como principal objetivo a implementação de um frontend utilizando o framework React, que oferece uma abordagem declarativa e eficiente para a criação de interfaces de usuário. No backend, foi utilizado o Express, possibilitando a construção de APIs robustas e integradas ao frontend. O trabalho também enfatizou a integração entre o frontend e o backend, utilizando novas tecnologias e ferramentas que garantem uma comunicação eficiente entre as camadas do sistema.

Além disso, foram aplicados testes automatizados com Jest para garantir a qualidade e confiabilidade do código, e o uso de Docker foi explorado para a criação de repositórios e imagens, assegurando um ambiente de desenvolvimento consistente e portátil. O versionamento do código foi gerido por meio do GitHub, seguindo práticas modernas de colaboração e controle de mudanças, fundamentais para o trabalho em equipe e a organização do projeto.

Este projeto representa um marco no processo de aprendizado, reunindo conceitos de desenvolvimento frontend e backend, integração de sistemas, testes de software, e práticas de gestão e versionamento. A seguir, serão apresentados os objetivos, métodos e resultados que evidenciam o progresso alcançado nesta jornada.

## **2. Objeto de Desenvolvimento**

O objeto de desenvolvimento deste semestre foi o Projeto EcosRev, uma plataforma web projetada para promover a sustentabilidade e a conscientização

ambiental, incentivando o descarte adequado de resíduos eletrônicos. O sistema permite que os cidadãos realizem trocas de seus resíduos por pontos, os quais podem ser acumulados e utilizados para resgatar benefícios diversos, como ingressos para eventos, passagens de ônibus, descontos em produtos e serviços, entre outros.

Por ser uma plataforma web, o EcosRev é acessível por meio de navegadores de internet, o que amplia sua disponibilidade e facilita o acesso por usuários em diferentes dispositivos, como computadores e smartphones. A solução combina tecnologias modernas para oferecer uma experiência fluida e intuitiva, incentivando a participação ativa dos cidadãos e contribuindo para a preservação ambiental.

O desenvolvimento técnico envolveu a construção de um frontend dinâmico em React, garantindo interfaces interativas e responsivas, e um backend robusto utilizando Express, permitindo a gestão eficiente das funcionalidades e a integração com bancos de dados. O projeto também seguiu boas práticas de desenvolvimento, como a utilização do GitHub para versionamento, Docker para criação de ambientes consistentes, e testes automatizados com Jest para assegurar a qualidade e confiabilidade do sistema.

O EcosRev representa uma inovação tecnológica com impacto ambiental e social positivo, conectando usuários a um ecossistema sustentável por meio de uma solução prática e acessível.

### 3. Acesso aos repositórios:

Item	Repositório
Front End GitHub	<a href="https://github.com/Ecosrev/PI-DSM-front">https://github.com/Ecosrev/PI-DSM-front</a>
Back End GitHub	<a href="https://github.com/Ecosrev/PI-DSM-back">https://github.com/Ecosrev/PI-DSM-back</a>
Imagem Docker Front End	<a href="https://hub.docker.com/r/yamaokak/ecosrev-frontend">https://hub.docker.com/r/yamaokak/ecosrev-frontend</a>

#### 4. Código Utilizado para Criação do Docker

O projeto **EcosRev** utiliza contêineres Docker para garantir consistência e portabilidade no ambiente de desenvolvimento e execução. Abaixo estão os comandos utilizados para criar e gerenciar as imagens e os contêineres do frontend e backend, além de integrar ambos por meio do Docker Compose.

##### Frontend

##### 1. Construir a Imagem

Para criar a imagem do frontend:

```
docker build -t ecosrev-frontend-app .
```

##### 2. Verificar Imagens Criadas

Confirme a criação da imagem:

```
docker images
```

##### 3. Executar o Contêiner

Execute o contêiner do frontend vinculando a porta local 3000 e configurando a variável de ambiente para a API:

```
docker run -p 3000:3000 -e  
NEXT_PUBLIC_API_URL=http://localhost:4000/api
```

Imagem do frontend no Docker Hub:

[EcosRev Frontend no Docker Hub](#)

## Backend

### 1. Construir a Imagem

Para criar a imagem do backend:

```
docker build -t ecosrev-backend .
```

### 2. Executar o Contêiner

Execute o contêiner do backend vinculando a porta local 4000 e utilizando as variáveis de ambiente definidas no arquivo .env:

```
docker run -p 4000:4000 --env-file .env ecosrev-backend
```

Imagem do backend no Docker Hub:

[EcosRev Backend no Docker Hub](#)

## Integração Frontend e Backend

Para combinar as imagens do frontend e backend em um único ambiente, utilizamos o **Docker Compose**. O comando abaixo realiza o build e a execução simultânea dos contêineres:

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

Esses comandos garantem que o **EcosRev** funcione de maneira integrada e consistente, tanto em ambientes de desenvolvimento quanto em produção.

## 5. Docker Composer

No código abaixo, configura a aplicação utilizando o Docker Compose. Definimos os serviços que compõem a aplicação (backend e frontend), suas respectivas imagens de container, portas de exposição, dependências entre serviços e configurações de rede. Ao executar o comando `docker-compose up`, o Docker Compose irá criar e iniciar todos os containers descritos no arquivo, estabelecendo as conexões e dependências necessárias para o funcionamento da aplicação. Essa abordagem simplifica a gestão de ambientes de desenvolvimento, testes e produção, permitindo a criação de um ambiente consistente e reproduzível.

```
services:
  backend:
    image: ecosrev-backend
    container_name: ecosrev-backend
    ports:
      - "4000:4000"
    environment:
      - MONGODB_URI=${MONGODB_URI}
      - MONGODB_DB=${MONGODB_DB}
      - EXPIRES_IN=${EXPIRES_IN}
      - SECRET_KEY=${SECRET_KEY}
      - PORT=${PORT}
    restart: unless-stopped

  frontend:
    image: ecosrev-frontend-app
    container_name: ecosrev-frontend
    ports:
      - "3000:3000"
    depends_on:
      - backend
    restart: unless-stopped

networks:
  default:
    driver: bridge
```





docker-composer.yml