



Proyecto

Computación
Tolerante a Fallas

Uziel Cornejo Olivares
Carlos Martín García Madrigal
Christopher Iván Andrade Mendoza

¿De que trata nuestra aplicación?

Nuestra aplicación, se trata de una web App para poder realizar reservaciones en un restaurante de una manera muy sencilla para el usuario común con un conocimiento básico de como utilizar una computadora.





¿Qué es Docker?

Plataforma que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute.

Docker o Docker Compose

La diferencia clave entre docker run y docker-compose es que docker run se basa completamente en la línea de comandos, mientras que docker-compose lee los datos de configuración de un archivo YAML.

Además, docker run solo puede iniciar un contenedor a la vez, mientras que docker-compose configurará y ejecutará varios.



Dockerfile

Build
→



Docker
Image

Run
→



Docker
Container

Kubernetes

Plataforma de sistema distribuido de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores.





¿Qué es Istio?

Istio es una malla de servicios (es decir, una capa de redes de servicios modernizada) que ofrece una manera transparente e independiente de cualquier lenguaje de automatizar las funciones de red de una aplicación de forma flexible y sencilla.



OpenShift

OpenShift Container Platform, es un producto en la nube de plataforma como servicio de Red Hat. Los desarrolladores pueden usar Git para desplegar sus aplicaciones Web en los diferentes lenguajes de la plataforma.



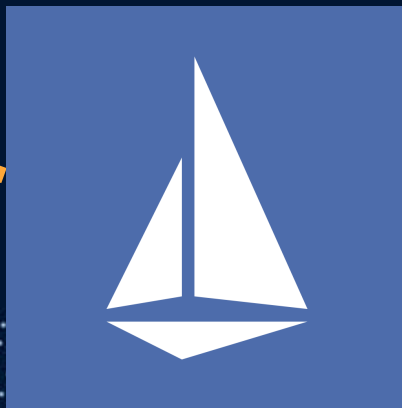
Estructura del proyecto



docker



kubernetes





Estructura de --- **la web App**

Home



Tenemos una página de inicio muy intuitiva que te da información sobre el restaurante y para acceder a las funciones de la página es necesario registrarse y loguearse en ella.

Registro y Login

REGISTRARSE

Nombre de usuario

Correo electrónico

Nombre(s)

Apellido(s)

Contraseña

Confirmar contraseña

¿Ya tienes una cuenta? [Iniciar sesión](#)

INICIAR SESIÓN

Por favor introduce tu nombre de usuario y contraseña!

Nombre de usuario

Contraseña

¿No tienes una cuenta? [Registrarse](#)

Tenemos unos formularios para estas funciones las cuales son tolerante a fallas ya que con django tenemos una función que se llama contrib auth que son decoradores que nos ayudan a la validación de posibles errores de nuestro servicio .

Reservaciones

Reservacion

Nombre

Email

Numero de telefono

Numero de personas

Fecha

Hora

CONFIRMA TU RESERVA

Tenemos una función que nos permite realizar reservaciones al restaurante la cual al momento que esta es generada para mayor seguridad para el consumidor le proporciona un código QR

Comentarios

Tenemos otra función que nos ayuda para saber el feedback que proporcionan los clientes

Me pareció muy bien su servicio

Carlos

MAY 23, 2022, 4:41 A.M.

Me pareció excelente

pepe

MAY 24, 2022, 12:58 A.M.

excelente todo

Deja tu comentario

Content

Content

CONFIRMA TU COMENTARIO

Base de datos

Nuestra base de datos es la que nos proporciona django al iniciar con nuestro proyecto la cual se llama sqlite3, es muy intuitiva y facil de utilizar ya que a traves de migrates puedes hacer modificaciones estructurales de ella y en su panel de administración nos permite observar, hacer cambios o eliminar datos que esten dentro de ella permitiendonos un buen control en todos los apartados de nuestra web

Resturant AdminPanel

Site administration

ABOUTS	
Historia	+ Add Change
Staff	+ Add Change
Virtudes	+ Add Change

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change

BLOG	
Categorias	+ Add Change
Comentarios	+ Add Change
Sitio	+ Add Change

DJANGO SUMMERNOTE	
Attachments	+ Add Change

MEALS	
Categorias	+ Add Change
Platillos	+ Add Change

QR	
Qrs	+ Add Change

REGISTER	
Clients	+ Add Change

RESERVATION	
Reservaciones	+ Add Change

Recent actions

My actions

- [✖ jaime](#)
Reservacion
- [✖ jaime](#)
User
- [✖ me parecia bien](#)
Comentario
- [✖ ale](#)
Reservacion
- [✖ Jaimes](#)
Reservacion
- [✖ Peje](#)
Reservacion
- [✖ xd](#)
Comentario
- [✖ dsad](#)
Comentario
- [✖ fdgdfg](#)
Comentario
- [✖ pepe](#)
User

USO

En el siguiente link mostraremos un video de la página funcionando:

<https://youtu.be/0z8B3hOwRAk>



Creación de imagen con Dockerfile

Para poder crear nuestra imagen, necesitamos primero crear el Dockerfile del microservicio a ejecutar



Creación de archivo Dockerfile

```
Dockerfile x
1 FROM python:3.8-slim-buster
2
3 WORKDIR /app
4
5 COPY ./requirements.txt ./
6 RUN pip install -r requirements.txt
7
8 COPY ./ ./
9
10 CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
11
```

Construcción de imagen para Docker

Una vez creado el Dockerfile, construimos la imagen docker con el siguiente comando:

docker build -t restaurant .

```
C:\Users\PC\Desktop\Nueva carpeta (11)\Nueva carpeta\src>docker build -t restaurant .
[+] Building 42.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 225B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim-buster        2.9s
=> [internal] load build context                                                  0.3s
=> => transferring context: 13.91MB                                             0.3s
=> [1/5] FROM docker.io/library/python:3.8-slim-buster@sha256:db90d6607a1c2ccb9bbc8a6fdf286d183c44bf8e9c92193ecd 24.0s
=> => resolve docker.io/library/python:3.8-slim-buster@sha256:db90d6607a1c2ccb9bbc8a6fdf286d183c44bf8e9c92193ecd 0.0s
=> => sha256:eae6fe00ee3c3d4a62faa08362ff949fcf2fcb418bcc5ff932c43613a1122c8b 11.29MB / 11.29MB 9.2s
=> => sha256:db90d6607a1c2ccb9bbc8a6fdf286d183c44bf8e9c92193ecd423254e85680ce 988B / 988B 0.0s
=> => sha256:77c3488ce5476695aa0b018e6dcfe6c4a2bf2e5d1e4781a5e20ae0e8086a9b58 1.37kB / 1.37kB 0.0s
=> => sha256:5cc94b67760dc542cb56567c7d7f5d9d19d9ec4bf6d1b5494fc7a65d4e822058 7.53kB / 7.53kB 0.0s
=> => sha256:32820e52a00eb22dc76d70d992be7082cd24b636cfb597ff3951d29a821b46b9 27.14MB / 27.14MB 21.4s
=> => sha256:6563993b0507bc602c27e376c26abd7766a8c8fa094da83beac1c0cfc8b366b 2.78MB / 2.78MB 2.6s
=> => sha256:bbcf4b417983bad30035c8d28bae1dbca13f8e7636def13e2d090caec61c9134 234B / 234B 2.7s
=> => sha256:b2f1e8a99da34fd5fdbc38fcf88bd78ab7d248f3d2feafa60b5375dfb5f13e0f 3.18MB / 3.18MB 5.4s
=> => extracting sha256:32820e52a00eb22dc76d70d992be7082cd24b636cfb597ff3951d29a821b46b9 1.3s
=> => extracting sha256:6563993b0507bc602c27e376c26abd7766a8c8fa094da83beac1c0cfc8b366b 0.2s
=> => extracting sha256:eae6fe00ee3c3d4a62faa08362ff949fcf2fcb418bcc5ff932c43613a1122c8b 0.5s
=> => extracting sha256:bbcf4b417983bad30035c8d28bae1dbca13f8e7636def13e2d090caec61c9134 0.0s
=> => extracting sha256:b2f1e8a99da34fd5fdbc38fcf88bd78ab7d248f3d2feafa60b5375dfb5f13e0f 0.3s
=> [2/5] WORKDIR /app                                                            0.6s
=> [3/5] COPY ./requirements.txt ./                                              0.0s
=> [4/5] RUN pip install -r requirements.txt                                     13.5s
=> [5/5] COPY ./ ./                                                             0.1s
=> exporting to image                                                            0.8s
=> => exporting layers                                                            0.8s
=> => writing image sha256:93b04070f90a7b30953f7b93255b8d319b43342fa2595a294ac042244579fd85 0.0s
=> => naming to docker.io/library/restaurant                                   0.0s
```

Ejecutamos imagen de Docker

Para verificar que la imagen se ha creado correctamente, lo corremos con el siguiente comando:

docker run -p 8000:8000 restaurant

```
C:\Windows\System32\cmd.exe - docker run -p 8000:8000 restaurant
C:\Users\PC\Desktop\Nueva carpeta (11)\Nueva carpeta\src>docker run -p 8000:8000 restaurant
[26/Nov/2022 00:15:52] "GET / HTTP/1.1" 200 13040
[26/Nov/2022 00:16:53] "GET /logout/ HTTP/1.1" 302 0
[26/Nov/2022 00:16:53] "GET / HTTP/1.1" 200 12956
[26/Nov/2022 00:16:55] "GET /login/ HTTP/1.1" 200 2779
[26/Nov/2022 00:16:55] "GET /static/main.css HTTP/1.1" 404 1749
[26/Nov/2022 00:17:08] "POST /login/ HTTP/1.1" 302 0
[26/Nov/2022 00:17:08] "GET / HTTP/1.1" 200 13040
[26/Nov/2022 00:17:19] "GET /reserve_table/ HTTP/1.1" 200 8745
[26/Nov/2022 00:17:59] "GET /admin/ HTTP/1.1" 200 10738
[26/Nov/2022 00:17:59] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 412
[26/Nov/2022 00:17:59] "GET /static/admin/css/base.css HTTP/1.1" 200 16225
[26/Nov/2022 00:17:59] "GET /static/admin/css/responsive.css HTTP/1.1" 200 17976
[26/Nov/2022 00:17:59] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[26/Nov/2022 00:17:59] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331
[26/Nov/2022 00:17:59] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380
[26/Nov/2022 00:17:59] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 82564
[26/Nov/2022 00:17:59] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 81348
[26/Nov/2022 00:17:59] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 80304
[26/Nov/2022 00:18:10] "GET /admin/reservation/reservation/ HTTP/1.1" 200 4408
[26/Nov/2022 00:18:10] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6170
[26/Nov/2022 00:18:10] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 363
[26/Nov/2022 00:18:10] "GET /static/admin/js/core.js HTTP/1.1" 200 7135
[26/Nov/2022 00:18:10] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 6918
[26/Nov/2022 00:18:10] "GET /admin/jsi18n/ HTTP/1.1" 200 3185
[26/Nov/2022 00:18:10] "GET /static/admin/js/actions.js HTTP/1.1" 200 6538
[26/Nov/2022 00:18:10] "GET /static/admin/js/unlabeled.js HTTP/1.1" 200 8072
```

Subir contenedor a DockerHub



Inicio de sesión

Si no tenemos cuenta, nos registramos en <https://hub.docker.com/>



Subimos la imagen


Ya teniendo construida nuestra imagen de Docker, ahora la subimos a DockerHub con un Push to Hub



Comprobamos que se subió


Para comprobarlo nos dirigimos a nuestro repositorio en Docker Hub

Subir contenedor a DockerHub


 docker hub

Explore Repositories Organizations Help ▾

Upgrade

 chrisiam21 ▾

Explore > uzielcornejo/restaurante > 2686197d51fb >



uzielcornejo/restaurante:2686197d51fb

DIGEST: sha256:a3121bdca294d2777f278b0ba4cd8c54fbef088d9ef10626db8fa135b112577

OS/ARCH	COMPRESSED SIZE ⓘ	LAST PUSHED	TYPE
linux/amd64	103.03 MB	a day ago by uzielcornejo	Image

IMAGE LAYERS ⓘ

1	ADD file ... in /	25.88 MB
2	CMD ["bash"]	0 B
3	ENV PATH=/usr/local/bin:/usr/local/sbin:/usr/local/...	0 B
4	ENV LANG=C.UTF-8	0 B
5	/bin/sh -c set -eux; apt-get	2.65 MB
6	ENV GPG_KEY=E3FF2839C048B25C084DEBE9B26995E310250568	0 B
7	ENV PYTHON_VERSION=3.8.15	0 B

Command

ADD file:9461639b945ced6fb6164491a7e0131261a1b7d69264cce516e75c71e4df22d2 in /



Kubernetes

Para utilizar kubernetes,
necesitaremos de la
herramienta minikube

Iniciamos minikube

```
(virtual) Uziels-MacBook-Pro:src uzi$ minikube start
🐳 minikube v1.28.0 on Darwin 12.6
🌟 Using the docker driver based on existing profile
👍 Starting control plane node minikube in cluster minikube
🚀 Pulling base image ...
🔄 Updating the running docker "minikube" container ...
🌐 Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
🔊 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: default-storageclass
🏠 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
(virtual) Uziels-MacBook-Pro:src uzi$ █
```

Ver el puerto donde se encuentra la imagen Docker

Para ver los puertos donde se encuentra nuestra imagen utilizamos el comando:

docker ps

```
(virtual) Uziels-MacBook-Pro:src uzi$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2be49ec5afcc	74f61a5d91f8	"python manage.py ru..."	24 hours ago	Up 24 hours		lucid_lehm
ann						
8e75b2fd8b09	restaurante:latest	"python manage.py ru..."	26 hours ago	Up 26 hours		tender_hel
lman						
135d493ee38d	kicbase/stable:v0.0.36	"/usr/local/bin/entr..."	2 weeks ago	Up 27 hours	0.0.0.0:61010->22/tcp, 0.0.0.0:61011->2376/tcp, 0.0.0.0:61013->5000/tcp, 0.0.0.0:61009->8443/tcp, 0.0.0.0:61012->32443/tcp	minikube

```
(virtual) Uziels-MacBook-Pro:src uzi$
```

Verificación de los pods

Para verificar que los pods se encuentren corriendo correctamente utilizamos el siguiente comando:

Kubectl get pod

```
(virtual) Uziels-MacBook-Pro:istio-1.16.0 uzi$ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
adservice-67b89697dd-t6cl6	2/2	Running	12 (17m ago)	13d
cartservice-5699fd487d-wzghh	2/2	Running	14 (17m ago)	13d
checkoutservice-8794c6fb-dlx fz	2/2	Running	11 (17m ago)	13d
currencyservice-7ddb4c6866-ptwgx	2/2	Running	12 (17m ago)	13d
emailservice-6685d77684-h6m6b	2/2	Running	13 (17m ago)	13d
frontend-7966dc59fb-lwbwd	2/2	Running	15 (17m ago)	13d
loadgenerator-6c7c5f96b-vdpb7	2/2	Running	10 (17m ago)	13d
paymentservice-569f84557b-tbtsv	2/2	Running	13 (17m ago)	13d
productcatalogservice-d87dfdd44-fmqzd	2/2	Running	11 (17m ago)	13d
python-webapp-7f74bcd69c-6trf9	2/2	Running	0	71s
python-webapp-7f74bcd69c-fn5p2	2/2	Running	0	71s
recommendationservice-65487869cd-mhprx	2/2	Running	11 (15m ago)	13d
redis-cart-7f557cdb55-k8pzq	2/2	Running	13 (17m ago)	13d
shippingservice-5cb87cffb6-t5t2p	2/2	Running	11 (17m ago)	13d

```
(virtual) Uziels-MacBook-Pro:istio-1.16.0 uzi$
```

Verificación de los servicios

Obtenemos el servicio en una IP interna:

Kubectl get services

```
(virtual) Uziels-MacBook-Pro:istio-1.16.0 uzi$ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
adservice	ClusterIP	10.105.56.235	<none>	9555/TCP	13d
cartservice	ClusterIP	10.107.67.176	<none>	7070/TCP	13d
checkoutservice	ClusterIP	10.102.104.183	<none>	5050/TCP	13d
currencyservice	ClusterIP	10.102.205.145	<none>	7000/TCP	13d
emailservice	ClusterIP	10.103.48.144	<none>	5000/TCP	13d
frontend	ClusterIP	10.110.216.231	<none>	80/TCP	13d
frontend-external	LoadBalancer	10.104.172.220	<pending>	80:32675/TCP	13d
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	20d
paymentservice	ClusterIP	10.111.98.93	<none>	50051/TCP	13d
productcatalogservice	ClusterIP	10.109.131.43	<none>	3550/TCP	13d
recommendationservice	ClusterIP	10.107.102.99	<none>	8080/TCP	13d
redis-cart	ClusterIP	10.106.99.47	<none>	6379/TCP	13d
shippingservice	ClusterIP	10.108.252.245	<none>	50051/TCP	13d

```
(virtual) Uziels-MacBook-Pro:istio-1.16.0 uzi$
```


Ejecutamos el panel minikube

Con el siguiente comando nos dirigimos al dashboard de minikube:

minikube dashboard

```
(virtual) Uziels-MacBook-Pro:istio-1.16.0 uzi$ minikube dashboard
```

```
✖ Enabling dashboard ...
```

```
  ■ Using image docker.io/kubernetesui/dashboard:v2.7.0
```

```
  ■ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
```

```
⚠ Some dashboard features require the metrics-server addon. To enable all features please run:
```

```
minikube addons enable metrics-server
```

```
🟡 Verifying dashboard health ...
```

```
🚀 Launching proxy ...
```

```
🟡 Verifying proxy health ...
```

```
🔗 Opening http://127.0.0.1:59140/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```

```
□
```

Ya podemos acceder al panel minikube



Workload Status

Expand card




Deployments

Name	Images	Labels	Pods	Created ↑
python-webapp	uzielcornejo/restaurante:2686197d51fb	app: web	2 / 2	10 minutes ago
currencyservice	gcr.io/google-samples/microservices-de mo/currencyservice:v0.4.1	-	1 / 1	13 days ago
adservice	gcr.io/google-samples/microservices-de mo/adservice:v0.4.1	-	1 / 1	13 days ago
loadgenerator	gcr.io/google-samples/microservices-de mo/loadgenerator:v0.4.1	-	1 / 1	13 days ago
paymentservice	gcr.io/google-samples/microservices-de mo/paymentservice:v0.4.1	-	1 / 1	13 days ago
productcatalogservice	gcr.io/google-samples/microservices-de mo/productcatalogservice:v0.4.1	-	1 / 1	13 days ago
cartservice	gcr.io/google-samples/microservices-de mo/cartservice:v0.4.1	-	1 / 1	13 days ago
redis-cart	redis:alpine	-	1 / 1	13 days ago
shippingservice	gcr.io/google-samples/microservices-de mo/shippingservice:v0.4.1	-	1 / 1	13 days ago
checkoutservice	gcr.io/google-samples/microservices-de mo/checkoutservice:v0.4.1	-	1 / 1	13 days ago


1 - 10 of 13 |< < > >

Ya podemos acceder al panel minikube

Pods

Name	Images	Labels	Node	Status	Restarts
 python-webapp-7f74bcd69c-6trf9	uzielcornejo/restaura 86197d51fb	pod-template-has h: 7f74bcd69c	minikube	Running	0
	docker.io/istio/proxy 6.0	security.istio.io/tls Mode: istio			
 python-webapp-7f74bcd69c-fn5p2	uzielcornejo/restaura 86197d51fb	pod-template-has h: 7f74bcd69c	minikube	Running	0
	docker.io/istio/proxy 6.0	security.istio.io/tls Mode: istio			
 redis-cart-7f557cdb55-k8pzq	redis:alpine	pod-template-has h: 7f557cdb55	minikube	Running	13
	docker.io/istio/proxy 2:1.15.3	security.istio.io/tls Mode: istio			

Nodes

Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	CPU capacity (cores)	Memory requests (bytes)	Memory limits (bytes)	Memory capacity (bytes)	Pods
 minikube	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64 Show all	True	4.34 (54.25%)	32.83 (410.31%)	8.00	5.44Gi (70.88%)	18.65Gi (242.99%)	7.67Gi	29 (26.36%)

Istio



Istio

Connect, secure, control, and observe services.

Generamos los archivos de configuración

Aplicamos la configuración de los archivos generados con kubernetes

kubectl apply -f kubernetes-manifest.yaml

```
Uziels-MacBook-Pro:istio-1.16.0 uzi$ kubectl apply -f kubernetes-manifests.yaml
deployment.apps/emailservice created
service/emailservice created
deployment.apps/checkoutservice created
service/checkoutservice created
deployment.apps/recommendationservice created
service/recommendationservice created
deployment.apps/frontend created
service/frontend created
service/frontend-external created
deployment.apps/paymentservice created
service/paymentservice created
deployment.apps/productcatalogservice created
service/productcatalogservice created
deployment.apps/cartservice created
service/cartservice created
deployment.apps/loadgenerator created
deployment.apps/currencyservice created
service/currencyservice created
deployment.apps/shippingservice created
service/shippingservice created
deployment.apps/redis-cart created
service/redis-cart created
deployment.apps/adservice created
service/adservice created
Uziels-MacBook-Pro:istio-1.16.0 uzi$
```

Verificamos las instancias creadas de los pods

kubectl get pod

```
Uziels-MacBook-Pro:istio-1.16.0 uzi$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
adservice-67b89697dd-9669n         0/1     Running   0           2m42s
cartservice-5699fd487d-rkxbx       1/1     Running   0           2m43s
checkoutservice-8794c6fb-zkcfr      1/1     Running   0           2m43s
currencyservice-7ddb4c6866-bbgmb    1/1     Running   0           2m43s
emailservice-6685d77684-f6xjx      1/1     Running   0           2m43s
frontend-7966dc59fb-fm4gb          1/1     Running   0           2m43s
loadgenerator-6c7c5f96b-xk42d      1/1     Running   0           2m43s
paymentservice-569f84557b-vrnlb     1/1     Running   0           2m43s
productcatalogservice-d87dfdd44-rl2nn 1/1     Running   0           2m43s
recommendationservice-65487869cd-w9vfr 1/1     Running   0           2m43s
redis-cart-7f557cdb55-khw6l        1/1     Running   0           2m42s
shippingservice-5cb87cffb6-m7nxv    1/1     Running   0           2m42s
Uziels-MacBook-Pro:istio-1.16.0 uzi$
```


Software de Istio

Verificamos que se
hayan creado las
instancias de los pods

**kubectl get svc -n
istio-system**

```
C:\Istio\istio-1.15.3>kubectl get svc -n istio-system
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana	ClusterIP	10.105.149.38	<none>	3000/TCP	22s
istio-ingressgateway	LoadBalancer	10.102.54.160	<pending>	15021:31429/TCP,80:31357/TCP,443:32596/TCP	47h
istiod	ClusterIP	10.106.27.225	<none>	15010/TCP,15012/TCP,443/TCP,15014/TCP	47h
jaeger-collector	ClusterIP	10.97.211.121	<none>	14268/TCP,14250/TCP,9411/TCP	21s
kiali	ClusterIP	10.101.51.188	<none>	20001/TCP,9090/TCP	21s
prometheus	ClusterIP	10.106.5.28	<none>	9090/TCP	20s
tracing	ClusterIP	10.98.45.228	<none>	80/TCP,16685/TCP	22s
zipkin	ClusterIP	10.104.15.58	<none>	9411/TCP	22s

Acceder a Kiali

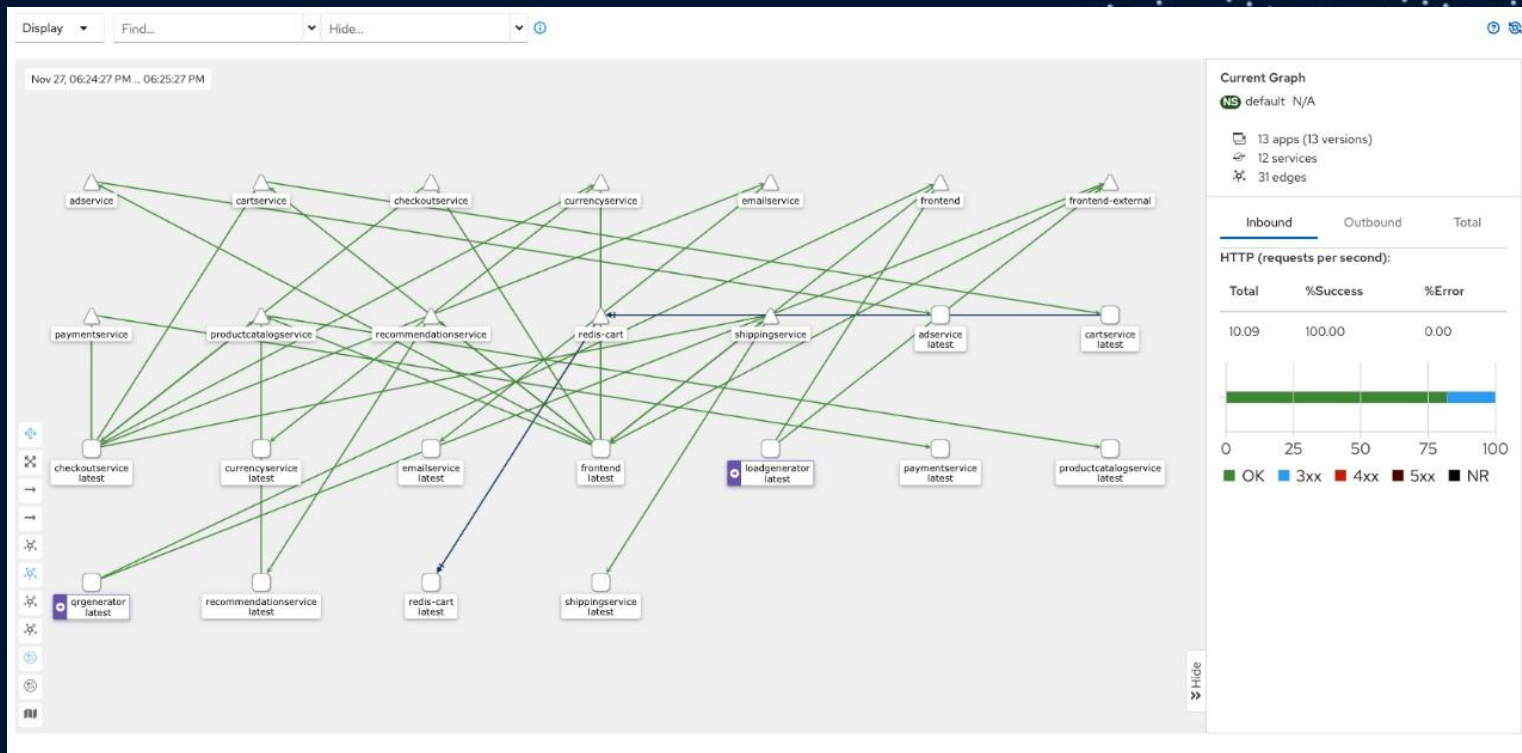
Accedemos a Kiali, una
consola de
administración para Istio

**kubectl port-forward
svc/kiali -n istio-system
20001**

```
C:\Istio\istio-1.15.3>kubectl get svc -n istio-system
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana	ClusterIP	10.105.149.38	<none>	3000/TCP	22s
istio-ingressgateway	LoadBalancer	10.102.54.160	<pending>	15021:31429/TCP,80:31357/TCP,443:32596/TCP	47h
istiod	ClusterIP	10.106.27.225	<none>	15010/TCP,15012/TCP,443/TCP,15014/TCP	47h
jaeger-collector	ClusterIP	10.97.211.121	<none>	14268/TCP,14250/TCP,9411/TCP	21s
kiali	ClusterIP	10.101.51.188	<none>	20001/TCP,9090/TCP	21s
prometheus	ClusterIP	10.106.5.28	<none>	9090/TCP	20s
tracing	ClusterIP	10.98.45.228	<none>	80/TCP,16685/TCP	22s
zipkin	ClusterIP	10.104.15.58	<none>	9411/TCP	22s

Funcionamiento de kiali!





OpenShift

Plataforma para desplegar la aplicación Web

Añadimos el proyecto para ejecutarse en la plataforma OpenShift

Primero debemos acceder a OpenShift creando una cuenta.

A screenshot of the Red Hat login page. It has a white background with the text 'Iniciar sesión en su cuenta de Red Hat' at the top. Below it is a text input field labeled 'Nombre de usuario o correo electrónico de Red Hat'. Under the field is a red button with the text 'Siguiendo'. Below the button are several social media icons (GitHub, Docker, LinkedIn, Twitter, Facebook, Google, Microsoft). At the bottom, there is a link 'Regístrate para crear una cuenta con Red Hat' and a smaller link '¿Olvidó su contraseña?'.

Una vez teniendo una cuenta añadimos el proyecto desde un repositorio dirigiéndonos al apartado “+Add”

Y seleccionamos “Get from Git” para importar nuestro proyecto desde nuestro repositorio en Github

A screenshot of a dialog box titled 'Container images'. It contains the text 'Deploy an existing Image from an Image registry or Image stream tag'.A screenshot of the OpenShift 'Developer' dashboard. It has a dark theme with the Red Hat logo and 'OpenShift Dedicated' text at the top. Below that is a 'Developer' header with a dropdown arrow. At the bottom, there is a '+Add' button.

Configuramos la importación del proyecto a OpenShift

Colocamos la dirección del repositorio a importar



Import from Git

Git

Git Repo URL *

`https://github.com/chrisIAM21/ProyectoN.git` ✓

Validated

> [Show advanced Git options](#)

✓ **Multiple import strategies detected**
The Dockerfile at Dockerfile is recommended.

Dockerfile [Edit Import Strategy](#)

General

Configuramos la importación del proyecto a OpenShift

Configuramos la aplicación



Advanced options

Target port

8080

Target port for traffic.

☒ Create a route
Exposes your component at a public URL.

▼ Hide advanced Routing options

Hostname

carlos-restaurante.herokuapp.com

Public hostname for the route. If not specified, a hostname is generated.

Path

/

Path that the router watches to route traffic to the service.

Security

☐ Secure Route
Routes can be secured using several TLS termination types for serving certificates.

Labels

app.io/type=frontend

Additional labels which are only added to the Route resource.

Click on the names to access advanced options for Health checks, Build configuration, Deployment, Scaling, Resource limits, and Labels.

Y seleccionamos "create" para que se empiece a construir nuestra aplicación web



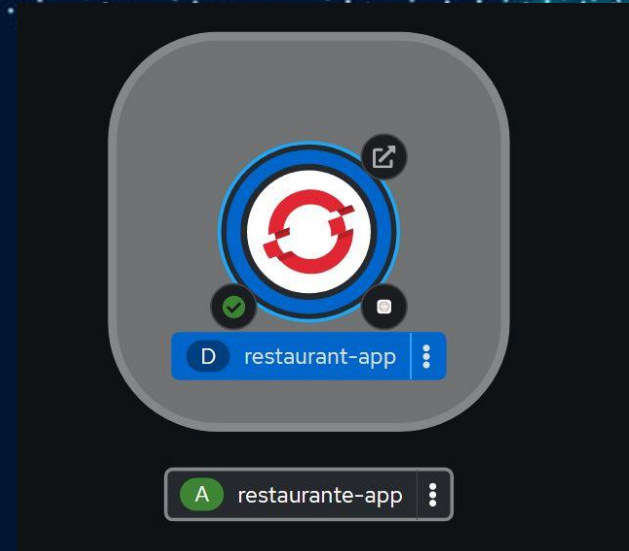
Create Cancel

Nuestra aplicación ya está lista!

Ya importado y creada nuestra aplicación, podemos monitorearla y verla desde la plataforma OpenShift

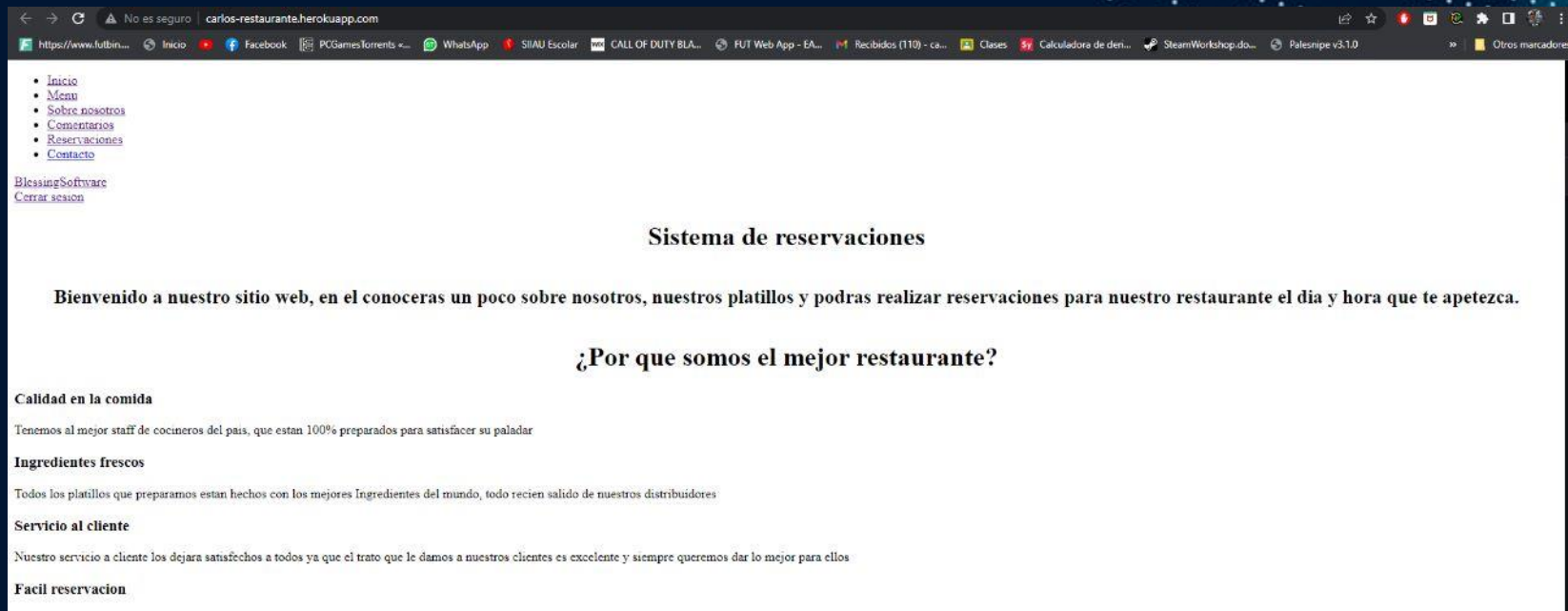


Ahora nuestra aplicación web se volvió mucho más robusta a tolerar posibles fallas, ya que si esta deja de funcionar, el mismo sistema se encarga de recuperar y volver a subir la imagen



Enlace: <http://carlos-restaurante.herokuapp.com/>

Aplicación funcionando desde OpenShift



The screenshot shows a web browser window with the address bar displaying "carlos-restauranta.herokuapp.com". The browser's address bar also shows "No es seguro" (Not safe). The browser's tabs include "https://www.futbin...", "Inicio", "Facebook", "PGamesTorrents", "WhatsApp", "SIAU Escolar", "CALL OF DUTY BLA...", "FUT Web App - EA...", "Recibidos (110) - ca...", "Clases", "Calculadora de den...", "SteamWorkshop.do...", "Palesnipe v3.1.0", and "Otros marcadores".

The website content includes a navigation menu with links: Inicio, Menu, Sobre nosotros, Comentarios, Reservaciones, and Contacto. Below the menu, there are links for "BlessingSoftware" and "Cerrar session".

Sistema de reservaciones

Bienvenido a nuestro sitio web, en el conoceras un poco sobre nosotros, nuestros platillos y podras realizar reservaciones para nuestro restaurante el dia y hora que te apetezca.

¿Por que somos el mejor restaurante?

Calidad en la comida

Tenemos al mejor staff de cocineros del pais, que estan 100% preparados para satisfacer su paladar

Ingredientes frescos

Todos los platillos que preparamos estan hechos con los mejores Ingredientes del mundo, todo recien salido de nuestros distribuidores

Servicio al cliente

Nuestro servicio a cliente los dejara satisfechos a todos ya que el trato que le damos a nuestros clientes es excelente y siempre quremos dar lo mejor para ellos

Facil reservacion

Aplicación funcionando desde OpenShift

← → No es seguro | carlos-restaurante.herokuapp.com/login/

https://www.futbin... Inicio Facebook PCGamesTorrents WhatsApp SIAU Escoler CALL OF DUTY BLA... FUT Web App - EA... Recibidos (110) - ca... Clases Calculadora de deri... SteamWorkshop...

INICIAR SESIÓN

Por favor introduce tu nombre de usuario y contraseña!

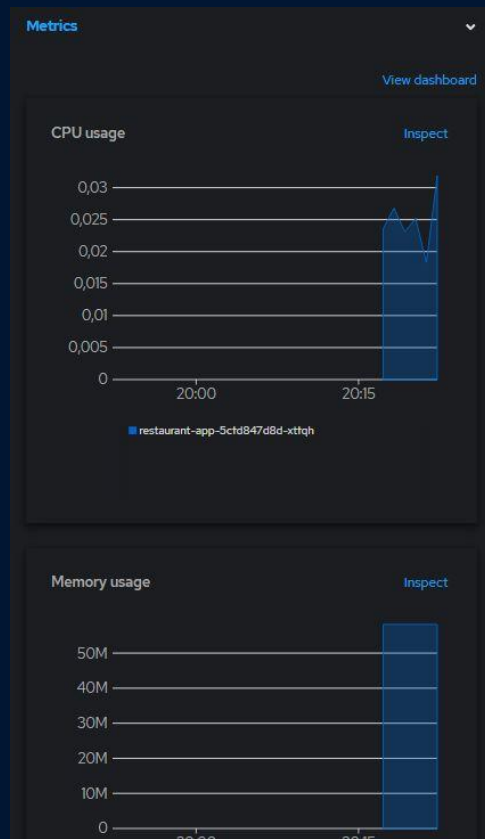
Nombre de usuario

Contraseña

Iniciar sesión

¿No tienes una cuenta? [Regístrase](#)

Monitoreo de la web App desde OpenShift



restaurant-app Actions

Details Resources Observe

1 Pod

Name
restaurant-app

Update strategy
RollingUpdate

Namespace
NS carlos-garcia09-dev

Max unavailable
25% of 1 pod

Labels Edit

- app=restaura...
- app.kub...=restaura...
- app.kub...=restaura...
- app.kub...=restaura...
- app.kub...=restaura...

Max surge
25% greater than 1 pod

Progress deadline seconds
600 seconds

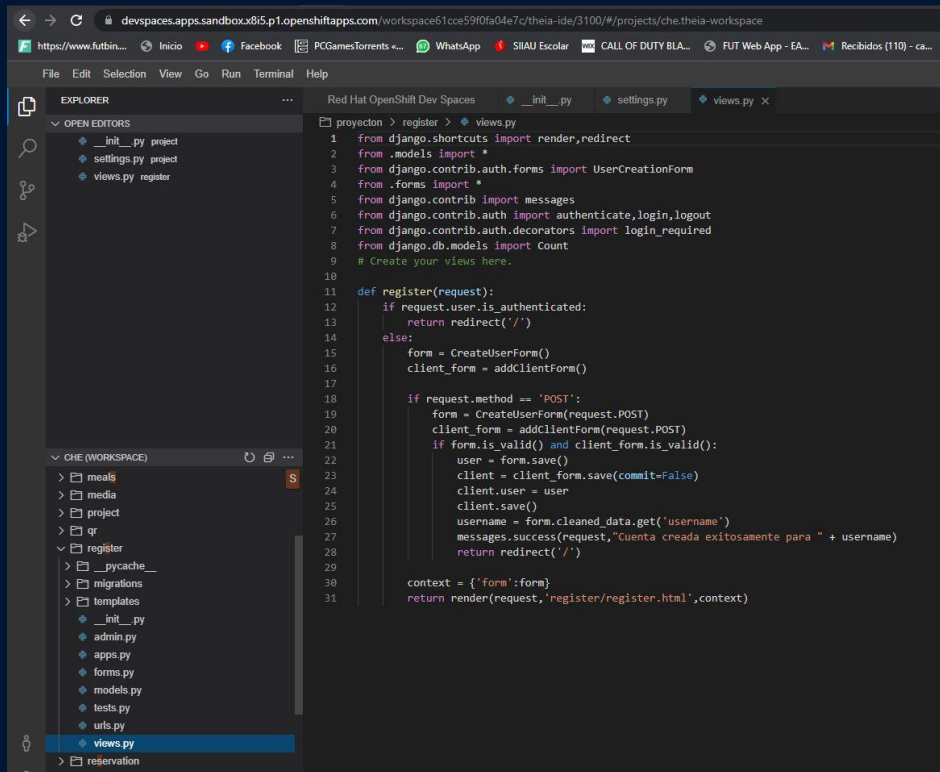
Min ready seconds
Not configured

Pod selector
Q app=restaurant-app

PodDisruptionBudgets
No PodDisruptionBudgets

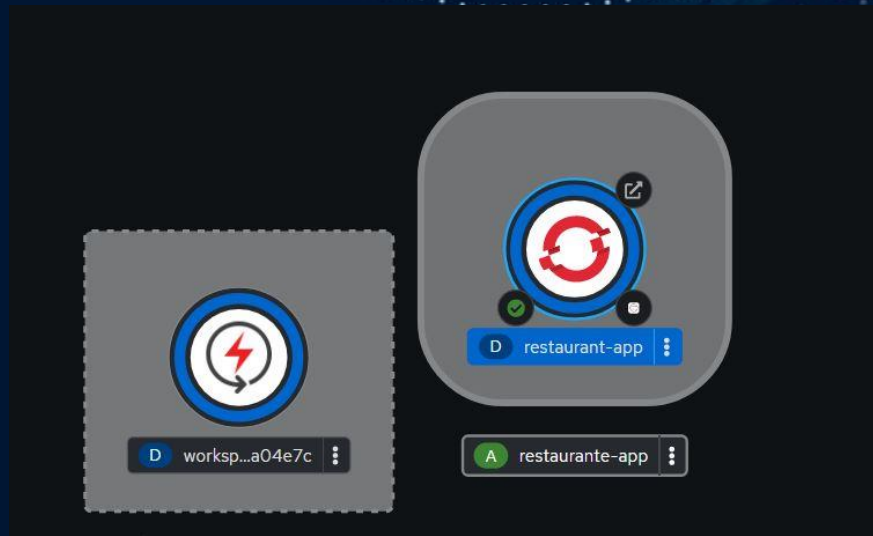
Node selector
No selector

Editar el código de nuestro proyecto desde OpenShift



The screenshot shows the OpenShift web console interface. The top navigation bar includes the OpenShift logo and a search bar. The main content area is divided into two panes. The left pane, titled 'EXPLORER', shows a file tree for the project 'restaurant-app'. The right pane, titled 'EDITOR', shows the code for 'views.py'. The code is a Django view function that handles the registration process. It includes imports for Django's authentication and forms modules, and it defines a 'register' function that checks if the user is authenticated and creates a new user if not.

```
1 from django.shortcuts import render, redirect
2 from .models import *
3 from django.contrib.auth.forms import UserCreationForm
4 from .forms import *
5 from django.contrib import messages
6 from django.contrib.auth import authenticate, login, logout
7 from django.contrib.auth.decorators import login_required
8 from django.db.models import Count
9 # Create your views here.
10
11 def register(request):
12     if request.user.is_authenticated:
13         return redirect('/')
14     else:
15         form = CreateUserForm()
16         client_form = addClientForm()
17
18         if request.method == 'POST':
19             form = CreateUserForm(request.POST)
20             client_form = addClientForm(request.POST)
21             if form.is_valid() and client_form.is_valid():
22                 user = form.save()
23                 client = client_form.save(commit=False)
24                 client.user = user
25                 client.save()
26                 username = form.cleaned_data.get('username')
27                 messages.success(request, "Cuenta creada exitosamente para " + username)
28                 return redirect('/')
29
30 context = {'form': form}
31 return render(request, 'register/register.html', context)
```





ChaosMonkey

Eliminador de Clusters

Instalamos Cheeky Monkey (Chaos Tools)

1) Visitamos el siguiente repositorio:

<https://github.com/richstokes/cheekymonkey>

2) Descargamos el repositorio de richstokes a nuestro ordenador, el cual contiene todos los archivos para ejecutar Cheeky Monkey.

3) Dentro de la carpeta de los archivos, ejecutamos el siguiente comando en una terminal:

pip install -r requirements.txt

De esta forma se instalará todo lo necesario para ejecutar el programa

```
Uziels-MacBook-Pro:cheekymonkey-master uzi$ pip install -r requirements.txt
DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion at https://github.com/Homebrew/homebrew-core/issues/76621
Collecting arcade==2.5.2
  Downloading arcade-2.5.2-py3-none-any.whl (38.3 MB)
    38.3/38.3 MB 6.7 MB/s eta 0:00:00
Collecting kubernetes==11.0.0
  Downloading kubernetes-11.0.0-py3-none-any.whl (1.5 MB)
    1.5/1.5 MB 6.0 MB/s eta 0:00:00
Collecting pymunk==5.7
  Downloading pymunk-5.7-py2.py3-none-macosx_10_13_x86_64.whl (201 kB)
    202.0/202.0 kB 3.6 MB/s eta 0:00:00
Collecting pillow==8.0
  Downloading Pillow-8.4.0-cp39-cp39-macosx_10_10_x86_64.whl (3.0 MB)
    3.0/3.0 MB 5.5 MB/s eta 0:00:00
Collecting pygame<2, >=1.5.11
  Downloading pygame-1.5.27-py3-none-any.whl (1.1 MB)
    1.1/1.1 MB 5.6 MB/s eta 0:00:00
Collecting pyyaml==5.3
  Downloading PyYAML-5.4.1-cp39-cp39-macosx_10_9_x86_64.whl (259 kB)
    259.3/259.3 kB 3.6 MB/s eta 0:00:00
Collecting channel==1.7
```

Ejecutamos Cheeky Monkey (Chaos Tools)

Una vez instalados los requerimientos, ya podemos correr el programa con el siguiente comando:

```
python cheekymonkey.py
```



Demostración Cheeky Monkey

Video:

<https://youtu.be/17w8nXz66dc>



Destruimos pods del proyecto (Cheeky Monkey)

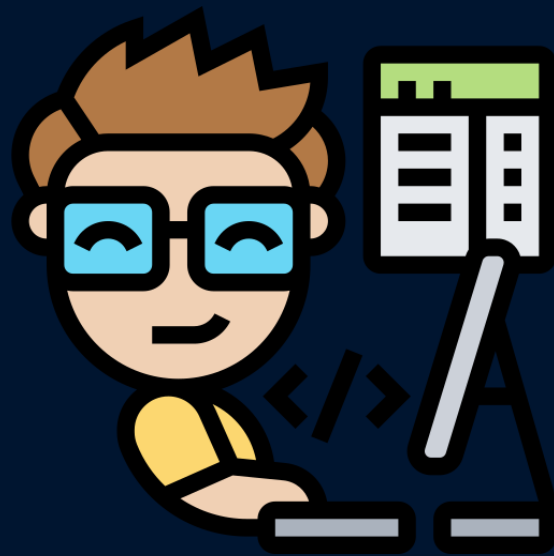
Last pod killed: redis-cart-7f557cdb55-khw6l

```
Uziels-MacBook-Pro:istio-1.16.0 uzi$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
adservice-67b89697dd-9rkhh          1/1     Running   0           4m49s
cartservice-5699fd487d-rkxbx        1/1     Running   1 (32m ago) 4d22h
checkoutservice-8794c6fb-zkcfr       1/1     Running   1 (32m ago) 4d22h
currencyservice-7ddb4c6866-r5twj     1/1     Running   0           4m45s
emailservice-6685d77684-f6xjx       1/1     Running   1 (32m ago) 4d22h
frontend-7966dc59fb-fm4gb           1/1     Running   1 (32m ago) 4d22h
loadgenerator-6c7c5f96b-xk42d        1/1     Running   1 (32m ago) 4d22h
paymentservice-569f84557b-46cln      1/1     Running   0           3m13s
productcatalogservice-d87dfdd44-rl2nn 1/1     Running   1 (32m ago) 4d22h
recommendationservice-65487869cd-w9vfr 1/1     Running   1 (32m ago) 4d22h
redis-cart-7f557cdb55-5skpf          1/1     Running   0           8m49s
shippingservice-5cb87cffb6-m7nxv     1/1     Running   1 (32m ago) 4d22h
Uziels-MacBook-Pro:istio-1.16.0 uzi$
```

```
Uziels-MacBook-Pro:cheekymonkey-master uzi$ python cheekymonkey.py
INFO: Starting in online mode
INFO: Excluding namespaces:
2022-12-02 17:50:17.178 Python[59770:1465003] ApplePersistenceIgnoreState: Existing state will not be touched. New state will be written to /var/folders/qb/f74k7v_j0wggdlpfz0tys2nc0000gn/T/org.python.python.savedState
INFO: Attempting to connect to Kubernetes API host..
INFO: Connected to Kubernetes host: https://127.0.0.1:64768
INFO: Number of pods: 21
INFO: Creating 21 crates
INFO: Updating pod list from: https://127.0.0.1:64768
INFO: Number of pods: 21
INFO: Killing Random pod: kube-apiserver-minikube from namespace: kube-system
INFO: Updating pod list from: https://127.0.0.1:64768
INFO: Number of pods: 21
INFO: Killing Random pod: adservice-67b89697dd-9669n from namespace: default
INFO: Updating pod list from: https://127.0.0.1:64768
INFO: Number of pods: 21
INFO: Killing Random pod: currencyservice-7ddb4c6866-bbgmb from namespace: default
INFO: Increased punch force: 1210.0
INFO: Increased punch force: 1331.0
INFO: Increased punch force: 1464.1000000000001
INFO: Updating pod list from: https://127.0.0.1:64768
INFO: Number of pods: 21
INFO: Killing Random pod: paymentservice-569f84557b-vrnlb from namespace: default
```

Conclusión

Haber desarrollado este proyecto utilizando diferentes herramientas desde Docker en conjunto con kubernetes, e Istio, nos brinda una aplicación mucho más robusta en cuanto a ser tolerante a posibles fallas, e incluso mucho más sencilla de monitorear utilizando otras herramientas como Kiali y OpenShift, además, esta estructura nos ayudará a ser más ágiles y conseguir atender nuestras necesidades de una forma estructurada y probada.



Repositorio del proyecto

<https://github.com/chrisIAM21/ProyectoFinal>