
SIMULACION DE ATENCION AL CLIENTE

202006681 – Kevin Haroldo Albizures Sirín

Resumen

Se realizó un software con las características necesarias para simular la atención de clientes utilizando TDAs en todo el proyecto para la correcta funcionalidad.

Por medio de el paradigma de orientación a objetos se consiguió manipular cada una de las empresas, puntos de atención, escritorios, servicios y clientes.

El propósito de este software es el de incentivar al usuario a aprender a manejar listas que no sean nativas de cualquier lenguaje, aumentando así la capacidad abstracta de quien pueda probar el software.

La forma en que muestra la información obtenida es por medio de Graphviz.

Palabras clave

Iteraciones: Son ciclos que realiza el programa del tejido de cada paciente.

Png: Formato gráfico.

Xml: Lenguaje de marcado, será el archivo que recibirá el programa.

Abstract

A software was made with the necessary characteristics to simulate customer service using TDAs throughout the project for the correct functionality.

By means of the object orientation paradigm, it was possible to manipulate each of the companies, service points, desks, services and clients.

The purpose of this software is to encourage the user to learn to handle lists that are not native to any language, thus increasing the abstract capacity of those who can test the software.

The way it displays the information obtained is through Graphviz.

Keywords

Iterations: These are cycles carried out by the tissue program for each patient.

Png: Graphic format.

Xml: Markup language, it will be the file that the program will receive.

Introducción

EL objetivo del software es poder manipular la información de diversas empresas las cuales poseen un servicio de atención de forma presencial, es por eso que se implemento una simulación de clientes para comprobar su perfecto funcionamiento.

El software solventa la falla de no llevar un control a la hora de recibir clientes dentro de un punto de atención, dejando impaciente a los usuarios sin saber el tiempo que les demorara ser atendidos, es por eso que surge la necesidad de automatizar y digitalizar la información que a los clientes les será de agrado poder observar.

El problema se abordó a través de la lógica de los archivos de lectura, pues esto brindo un panorama concreto sobre la estructura que se iba a implementar luego con TDAs.

El tema por el cual se trabajó listas con TDAs, fue con el propósito de ampliar la capacidad abstracta y el entendimiento de los temas relacionados a orientación a objetos.

Desarrollo De La Aplicación

Se comenzó implementando un menú que registrará las funcionalidades básicas del programa, partiendo de esta base cada opción brindan menús diferentes.

Configurar empresa: En este apartado abrirá otro menú el cual contendrá todas las funcionalidades requeridas para poder almacenar la información de nuevas empresas, puntos de atención, escritorios y servicios.

Selección de Empresa y Punto de Atención: En el apartado se realizó un menú dinámico en función de la cantidad de empresas almacenadas, cumple con la función de poder elegir el objeto empresa y el objeto punto de atención requeridos, si no se cargan los archivos de empresas, no se podrá ingresar a este apartado.

Manejo de Puntos de Atención: En este apartado luego de escoger los objetos de empresa y punto de atención, se nos podrá abrir un nuevo menú con las funcionalidades para poder ingresar clientes y poderlos atender individualmente o masivamente, también podemos desactivar y activar escritorios, además podemos simular una cola de clientes para poder registrar los valores del tiempo en máximos y mínimos, con el objetivo de poder ser más dinámicos con la información, también se puede generar un archivo por medio de Graphviz para poder mostrar el contenido de los escritorios desactivados y activados, también la cola de clientes.

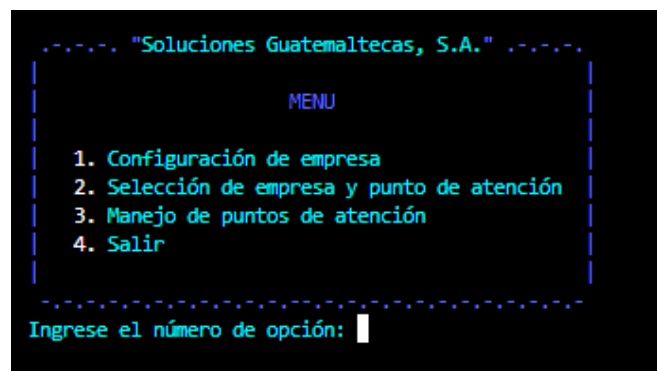


Figura 1. Menú Principal.

Fuente: elaboración propia.

Para trabajar con el contenido de las listas, fue necesario crear cinco clases distintas de con sus respectivos métodos y atributos, cada clase

almacenaba los valores en conjunto de toda la información de cada empresa.

```

345 #2 Nodo Empresa
346 > class Empresa():...
452
453 #3 Puntos de atención
454 > class Puntos_Atencion():...
569
570 #4 Servicios
571 > class Servicios():...
579
580 #5 Escritorios
581 > class Escritorios():...
599
600 #6 Nodo Clientes
601 > class Cliente():...
612

```

Figura 2. Clases Utilizadas.

Fuente: elaboración propia.

El orden en que se fueron implementando las clases sería el siguiente:

1. ListaEmpresas: Dentro de ella se almaceno la clase Empresa.
2. Empresa: Dentro de ella se almaceno los servicios y los puntos de atención.
3. Puntos de Atención: Se almacenaron las clases de escritorios y clientes.

Se hizo implementación de Tkinter para poder abrir el explorador de archivos únicamente.

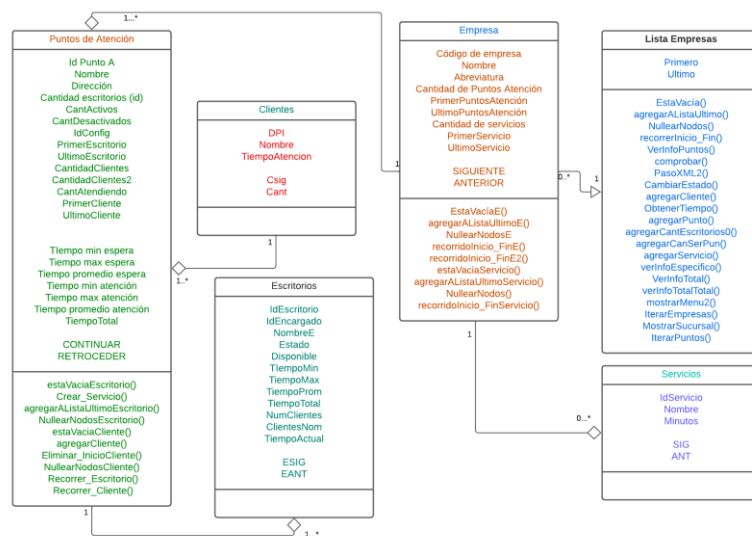


Figura 3. Diagrama de Clases.

Fuente: elaboración propia.

Generación de graficas: Por medio de Graphiz se guardo la información de todos los tiempos de cada escritorio así como la cola de clientes que se iba reduciendo conforme se iban atendiendo.

Conclusiones

Sin duda usar listas nativas en Python ahorran una cantidad enorme de código, sin embargo, a la hora de aplicar punteros se puede llegar a comprender el funcionamiento de la lista nativa.

Los TDAs nos brindan demasiadas posibilidades a la hora de querer manipular la información con un enfoque demasiado dinámico, pues nos otorga la capacidad de poder conjugar la información de diversas maneras.

El paradigma de POO es un método que brinda demasiada utilidad a la hora de trabajar con información encerrada en categorías.

Los archivos de marcado como lo es xml, markdown, etc. Brindan grandes posibilidades poder estructurar la información de forma que sea más rápida su lectura por programas secundarios.

Graphviz es una herramienta que brinda todo lo esencial para poder graficar información y poderla anidar de forma sencilla y eficiente.

Referencias bibliográficas

- Guagliano, C. (2019). *Programación en Python II: Programación orientada a objetos*. RedUsers
- *Python 3: los fundamentos del lenguaje*. (2016). Ediciones ENI.