
SEGUNDO PROYECTO IPC2

201902363 – CHRISTOPHER IVÁN MONTERROSO ALEGRIA

Resumen

Aplicación creada para la gestión de empresas, sus transacciones y sus puntos de atención ubicadas en diferentes zonas, de los cuales tendrán escritorios de atención para los clientes, para estos la aplicación puede ser cargada a través de archivos tipo .xml ó crear las empresas con sus respectivas funciones para visualizar de forma gráfica el comportamiento de los escritorios y las filas de espera de los clientes o bien realizar una simulación completa de una fila de espera.

Palabras clave

Lista enlazada, Nodo, POO, Optimización.

Abstract

Application created for epidemiological research in Guatemala in order to calculate the patterns of Application created for the management of companies, their transactions and their service points located in different areas, of which they will have customer service desks, for these the application can be loaded through .xml type files or create the companies with their respective functions to graphically visualize the behavior of the desks and the queues of the clients or carry out a complete simulation of a queue.

Keywords

Linked list, Node, OOP, Optimization.

Introducción

La aplicación lee un primer archivo xml haciendo uso de la librería `xml.etree.ElementTree`, extrayendo la información, guardando en una lista dinámica, la empresa, sus transacciones, sus puntos de atención, los escritorios del punto de atención y un segundo archivo para la prueba y simulación de la cola de espera, el cual extrae la información guardándola en una lista dinámica perteneciente a las listas anteriormente dichas, el dpi del cliente, su nombre, y las transacciones que hará. Una vez cargados los dos archivos se permite escoger la empresa y su punto de atención los cuales se manejarán desde un menú con distintas opciones las cuales serán mostradas en el desarrollo del tema. Para las gráficas se usó la librería `graphviz`.

Desarrollo del tema

La aplicación inicia en su menú principal, el cuál nos muestra las siguientes opciones:

1. Carga de archivo del sistema.
2. Carga de archivo para la prueba inicial.
3. Crear nueva empresa.
4. Ver empresas.
5. Limpiar sistema.
1. Salir

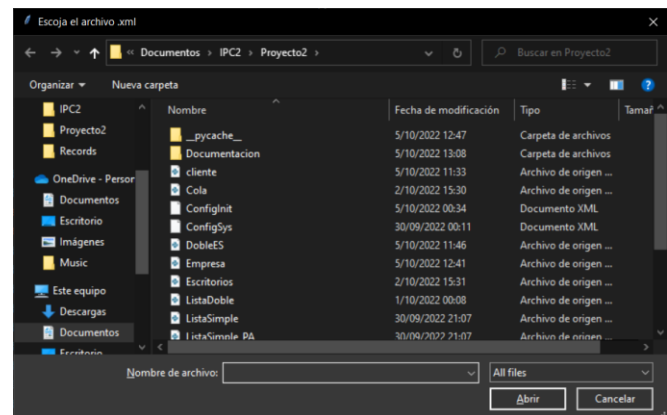
```

=====
|--Soluciones Guatemaltecas S.A.--|
=====
-----Menú principal-----
1. Cargar archivo C-S
2. Cargar archivo C-I-P
3. Crear Nueva empresa
4. Ver empresas
5. Limpiar sistema
0. Salir

Ingrese la opción a realizar
-> 4

```

La opción 1 y 2 permite cargar los datos de las empresas y los clientes para la simulación para esto presenta una ventana emergente la cual permite buscar los archivos a cargar.



Una vez cargado eligiendo la opción 4 se podrá visualizar los datos de la empresa o en caso de que se requiera una crear una empresa se elige la opción 3.

```

Ingrese la opción a realizar
-> 4

=====
|--Soluciones Guatemaltecas S.A.--|
=====
-----Listado de empresas-----
ID: BNR Nombre: Banco de Desarrollo Rural Abreviatura: Banrural
ID: CLR Nombre: Claro Abreviatura: Claro

Ingrese el ID de la empresa para visualizar
sus puntos de atención ó presione 0 para regresar
-> 0

```

La carga de datos tiene un funcionamiento complejo, puesto que usa la librería `xml`, para su carga, se utilizó POO para crear un objeto para la empresas y derivados.

```

def file_upload(type):
    if type==1:
        try:
            archivo=open('Escoja el archivo .xml', 'r')
            tree = ET.parse(archivo)
            root=tree.getroot()
            for empresa in root:
                id = empresa.attrib['id']
                nombre = empresa.attrib['nombre']
                abr = empresa.attrib['abr']
                empresa.CreateEmpresa(id,nombre,abr)
                for puntoatencion in empresa[2]:
                    idpuntoatencion= puntoatencion.attrib['id']
                    nombre_puntoatencion= puntoatencion.attrib['nombre']
                    direccion= puntoatencion.attrib['direccion']
                    empresa.getEmpresa(id).lista_PAs.InsertarPunto_Atencion(idp,nombre_puntoatencion,direccion)
                for escritorio in puntoatencion[2]:
                    id_escritorio=escritorio.attrib['id']
                    identificacion=escritorio.attrib['identificacion']
                    encargado=escritorio.attrib['encargado']
                    escritorio.lista_escritorios.InsertarEscritorio(id_escritorio,identificacion,encargado,falso)
            for transacciones in empresa[3]:
                id_transaccion=transacciones.attrib['id']
                nombre_transaccion=transacciones.attrib['nombre']
                tiempo=transacciones.attrib['tiempo']
                empresa.getEmpresa(id).lista_transacciones.InsertarTransaccion(id_transaccion,nombre_transaccion,tiempo)
            print(" [ARCHIVO CARGADO CORRECTAMENTE] ")

```

Para visualizar los puntos de atención de cada empresa y sus transacciones, se introduce el **ID** de la empresa, este caso se escogió CLR de Claro para el ejemplo de este manual.

Sí se requiere regresar se introduce el número 0.

```
-----Listado de empresas-----
ID: BNR Nombre: Banco de Desarrollo Rural Abreviatura: Banrural
ID: CLR Nombre: Claro Abreviatura: Claro

Ingrese el ID de la empresa para visualizar
sus puntos de atención ó presione 0 para regresar
-> CLR

=====
|--Soluciones Guatemaltecas S.A.--|
=====
Empresa: Claro
-----Puntos de atención-----
ID: VN Nombre: Claro Villa Nueva Dirección: Centro comercial zona 1
ID: AMTLN Nombre: Claro Amatitlan Dirección: Centro comercial Flores

-----Transacciones-----
ID: COCA Nombre: Contratar casa claro Tiempo: 10 min
ID: CASE Nombre: Cancelar servicios e irse a tigo que es mejor :V Tiempo: 8 min
ID: PAME Nombre: Pagar mensualidad del servicio Tiempo: 5 min
ID: COPR Nombre: Comprar productos de la tienda Tiempo: 10 min

Ingrese el ID del punto de atención para visualizar sus acciones de manejo ó 0 para regresar.
-> 0
```

Al introducir el ID de la empresa se visualizan primero los puntos de atención con su respectivo ID, nombre y dirección, segundo las transacciones con su respectivo ID, nombre y tiempo de cada transacción.

Para continuar se debe introducir el ID del punto de atención, en este caso se introduce AMTLN para el ejemplo de este manual.

```
Ingrese el ID del punto de atención para visualizar sus acciones
-> AMTLN

=====
|--Soluciones Guatemaltecas S.A.--|
=====
Punto de atención: Claro Amatitlan
-----Manejo-----
1. Estado del punto de atención
2. Activar escritorio
3. Desactivar escritorio
4. Atender cliente
5. Solicitud atención
6. Simular actividad
0. Regresar

Ingrese la opción a realizar
-> 1
```

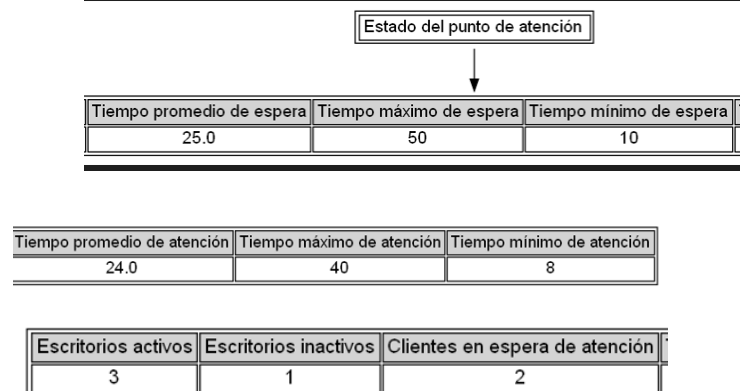
Aquí se muestra el siguiente menú de opciones de manejo del punto de atención, en el cuál tenemos las siguientes opciones:

1. Estado del punto de atención

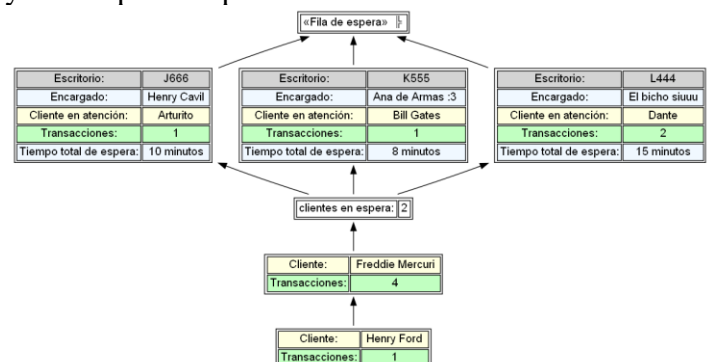
2. Activar escritorio.
3. Desactivar escritorio.
4. Atender cliente.
5. Solicitud de atención.
6. Simular actividad.
0. Regresar.

1. Estado del punto de atención:

Esta opción muestra primero de forma gráfica la información del punto de atención; la cantidad de escritorios activos e inactivos, clientes en espera, tiempo de espera y de atención, máximo, mínimo y promedio respectivamente.



Siguiente de la gráfica anterior, se muestra la gráfica de la cola o fila de espera del punto de atención, mostrando la identificación del escritorio y su encargado junto del cliente que está siendo atendido y el tiempo de espera.



```

Generando gráfica

---Estado del punto de atención---
Atención: Se guardo el archivo con el nombre: Info. Punto atención.png

[Presione ENTER]

---Estado escritorios activos---
Atención: Se guardo el archivo con el nombre: Fila de espera.png

[Presione ENTER]

```

Al terminar esto vuelve al menú de manejo del punto de atención.

2. Activar escritorio:

Con la gráfica mostrada anteriormente se pueden visualizar los escritorios activos e inactivos, con lo cual se puede saber cuántos pueden ser activados.

De igual forma se muestran en pantalla los escritorios activos e inactivos. Para activar un escritorio debe estar inactivo e introducir el ID del escritorio.

```

-> 2
ID: I700 Identificación: I777 Encargado: Nicola Tesla Estado: Inactivo
ID: J600 Identificación: J666 Encargado: Henry Cavil Estado: Activa
ID: K500 Identificación: K555 Encargado: Ana de Armas :3 Estado: Activa
ID: SIUU Identificación: L444 Encargado: El bicho siuuu Estado: Activa

Ingrese el ID del escritorio del cual desea activar.
-> I700

Escritorio Activado

```

3. Desactivar escritorio:

Con la gráfica mostrada anteriormente se pueden visualizar los escritorios activos e inactivos, con lo cual se puede saber cuántos pueden ser activados.

De igual forma se muestran en pantalla los escritorios activos e inactivos. Para desactivar un escritorio debe estar activo y no tener un cliente siendo atendido e introducir el ID del escritorio.

```

ID: I700 Identificación: I777 Encargado: Nicola Tesla Estado: Activa
ID: J600 Identificación: J666 Encargado: Henry Cavil Estado: Activa
ID: K500 Identificación: K555 Encargado: Ana de Armas :3 Estado: Activa
ID: SIUU Identificación: L444 Encargado: El bicho siuuu Estado: Activa

Ingrese el ID del escritorio del cual desea desactivar.
-> I700

Escritorio Desactivado

```

4. Atender Cliente:

Esta opción realiza las transacciones del cliente en escritorio más próximo a salir en base a su tiempo de espera, mostrando el escritorio y el cliente atendido, dejando libre el escritorio para que pase el siguiente cliente en espera, para comprobar esto se puede presionar la opción 1 para ver de forma gráfica el proceso.

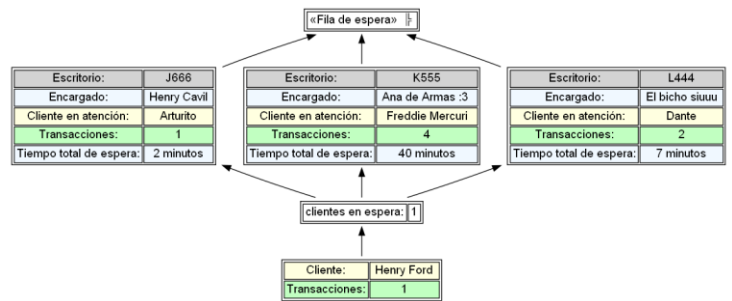
```

Ingrese la opción a realizar
-> 4
Cliente: Bill Gates atendido.
Escritorio K555 libre.

Siguiente en pasar: Freddie Mercury

[Presione ENTER]

```



Fila o cola de espera actualizada.

5. Solicitud de atención:

Esta opción nos permite agregar un nuevo cliente a la cola o escritorio, requiero que sea introducido su DPI, nombre y cantidad de transacciones.

Para escoger la transacción se debe introducir el ID de la transacción las cuales serán mostradas en pantalla.

```

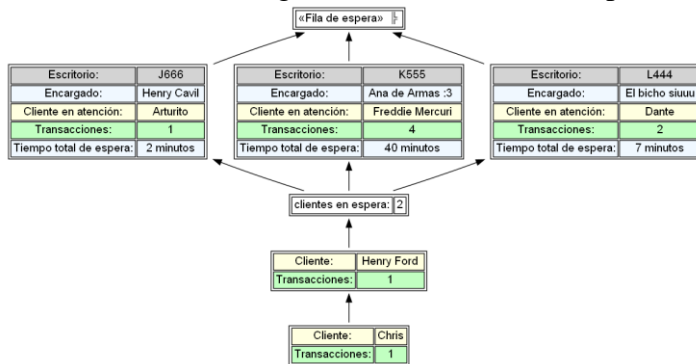
--Nuevo cliente--
DPI: 3033488440108
Nombre: Chris
ID: COCA Nombre: Contratar casa claro Tiempo: 10 min
ID: CASE Nombre: Cancelar servicios e irse a tigo que es mejor :V Tiempo: 8 min
ID: PAME Nombre: Pagar mensualidad del servicio Tiempo: 5 min
ID: COPR Nombre: Comprar productos de la tienda Tiempo: 10 min

Cantidad de transacciones: 1
Ingrese el ID de la transacción: PAME

Cliente añadido a la cola

```

Una vez agregado un cliente se puede volver a visualizar de forma gráfica la fila o cola de espera.



Como se puede visualizar, el ultimo cliente agregado ya se muestra en el gráfico de la fila de espera.

6. Simular actividad:

Esta opción ejecuta de forma automática el proceso de la fila o cola de espera, atendiendo a todos los clientes en el orden ya especificado y mostrando cuántos escritorios están activos o inactivos y cuántos clientes atendió cada escritorio.

Simulación de actividad

Cliente: Arturito atendido.
Escritorio J666 libre.

Siguiente en pasar: Henry Ford

Cliente: Dante atendido.
Escritorio L444 libre.

Siguiente en pasar: Chris

Cliente: Henry Ford atendido.
Escritorio J666 libre.

Cliente: Chris atendido.
Escritorio L444 libre.

Cliente: Freddie Mercuri atendido.
Escritorio K555 libre.

Cientes atendidos en el punto de servicio: 6

Escritorios activos: 3
Escritorios inactivos: 1

2 clientes atendidos en el escritorio: J666
2 clientes atendidos en el escritorio: K555
2 clientes atendidos en el escritorio: L444

Simulación terminada

Por último, en el menú principal se tiene la opción limpiar sistema la reestablece la información de los puntos de atención, sin el archivo de prueba inicial.

Conclusiones

La utilización de listas enlazadas creadas desde cero, son más funcionales que las listas por defecto de un lenguaje de programación.

La optimización con el uso de listas es complicada si no se usa una lista de tipo de pila

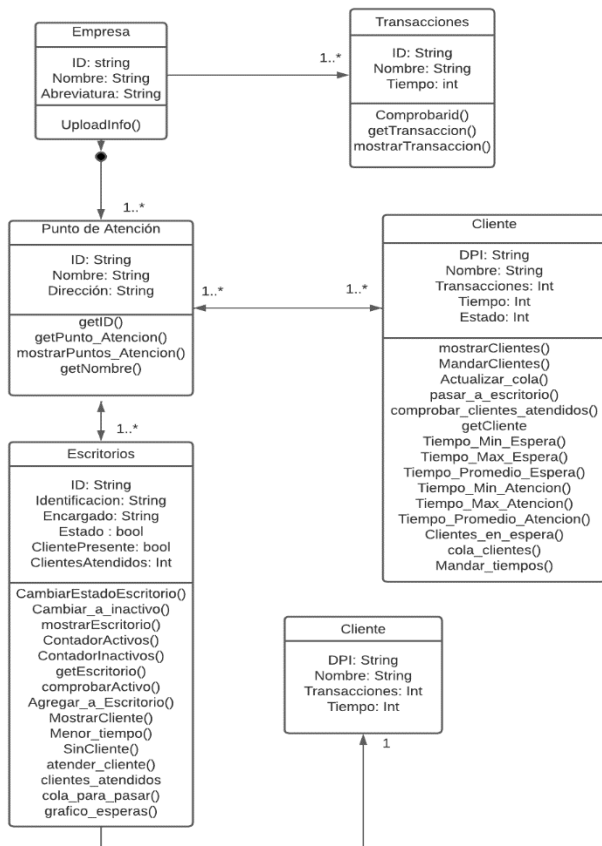
Referencias bibliográficas

MÉNDEZ, Mariano. 75.41 Algoritmos y Programación II Tda Lista y sus Derivados.

OJEDA, Luis Roberto. Tda Programacion Orientado a Objetos en Turbo. Univ. Nacional de Colombia.

Anexos:

Diagrama de clases:



Métodos de main.py :

```

def file_upload(type): #tipo tipo de archivo xml a subir
    if type==1:
        try:
            archivo=easygui.fileopenbox(title="Escoge el archivo .xml",filetypes=("Archivos xml","*.xml"))
            tree = ET.parse(archivo)
            root=tree.getroot()
            for empresa in root:
                id = empresa.attrib['id']
                nombre =empresa[0].text
                abr = empresa[1].text
                Empresa.crearEmpresa(id,nombre, abr)
                for puntosAtencion in empresa[2]:
                    idPA=puntosAtencion.attrib['id']
                    Nombre_PA=puntosAtencion[0].text
                    direccion=puntosAtencion[1].text
                    Empresa.getEmpresa(id).lista_Pds.InsertarPunto_Atencion(idPA,Nombre_PA,direccion)
                    Escritorio=Empresa.getEmpresa(id).lista_Pds.getPunto_Atencion(idPA)
                    for escritorios in puntosAtencion[2]:
                        id_escritorios=escritorios.attrib['id']
                        identificacion=escritorios[0].text
                        Encargado=escritorios[1].text
                        Escritorio.lista_escritorios.InsertarEscritorio(id_Escritorio,identificacion,Encargado,False)
                for transacciones in empresa[3]:
                    id_transaccion=transacciones.attrib['id']
                    nombre_transaccion=transacciones[0].text
                    tiempo=transacciones[1].text
                    Empresa.getEmpresa(id).lista_Transacciones.InsertarTransaccion(id_transaccion,nombre_transaccion,tiempo)
        except:
            print(" [ARCHIVO CARGADO CORRECTAMENTE]")
  
```

```

if type==2:
    try:
        tree = ET.parse(easygui.fileopenbox(title="Escoge el archivo .xml",filetypes=("Archivos xml","*.xml")))
        root=tree.getroot()
        for configuracion in root:
            id_empresa=configuracion.attrib['id_empresa']
            id_punto=configuracion.attrib['id_punto']
            Escritorio=Empresa.getEmpresa(id_empresa).lista_Pds.getPunto_Atencion(id_punto).lista_escritorios
            cliente=Empresa.getEmpresa(id_empresa).lista_Pds.getPunto_Atencion(id_punto).lista_clientes
            Transaccion=Empresa.getEmpresa(id_empresa).lista_Transacciones
            for escritorios in configuracion[0]:
                activar =escritorios.attrib['id_escritorio']
                Escritorio.CambiarEstadoEscritorio(activar)
            for clientes in configuracion[1]:
                total=0
                ntransacciones=0
                sumando = 0
                for transacciones in clientes[1]:
                    ntransacciones=Int(transacciones.attrib['cantidad'])
                    sumando = sumando+(ntransacciones *Int(Transaccion.getTransaccion(transacciones.attrib['idtransaccion'])))
                total = total+ntransacciones
                dpi=clientes.attrib['dpi']
                nombrecliente=clientes[0].text
                Cliente.InsertarCliente(dpi,nombrecliente,total,sumando,0)
                if Escritorio.comprobarActivo(Escritorio.getActivos())==True:
                    Escritorio.AgregarCliente(Escritorio.getActivos(),cliente.MandarCliente(dpi))
            print(" [ARCHIVO CARGADO CORRECTAMENTE]")
        except:
            print(" [Error]")
  
```

```

def crearEmpresa():
    codigo_Empresa = input("ID de la empresa: ")
    Nombre_Empresa = input("Nombre de la empresa: ")
    abreviatura = input("Abreviatura: ")
    Empresa.crearEmpresa(codigo_Empresa,Nombre_Empresa,abreviatura)# Creamos una empresa
    PA= Empresa.getEmpresa(codigo_Empresa)
    #Pedimos la cantidad de puntos de atención que tendrá la empresa
    cantidad_puntos_atencion= int(input(Fore.RED+"Cantidad de puntos de atención: "))
    print(Fore.GREEN+"Ingrese la información de los puntos de servicio de: [",Nombre_Empresa,")")
    for i in range (cantidad_puntos_atencion):
        codigo_punto=int(input(Fore.WHITE+"\nID del punto de servicio: "))
        nombre_punto = input("Nombre del punto de servicio: ")
        direccion= input("Dirección del punto de servicio: ")
        PA.lista_Pds.InsertarPunto_Atencion(codigo_punto,nombre_punto,direccion)
        Escritorio = PA.lista_Pds.getPunto_Atencion(codigo_punto)
        #Pedimos la cantidad de Escritorios del punto de servicio
        cantidad_escritorios= int(input(Fore.RED+"\nCantidad de escritorios del punto de servicio: "))
        print(Fore.GREEN+"Ingrese la información de los escritorios del punto de servicio : [",nombre_punto,")")
        for k in range(cantidad_escritorios):
            codigo_escritorio= int(input(Fore.WHITE+"\nID del escritorio: "))
            identificacion= input("Identificación del escritorio: ")
            encargado= input("Encargado del escritorio: ")
            Escritorio.lista_escritorios.InsertarEscritorio(codigo_escritorio,identificacion,encargado,False)
        cantidad_transacciones= int(input(Fore.RED+"\nCantidad de transacciones: "))
        print(Fore.GREEN+"Ingrese la información de las transacciones de: [",Nombre_Empresa,")")
        for s in range (cantidad_transacciones):
            codigo_transaccion = int(input(Fore.WHITE+"\nID de la transacción: "))
            nombre_transaccion = input("Nombre de la transacción: ")
            tiempo = int(input("Tiempo de la transacción(minutos): "))
            PA.lista_Transacciones.InsertarTransaccion(codigo_transaccion,nombre_transaccion,tiempo)
        print("\nEmpresa añadida correctamente")
  
```



```

def limpiar_sistema(archivo):
    try:
        tree=ET.parse(archivo)
        root=tree.getroot()
        for empresa in root:
            id = empresa.attrib['id']
            nombre =empresa[0].text
            abr = empresa[1].text
            limpiar.createEmpresa(id,nombre, abr)
            for puntosAtencion in empresa[2]:
                idPA=puntosAtencion.attrib['id']
                Nombre_PA=puntosAtencion[0].text
                direccion=puntosAtencion[1].text
                limpiar.getEmpresa(id).lista_Pds.InsertarPunto_Atencion(idPA,Nombre_PA,direccion)
            Escritorio=limpiar.getEmpresa(id).lista_Pds.getPunto_Atencion(idPA)
            for escritorios in puntosAtencion[2]:
                id_Escritorio=escritorios.attrib['id']
                identificacion=escritorios[0].text
                Encargado=escritorios[1].text
                Escritorio.lista_escritorios.insertarEscritorio(id_Escritorio,identifiacion,Encargado)
            Empresa.getEmpresa(id).lista_Pds=limpiar.getEmpresa(id).lista_Pds
    except:
        print(" [ERROR]")

def Ver_Empresas():
    print(" =====")
    print(" ||--Soluciones Guatemaltecas S.A.--||")
    print(" =====")
    print(" -----listado de empresas-----")
    Empresa.mostrarEmpresas()
    print("\n Ingrese el ID de la empresa para visualizar\nsus puntos de atención ó presione 0 para regresar")

def Ver_puntos_atencion(codigo):
    print(" =====")
    print(" ||--Soluciones Guatemaltecas S.A.--||")
    print(" =====")
    print(" Empresa: ",Empresa.getNombre(codigo))
    print(" -----Puntos de atención-----")
    Empresa.getPuntos_Atencion(codigo).lista_Pds.mostrarPuntos_Atencion()
    print("\n -----Transacciones-----")
    Empresa.getEmpresa(codigo).lista_Transacciones.mostrarTransaccion()
    print("\nIngrese el ID del punto de atención para visualizar sus acciones de manejo ó 0 para regresar")

def Manejo_puntosDeAtencion(codigo,codigo2):
    print(" =====")
    print(" ||--Soluciones Guatemaltecas S.A.--||")
    print(" =====")
    print(" Punto de atención: ",Empresa.getEmpresa(codigo).lista_Pds.getNombre_Punto_Atencion(codigo2))
    print(" -----Manejo-----")
    print(" 1. Estado del punto de atención")
    print(" 2. Activar escritorio ")
    print(" 3. Desactivar escritorio ")
    print(" 4. Atender cliente ")
    print(" 5. Solicitud atención")
    print(" 6. Simular actividad")
    print(" 0. Regresar")
    print("\n Ingrese la opción a realizar")

```

```
def Nueva_Empresa():  
    print("=====")  
    print("||--Soluciones Guatemaltecas S.A.--||")  
    print("=====")  
    print("-----Crear nueva empresa-----")  
    print("\n\tIngrese los datos de la empresa.\n")  
  
def MenuPrincipal():  
    print("=====")  
    print("█---Soluciones Guatemaltecas S.A.---█")  
    print("=====")  
    print("-----Menú principal-----")  
    print("\t\t1. Cargar archivo C-S")  
    print("\t\t2. Cargar archivo C-I-P")  
    print("\t\t3. Crear Nueva empresa")  
    print("\t\t4. Ver empresas")  
    print("\t\t5. Limpiar sistema")  
    print("\t\t0. Salir")  
    print("\n\t\t\t\tIngrese la opción a realizar")
```

```

while APPAPP: #Inicio de la app
    MenuPrincipal()
    try:
        choice = int(input("          -> ")) #Elección
        if choice>6:
            print("La opción no es válida\n")
        else:
            if choice==0:
                print("          [Ejecución terminada]")
                startAPP= False
            elif choice==1: #Carga archivo de configuración del sistema
                data= file_upload(1)
            elif choice==2: #Carga archivo inicial para la prueba
                file_upload(2)
            elif choice==3:
                Nueva_Empresa()
                crearEmpresa()
            elif choice==4:
                verEmpresas= True
                while verEmpresas:
                    print("")
                    Ver_Empresas()
                    id_empresa = input("          -> ")
                    if id_empresa=="0":
                        verEmpresas=False
                        #confirma que el código sea válido
                    else:
                        if Empresa.getCodigo(id_empresa)==True:
                            empresav= Empresa.getPuntos_Atencion(id_empresa).lista_Pds
                            transacciones=Empresa.getEmpresa(id_empresa).lista_Transacciones
                            codigo = id_empresa
                            verPuntosAtencion= True
                            while verPuntosAtencion:

```

```

elif opcion=="3":
    empresav.getPunto_Atencion(id_PA).lista_escritorios.mostrarEscritorio()

    print("\nIngrese el ID del escritorio del cual desea desactivar.")
    id_escritorio= input("    -> ")
    print("")
    if empresav.getPunto_Atencion(id_PA).lista_escritorios.getID_Escritorio(id_escritorio)==True:
        if empresav.getPunto_Atencion(id_PA).lista_escritorios.Cambiar_a_inactivo(id_escritorio)==True:
            print("\n        Escritorio Desactivado")
        else:
            print("\n        Atención -> ¡Este escritorio ya está inactivo!")
    else:
        print("        No se ha logrado completar la acción.")

elif opcion=="4":
    Cliente=empresav.getPunto_Atencion(id_PA).lista_clientes
    escritorio=empresav.getPunto_Atencion(id_PA).lista_escritorios
    empresav.getPunto_Atencion(id_PA).lista_clientes.ActualizarCola(escritorio.atender_cliente())
    if Cliente.len0:
        if Cliente.clientes_en_espera()>0:
            for i in range (escritorio.SinCliente()):
                escritorio.AgregarCliente(escritorio.getActivos()),Cliente.MandarCliente(Cliente.pasar_a_escritorio(i))
            input("\n        [Presione ENTER] \n")
        else:
            print("        No hay clientes para atender")
    elif opcion=="5":
        Escritorio=empresav.getPunto_Atencion(id_PA).lista_escritorios
        sumando = 0
        print("        --Nuevo cliente--")
        dpi=input("        DPI: ")
        nombre= input("        Nombre: ")
        transacciones.mostrarTransaccion()
        ntransacciones= int(input("\n        Cantidad de transacciones: "))
        for i in range (ntransacciones):
            id_transaccion= input("        Ingrese el ID de la transacción: ")
            sumando +=sumando+int(transacciones.gettransaccion(id_transaccion))
        empresav.getPunto_Atencion(id_PA).lista_clientes.InsertarCliente(dpi,nombre,ntransacciones,sumando,0)
        if Escritorio.comprobarActivo(escritorio.getActivos())==True:
            Escritorio.AgregarCliente(escritorio.getActivos()),empresav.getPunto_Atencion(id_PA).lista_clientes.MandarCliente(dpi)
            print("Cliente pasa a escritorio")
        else:
            print("\n        Cliente añadido a la cola *")
    elif opcion=="6":
        Cliente=empresav.getPunto_Atencion(id_PA).lista_clientes
        escritorio=empresav.getPunto_Atencion(id_PA).lista_escritorios
        print("        [Simulación de actividad]")
        print("        [Simulación de actividad]")

simular=True
while simular:
    empresav.getPunto_Atencion(id_PA).lista_clientes.ActualizarCola(escritorio.atender_cliente())
    if Cliente.len0:
        if Cliente.clientes_en_espera()>0:
            for i in range (escritorio.SinCliente()):
                escritorio.AgregarCliente(escritorio.getActivos()),Cliente.MandarCliente(Cliente.pasar_a_escritorio(i))
        if Cliente.comprobar_clientes_atendidos()==True:
            print("\n        Escritorios activos: ",escritorio.ContadorActivos())
            print("        Escritorios inactivos: ",escritorio.ContadorInactivo())
            print("")
            escritorio.clientes_atendidos()
            simular=False
            print("        [Simulación terminada]")
            print("        [Presione ENTER] \n")
        else:
            print("        Opción inválida")
    else:
        print("        ID incorrecto")
    elif opcion=="7":
        print("        ID incorrecto")
    elif opcion=="8":
        limpiar_sistema(data)
        print("\n        [Sistema reiniciado exitosamente]")
        print("        [Sistema reiniciado exitosamente]")
except:
    print("Opción inválida")

```