

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS Y SISTEMAS

---

## MANUAL TECNICO DE INSTALACION Y CONFIGURACION

# SISTEMAS OPERATIVOS 1

<b>Inti Andrés Samayoa Ortiz</b>	<b>201504002</b>
<b>Pascual Domingo Diego</b>	<b>201403504</b>
<b>Bernald Renato Paxtor Perén</b>	<b>201313828</b>

ing. Sergio Méndez  
Aux. Fernando Mazariegos

## **INTRODUCCION**

Se creó una arquitectura de sistemas distribuida genérica que muestra estadísticas en tiempo real mediante Kubernetes y tecnologías nativas en la nube. Se utilizaron herramientas que genera tráfico utilizando Locust y Python como lenguaje de programación. Este tráfico es recibido por el balanceador de cargas, otras de las herramientas que se utilizó son Git y docker. Este proyecto se aplicó a los casos infectados actuales de COVID-19 en todo el mundo.

## PROCESO DE DESARROLLO

A continuación se describen de la forma como fue el proceso de desarrollo o instalación y configuración del sistema.

### **Instalación de Kubernetes:**

para instalar Kubernetes se debe descargar el archivo en la dirección que se les proporciona en este apartado: `wget` `curl` `-LO`

`https://storage.googleapis.com/kubernetes-release/release/v1.18.0/bin/linux/amd64/kubect`

después de la descarga ejecutar los siguientes comandos en la consola para la instalación.

```
chmod +x ./kubectl
```

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

### **instalar google cloud sdk**

entrar en el siguiente enlace

[https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk-297.0.0-linux-x86\\_64.tar.gz](https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk-297.0.0-linux-x86_64.tar.gz)

ejecutar en consola

- `tar zxvf google-cloud-sdk-297.0.0-linux-x86_64.tar.gz` `google-cloud-sdk`
- `./google-cloud-sdk/install.sh`
- `#reiniciar consola`
- `gcloud init`
- ingresar credenciales

## Crear Cluster de Kubernetes

en la misma consola se debe pegar los siguientes comandos

- `gcloud container clusters create proyecto2-cluster --num-nodes=1 --tags=allin,allout --enable-legacy-authorization --enable-basic-auth --issue-client-certificate --machine-type=n1-standard-2 --no-enable-network-policy --preemptible --disk-size=20 --enable-autoscaling --max-nodes=3 --min-nodes=1 --zone us-central1-c`
- `gcloud container clusters get-credentials proyecto2-cluster --zone=us-central1-c`

## crear HELM Y NGINX

ejecutar los siguientes comandos

- `wget https://get.helm.sh/helm-v3.4.2-linux-amd64.tar.gz`
- `tar -zxvf helm-v3.4.2-linux-amd64.tar.gz`
- `mv linux-amd64/helm /usr/local/bin/helm`
- `helm repo add stable https://charts.helm.sh/stable`
- `kubectl create namespace nginx-ingress`
- `helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx`
- `helm repo update`
- `helm install nginx-ingress ingress-nginx/ingress-nginx -n nginx-ingress`

## Crear Linkerd

ejecutar los siguientes comandos en consola

- `curl -sL https://run.linkerd.io/install | sh`
- `PATH=$PATH:$HOME/.linkerd2/bin #add line to ~/.profi`
- `linkerd install | kubectl apply -f -`

## Crear NGINX

ejecutar los comandos

- `kubectl -n nginx-ingress get deployment nginx-ingress-ingress-nginx-controller -o yaml | linkerd inject --ingress - | kubectl apply -f -`
- `kubectl -n nginx-ingress describe pods nginx-ingress-ingress-nginx-controller-68657784f8-97sws | grep "linkerd.io/inject: ingress"`

## Crear Blue y Green

- `kubectl create -f dummy.yaml`
- `kubectl get -n project deploy -o yaml \`  
| `linkerd inject - \`  
| `kubectl apply -f -`

## Crear webPage

- `kubectl create namespace webpage`
- `kubectl create deployment pagina -n webpage`  
`--image=bernaldrpp/pagina`
- `kubectl -n webpage expose deployment pagina --port 3000 --target-port=3000 --type NodePort --name=pagina-svc`

## Crear Servidor grpc (client/server)

- `kubectl create namespace grpc`
- `kubectl -n grpc create deployment grpc-servidor`  
`--image=pddomingo/grpc_servidor2`
- `kubectl -n grpc expose deployment grpc-servidor --port=9090 --target-port=9090 --type=NodePort --name=grpc-servidor-svc`

## Crear Servidor Redis

- `kubectl create namespace pubsub`
- `kubectl create deployment golang-sub -n pubsub --image=inrock96/golang-sub`
- `kubectl -n pubsub expose deployment golang-sub --port=9091 --target-port=9091 --type NodePort --name=golang-sub-srv`

# Visualización (sitio web) de tiempo real de los datos insertados.

Covid-19

Todos los casos

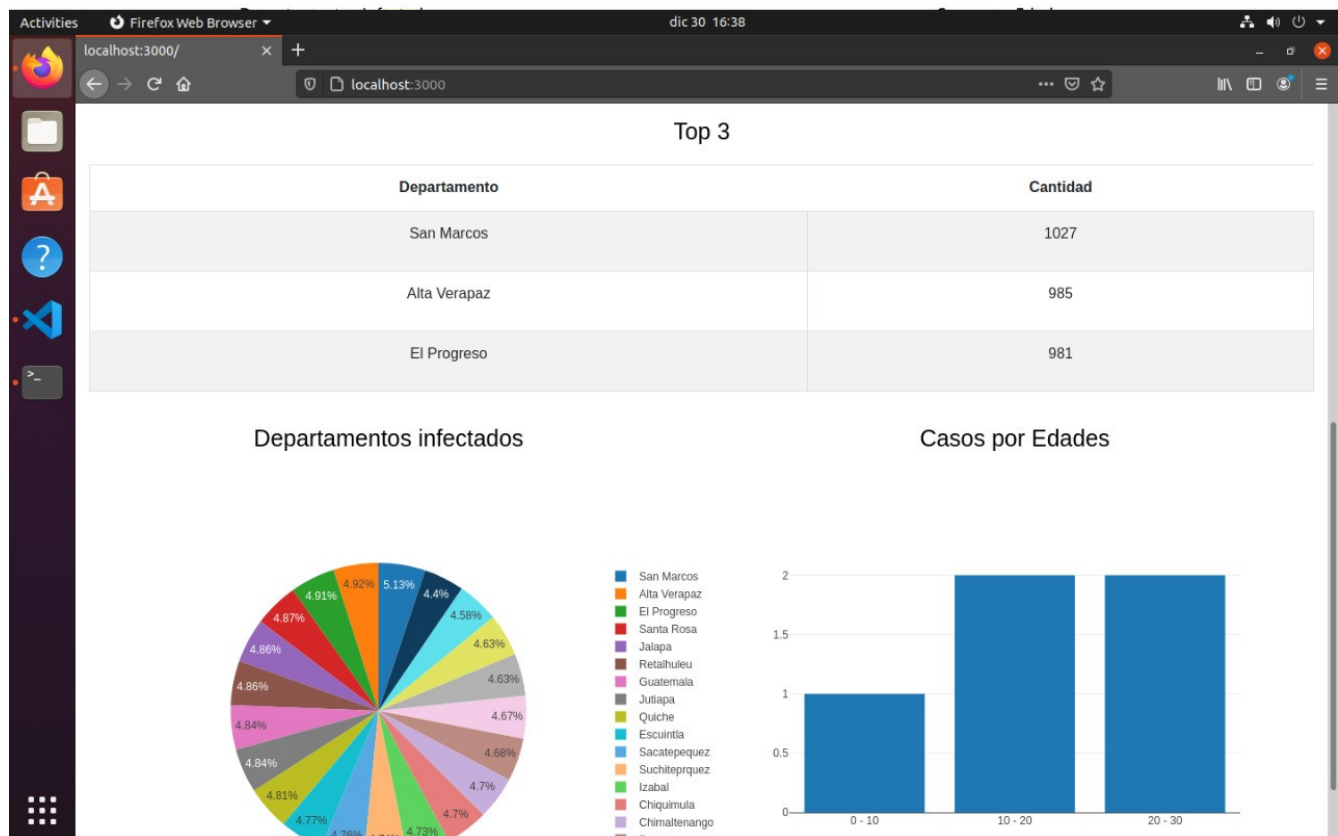
Nombre	Edad	Departamento	Forma de Contagio	Estado
juan	16	peten	imported	on
paco	26	sacatepequez	imported	deceased
pedro	26	zacapa	imported	active
prueba docker	26	zacapa	imported	active
prueba docker ambos	26	zacapa	imported	active
Jose Alberto test1	21	Escuintla1	imported	active
test2	22	test2	imported	active
test3	22	test3	imported	active

Ultimo Caso

Nombre	Edad	Departamento	Forma de Contagio	Estado
pedro	26	zacapa	imported	active
paco	26	sacatepequez	imported	deceased
juan	16	peten	imported	on

Top 3

Departamento	Cantidad
zacapa	3
guate	2
Escuintla1	1



# Observabilidad y Monitoreo

supervision del estado acutal mediante los servicios prometheus y grafana

supervisar bases de datos NoSql mediante  
visualizar informacion mediante grafana

