Paradigmas de la programación - Final I

Eduardo Gómez

Esta aplicación fue contruida integramente con Python en su versión 3.7.9. El repositorio al proyecto junto a todos los ficheros originales se encuentran alojados en github.

Caracteristicas implementadas

- Registro de paquetes
- Registro de clientes
- Registro de empleados
- Registro de transportes
- Asignación automática de paquetes a transportes
- Calculo de costo de transpor

> *Si bien el sistema se encuentra avanzado, se necesitan agregar funciones para volverlo completamente funcional. Los problemas con los cuales me encontré están detallados al final del documento*

Organización

Cada fichero se encuentra cuidadosamente documentado con la siguiente estructura:

1. Descripción breve
2. Parametros
3. Argumentos y tipo de dato a recibir
4. Valor de retorno de la función

El proyecto cuenta con la siguiente estructura de directorios:

Core

El Core del sistema se encuenta modularizado de la siguiente manera:

- Class - Todas las clases que conforman el sistema
- Controller - La parte operativa o el controlador
- Database - Paquete que interactua con la base de datos

View

Con el fin de lograr una mayor modularización se estructura la interfaz de la siguiente manera:

1. **Componentes** En esta clase se encuentran los métodos necesarios para crear componentes en la interfaz grafica, tales como botones, texto, etc. Esta clase hereda las funciones de la biblioteca tkinter.
2. **Interfaz** En esta clase se encuentra la integración de la GUI y la Aplicación del sistema

GUI, esta a su vez cuenta con divisiones según el módulo del sistema. Alguno de ellos son:

- **Menu Principal:** En este se encuentran las opciones descriptas a continuación. La utilización de la interfaz consite en un selector numérico el cual permite ingresar a las opciones desplegadas.
- **Administración:**
- - Agregar Empleado: en este apartado se nos permite agregar un empleado a la empresa
- - Agregar Cliente: se solicitan los datos para el registro de un nuevo cliente. Los campos nombre, apellido y ci, son obligatorios para el correcto registro.
- **Paquetería:**
- - Registrar Paquete: todos los campos solicitados son obligatorios, con excepción del valor de articulo para los paquetes menores a 2Kg. *El input debe ser en gramos.*
- - Paquetes en transito: muestra los paquetes que ya poseen un transportes y esperan su entrega a destino
- - Paquetes pendientes: se nos muestra la cantidad de paquetes pendientes de transportes
- - Paquetes entregados: lista los paquetes que ya fueron entregados. **Función pendiente de implementación*
- **Transportes:**
- - Registrar transportes:todos los campos solicitados son obligatorios a excepción de la fecha de llegada a destino. En caso de no introducirse una fecha se asumirá el tiempo habitual de demora.
- - Ver transportes disponibles: se muestra con todos los transportes disponibles.

Controller

Este paquete se encarga de la interacción entre los objetos instanciados y la interfaz del sistema, así también se encarga de gestionar las llamadas a la base de datos y la validación de los datos. Todas las funciones se encuentran en el fichero App.py

Test

A modo de facilitar la correción se facilitaron algunos archivos para realizar test a la capa de control

Fichero	Acción a realizar
App	Test de todas la funciones del sistema
Empresa	Prueba de los métodos de la clase Empresa
Ticket	Prueba el módulo responsable de la generación de tickets

Instalación

Debian 10 (buster) o distribuciones basadas en Debian:

```
sudo apt install virtualenv
python3 -m env /app
source env/bin/activate

#copiar el .zip dentro del entorno virutal creado
unzip archivo.zip
python3 index.py

#se debe utilizar el interprete del entorno virtual para evitar errores de importación
```

Problemas a resolver

1. La persistencia de objetos no pudo ser implementada debido a un error desconocido al momento de ejecutar la biblioteca de pickle. *las lineas fueron comentadas para evitar los errores durante la ejecución
2. Las funciones de "entrega de paquetes" aún no fue implementado
3. Se necesita programar un evento para indicar que los embarques ya salieron del origen
4. Se requiere un módulo que actualice de manera constante la fecha global del sistema. *probablemente sea implementado para el trabajo final