



ARQUITECTURAS DE NUBE PARA Internet of Things



PROGRAMA IBEROAMERICANO DE CIENCIA
Y TECNOLOGÍA PARA EL DESARROLLO

Agenda

Sesión 1: 2 horas sincrónicas + 4 horas independientes

1. Introducción a IoT y arquitecturas
2. Introducción y principios de Nube
3. Manejo del laboratorio
4. Ejercicios de introducción (trabajo independiente)

Sesión 2: 2 horas sincrónicas + 4 horas independientes

1. Arquitecturas de Nube – Generalidades
2. El Agente/Orquestador/Broker
3. Sistemas de almacenamiento
4. Sistemas de ETL
5. Sistemas de Toma de decisiones
6. Sistemas de Visualización
7. Ejercicios de Agentes y escritura de datos (trabajo independiente)

Sesión 3: 2 horas sincrónicas + 4 horas independiente

1. Sistemas de Almacenamiento – Modos de almacenamiento/arquitecturas con ventajas y desventajas

Sesión 4: 2 horas sincrónicas + 4 horas independiente

1. Sistemas de ETL
2. Visualización de datos
3. Taller de ETL y visualización de datos – Ejercicio PM2.5 (trabajo independiente)

Sesión 5: 2 horas sincrónicas + 4 horas independiente

1. Sistemas de toma de decisiones
2. Interfaces de usuario y desarrollo de apps
3. Taller de toma de decisiones y desarrollo de apps (preventivos y reactivos)

Sesión 6: 2 horas sincrónicas + 4 horas independiente

1. Integración de la arquitectura con FiWARE

Sesión 7: 8 horas presenciales

1. Montaje del proyecto presencial
2. Arquitecturas de alta disponibilidad en Nube
3. Ejercicios prácticos de montaje con sensores vía WiFi
4. Dimensionamiento de procesamiento y aspectos financieros de soluciones de Nube

MONTAJE DEL DIA DE HOY



PROGRAME UN DISPOSITIVO END DEVICE USANDO CUALQUIER RECURSO DE HARDWARE DISPONIBLE



CONSTRUYA UNA ARQUITECTURA BÁSICA DE PRUEBA PARA AFINAR LOS COMPONENTES DE LA ARQUITECTURA



LLEVE ESA ARQUITECTURA A UN NIVEL DE ALTA DISPONIBILIDAD

ORDEN DEL DESARROLLO SUGERIDO

EJERCICIO UNO – RESUMEN CURSO - BÁSICO

1. CREE UNA IP ELÁSTICA
2. PROGRAME SU MICRO Y DEFINA EL NOMBRE DEL SENSOR
3. MONTE UNA ARQUITECTURA DE PRUEBA (SUGERENCIA – USE CONTENEDORES EN UN EC2 PARA MONTAR UN FIWARE ORION, MONGO, QUANTUM LEAP, CRATE Y GRAFANA)
4. CONFIGURE LA ARQUITECTURA CON UNA ENTIDAD Y SUSCRIPCIÓN
5. ENVIE DATOS Y GRAFIQUE

IMPLEMENTACIÓN DEL SENSOR

```
#include <Arduino.h>
#include <max6675.h>
#include <ESP8266WiFi.h>

//definiendo los pines del microcontrolador
int thermoDO = 4;
int thermoCS = 0;
int thermoCLK = 2;

//definiendo la red WiFi

const char* ssid = "dlink"; // el ssid
const char* password = ""; // el password

WiFiClient client;

const char* server = "34.207.168.149"; //la ip del servidor
MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);
```

EJEMPLO DE UN ESP8266
CON UNA TERMOCUPLA
MAX6675

VARÍE EL DESARROLLO SUYO DE
ACUERDO CON EL SENSOR QUE
TENGA A A DISPOSICIÓN Y
CONFIGURE SU RED WIFI

IMPLEMENTACIÓN DEL SENSOR

```
void send_data();  
void setup() {  
  // MONITOR SERIAL:  
  Serial.begin(115200);  
  // modo del wifi para mandar data  
  WiFi.mode(WIFI_STA);  
  wifi_set_macaddr(STATION_IF, &MACAddr[0]);  
  WiFi.begin(ssid,password);  
  Serial.println("iniciando conexion");  
  while (WiFi.status() != WL_CONNECTED) {  
    Serial.print(".");  
    delay(100);  
  }  
  Serial.println("Conectado");  
}
```

EJEMPLO DE UN ESP8266
CON UNA TERMOCUPLA
MAX6675

VARÍE EL DESARROLLO SUYO DE
ACUERDO CON EL SENSOR QUE
TENGA A A DISPOSICIÓN Y
CONFIGURE SU RED WIFI

IMPLEMENTACIÓN DEL SENSOR

```
void loop() {  
  Serial.println("Midiendo");  
  send_data();  
  delay(5000);  
}
```

EJEMPLO DE UN ESP8266
CON UNA TERMOCUPLA
MAX6675

VARÍE EL DESARROLLO SUYO DE
ACUERDO CON EL SENSOR QUE
TENGA A A DISPOSICIÓN Y
CONFIGURE SU RED WIFI

IMPLEMENTACIÓN DEL SENSOR

```
void send_data()
{
    String temp = String(thermocouple.readCelsius());
    String postData = String("{\"temperatura\":{\"value\":"+
    temp +",\"type\":\"Float\"}}");
    if (client.connect(server,1026)){
        Serial.println("Conectado al server");
        client.println("PATCH /v2/entities/sensor03/attrs
        HTTP/1.1");
        client.print("Host: ");
        client.println(server);
        client.println("Content-Type: application/json");
        client.print("Content-Length: ");
        client.println(postData.length());
        client.println("");
        client.println(postData);
        Serial.println(postData);
    }
}
```

EJEMPLO DE UN ESP8266
CON UNA TERMOCUPLA
MAX6675

VARÍE EL DESARROLLO SUYO DE
ACUERDO CON EL SENSOR QUE
TENGA A A DISPOSICIÓN Y
CONFIGURE SU RED WIFI

IMPLEMENTACIÓN DEL SENSOR

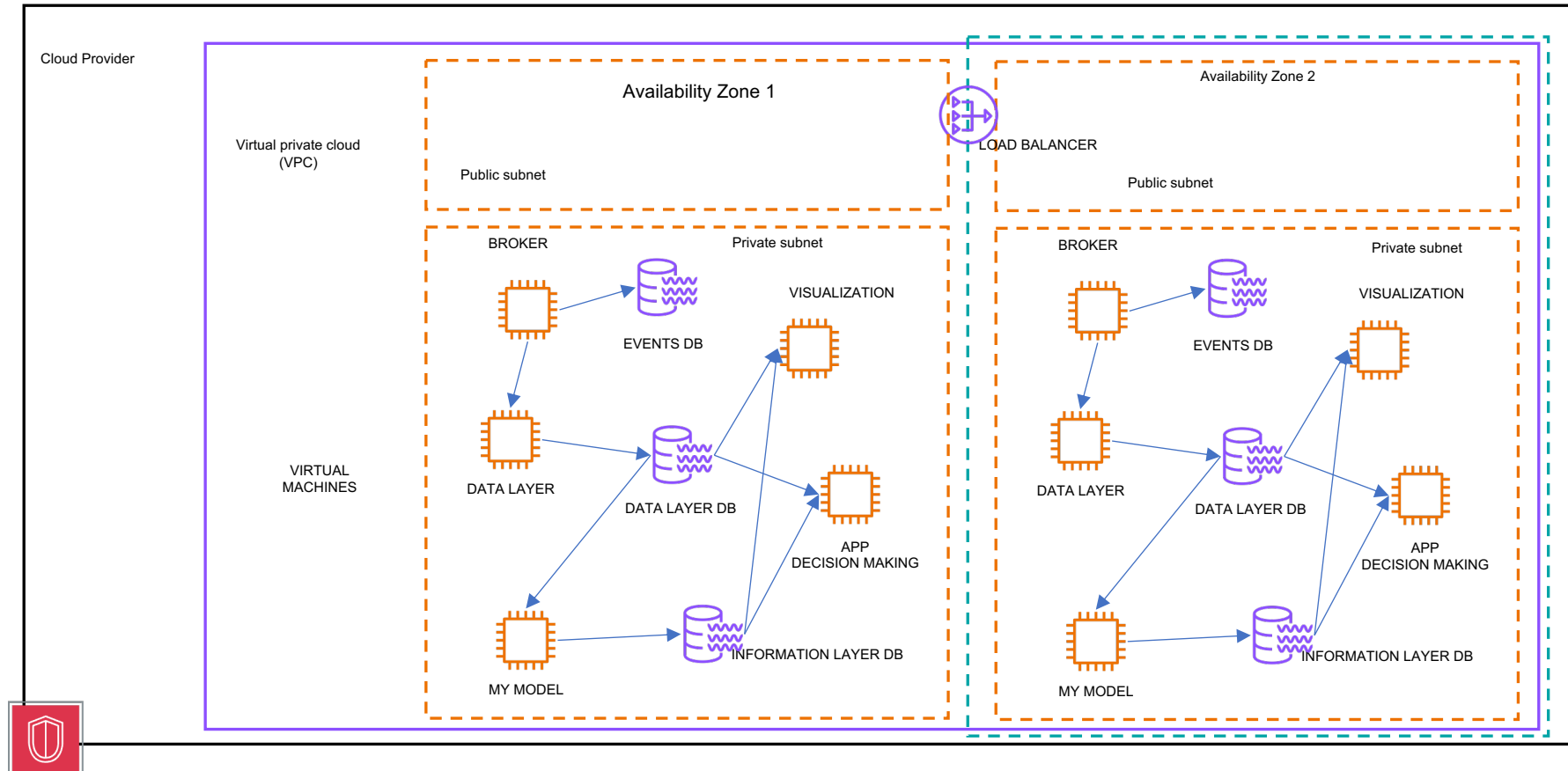
```
else {  
  Serial.println("Error de conexion");  
}  
delay(500);  
while (client.available())  
{  
  String line = client.readStringUntil('\n');  
  Serial.println(line);  
}  
}
```

EJEMPLO DE UN ESP8266
CON UNA TERMOCUPLA
MAX6675

VARÍE EL DESARROLLO SUYO DE
ACUERDO CON EL SENSOR QUE
TENGA A A DISPOSICIÓN Y
CONFIGURE SU RED WIFI

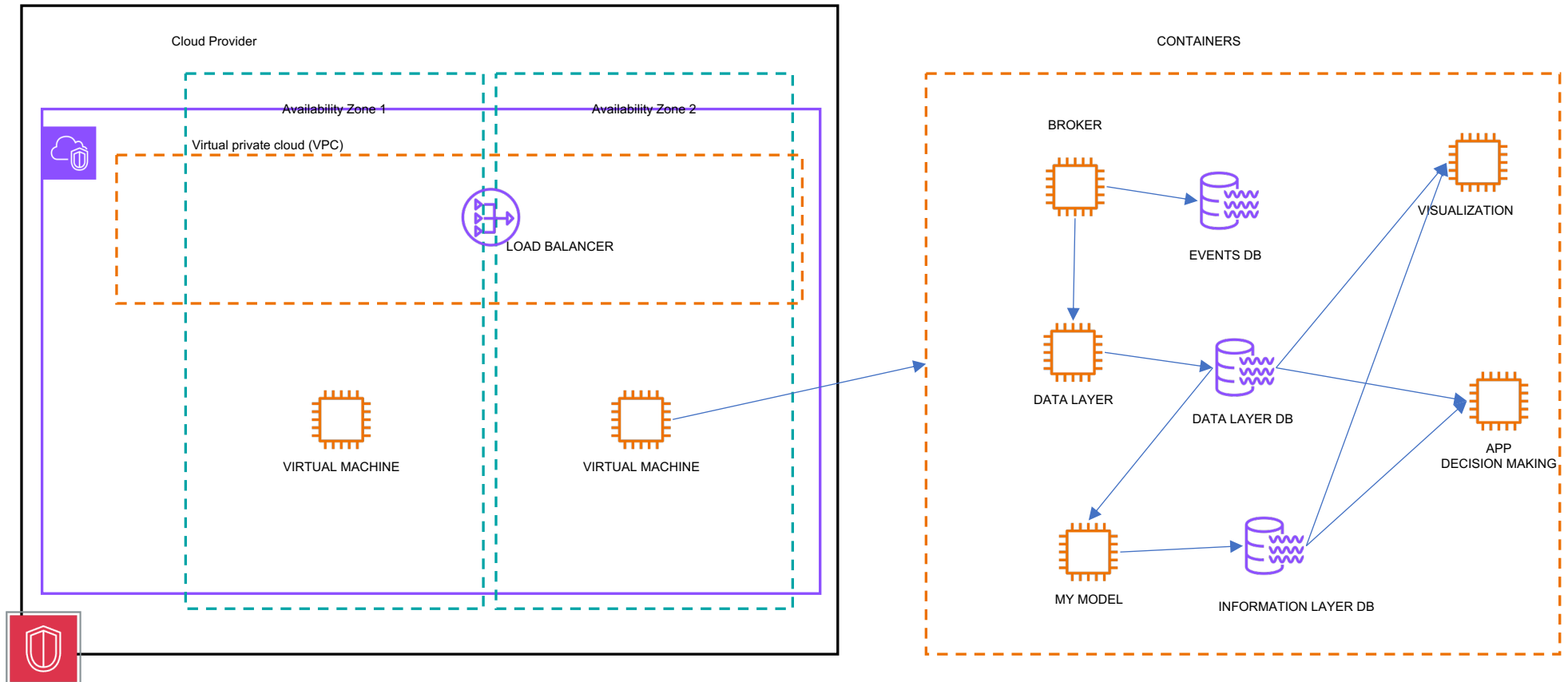
ARQUITECTURA EJERCICIO FINAL OPCION 1

DISCRETE IMPLEMENTATION



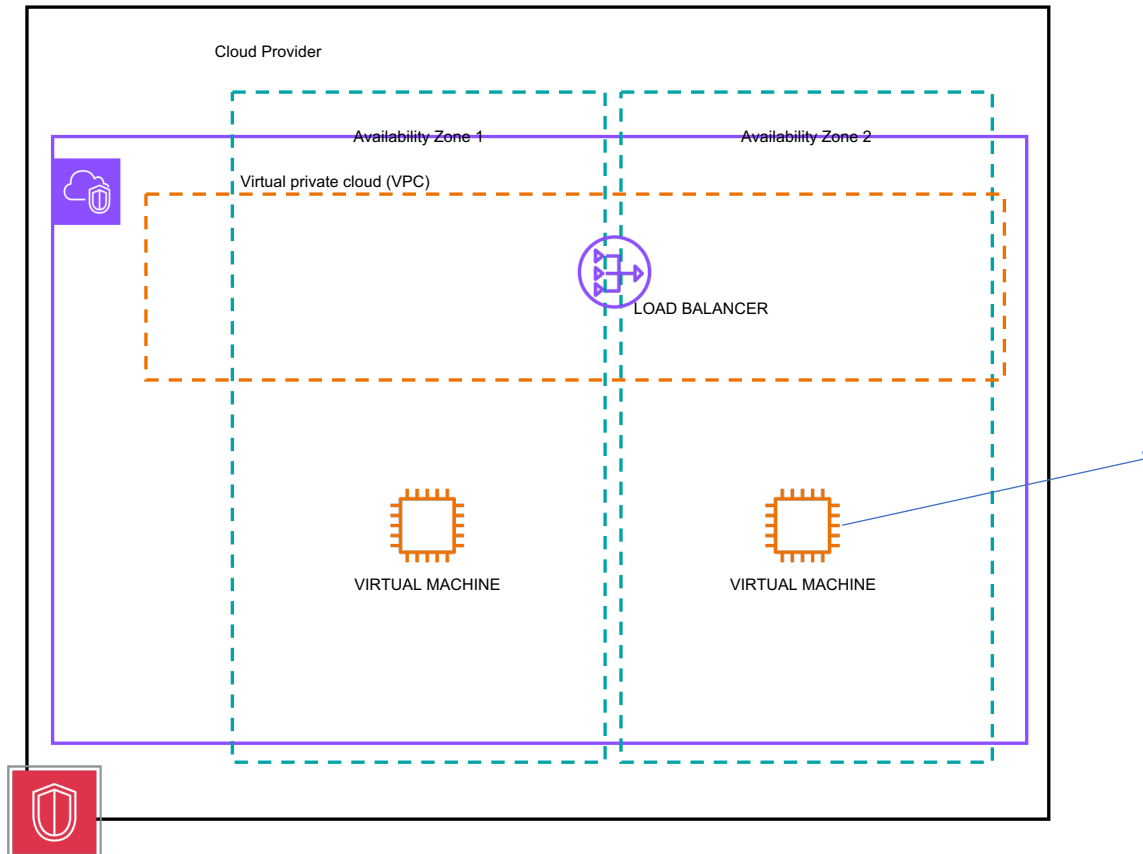
ARQUITECTURA EJERCICIO FINAL OPCION 2

AGNOSTIC IMPLEMENTATION

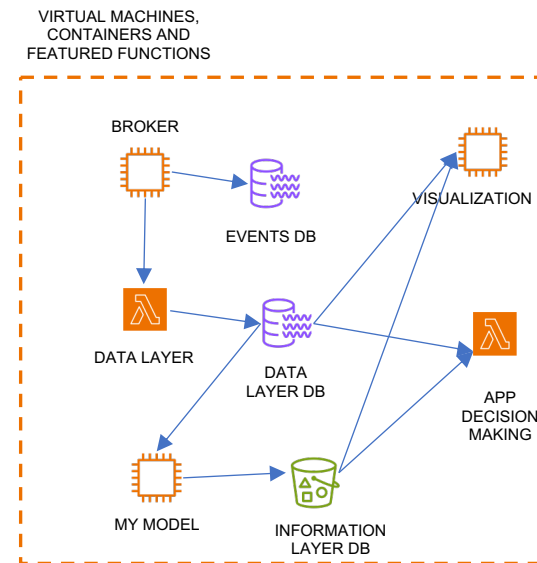


ARQUITECTURA EJERCICIO FINAL OPCION 3

HYBRID

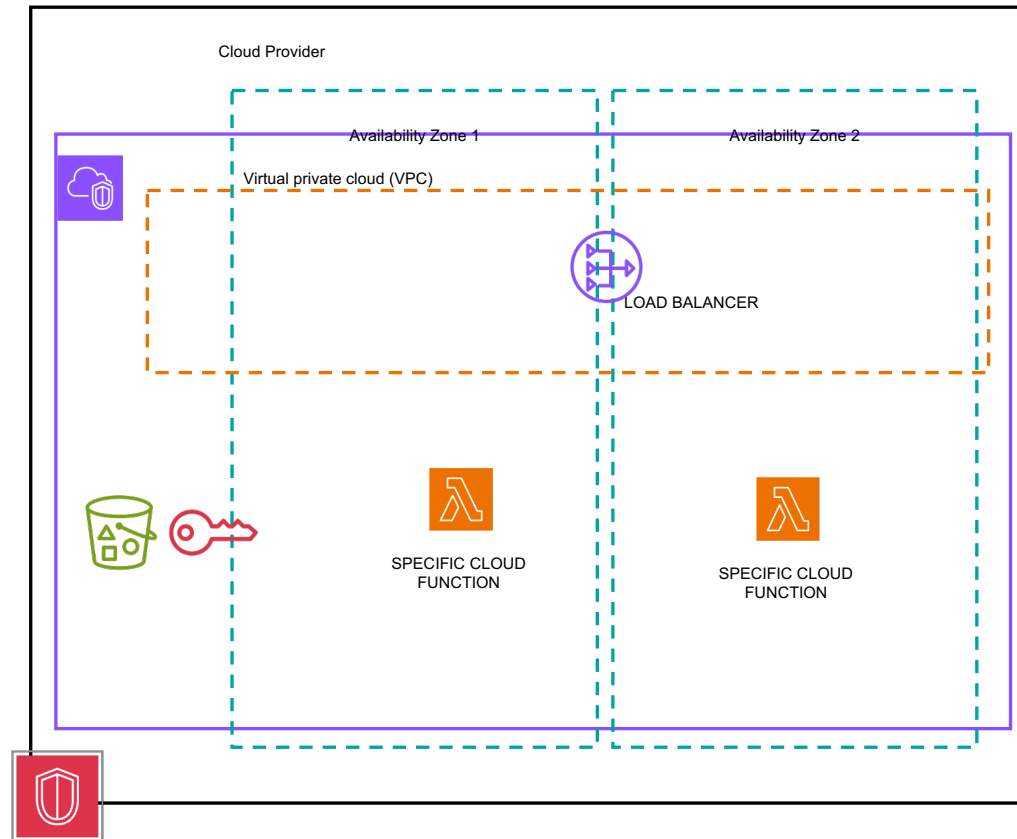


HYBRID IMPLEMENTATION



ARQUITECTURA EJERCICIO FINAL OPCION 4

FEATURED IMPLEMENTATION



COMO EVOLUCIONAR EN ÉSTA ÁREA DEL CONOCIMIENTO

1. HAGA LOS DISEÑOS DISCRETOS LA PRIMERA VEZ
2. APRENDA BASH SCRIPT
3. AUTOMATICE EN UN SERVIDOR LAS ACCIONES REPETITIVAS
4. MONTE LA INFRAESTRUCTURA PASO A PASO Y REPITA HASTA ENTENDER COMPONENTES Y OPERACIÓN

1. HAGA LOS DISEÑOS EN CONTENEDORES
2. AUTOMATICE CON DOCKER FILE Y APRENDA A SUBIR Y BAJAR CONTENDORES Y SERVICIOS
3. MONTE LA INFRAESTRUCTURA USANDO CODIGOS EN DOCKERFILE Y BASH SCRIPT

COMO EVOLUCIONAR EN ÉSTA ÁREA DEL CONOCIMIENTO

1. APRENDA TERRAFORM
 1. AUTOMATIZAR DESPLIEGUE
2. APRENDA ANSIBLE
 1. AUTOMATIZAR CONFIGURACIÓN DE LAS VM
3. COMBINE LOS DOS ELEMENTOS EN SUS DESPLIEGUES

1. APRENDA A USAR KUBERNETES
2. CONSTRUYA CLUSTERS DE ALTA DISPONIBILIDAD
3. INTRODUCZA EL USO DE REPOSITORIOS

COMO EVOLUCIONAR EN ÉSTA ÁREA DEL CONOCIMIENTO

1. APRENDA JEINKINS
2. AUTOMATICE LA ARQUITECTURA
3. APRENDA DESARROLLO
4. AUTOMATICE DESARROLLO

