CISCO DEVNET

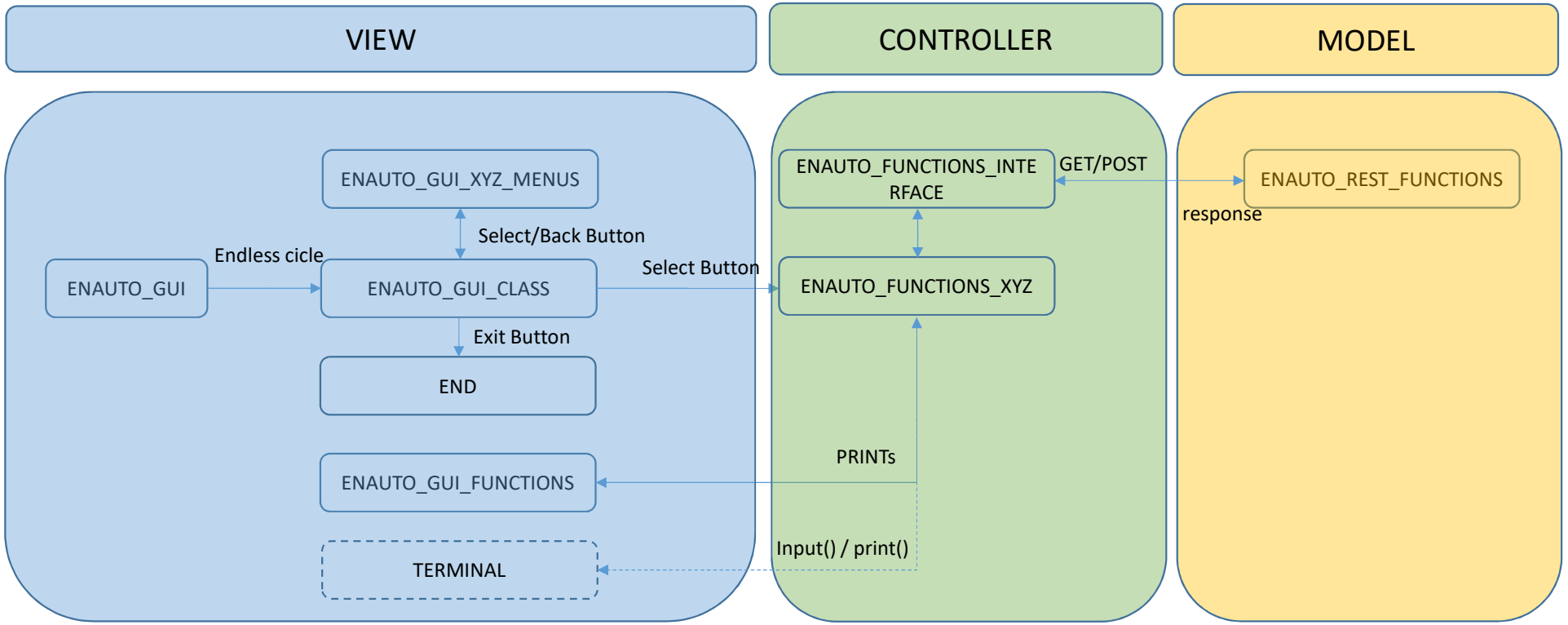
LEARN CODE INSPIRE CONNECT

- Software orientado al aprendizaje
- Escrito en totalmente en Python
- Ayuda a prepararse para el examen 300-435 ENAUTO de CISCO
- Usa SANDBOX ALWAYS ON de CISCO
- Diseñado para agregar funciones o cambiar parámetros fácilmente
- Código abierto para ser modificado, adaptado y mejorado

- SANDBOX IOS XE
- SANDBOX DNA CENTER
- SANDBOX SD-WAN
- SANDBOX MERAKI

cisco DevNet

MODELO GENERAL DEL CÓDIGO



cisco DevNet

VIEW (GUI)
INTERFAZ COMÚN

enauto_gui_ebo

librerías

```
import tkinter as tk
from enauto_gui_class_ebo import *
```

código

```
app=tk.Tk()
window=MAIN_WINDOW(app)
app.mainloop()
```

enauto_gui_class_ebo

CLASE WINDOW (__init__)

```
def __init__(self,window):
    #VENTANA
    window.geometry("1300x500")
    window.title("ENAUTO SCRIPS")
    #CUADRO DE TEXTO
    text_box=tk.Text(window,width=159,height=25)
    text_box.place(x=0,y=0)
    text_box.insert("1.0",MAIN_MENU)
    text_box.config(state="disabled")
    #SCROLLBAR
    scrollbar=tk.Scrollbar(window,orient="vertical")
    scrollbar.place(x=1280,y=0,height=400)
    #CONFIGURO SCROLLBAR
    text_box.config(yscrollcommand = scrollbar.set)
    scrollbar.config(command=text_box.yview)
    #CUADRO DE ENTRADA
    entry=tk.Entry(window,fg="blue",bg="white", width=25)
        entry.place(x=5,y=410)
```

librerías

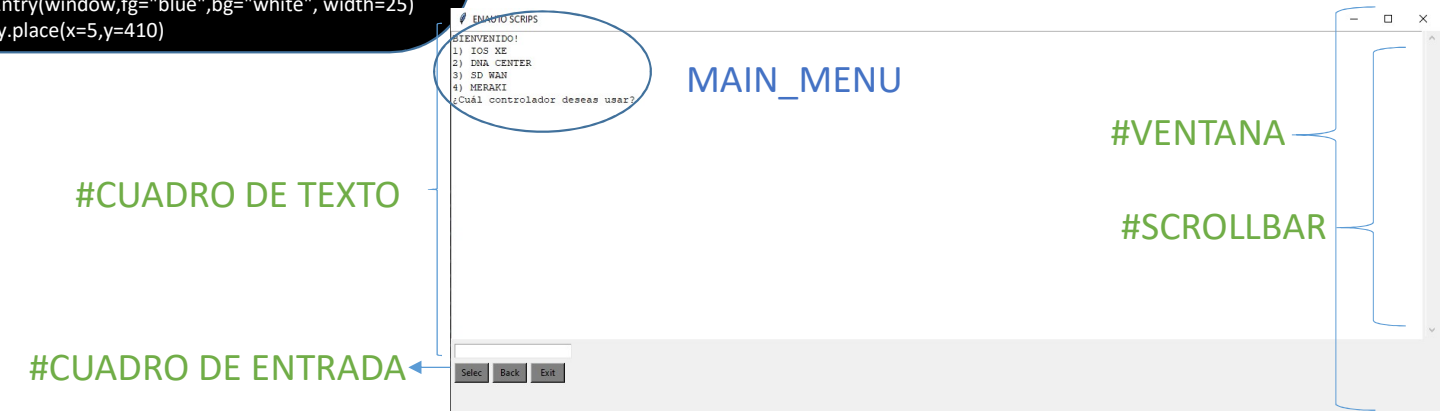
```
#------USER INTERFACE (VIEW)----------
import tkinter as tk

from enauto_gui_iosxe_menus_ebo import *
from enauto_gui_dnac_menus_ebo import *
from enauto_gui_sdwan_menus_ebo import *
from enauto_gui_meraki_menus_ebo import *

#------CONTROLLER--------------------
from enauto_functions_iosxe_ebo import *
from enauto_functions_dnac_ebo import *
from enauto_functions_sdwan_ebo import *
from enauto_functions_meraki_ebo import *
```

PARA MENÚS

PARA REQUESTS

MENÚS PRINCIPALES

```
MAIN_MENU
ERROR_MENU
```



MAIN_MENU

#CUADRO DE TEXTO

#CUADRO DE ENTRADA

#VENTANA

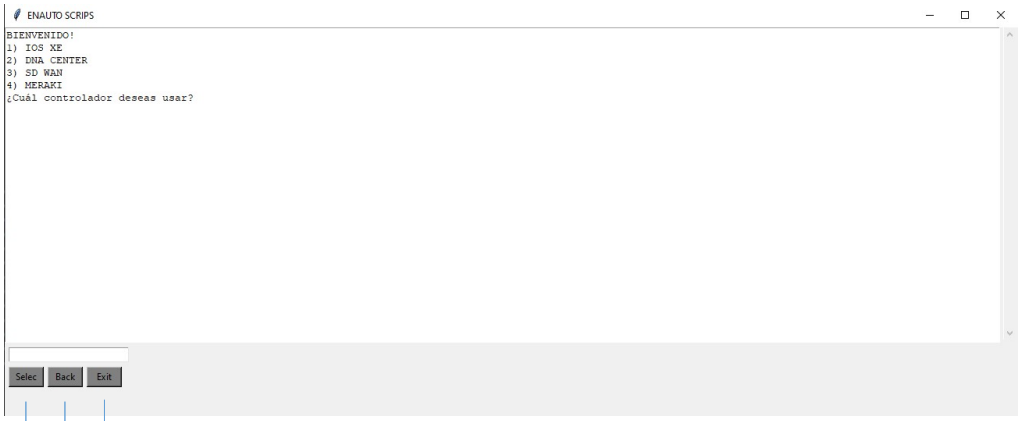
#SCROLLBAR

VIEW (GUI)
INTERFAZ COMÚN

enauto_gui_class_ebo

CLASE WINDOW (__init__) (Cont.)

```
#BOTON FW
button_fw=tk.Button(window,text="Selec",
      fg="black",bg="grey",
      width=5,height=1,
      command=lambda:self.PRINT_MENU_FW(entry,text_box))
button_fw.place(x=5,y=435)
#BOTON BACK
button_back=tk.Button(window,text="Back",
      fg="black",bg="grey",
      width=5,height=1,
      command=lambda:self.PRINT_MENU_BW(text_box))
button_back.place(x=55,y=435)
#BOTON EXIT
button_exit=tk.Button(window,text="Exit",
      fg="black",bg="grey",
      width=5,height=1,
      command=window.destroy)
button_exit.place(x=105,y=435)
```



VARIABLES GLOBALES

```
ans=""
headers=""
menu=""
menu2=""
```

BUTTONS CALLBACK FUNCTIONS

```
def PRINT_MENU_FW(self,entry,text_box):
    #VARIABLE GLOBAL
    global ans
    ans1=ans+entry.get()
    self.SELECT_AND_PRINT(ans1,text_box)
    ans=ans1
    entry.delete(0,tk.END)
```

```
def PRINT_MENU_BW(self,text_box):
    #VARIABLE GLOBAL
    global ans
    ans=ans[:-1]
    self.SELECT_AND_PRINT(ans,text_box)
```

- Al presionar “Select”, guarda en ans1 lo había antes más el nuevo dígito.
- Al presionar “Back”, quita el último dígito.
- En ambos casos ejecuta el método “SELECT AND PRINT”

cisco DevNet

VIEW (GUI)
IOS XE

enauto_gui_class_ebo
MÉTODO "SELECT AND PRINT"

MÉNÚS

```
#----MENU NIVEL 0-----
if ans1=="":menu=MAIN_MENU
#----IOS XE MENU NIVEL 1-3----
elif ans1=="1":menu=IOSXE_L1_MENU
elif ans1=="11" or ans1=="12":menu=IOSXE_L2_MENU
elif ans1=="111" or ans1=="121":menu=IOSXE_NETCONF_MENU
elif ans1=="112" or ans1=="122":menu=IOSXE_RESTCONF_MENU
```

enauto_gui_iosxe_menus_ebo

REQUEST

```
#----IOS XE RUN NIVEL 4------
elif (ans1=="1112" or ans1=="1212"):IOSXE_NETCONF_GET_YANG_INTERFACES(ans1,text_box)
elif (ans1=="1113" or ans1=="1213"):IOSXE_NETCONF_GET_INTERFACES(ans1,text_box)
elif (ans1=="1114" or ans1=="1214"):IOSXE_NETCONF_ADD_INTERFACES(ans1,text_box)
elif (ans1=="1115" or ans1=="1215"):IOSXE_NETCONF_MERGE_DESCRIPTION(ans1,text_box)
elif (ans1=="1116" or ans1=="1216"):IOSXE_NETCONF_EDIT_INTERFACES(ans1,text_box)
elif (ans1=="1117" or ans1=="1217"):IOSXE_NETCONF_DEL_INTERFACES(ans1,text_box)
elif (ans1=="1121" or ans1=="1221"):headers=IOSXE_RESTCONF_GET_HEADERS(ans1,text_box)
elif (ans1=="1122" or ans1=="1222") and headers!="":IOSXE_RESTCONF_GET_YANG_INTERFACES(ans1,headers,text_box)
elif (ans1=="1123" or ans1=="1223") and headers!="":IOSXE_RESTCONF_GET_INTERFACES(ans1,headers,text_box)
elif (ans1=="1124" or ans1=="1224") and headers!="":IOSXE_RESTCONF_ADD_INTERFACES(ans1,headers,text_box)
elif (ans1=="1125" or ans1=="1225") and headers!="":IOSXE_RESTCONF_MERGE_DESCRIPTION(ans1,headers,text_box)
elif (ans1=="1126" or ans1=="1226") and headers!="":IOSXE_RESTCONF_EDIT_INTERFACES(ans1,headers,text_box)
elif (ans1=="1127" or ans1=="1227") and headers!="":IOSXE_RESTCONF_DEL_INTERFACES(ans1,headers,text_box)
```

enauto_functions_iosxe_ebo

cisco DevNet

VIEW (GUI)
DNA CENTER (Parte 1)

enauto_gui_class_ebo
MÉTODO "SELECT AND PRINT"

MÉNÚS

```
#----DNAC MENU NIVEL 1-3----------
elif ans1=="2":menu=DNAC_L1_MENU
elif ans1=="21" or ans1=="22":menu=DNAC_L2_MENU
elif ans1=="212" or ans1=="222":menu=DNAC_SITE_MENU
elif ans1=="213" or ans1=="223":menu=DNAC_TOPOLOGY_MENU
elif ans1=="214" or ans1=="224":menu=DNAC_DEVICES_MENU
elif ans1=="215" or ans1=="225":menu=DNAC_CLIENTS_MENU
elif ans1=="216" or ans1=="226":menu=DNAC_CONFIG_TEMPLATES_MENU
```

enauto_gui_dnac_menus_ebo

REQUEST

```
elif ans1=="211" or ans1=="221":headers=DNAC_AUTHENTICATION(ans1,text_box)
elif (ans1=="2121" or ans1=="2221") and headers!="":DNAC_SITE(ans1,headers,text_box)
elif (ans1=="2122" or ans1=="2222") and headers!="":DNAC_SITE_HEALTH(ans1,headers,text_box)
elif (ans1=="2131" or ans1=="2231") and headers!="":DNAC_TOPOLOGY_PHYSICAL(ans1,headers,text_box)
elif (ans1=="2132" or ans1=="2232") and headers!="":DNAC_TOPOLOGY_SITE(ans1,headers,text_box)
elif (ans1=="2133" or ans1=="2233") and headers!="":DNAC_TOPOLOGY_VLAN(ans1,headers,text_box)
elif (ans1=="2141" or ans1=="2241") and headers!="":DNAC_DEVICES_NETWORK(ans1,headers,text_box)
elif (ans1=="2142" or ans1=="2242") and headers!="":DNAC_DEVICES_INTERFACES(ans1,headers,text_box)
elif (ans1=="2151" or ans1=="2251") and headers!="":DNAC_CLIENT_HEALTH(ans1,headers,text_box)
elif (ans1=="2152" or ans1=="2252") and headers!="":DNAC_CLIENT_ENRICHMENT_DETAILS(ans1,headers,text_box)
elif (ans1=="2153" or ans1=="2253") and headers!="":DNAC_ISSUES_ENRICHMENT_DETAILS(ans1,headers,text_box)
elif (ans1=="2161" or ans1=="2261") and headers!="":DNAC_TEMPLATES_PROJECT_LIST(ans1,headers,text_box)
elif (ans1=="2162" or ans1=="2262") and headers!="":DNAC_TEMPLATES_TEMPLATE_LIST(ans1,headers,text_box)
elif (ans1=="2163" or ans1=="2263") and headers!="":DNAC_TEMPLATES_TEMPLATE_DETAILS(ans1,headers,text_box)
elif (ans1=="2164" or ans1=="2264") and headers!="":DNAC_TEMPLATES_TEMPLATE_CREATE(ans1,headers,text_box)
elif (ans1=="2165" or ans1=="2265") and headers!="":DNAC_TEMPLATES_TEMPLATE_DEPLOY(ans1,headers,text_box)
elif (ans1=="2166" or ans1=="2266") and headers!="":DNAC_TEMPLATES_TEMPLATE_DEPLOY_STATUS(ans1,headers,text_box)
```

enauto_functions_dnac_ebo

cisco DevNet

VIEW (GUI)
DNA CENTER (Parte 2)

enauto_gui_class_ebo
MÉTODO "SELECT AND PRINT"

MÉNÚS

```
elif ans1=="217" or ans1=="227":menu=DNAC_COMMAND_RUNNER_MENU
elif ans1=="218" or ans1=="228":menu=DNAC_NETWORK_DISCOVERY_MENU
elif ans1=="219" or ans1=="229":menu=DNAC_PATH_TRACE_MENU
elif ans1=="21A" or ans1=="22A":menu=DNAC_TASK_MENU
elif ans1=="21B" or ans1=="22B":menu=DNAC_FILE_MENU
elif ans1=="21C" or ans1=="22C":menu=DNAC_EVENT_MENU
```

enauto_gui_dnac_menus_ebo

REQUEST

```
elif (ans1=="2171" or ans1=="2271") and headers!="":DNAC_COMMAND_RUNNER_LEGIT_READS(ans1,headers,text_box)
elif (ans1=="2172" or ans1=="2272") and headers!="":DNAC_COMMAND_RUNNER_READ_REQUEST(ans1,headers,text_box)
elif (ans1=="2181" or ans1=="2281") and headers!="":DNAC_NETWORK_DISCOVERY_COUNT(ans1,headers,text_box)
elif (ans1=="2182" or ans1=="2282") and headers!="":DNAC_NETWORK_DISCOVERY_LIST(ans1,headers,text_box)
elif (ans1=="2183" or ans1=="2283") and headers!="":DNAC_NETWORK_DISCOVERY_SUMMARY(ans1,headers,text_box)
elif (ans1=="2191" or ans1=="2291") and headers!="":DNAC_PATH_TRACE_INIT(ans1,headers,text_box)
elif (ans1=="2192" or ans1=="2292") and headers!="":DNAC_PATH_TRACE_GET_FLOW_ANALYSIS(ans1,headers,text_box)
elif (ans1=="2193" or ans1=="2293") and headers!="":DNAC_PATH_TRACE_GET_FLOW_ANALYSIS_ID(ans1,headers,text_box)
elif (ans1=="2194" or ans1=="2294") and headers!="":DNAC_PATH_TRACE_DELETE_FLOW_ANALYSIS_ID(ans1,headers,text_box)
elif (ans1=="21A1" or ans1=="22A1") and headers!="":DNAC_TASK_LIST(ans1,headers,text_box)
elif (ans1=="21A2" or ans1=="22A2") and headers!="":DNAC_TASK_ID(ans1,headers,text_box)
elif (ans1=="21B1" or ans1=="22B1") and headers!="":DNAC_FILE_NAMESPACE_LIST(ans1,headers,text_box)
elif (ans1=="21B2" or ans1=="22B2") and headers!="":DNAC_FILE_NAMESPACE_ID(ans1,headers,text_box)
elif (ans1=="21B3" or ans1=="22B3") and headers!="":DNAC_FILE_ID(ans1,headers,text_box)
elif (ans1=="21C1" or ans1=="22C1") and headers!="":DNAC_EVENT_GET(ans1,headers,text_box)
elif (ans1=="21C2" or ans1=="22C2") and headers!="":DNAC_EVENT_SUBSCRIPTION(ans1,headers,text_box)
elif (ans1=="21C3" or ans1=="22C3") and headers!="":DNAC_EVENT_NOTIFICATIONS(ans1,headers,text_box)
elif (ans1=="21C4" or ans1=="22C4") and headers!="":DNAC_EVENT_POST_SUSCRIPTION(ans1,headers,text_box)
```

enauto_functions_dnac_ebo

cisco DevNet

VIEW (GUI)
SD WAN (Parte 1)

enauto_gui_class_ebo
MÉTODO "SELECT AND PRINT"

MÉNÚS

```
#----SDWAN MENU NIVEL 1-3---------
elif ans1=="3":menu=SDWAN_L1_MENU
elif ans1=="31" or ans1=="32":menu=SDWAN_L2_MENU
elif ans1=="312" or ans1=="322":menu=SDWAN_CERTIFICATION_MANAGEME
NT_MENU
elif ans1=="313" or ans1=="323":menu=SDWAN_DEVICE_INVENTORY_MENU
elif ans1=="314" or ans1=="324":menu=SDWAN_ADMINISTRATION_MENU
```

enauto_gui_sdwan_menus_ebo

REQUEST

```
#----SDWAN RUN NIVEL 3-4-----
elif ans1=="311" or ans1=="321":headers=SDWAN_AUTHENTICATION(ans1,text_box)
elif (ans1=="3121" or ans1=="3221") and headers!="":SDWAN_CERTIFICATION_DEVICE_DETAILS(ans1,headers,text_box)
elif (ans1=="3122" or ans1=="3222") and headers!="":SDWAN_CERTIFICATION_VSMART_LIST(ans1,headers,text_box)
elif (ans1=="3123" or ans1=="3223") and headers!="":SDWAN_CERTIFICATION_VEDGE_LIST(ans1,headers,text_box)
elif (ans1=="3124" or ans1=="3224") and headers!="":SDWAN_CERTIFICATION_CSR_DETAILS(ans1,headers,text_box)
elif (ans1=="3131" or ans1=="3231") and headers!="":SDWAN_DEVICE_INVENTORY_VEDGES(ans1,headers,text_box)
elif (ans1=="3132" or ans1=="3232") and headers!="":SDWAN_DEVICE_INVENTORY_CONTROLLERS(ans1,headers,text_box)
elif (ans1=="3133" or ans1=="3233") and headers!="":SDWAN_DEVICE_INVENTORY_VEDGES_STATUS(ans1,headers,text_box)
elif (ans1=="3134" or ans1=="3234") and headers!="":SDWAN_DEVICE_INVENTORY_MNG_SYS_IP(ans1,headers,text_box)
elif (ans1=="3141" or ans1=="3241") and headers!="":SDWAN_ADMINISTRATION_AUDITLOG(ans1,headers,text_box)
elif (ans1=="3142" or ans1=="3242") and headers!="":SDWAN_ADMINISTRATION_USER(ans1,headers,text_box)
elif (ans1=="3143" or ans1=="3243") and headers!="":SDWAN_ADMINISTRATION_USER_GROUP(ans1,headers,text_box)
```

enauto_functions_sdwan_ebo

VIEW (GUI)
SD WAN (Parte 2)

enauto_gui_class_ebo
MÉTODO "SELECT AND PRINT"

MÉNÚS

```
elif ans1=="315" or ans1=="325":menu=SDWAN_CONFIGURATION_MENU
elif ans1=="316" or ans1=="326":menu=SDWAN_MONITORING_MENU
```

enauto_gui_sdwan_menus_ebo

REQUEST

```
elif (ans1=="3151" or ans1=="3251") and headers!="":SDWAN_CONFIGURATION_DEVICE_FEATURELIST(ans1,headers,text_box)
elif (ans1=="3152" or ans1=="3252") and headers!="":SDWAN_CONFIGURATION_DEVICE_TEMPLATE(ans1,headers,text_box)
elif (ans1=="3153" or ans1=="3253") and headers!="":SDWAN_CONFIGURATION_GET_FEATURE_TEMPLATE(ans1,headers,text_box)
elif (ans1=="3154" or ans1=="3254") and headers!="":SDWAN_CONFIGURATION_PUT_FEATURE_TEMPLATE(ans1,headers,text_box) #NO EXISTE FUNC TODAVÍA
elif (ans1=="3155" or ans1=="3255") and headers!="":SDWAN_CONFIGURATION_DEL_FEATURE_TEMPLATE(ans1,headers,text_box) #NO EXISTE FUNC TODAVÍA
elif (ans1=="3156" or ans1=="3256") and headers!="":SDWAN_CONFIGURATION_DEVICE_RUNCONFIG_TEMPLATE(ans1,headers,text_box)
elif (ans1=="3157" or ans1=="3257") and headers!="":SDWAN_CONFIGURATION_GET_CLI_CONTROLLER_DEV_CONFIG_TEMP(ans1,headers,text_box)
elif (ans1=="3158" or ans1=="3258") and headers!="":SDWAN_CONFIGURATION_GET_CLI_VMANAGE_DEV_CONFIG_TEMP(ans1,headers,text_box)
elif (ans1=="3159" or ans1=="3259") and headers!="":SDWAN_CONFIGURATION_POS_CLI_DEV_CONFIG_TEMP(ans1,headers,text_box)  #NO EXISTE FUNC TODAVÍA
elif (ans1=="315A" or ans1=="325A") and headers!="":SDWAN_CONFIGURATION_GET_ATTACHED_DEV_TEMP(ans1,headers,text_box)    #NO EXISTE FUNC TODAVÍA
elif (ans1=="315B" or ans1=="325B") and headers!="":SDWAN_CONFIGURATION_POS_ATTACHED_DEV_TEMP(ans1,headers,text_box)    #NO EXISTE FUNC TODAVÍA
elif (ans1=="3161" or ans1=="3261") and headers!="":SDWAN_MONITORING_ALARMS(ans1,headers,text_box)
elif (ans1=="3162" or ans1=="3262") and headers!="":SDWAN_MONITORING_ALARMS_STATS(ans1,headers,text_box)
elif (ans1=="3163" or ans1=="3263") and headers!="":SDWAN_MONITORING_DEVICE(ans1,headers,text_box)
elif (ans1=="3164" or ans1=="3264") and headers!="":SDWAN_MONITORING_DEVICE_STATUS(ans1,headers,text_box)
elif (ans1=="3165" or ans1=="3265") and headers!="":SDWAN_MONITORING_DEVICE_COUNTERS(ans1,headers,text_box)
elif (ans1=="3166" or ans1=="3266") and headers!="":SDWAN_MONITORING_DATA_DEV_STATS(ans1,headers,text_box)
elif (ans1=="3167" or ans1=="3267") and headers!="":SDWAN_MONITORING_EVENT_LISTENERS(ans1,headers,text_box)
```

enauto_functions_sdwan_ebo

cisco DevNet

VIEW (GUI)
SD WAN (Parte 3)

enauto_gui_class_ebo
MÉTODO "SELECT AND PRINT"

MÉNÚS

```
elif ans1=="317" or ans1=="327":menu=SDWAN_RT_MONITORING_MENU
elif ans1=="318" or ans1=="328":menu=SDWAN_TROUBLESHOOT_MENU
```

enauto_gui_sdwan_menus_ebo

REQUEST

```
elif (ans1=="3171" or ans1=="3271") and headers!="":SDWAN_RT_MONITORING_ARP(ans1,headers,text_box)
elif (ans1=="3172" or ans1=="3272") and headers!="":SDWAN_RT_MONITORING_CONTROL_SUMMARY(ans1,headers,text_box)
elif (ans1=="3173" or ans1=="3273") and headers!="":SDWAN_RT_MONITORING_CONTROL_WAN_IF(ans1,headers,text_box)
elif (ans1=="3174" or ans1=="3274") and headers!="":SDWAN_RT_MONITORING_DHCP_IF(ans1,headers,text_box)
elif (ans1=="3175" or ans1=="3275") and headers!="":SDWAN_RT_MONITORING_DHCP_SERVER(ans1,headers,text_box)
elif (ans1=="3176" or ans1=="3276") and headers!="":SDWAN_RT_MONITORING_DHCPV6_IF(ans1,headers,text_box)
elif (ans1=="3177" or ans1=="3277") and headers!="":SDWAN_RT_MONITORING_IF(ans1,headers,text_box)
elif (ans1=="3178" or ans1=="3278") and headers!="":SDWAN_RT_MONITORING_ROUTE_TABLE(ans1,headers,text_box)
elif (ans1=="3181" or ans1=="3281") and headers!="":SDWAN_TSHOOT_ACTIVE_COUNT_STATUS_DEV(ans1,headers,text_box)
elif (ans1=="3182" or ans1=="3282") and headers!="":SDWAN_TSHOOT_CLEAN_STATUS_DEV(ans1,headers,text_box)
elif (ans1=="3183" or ans1=="3283") and headers!="":SDWAN_TSHOOT_GROUP(ans1,headers,text_box)
elif (ans1=="3184" or ans1=="3284") and headers!="":SDWAN_TSHOOT_GROUP_DEVICE(ans1,headers,text_box)
elif (ans1=="3185" or ans1=="3285") and headers!="":SDWAN_TSHOOT_GROUP_DEVICES(ans1,headers,text_box)
elif (ans1=="3186" or ans1=="3286") and headers!="":SDWAN_TSHOOT_GROUP_MAP_DEVICES(ans1,headers,text_box)
elif (ans1=="3187" or ans1=="3287") and headers!="":SDWAN_TSHOOT_GROUP_MAP_DEVICES_LINKS(ans1,headers,text_box)
```

enauto_functions_sdwan_ebo

cisco DevNet

VIEW (GUI)
MERAKI

enauto_gui_class_ebo
MÉTODO "SELECT AND PRINT"

MÉNÚS

```
#----MERAKI MENU NIVEL 1-3----
elif ans1=="4":menu=MERAKI_L1_MENU
elif ans1=="41" or ans1=="42":menu=MERAKI_L2_MENU
elif ans1=="412" or ans1=="422":menu=MERAKI_DASHBOARD_MENU
```

enauto_gui_meraki_menus_ebo

REQUEST

```
#----MERAKI RUN NIVEL 3-4-----
elif (ans1=="411" or ans1=="421"):headers=MERAKI_GET_HEADERS(ans1,text_box)
elif (ans1=="4121" or ans1=="4221") and headers!="":MERAKI_DASHBOARD_ORGANIZATIONS(ans1,headers,text_box)
elif (ans1=="4122" or ans1=="4222") and headers!="":MERAKI_DASHBOARD_NETWORKS(ans1,headers,text_box)
elif (ans1=="4123" or ans1=="4223") and headers!="":MERAKI_DASHBOARD_DEVICES(ans1,headers,text_box)
elif (ans1=="4124" or ans1=="4224") and headers!="":MERAKI_DASHBOARD_SSID(ans1,headers,text_box)
```

enauto_functions_meraki_ebo

```
#----------------------
#---PRINTS MENUS ONLY---
#----------------------
if menu!=menu2:                          #si son IGUALES, se ejecutó un REQUEST (no imprimo menú)
   text_box.config(state="normal")   #habilito la escritura
   text_box.delete("1.0", tk.END)     #para borrar el texto
   text_box.insert("1.0",menu)        #imprimo menú
   text_box.config(state="disabled") #solo lectura
   menu2=menu
else:
   menu2=""
```

- Se ejecuta cuando se tiene que imprimir algún MENÚ, es decir, cuando el menú actual es diferente al anterior.
- Cuando se ejecuta un REQUEST, el menú actual es el mismo que el anterior.

CONTROLLER
AUTHENTICATION (Parte 1)

enauto_functions_interface_ebo

SANDBOX_AUTHENTICATION(ans1,text_box)

enauto_functions_iosxe_ebo

```
def IOSXE_RESTCONF_GET_HEADERS(ans1,text_box):
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

enauto_functions_dnac_ebo

```
#---------AUTHENTICATION------------
def DNAC_AUTHENTICATION(ans1,text_box):
    #SELECCIONO ENTRE LOS DOS DNAC
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

enauto_functions_sdwan_ebo

```
#---------AUTHENTICATION------------
def SDWAN_AUTHENTICATION(ans1,text_box):
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

enauto_functions_meraki_ebo

```
def MERAKI_GET_HEADERS(ans1,text_box):
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

IOS XE AUTHENTICATION

```
#SELECCIONO ENTRE LOS DOS IOS XE
if ans1[:2]=="11":
    user="developer"
    password="C1sco12345"
#elif ans1[:2]=="12": #PARA UN SEGUNDO SANDBOX DE IOS XE
```

```
#SI ES IOS XE
if ans1[0]=="1":
    authorization=base64.b64encode(str(user+":"+password).encode('UTF-8'))
    headers={"Content-type": "application/yang-data+json",
        "Accept": "application/yang-data+json",
        "Authorization": "Basic "+authorization.decode('utf-8')}
    PRINT_HEADERS(headers,text_box)
    return headers
```

MERAKI AUTHENTICATION

```
#SELECCIONO ENTRE LOS DOS MERAKI
elif ans1[:2]=="41":
    api_key="15da0c6ffff295f16267f88f98694cf29a86ed87"
elif ans1[:2]=="42":
    api_key="6bec40cf957de430a6f1f2baa056b99a4fac9ea0"
```

```
#SI ES MERAKI
elif ans1[0]=="4":
    headers = {"content-type": "application/json",
            "X-Cisco-Meraki-API-Key": api_key}
    PRINT_HEADERS(headers,text_box)
    return headers
```

enauto_gui_functions_ebo

```
def PRINT_HEADERS(headers,text_box):
    text_box.config(state="normal")             #habilito la escritura
    text_box.delete("1.0", tk.END)              #para borrar el texto
    text_box.insert("1.0","Headers: "+str(headers)+"\n")
    text_box.config(state="disabled")           #solo lectura
```

- En el caso de IOS XE y MERAKI, no se requiere un POST para obtener un token.
- Únicamente se crea el HEADER que será usado en los otros REQUEST.
- La función PRINT_HEADERS imprime el header creado en el tex_box.

CONTROLLER
AUTHENTICATION (Parte 2)

enauto_functions_interface_ebo
SANDBOX_AUTHENTICATION(ans1,text_box)

enauto_functions_iosxe_ebo

```
def IOSXE_RESTCONF_GET_HEADERS(ans1,text_box):
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

enauto_functions_dnac_ebo

```
#---------AUTHENTICATION------------
def DNAC_AUTHENTICATION(ans1,text_box):
    #SELECCIONO ENTRE LOS DOS DNAC
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

enauto_functions_sdwan_ebo

```
#---------AUTHENTICATION------------
def SDWAN_AUTHENTICATION(ans1,text_box):
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

enauto_functions_meraki_ebo

```
def MERAKI_GET_HEADERS(ans1,text_box):
    headers=SANDBOX_AUTHENTICATION(ans1,text_box)
    return headers
```

DNAC AUTHENTICATION (Paso 1)

```
#SELECCIONO ENTRE LOS DOS DNAC
elif ans1[:2]=="21":
    url="https://sandboxdnac2.cisco.com:443/dna/intent/api/"
    user="dnacdev"
    password="D3v93T@wK!"
elif ans1[:2]=="22":
    url="https://sandboxdnac.cisco.com:443/dna/intent/api/"
    user="devnetuser"
    password="Cisco123!"
```

```
#SI ES DNA CENTER
elif ans1[0]=="2":
    url=url.replace("intent","system")
    authorization=base64.b64encode(str(user+":"+password).encode('UTF-8'))
    headers={"Content-type": "application/json",
          "Authorization": "Basic "+authorization.decode('utf-8')}
    data=""
    services="v1/auth/token"
```

SDWAN AUTHENTICATION (Paso 1)

```
#SELECCIONO ENTRE LOS DOS SD WAN
elif ans1[:2]=="31":
    url="https://sandbox-sdwan-1.cisco.com:443/"
    j_username="devnetuser"
    j_password="RG!_Yw919_83"
elif ans1[:2]=="32":
    url="https://sandboxsdwan.cisco.com:8443/"
    j_username="devnetuser"
    j_password="C1sco12345"
```

```
#SI ES SDWAN
elif ans1[0]=="3":
    headers={"Content-Type": "application/x-www-form-urlencoded"}
    data = "j_username="+j_username+"&j_password="+j_password
    services="j_security_check"
```

POST FOR A TOKEN (Paso 2)

```
#PIDO EL TOKEN
response=POST_EBO(url=url,services=services,headers=headers,data=data)
#IMPRIMO EN EL TEXT BOX
PRINT_STATUS_CODE(response,text_box)
```

NUEVO HEADER (Paso 3)

```
#SI ES DNA CENTER
if ans1[0]=="2":
    token=response.json()["Token"]
    headers={"content-type": "application/json",
          "x-auth-token": token}
#SI ES SDWAN
elif ans1[0]=="3":
    j_session_id=dict(response.cookies)['JSESSIONID']
    headers={"Cookie": "JSESSIONID="+j_session_id}
return headers
```

cisco DevNet

CONTROLLER
GET REQUEST

enauto_functions_iosxe_ebo

```
def IOSXE_RESTCONF_GET_YANG_INTERFACES(ans1,headers,text_box):
    services="netconf-state/capabilities"
    response=GET_FUNCTION_EBO(ans1,services,headers,text_box)
    PRINT_RESPONSE_JSON(response,text_box)
```

enauto_functions_dnac_ebo

```
def DNAC_SITE(ans1,headers,text_box):
    response=GET_FUNCTION_EBO(ans1=ans1,services="v1/site",headers=headers,text_box=text_box)
    PRINT_TABLE_IN_TEXT(text_box,response.json()['response'],
                size1="30",name1="Name",data1='name',
                size2="40",name2="Site Id",data2='id',
                size3="30",name3="Site Name Hierarchy",data3='siteNameHierarchy')
```

enauto_functions_sdwan_ebo

```
def SDWAN_CERTIFICATION_CSR_DETAILS(ans1,headers,text_box):
    #SELECCIONO ENTRE LOS DOS DNAC, EJECUTO EL GET, IMPRIMO RESPUESTA
    response=GET_FUNCTION_EBO(ans1=ans1,services="dataservice/certificate/csr/details",
                        headers=headers,text_box=text_box)
    #IMPRIMO EN EL TEXT BOX
    PRINT_CONTENT_JSON(response,text_box)
```

enauto_functions_meraki_ebo

```
def MERAKI_DASHBOARD_ORGANIZATIONS(ans1,headers,text_box):
    #SELECCIONO ENTRE LOS DOS DNAC, EJECUTO EL GET, IMPRIMO RESPUESTA
    response=GET_FUNCTION_EBO(ans1=ans1,services="organizations",headers="",text_box=text_box)
```

enauto_functions_interface_ebo
GET_FUNCTION_EBO(ans1,services,headers,text_box)

```
url=SANDBOX_INFO(ans1)
response=GET_EBO(url=url,services=services,headers=headers)
PRINT_STATUS_CODE(response,text_box)
return response
```

enauto_functions_interface_ebo
SANDBOX_INFO(ans)

```
#SELECCIONO ENTRE LOS DOS IOS XE
if ans1[:2]=="11":
    host="ios-xe-mgmt.cisco.com"
    port=9443   #restconf port
    url="https://"+host+":"+str(port)+"/restconf/data/"
#elif ans1[:2]=="12":
#SELECCIONO ENTRE LOS DOS DNAC
elif ans1[:2]=="21":url="https://sandboxdnac.cisco.com/dna/intent/api/"
elif ans1[:2]=="22":url="https://sandboxdnac2.cisco.com/dna/intent/api/"
#SELECCIONO ENTRE LOS DOS SDWAN
elif ans1[:2]=="31":url="https://sandbox-sdwan-1.cisco.com:443/"
elif ans1[:2]=="32":url="https://sandboxsdwan.cisco.com:8443/"
#SELECCIONO ENTRE LOS DOS MERAKI
elif ans1[:2]=="41":url="https://api.meraki.com/api/v1/"
elif ans1[:2]=="42":url="https://api.meraki.com/api/v0/"
return url
```

enauto_gui_functions_ebo

```
def PRINT_STATUS_CODE(response,text_box):
    #IMPRIMO EN EL TEXT BOX
    text_box.config(state="normal")              #habilito la escritura
    text_box.delete("1.0", tk.END)               #para borrar el texto
    text_box.insert("1.0","Request status: "+str(response.status_code)+"\n")
    text_box.config(state="disabled")
```

cisco DevNet

MODEL
POST AND GET REQUEST

enauto_rest_functions_ebo

POST_EBO(url=url,services=services,headers=headers,data=data

```
#---------MODEL----------
import requests
import json
import base64


#------REST FUNCTIONS----------------
def POST_EBO(url,services,headers,data):
    requests.packages.urllib3.disable_warnings()                #deshabilitar SSL certificate warnings
    url+=services          #concateno url completo
    resp=requests.post(url,headers=headers,data=data,verify=False)  #post con los datos anteriores
    return resp
def GET_EBO(url,services,headers):
    requests.packages.urllib3.disable_warnings()                #deshabilitar SSL certificate warnings
    url+=services                                       #concateno url completo
    resp = requests.get(url, headers=headers, verify=False)        #get con los datos anteriores
    return resp
```