

# Relatório de Laboratório 2025.S1-E1.01

Jonas de Araújo Luz Junior<sup>a,1</sup>, José de La Cruz Iraheta<sup>a,2</sup> and Pedro Jardelino Neto<sup>a,3</sup><sup>a</sup>Universidade de Fortaleza (Unifor)

Professor Nabor das Chagas Mendonça

**Resumo**—Este relatório descreve a execução da **Primeira Entrega Parcial** do trabalho prático proposto no Laboratório de Sistemas Distribuídos, cujo objetivo é provisionar um cluster Kubernetes local, instalar o service-mesh Istio e implantar a aplicação de micro-serviços *Online Boutique*.

**Keywords**—Relatório de laboratório, Sistemas distribuídos, Kubernetes, Istio, Docker

## 1. Especificação da Entrega

### 1.1. Tarefa 1 - Configuração do Ambiente

- Configurem um repositório Git compartilhado para o projeto.
- Instalem e configurem a distribuição Kubernetes local escolhida em suas máquinas (ou em uma máquina compartilhada pela equipe). Certifiquem-se de alocar recursos suficientes (CPU/-RAM).
- Instalem o Istio no cluster Kubernetes, utilizando o perfil de instalação demo ou default. Verifiquem a instalação.

### 1.2. Tarefa 2 - Implantação da Aplicação

- Obtenham os manifestos de implantação da aplicação *Online Boutique*.
- **Implantação base:** implantem a aplicação sem a injeção automática de sidecars do Istio (ou seja, em um namespace sem o rótulo `istio-injection=enabled`). Verifiquem se todos os serviços estão rodando e se a aplicação está acessível.
- **Implantação com Istio:** habilitem a injeção automática de sidecars do Istio para um novo namespace (e.g., `online-boutique-istio`) e implantar a aplicação novamente neste namespace. Verifiquem se os sidecars foram injetados (`kubectl get pods -n -o wide` deve mostrar 2/2 containers por pod) e se a aplicação continua acessível.

### 1.3. Entregáveis:

- Relatório Preliminar (2-3 páginas) [este] incluindo:
  - Formação da equipe e link para o repositório Git criado.
  - Evidência do sucesso (e.g., screenshots, logs) na instalação e configuração do ambiente Kubernetes local (incluir versão, recursos alocados).
  - Evidência do sucesso (e.g., screenshots, logs) na instalação do Istio (incluir versão, perfil utilizado).
  - Evidência do sucesso (e.g., screenshots, logs) na implantação da aplicação *Online Boutique* nos dois cenários (sem e com injeção do sidecar Istio).
  - Repositório Git atualizado com a estrutura inicial e quaisquer scripts/manifestos básicos utilizados.

## 2. Repositório Git

- Repositório oficial do projeto (código, manifestos, evidências): <https://github.com/jonasluz/DIA.kubernetes-istio/>

## 3. Ambiente de Trabalho

**Sistema** Fedora 41 (x86\_64) atualizado em 6 de maio de 2025.

**Recursos** 8 GB RAM, 4 vCPU, 60 GB SSD.

**Container Runtime** Docker 24.x (moby-engine) [1].

**Cluster** Minikube v1.35.0 com Kubernetes v1.32.0 [5].

**Istio** 1.22.0 (perfil demo) [3].

## 4. Passo a Passo da Instalação

Os comandos abaixo foram executados sequencialmente em shell bash. Cada etapa inclui uma breve explicação e um espaço reservado para evidência (log ou captura de tela).

### 4.1. Atualização do Sistema

#### 1. Atualizar pacotes e utilitários básicos

```
1 sudo dnf upgrade --refresh -y && sudo reboot
```

after reboot:

```
1 sudo dnf install -y curl wget git conntrack jq
```

### 4.2. Instalação do Docker

#### 1. Adicionar repositório Docker CE e instalar runtime

```
1 sudo dnf install -y dnf-plugins-core
2 sudo dnf config-manager --add-repo \
3   https://download.docker.com/linux/fedora/docker-ce
4   ↪ .repo
5 sudo dnf install -y docker-ce docker-ce-cli
6   ↪ containerd.io \
7   ↪ docker-buildx-plugin docker-compose-plugin
8 sudo systemctl enable --now docker
9 sudo usermod -aG docker $(whoami)
```

Evidência: Figura 1.

```
jonas@segat03:~/Projetos/DIA.kubernetes-istio$ docker --version
Docker version 28.1.1, build 4eba377
```

Figura 1. Evidência: Instalação do Docker

### 4.3. Instalação do kubectl

#### 1. Baixar binário compatível (v1.32.0) [4]

```
1 curl -LO https://dl.k8s.io/release/v1.32.0/bin/linux
2   ↪ /amd64/kubectl
3 sudo install -o root -g root -m 0755 kubectl /usr/
4   ↪ local/bin/
5 rm kubectl
```

Evidência: Figura 2.

```
jonas@segat03:~/Projetos/DIA.kubernetes-istio$ kubectl version --client
Client Version: v1.32.0
Kustomize Version: v5.5.0
```

Figura 2. Evidência: Instalação do kubectl

### 4.4. Instalação do Minikube

#### 1. Baixar Minikube v1.35.0 [5]

```
1 curl -LO https://github.com/kubernetes/minikube/
2   ↪ releases/download/v1.35.0/minikube-linux-amd
3   ↪ 64
```

```
2 sudo install minikube-linux-amd64 /usr/local/bin/
   ↪ minikube
3 rm minikube-linux-amd64
```

## 2. Inicializar cluster (driver Docker)

```
1 minikube start --driver=docker --cpus=4 --memory=819
   ↪ 2
```

Evidência: Figura 3.

```
jonas@segat03: ~/Projetos/DIA.kubernetes-istio$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

jonas@segat03: ~/Projetos/DIA.kubernetes-istio$ kubectl get nodes
NAME      STATUS   ROLES    AGE   VERSION
minikube  Ready    control-plane  3h22m  v1.32.0
jonas@segat03: ~/Projetos/DIA.kubernetes-istio$
```

Figura 3. Evidência: Instalação do Minikube.

## 4.6. Implantação da Online Boutique

### 1. Clonar repositório e implantar namespace boutique-base [2].

```
1 git clone --depth 1 https://github.com/jonasluz/
   ↪ microservices-demo.git
2 kubectl create namespace boutique-base
3 kubectl apply -f microservices-demo/release/
   ↪ kubernetes-manifests.yaml -n boutique-base
```

Evidência: Figuras 5 e 6

```
jonas@segat03: ~/Projetos/DIA.kubernetes-istio$ kubectl get pods -n boutique-base -w
NAME                                READY   STATUS    RESTARTS   AGE
adservice-8568877bf9-7z7vx         1/1     Running   0           3m49s
cartservice-f84bf7dd4-ntwqj        1/1     Running   0           3m50s
checkoutservice-5d9894c787-6fgnk   1/1     Running   0           3m52s
currencyservice-84459c6759-td99n    1/1     Running   0           3m50s
emailservice-6fb4dd89fc-nlqmw       1/1     Running   0           3m52s
frontend-754cdbf884-dcv2d          1/1     Running   0           3m51s
loadgenerator-696d89b74f-jvlxf      1/1     Running   0           3m50s
paymentservice-5575668b5c-jxmp2     1/1     Running   0           3m51s
productcatalogservice-59cf6d7b5-6bsvj 1/1     Running   0           3m51s
recommendationservice-589895488f-pfrdx 1/1     Running   0           3m52s
redis-cart-c4fc658fb-kb842         1/1     Running   0           3m50s
shippingservice-fb4c9695c-wqr9n     1/1     Running   0           3m49s
```

Figura 5. Evidência: Instalação da aplicação Online Boutique

## 4.5. Instalação do Istio

### 1. Download e instalação do Istioctl 1.22.0

```
1 curl -L https://istio.io/downloadIstio |
   ↪ ISTIO_VERSION=1.22.0 sh -
2 export PATH="$PATH:$HOME/istio-1.22.0/bin"
3 istioctl install --set profile=demo -y
4 istioctl verify-install
```

Evidência: Figura 4

```
jonas@segat03: ~/Projetos/DIA.kubernetes-istio$ istioctl verify-install
1 Istio control planes detected, checking --revision "default" only
✓ Deployment: istio-egressgateway.istio-system checked successfully
✓ Deployment: istio-ingressgateway.istio-system checked successfully
✓ Deployment: istiod.istio-system checked successfully
✓ Service: istio-egressgateway.istio-system checked successfully
✓ Service: istio-ingressgateway.istio-system checked successfully
✓ Service: istiod.istio-system checked successfully
✓ ConfigMap: istio.istio-system checked successfully
✓ ConfigMap: istio-sidecar-injector.istio-system checked successfully
✓ Pod: istio-egressgateway-5c7fd7dc64-4pr6x.istio-system checked successfully
✓ Pod: istio-ingressgateway-64b5467f4-cmh42.istio-system checked successfully
✓ Pod: istiod-c4b4bd85b-hdndb.istio-system checked successfully
✓ ServiceAccount: istio-egressgateway-service-account.istio-system checked successfully
✓ ServiceAccount: istio-ingressgateway-service-account.istio-system checked successfully
✓ ServiceAccount: istiod.istio-system checked successfully
✓ RoleBinding: istio-egressgateway-sds.istio-system checked successfully
✓ RoleBinding: istio-ingressgateway-sds.istio-system checked successfully
✓ RoleBinding: istiod.istio-system checked successfully
✓ Role: istio-egressgateway-sds.istio-system checked successfully
✓ Role: istio-ingressgateway-sds.istio-system checked successfully
✓ Role: istiod.istio-system checked successfully
✓ PodDisruptionBudget: istio-egressgateway.istio-system checked successfully
✓ PodDisruptionBudget: istio-ingressgateway.istio-system checked successfully
✓ PodDisruptionBudget: istiod.istio-system checked successfully
✓ MutatingWebhookConfiguration: istio-revision-tag-default.istio-system checked successfully
✓ MutatingWebhookConfiguration: istio-sidecar-injector.istio-system checked successfully
✓ ValidatingWebhookConfiguration: istio-validator.istio-system checked successfully
✓ ValidatingWebhookConfiguration: istiod-default-validator.istio-system checked successfully
✓ ClusterRole: istio-reader-clusterrole.istio-system.istio-system checked successfully
✓ ClusterRole: istiod-clusterrole.istio-system.istio-system checked successfully
✓ ClusterRole: istiod-gateway-controller.istio-system.istio-system checked successfully
✓ ClusterRoleBinding: istio-reader-clusterrole.istio-system.istio-system checked successfully
✓ ClusterRoleBinding: istiod-clusterrole.istio-system.istio-system checked successfully
✓ ClusterRoleBinding: istiod-gateway-controller.istio-system.istio-system checked successfully
✓ CustomResourceDefinition: authorizationpolicies.security.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: destinationrules.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: envoyfilters.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: gateways.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: peerauthentications.security.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: proxyconfigs.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: requestauthentications.security.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: serviceentries.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: sidecars.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: telemetry.telemetry.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: virtualservices.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: wasmplugins.extensions.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: workloadentries.networking.istio.io.istio-system checked successfully
✓ CustomResourceDefinition: workloadgroups.networking.istio.io.istio-system checked successfully
Checked 14 custom resource definitions
Checked 3 Istio Deployments
✓ Istio is installed and verified successfully
jonas@segat03: ~/Projetos/DIA.kubernetes-istio$
```

Figura 4. Evidência: Instalação do Istio

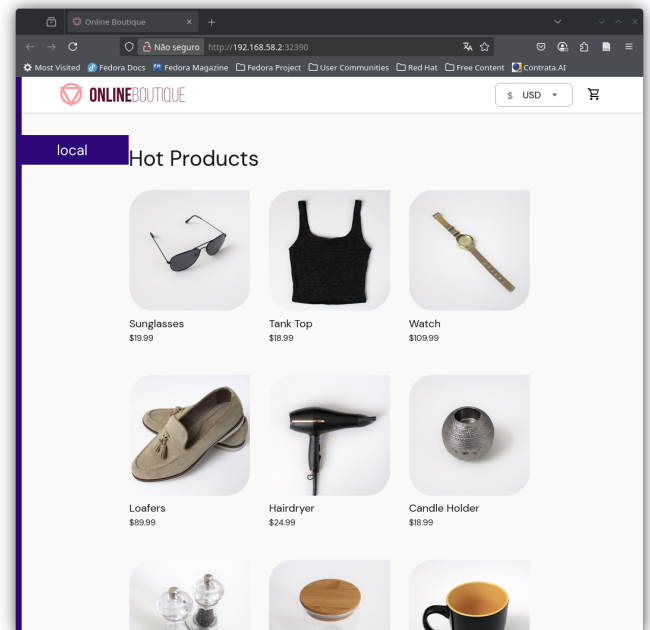


Figura 6. Evidência: Tela do navegador com a aplicação Online Boutique

### 2. Implantar versão com sidecars Istio (boutique-istio)

```
1 kubectl create namespace boutique-istio
2 kubectl label namespace boutique-istio istio-
   ↪ injection=enabled
3 kubectl apply -f microservices-demo/release/
   ↪ kubernetes-manifests.yaml -n boutique-istio
```

Evidência: Figuras 5 e 6

## 5. Problemas Encontrados e Soluções

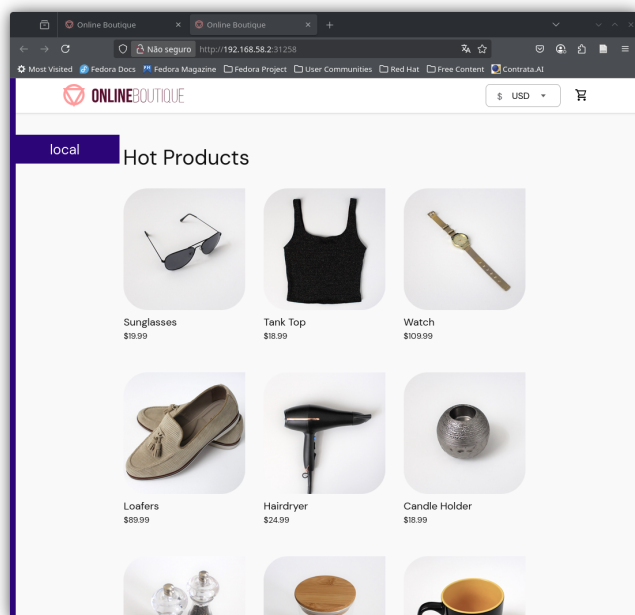
- **Driver Podman rootless instável:** optou-se pelo driver *Docker*, que funcionou sem ajustes extras.
- **Aviso de versão do kubectl:** resolvido instalando binário 1.32.

```

jomas@paga183:~/Projeto/DIA Kubernetes-Istio$ kubectl get pods -n boutique-istio -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE             NOMINATED NODE   READINESS GATES
adservice-856837bf9-56nf7           2/2    Running   0           44s   10.244.0.29     minikube         <none>            <none>
cartservice-f84b7d84-wddq           2/2    Running   0           46s   10.244.0.23     minikube         <none>            <none>
checkoutservice-5d9894c787-g4j69    2/2    Running   0           47s   10.244.0.18     minikube         <none>            <none>
currencyservice-84459c7f55-8gqkt     2/2    Running   0           45s   10.244.0.27     minikube         <none>            <none>
emailservice-6fb4d89fc-jzpq         2/2    Running   0           47s   10.244.0.19     minikube         <none>            <none>
frontend-754cd8f84-nvzqg            2/2    Running   0           46s   10.244.0.21     minikube         <none>            <none>
loadgenerator-996d8b674f-ijjfn       2/2    Running   0           46s   10.244.0.26     minikube         <none>            <none>
paymentservice-52756885c-55jfx      2/2    Running   0           46s   10.244.0.24     minikube         <none>            <none>
productcatalogservice-59cf6f7b5-lmj7  2/2    Running   0           46s   10.244.0.22     minikube         <none>            <none>
recommendationservice-589895488f-t4xv4 2/2    Running   0           47s   10.244.0.28     minikube         <none>            <none>
redis-cart-c4fd68f0-gj7j            2/2    Running   0           46s   10.244.0.25     minikube         <none>            <none>
shippingservice-fb4c9695c-rdn6v      2/2    Running   0           45s   10.244.0.28     minikube         <none>            <none>
jomas@paga183:~/Projeto/DIA Kubernetes-Istio$ kubectl get pods -n boutique-istio -o wide > evidences/kubernetes-istio.txt
jomas@paga183:~/Projeto/DIA Kubernetes-Istio$

```

**Figura 7.** Evidência: Instalação da aplicação Online Boutique com Istio



**Figura 8.** Evidência: Tela do navegador com a aplicação Online Boutique (com Istio)

- Outros problemas menores relacionados à disponibilização de pacotes e novos repositórios fonte de pacotes.

## 6. Conclusão

A primeira entrega foi concluída com êxito: cluster Kubernetes funcional em Minikube, Istio instalado e aplicação Online Boutique implantada em dois cenários (com e sem sidecar). As evidências coletadas encontram-se nos espaços reservados e em formato completo no diretório /evidences do repositório oficial.

## Referências

- [1] I. Docker, *Docker Engine & Docker Desktop*, <https://docs.docker.com/>, Acesso em 06mai2025, 2024.
- [2] Google Cloud Platform, *Online Boutique (Microservices Demo)*, <https://github.com/GoogleCloudPlatform/microservices-demo>, Acesso em 06mai2025, 2024.
- [3] The Istio Authors, *Istio Service Mesh*, <https://istio.io/>, Acesso em 06mai2025, 2024.
- [4] The Kubernetes Authors, *Kubernetes: Production-Grade Container Orchestration*, <https://kubernetes.io/>, Acesso em 06mai2025, 2024.
- [5] The Kubernetes Authors, *Minikube: Run Kubernetes Locally*, <https://minikube.sigs.k8s.io/>, Acesso em 06mai2025, 2024.