
PROYECTO 2 – PROGRAMA ADMINISTRACIÓN DE EMPRESAS – “SOLUCIONES GUATEMALTECAS S.A”

202110206 – Julio Alejandro Zaldaña Ríos

Resumen

En el presente proyecto, se da a conocer una forma para manejar información de empresas, en diferentes sucursales, manejo de clientes en escritorios dentro de sucursales, etc. Donde se hace uso de estructuras de datos (listas enlazadas) para crear y almacenar valores/datos desde consola y mediante la lectura de archivos XML. Todo utilizando el lenguaje de programación Python, siempre cumpliendo con los principios FIFO y LIFO, al igual que el uso de clases nodos y la programación orientada a objetos.

Palabras clave

Python

XML

Lista Enlazada

FIFO (first in- first out)

LIFO (Last in – first out)

Abstract

In the present project, it is shown a way to manage companies' information, in different subsidiaries, client management oriented inside the subsidiaries etc. Where it is used data structures (linked lists) to create and save data from console and by parsing XML files. Everything was done in the programming language Python. Always following the principles of FIFO and LIFO, and by the usage of node classes and Object-oriented-programming.

Keywords

Python

XML

Linked List

FIFO (first in- first out)

LIFO (Last in – first out)

Introducción

A continuación, se presentará un programa en el que primordialmente, se podrá manejar información de empresas como parte de una solución propuesta por “Soluciones Guatemaltecas S.A”.

Desarrollo del tema

Se trabaja para poder manejar información dentro de una empresa, que principalmente será dividida por distintos atributos importantes:

- Empresa:
 - Sucursales (Puntos de Atención)
 - Escritorios
 - Activo o Inactivo

Existirán clientes, que podrán ingresar y realizar distintas transacciones en las empresas.

El objetivo es que ese cliente sea atendido en un escritorio disponible y activo dentro de la empresa, en tiempo de ejecución.

Para el manejo del programa desarrollado, se centra en:

Mediante la utilización de Python, en un archivo llamado main, se definió un menú que se mostrará en consola, donde se contendrán hasta 18 opciones interactivas.

```
-----  
MENU - SOLUCIONES GUATEMALTECAS S.A  
1. Ingresar Empresa  
2. Selección de Empresas  
3. Eliminar Empresa  
4. Crear Puntos de Atención  
5. Ver puntos de atención por Empresas  
6. Crear escritorios  
7. Ver escritorios en sucursal de Empresa  
8. Crear una transaccion para una empresa  
9. Mostrar transacciones para empresas  
10. Cargar Archivo Configuracion Sistema  
-----  
11. Crear Cliente  
12. Ver Cola de Clientes  
13. Cargar Archivo Configuracion Inicial  
14. Activar Escritorio  
15. Desactivar Escritorio  
16. Mostrar Escritorios Activos  
17. Atender Cliente en Escritorio  
18. Salir  
-----  
Ingrese una opcion:
```

Figura 1. Menú de programa

Fuente: elaboración propia, 2022

Detalle del funcionamiento de cada opción:

1. Ingresar Empresa:

Se le preguntará al usuario, los siguientes datos:

1. Id para la empresa
2. Un nombre para la empresa
3. Abreviatura para la empresa

Así se generará y creará una empresa nueva.

2. **Selección de empresas:** Se podrá visualizar la lista de empresas que están creadas en el programa.

3. **Eliminar Empresa:** Se le preguntará el nombre de la empresa, que se desea eliminar del programa.

4. **Crear Puntos de Atención:** Se podrán crear puntos de atención dentro de la empresa. (Sucursales o puestos dentro de la empresa)

Se le preguntará al usuario, el id de la empresa donde quiere ingresar su punto de atención, con los siguientes datos:

1. Id para el punto de atención
2. Nombre para el punto de atención
3. Dirección para el punto de atención

5. **Ver Puntos de Atención por Empresas:** Se visualizarán los puntos de atención de una empresa que se seleccionará por medio de su id.

6. **Crear Escritorios:** Se podrán crear escritorios dentro de cualquiera punto de atención/sucursal que desee el usuario.

Para poder crear un escritorio se necesita que el usuario ingrese el id de la empresa y el id del punto de atención, a donde quiera que el escritorio pertenezca, se tendrá que llenar los siguientes datos:

1. Id del escritorio
2. Nombre de identificación de escritorio
3. Nombre del encargado del escritorio

7. **Ver escritorios en de sucursal de Empresa:** Permitirá al usuario poder visualizar la lista de escritorios que se encuentra dentro de la sucursal de una empresa.

Se tendrá que brindar el id de la empresa y del punto de atención, para poder visualizar los distintos escritorios dentro de las mismas.

8. **Crear una transacción para una empresa:**

Se le permitirá al usuario poder crear manualmente una transacción que se añadirá a la lista general de transacciones que pertenece a una empresa, se le piden los siguientes datos:

1. Id de transacción
2. Nombre de la transacción
3. Tiempo de duración de la transacción

9. **Mostrar transacciones para empresas:** Se podrá visualizar la lista de transacciones dentro de cualquier empresa.

Se puede acceder ingresando el id de la empresa, que se requiera ver.

10. **Cargar archivo de configuración de sistema:** Cargará el archivo XML, con los datos o valores que se pueden ingresar al programa para manejar muchos más datos.

11. **Crear cliente:** Se podrá crear un cliente manualmente, ingresando los siguientes datos:

1. DPI del cliente
2. Nombre del cliente
3. Transacción que realizará el cliente

12. **Ver Cola de Clientes:** Mostrará la lista de clientes dentro de la cola de espera, antes de acceder a un escritorio dentro de una sucursal.

13. Cargar Archivo Configuración Inicial:

Cargará un archivo XML, con los datos o valores que se pueden ingresar al programa para manejar muchos más datos. (Este contiene datos para clientes y escritorios activos)

14. Activar Escritorio: Se podrá activar un escritorio de forma manual. Donde se pedirá al usuario que ingrese, el id de la empresa, id del punto de atención y del id del escritorio que quiera activar.

15. Desactivar Escritorio: Se podrá desactivar un escritorio de forma manual. Donde se pedirá al usuario que ingrese el id de la empresa, id del punto de atención y el id del escritorio activo que quiera desactivar.

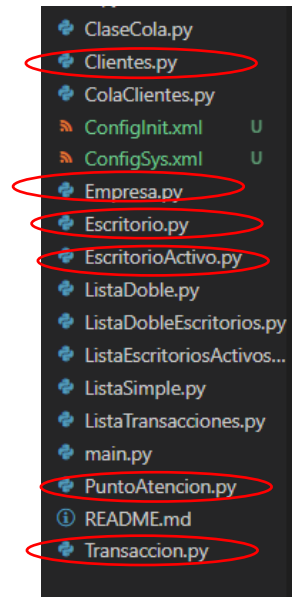
16. Mostrar Escritorios Activos: Se podrá visualizar una lista de escritorios activos dentro de la sucursal de la empresa.

17. Atender Cliente en Escritorio: Se le pedirá al usuario el id de la empresa, id del punto de atención y el id del escritorio donde se atenderá a un cliente de la cola de cliente de espera.

Luego se le permitirá al cliente retirarse.

18. Salir: Se saldrá del programa.

Se crearon distintas clases con atributos de los distintos objetos que se manejan dentro del programa:



Toda clase, tiene su propia estructura de datos:

- Cliente -> colaClientes / Clase Cola
- Empresa -> ListaSimple
- PuntoAtencion -> ListaDoble
- Escritorio -> ListaDobleEscritorios
- Transaccion -> ListaTransacciones
- EscritorioActivo -> ListaEscritoriosActivos

Siguiendo los conceptos de lista simples y listas dobles para poder manejar, crear y manipular distintos datos, ya sea por consola o por la carga de un archivo XML.

```
from Empresa import Empresa

#Lista para empresas

class ListaSimple():
    def __init__(self):
        self.inicio=None
        self.fin=None

    def añadirEmpresa(self,id,nombre,abreviatura):
        nuevo=Empresa(id,nombre,abreviatura)
        if self.inicio is None:
            self.inicio=nuevo
        else:
            tmp=self.inicio
            while tmp.siguiente is not None:
                tmp=tmp.siguiente
            tmp.siguiente=nuevo

    def imprimirEmpresas(self):
        print("-----")
        print(" La lista de empresas es:")
        print("-----")
        tmp=self.inicio
        while tmp is not None:
            print("id: ",tmp.id," Nombre: ",tmp.nombre," Abreviatura: ",tmp.abreviatura)
            tmp=tmp.siguiente
```

Figura 2. Ejemplo de la clase nodo de empresa, donde se definen sus atributos.

Fuente: elaboración propia, 2022

```
from ListaDoble import ListaDoble
from ListaDobleEscritorios import ListaEscritorio
from ListaTransacciones import ListaTransacciones

class Empresa:

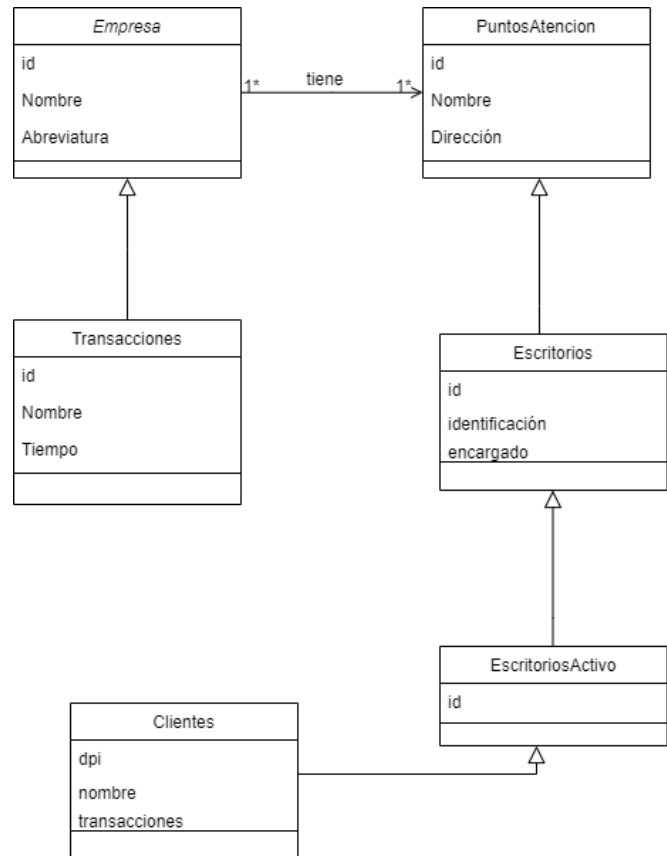
    def __init__(self, id, nombre, abreviatura):
        self.id=id
        self.nombre=nombre
        self.abreviatura=abreviatura
        self.lista_puntoAtencion=ListaDoble()
        self.lista_escritorios=ListaEscritorio()
        self.lista_transacciones=ListaTransacciones()
        self.siguiente=None

    def getPuntoAtencion(self):
        return self.lista_puntoAtencion
```

Figura 3. Ejemplo de clase lista simple para almacenar y crear empresas.

Fuente: elaboración propia, 2022

Diagramas de Clases



Conclusiones

Es muy útil saber cómo funciona y tener un buen manejo de la programación orientada a objetos, al igual que el manejo y uso de las listas simples, dobles y doblemente enlazadas, utilizando clases nodos, así como se trabajó en el proyecto, ya que facilita mucho al momento de manejar distintos y una gran cantidad de objetos.

Al igual que es importante reconocer el concepto y funcionamiento de una cola y de una pila, ya que estas tienen distintas aplicaciones importantes, que pueden llegar a ser muy útiles en diferentes contextos. Así como se utilizó en la cola de clientes, o en la pila de escritorios activos.

Referencias bibliográficas

Documentacion Python, (1, octubre,2022).

*Xml.etree.ElementTree – La API XML de
ElementTree*

<https://docs.python.org/es/3/library/xml.etree.elementtree.html>