



Registro de Empleados (Proyecto de Python)

Propósitos generales.

Este proyecto sencillo hecho en Python está hecho con la finalidad de practicar habilidades con el manejo de Interfaces Gráficas de Usuario (GUI's) y el manejo de archivos JSON. No pretende ser un sistema serio, al menos no a corto plazo.

Sobre el diseño de la aplicación.

Para empezar, el archivo `Trabajos.py` contiene a la clase Trabajador, que se encarga de representar a un empleado con los datos de nombre, apellido, sueldo, puesto, CURP, RFC y una ID única generada de acuerdo al puesto. Esta ID toma en cuenta cuantos empleados del mismo puesto se han registrado antes, para esto, hace uso de un contador (que es un archivo JSON), almacenado en la carpeta *ficheros*.

Por su parte, el archivo `Registro.py` se encarga de recibir los datos de un trabajador y almacenarlos en un fichero llamado `Trabajadores.json`. Es el intermediario entre los datos de un empleado y el almacén de información. Contiene un método para devolver un diccionario a partir de el archivo JSON.

La Interfaz de usuario.

La Interfaz de usuario está dividida en 3 clases, la principal, Interfaz.py, MuestraTrabajadores.py y RegistroInterfaz.py, todas ellas están contenidas en la carpeta *src/interfaces*. *Interfaz.py* se encarga de crear la raíz de la interfaz y proporcionarle al usuario la capacidad de moverse entre la “pestaña” de registro de nuevos trabajadores (de esto se encarga *RegistroInterfaz.py*) y la “pestaña” que muestra una tabla de los trabajadores ya existentes (de esto se encarga *MuestraTrabajadores.py*). Mencione pestaña entre comillas pues en realidad las interfaces se cambian mediante botones y no como pestañas de un navegador de internet.

Para registrar un empleado, RegistroInterfaz.py necesita de un nombre, apellido, sueldo, puesto (este dato se establece mediante un menú desplegable), CURP y RFC. Estos datos se corresponden al constructor de la clase Trabajador en Trabajos.py. Adicionalmente, se tiene la opción de guardar una foto en formato PNG, JPG o JPEG, que se debe seleccionar mediante un explorador de archivos (file dialog), esta foto se guarda como: “<ID_del_trabajador>+<extensión_de_archivo>” en la carpeta *src/ficheros/imagenes*. En caso de no seleccionar ninguna imagen, la aplicación guardará una imagen por defecto.

Para mostrar a los trabajadores existentes en una tabla, MuestraTrabajadores.py hace uso del fichero *Trabajadores.json* que se encuentra en la carpeta *src/ficheros*. Esta tabla muestra el ID, nombre, apellido, sueldo puesto y fecha de registro de un trabajador, si es que hay trabajadores almacenados en el fichero, de lo contrario, se mostrará una tabla vacía.

Requisitos.

Este proyecto fue desarrollado en Python3, por lo que se requiere de la versión 3.10.6 o superior para evitar problemas. Además, se debe instalar la biblioteca Tkinter, que se puede obtener con el siguiente comando en linux (distribuciones Debian):

```
sudo apt-get install python3-tk
```

La biblioteca Pillow (para instalarla, es necesario contar con el manejador de paquetes pip), que se puede instalar desde la terminal en linux (distribuciones Debian) con el siguiente comando:

```
pip install Pillow
```

Y la biblioteca ImageTk:

```
sudo apt-get install python3-pil.imagetk
```

Debido a que el programa fue desarrollado en Linux y la forma de acceder a las carpetas es distinto a Windows (pues Windows utilizar \ para ello) este proyecto no se ejecutará correctamente en Windows. En un futuro agregaré las optimizaciones necesarias para que se pueda ejecutar en Windows.

Ejecución del programa.

Para poder ejecutar el programa, se debe ejecutar el siguiente comando, a la altura de src, en terminal:

```
python3 Interfaces/Interfaz.py
```

Detalles varios.

Para tener acceso a todos las clases desde las interfaces, fue necesario importar "sys" y "os", pues se encontraban en directorios distintos.

El menú de selección de imagen para un empleado empieza desde la raíz del sistema, por lo que primero debemos desplazarnos a home/user para encontrar los directorios a los que estamos familiarizados.

Las fotos que se almacenarán estarán reajustadas a un tamaño fijo, por lo que se sugiere que las fotos de los empleados sean más o menos cuadradas.

Futuras actualizaciones.

- Tengo planeado agregar un buscador de empleados y una barra deslizable en MuestraTrabajadores.py, para que se pueda encontrar con más facilidad a algún empleado en caso de tener a muchos registrados.
- También agregaré varias pruebas que tomarán información de JSON placeholder, así como también agregaré una herramienta que cree un PDF con los datos de un trabajador junto con su foto y el nombre de la empresa, esto con el uso de APIs.
- En un futuro pretendo agregar una base de datos con toda la información que actualmente estoy guardando en la carpeta de ficheros. Estos ficheros serán solo caché cuando se agregue la base de datos.