



**UNIVERSIDAD NACIONAL DE ASUNCIÓN**  
**FACULTAD POLITÉCNICA**



**EDITOR DE ORGANIGRAMAS**

**GRUPO: Rect Slot**

**Integrantes:**

- 5648043, Atilio Sebastián Paredes Pérez.
- 5827754, Santiago Damián Sotelo Rejala.
- 6038491, Elias Marcelo Riveros Vera.
- 6052031, Fernando David Fleitas Cáceres.

**Materia:** Algoritmos & Estructuras de Datos II.

**Carrera:** Ingeniería en Informática.

**Semestre:** 2do.

**AÑO: 2023.**

**San Lorenzo – Paraguay.**

## INVESTIGACIÓN

- **PYQT5 y QT DESIGNER:**

PyQt5 es una biblioteca de enlace para el lenguaje de programación Python que permite crear aplicaciones de escritorio con interfaces gráficas de usuario (GUI, por sus siglas en inglés) utilizando el framework Qt. Qt es un framework multiplataforma ampliamente utilizado para desarrollar aplicaciones de software en C++ y ofrece una amplia gama de funcionalidades para la creación de interfaces de usuario avanzadas. PyQt5 proporciona una capa de abstracción sobre Qt y permite a los desarrolladores utilizar todas las capacidades de Qt en sus aplicaciones de Python.

- **GRAPHVIZ:**

Graphviz es un software abierto de libre distribución para graficar, que presenta información estructural en forma de diagramas y puede aplicarse en diversas áreas como el análisis de redes, bioinformática, ingeniería de software, bases de datos, diseño de sitios web, aprendizaje por computadora y tiene interfaces gráficas para otros dominios. Su modo de utilización se basa en el diseño de pequeños programas que toman descripciones de los diagramas de un lenguaje de texto simple y los dibuja en diversos formatos tales como archivos de imágenes, SVG, PDF ó para desplegarse en exploradores.

- **SQLITE:**

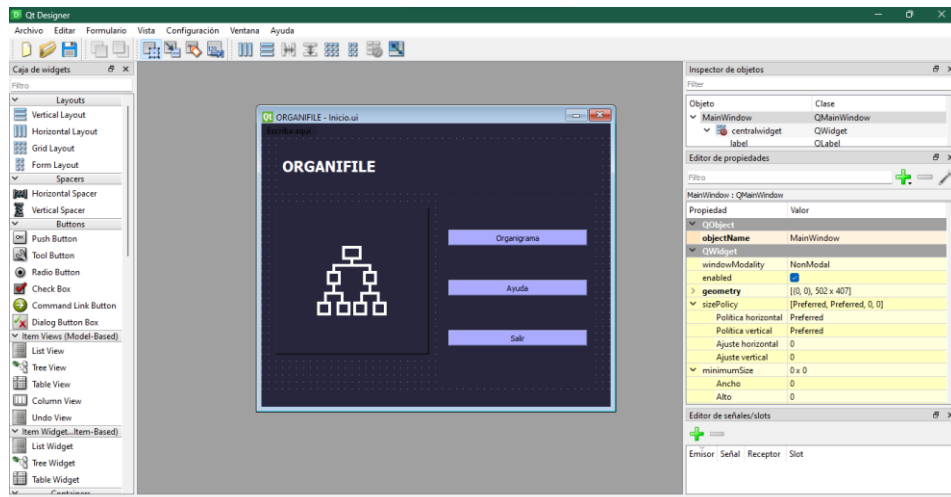
SQLite es una popular base de datos relacional, Se compone de una biblioteca en lenguaje C y se diferencia de otros sistemas de gestión de bases de datos al funcionar como un servidor propio e independiente, eliminando la necesidad de consultas y procesos separados. La biblioteca SQLite se genera y almacena directamente en el archivo de la base de datos, lo cual lo hace práctico y accesible para aplicaciones más simples, tanto en entornos desktop como mobile, sitios web ligeros y sistemas con un número reducido de usuarios. SQLite utiliza el lenguaje SQL (Structured Query Language) para realizar consultas y manejar la base de datos. También puede combinarse con scripts en Python para análisis de datos más complejos.

- **REPORT LAB:**

La librería "ReportLab" es una biblioteca de Python que permite generar documentos PDF de manera programática. Proporciona un conjunto de herramientas y funciones para crear y personalizar documentos PDF con contenido dinámico, como texto, tablas, imágenes y gráficos.

## ¿COMO UTILIZAMOS LAS LIBRERIAS?

El entorno gráfico de la aplicación se crea utilizando la herramienta de Qt designer, la cual ofrece una amplia gama de utilidades, funciones y opciones para organizar una interfaz de usuario de manera efectiva.



*Entorno de qtdesigner.*

Con PYQT5 nos encargamos de darle vida a cada botón que tenga la interfaz gráfica a través de clases y métodos.

```
48 class MainWindow(QMainWindow):
49     def __init__(self):
50         super(MainWindow, self).__init__()
51         self.setWindowTitle("Mi Aplicación")
52         self.graphics_view = MyGraphicsView()
53
54         # Cargar el archivo .ui
55         loadUi("main_window.ui", self)
56         database.connect()
57         rows = database.buscarData("Organigrama", f"id = {organigrama_activo}", ["nombre"])
58         if len(rows) != 0:
59             self.label_organigrama.setText(f"Organigrama: {rows[0][0]}")
60         else:
61             self.label_organigrama.setText("No ha seleccionado un organigrama")
62
63         # Conectar señales y slots si es necesario
64         self.crear_dependencia.clicked.connect(self.create_Dependencia)
65         self.crear_organigrama.clicked.connect(self.create_organigrama)
66         # self.abrir_organigrama.clicked.connect(self.open_organigrama)
67         self.agregar_persona.clicked.connect(self.abrir_form_persona)
68         self.action_PDF.triggered.connect(self.exportar_a_pdf)
69         self.action_IMAGEN.triggered.connect(self.exportar_a_imagen)
70         # self.actionInforme_por_dependencia.triggered.connect(self.Personal_Dependencia)
```

*Carga de la ventana principal y definición de botones.*

Sqlite3 se encarga de crear una estructura de base de datos para almacenar información sobre dependencias, personas (empleados) y el organigrama de una organización.

```

1  # Este archivo se ejecuta una vez para crear y configurar la base de datos
2  # * Creamos una tabla de dependencia
3  # * Creamos una tabla de personas en la que se almacenan los empleados
4
5  import sqlite3
6
7  con = sqlite3.connect("base1.db")
8  cur = con.cursor()
9
10 cur.execute(""" CREATE TABLE Dependencia (
11             id INTEGER NOT NULL PRIMARY KEY,
12             nombre TEXT,
13             manager_id INTEGER,
14             id_organigrama INTEGER,
15             FOREIGN KEY(id_organigrama) REFERENCES Organigrama(id)
16         )""")
17
18 cur.execute("""CREATE TABLE Persona (
19             id INTEGER NOT NULL PRIMARY KEY,
20             ci TEXT,
21             apellido TEXT,
22             nombre TEXT,
23             telefono TEXT,
24             direccion TEXT,
25             id_dependencia INTEGER,

```

*Creación de tablas en sqlite3.*

Mediante Graphviz realizamos la función de mostrar en pantalla los resultados del organigrama, sus conexiones, nombres y dependencias.

```

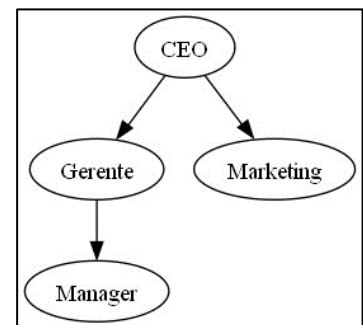
def generate_graph():
    dot = Digraph()
    return dot

def generate_node(graph, nombre_dependencia, label):
    graph.node(nombre_dependencia, label)

def connect_nodes(graph, node1_label, node2_label, edge_label):
    graph.edge(node1_label, node2_label, edge_label)

def generar_nodos(graph, id_dependencia):
    res_jefe = db.buscarData("Persona", f"id_dependencia = {id_dependencia}", ["id", "apellido", "nombre"])
    if len(res_jefe) != 0:
        id_jefe = res_jefe[0][0]
        apellido_jefe = res_jefe[0][1]
        nombre_jefe = res_jefe[0][2]
        if id_dependencia != 0:
            res_dep = db.buscarData("Dependencia", f"id = {id_dependencia}", ["nombre"])
            nombre_dep = res_dep[0][0]
        else:
            nombre_dep = "CEO"

```



*Ejemplo de ilustración creada con graphviz.*

*Código de conexión de nodos.*

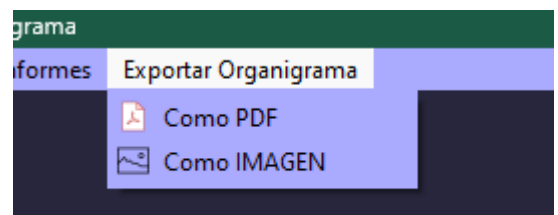
Report lab se utiliza para la generación del archivo PDF para exportar el organigrama y poder compartirlo.

```

def exportar_a_pdf(self):
    # Obtener el nombre del archivo y la ruta de la imagen
    nombre = self.lista_organigramas.currentText()
    ruta_imagen = f"INTERFAZ\\{nombre}.png"
    ruta_pdf = f"INTERFAZ\\{nombre}.pdf"
    # Abrir la imagen con QImage
    imagen = QImage(ruta_imagen)
    # Crear un objeto canvas de ReportLab para generar el PDF
    c = canvas.Canvas(ruta_pdf)
    # Definir las dimensiones del PDF basado en las dimensiones de la imagen
    ancho = imagen.width()
    alto = imagen.height()
    c.setPageSize((ancho, alto))

    # Dibujar la imagen en el PDF
    c.drawImage(ruta_imagen, 0, 0, ancho, alto)
    # Guardar el PDF y cerrar el objeto canvas
    c.save()
    # Abrir directamente el pdf
    QDesktopServices.openUrl(QUrl.fromLocalFile(f"INTERFAZ\\{nombre}.pdf"))

```



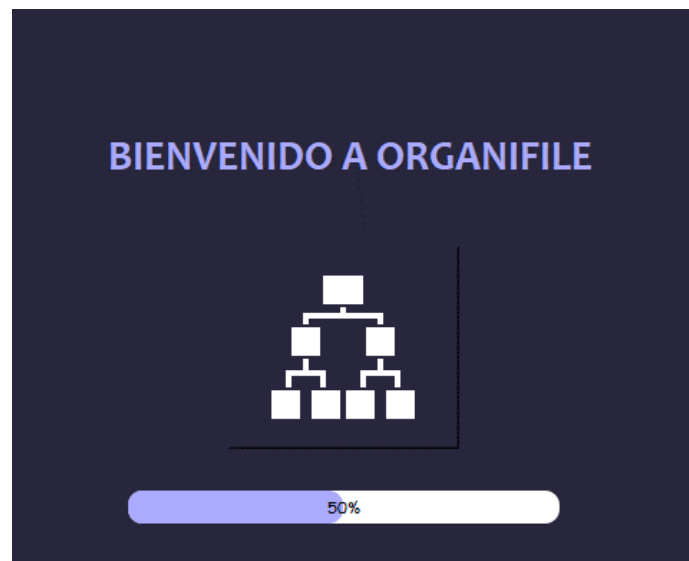
## DISEÑO

### Icono del programa



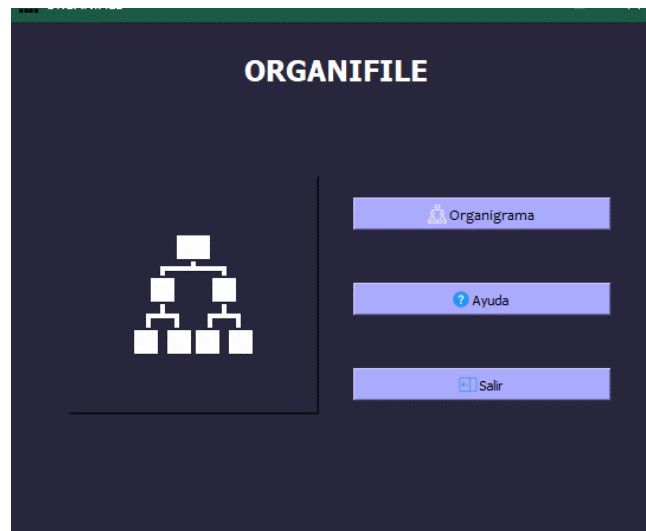
El icono del programa es sencillo representativo, y bastante minimalista, decidimos elegir un icono así para poder representar el significado del editor de organigramas, como organigrama y estructura organizativa.

### Pantalla de Carga



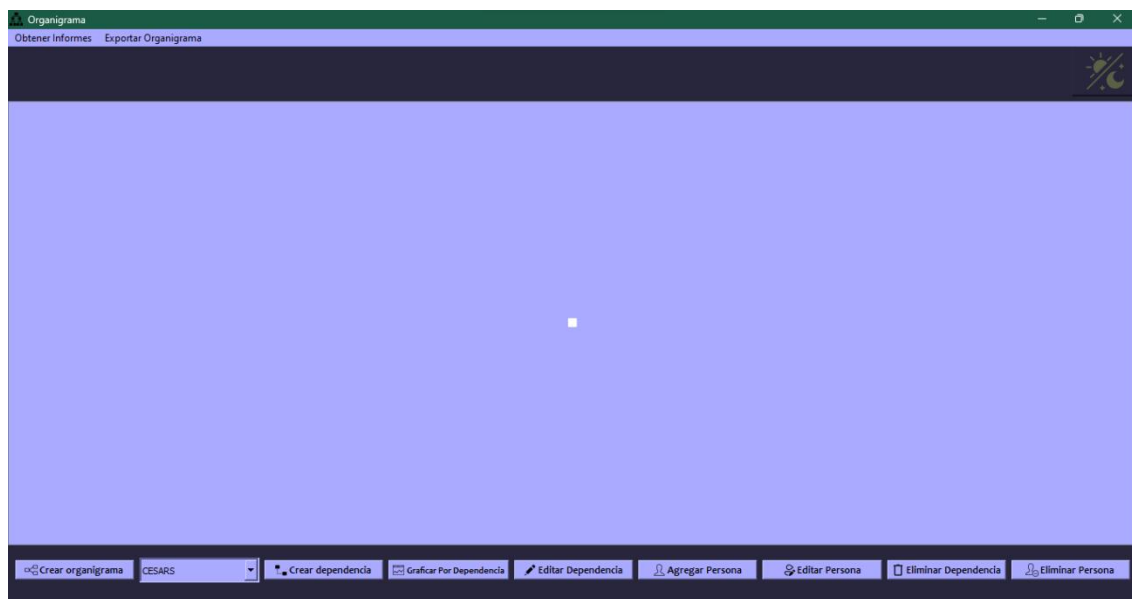
Esta sería la pantalla de carga del programa, creada para que sea más dinámica la presentación del proyecto.

## Pantalla de Inicio del Programa



Fue creada para que, al cargar toda la barra de progreso de la presentación, aparezca un menú, para elegir las opciones correspondientes.

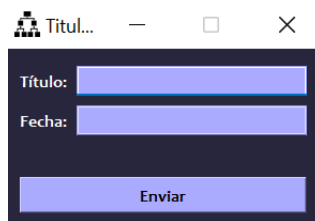
## Pantalla del Menú Principal



Esta es la interfaz del menú principal, inspirada en un editor manipulable y dinámico, está diseñada para poder visualizar el organigrama creado, poder manipular los nodos más fácilmente, agregar personas/dependencias, eliminar las personas/dependencias, obtener los informes y poder guardar en el formato deseado el organigrama creado, todo esto a solo unos clics de distancia.

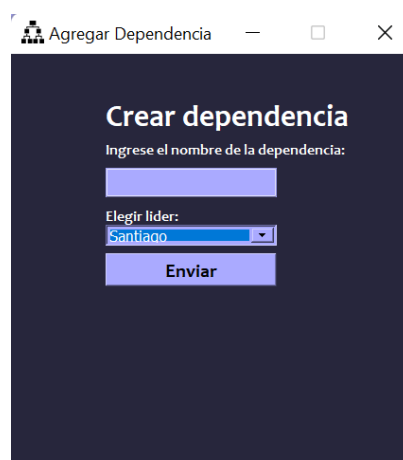
# FORMULARIOS

## Formulario para Crear el Organigrama



Simple y dinámico, recibe el título del organigrama y la fecha de este.

## Formulario para Crear Dependencia



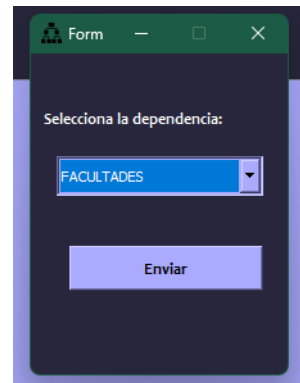
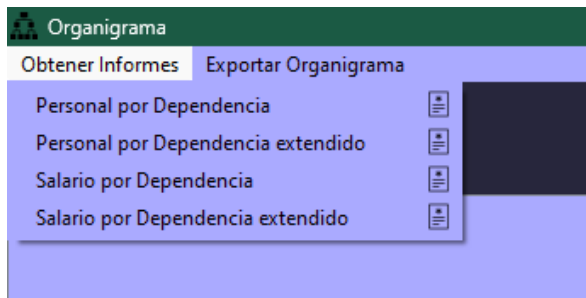
Este formulario se utiliza para crear la dependencia, diseñado para ser fácil de utilizar.

## Formulario para Agregar Persona

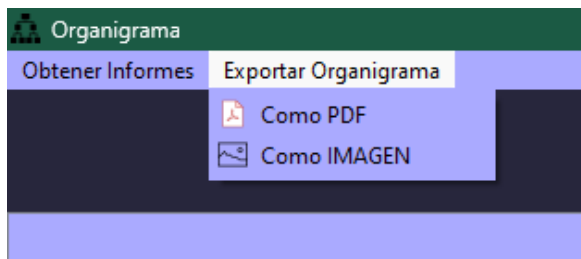


Con este formulario cargamos los datos solicitados por el programa y seleccionamos la dependencia mediante un selector desplegable llamado combo box.

## Menú de Gestión de Informes



Permite elegir el tipo de informe y sobre qué dependencia obtener los informes.



Exportar organigrama, permite graficar el organigrama completo como archivo PDF o formato imagen.



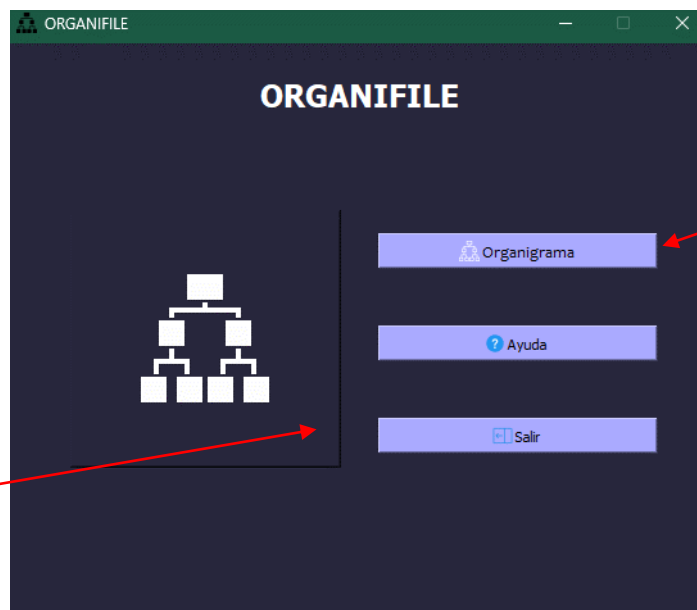
# MANUAL DE AYUDA

**Programa ORGANIFILE** permite la creación de un organigrama, editar sus componentes y enviarlos.

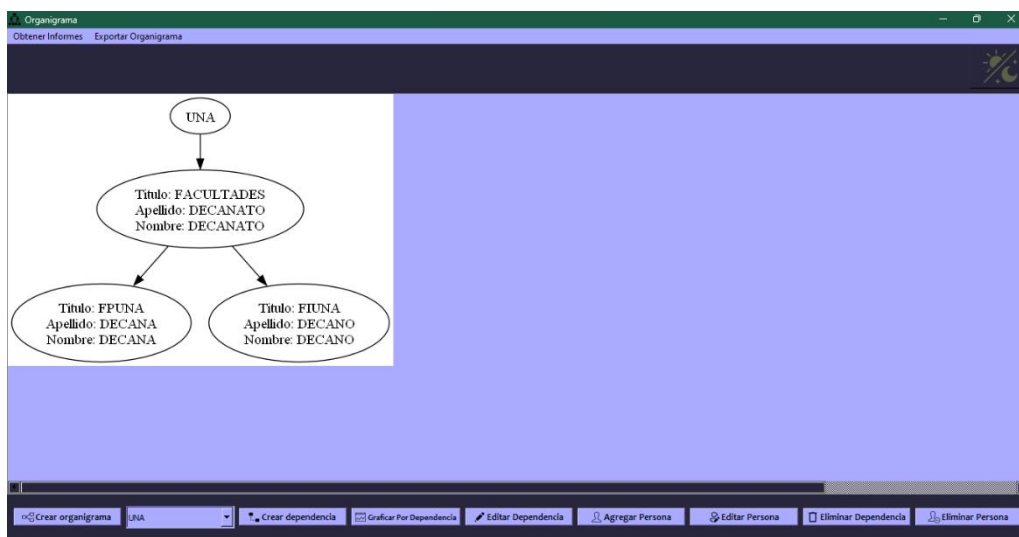
## Funcionalidades:

- Crear organigrama, elegir nombre y fecha de vigencia.
- Elegir el jefe de la organización.
- Crear dependencias.
- Modificar atributo de las dependencias.
- Eliminar dependencias.
- Agregar personas en las dependencias.
- Modificar datos de las personas.
- Eliminar personas de las dependencias.
- Obtener informes por dependencia.
- Crear gráfico del organigrama completo.

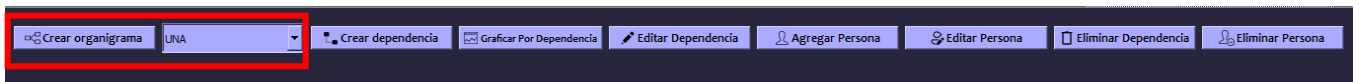
## INICIO



## ESPACIO DE TRABAJO



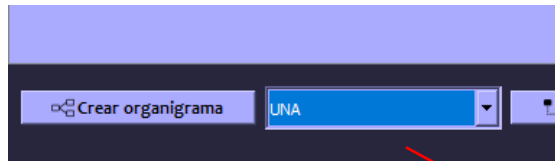
## MENÚ INFERIOR



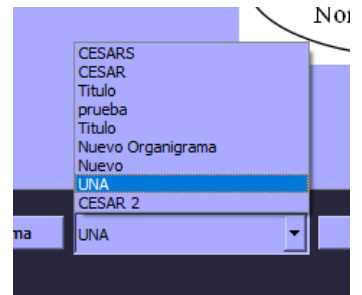
### CREACIÓN Y SELECCIÓN DE ORGANIGRAMA

**Crear organigrama:** despliega un formulario para ingresar el nombre y fecha del organigrama.

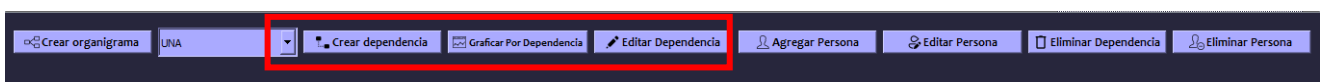
Luego de crear el organigrama sale un formulario para ingresar el jefe.



**Nombres:** permite elegir con qué organigrama trabajar.

Una ventana emergente titulada 'Titulo Del Organigrama'. Contiene dos campos de texto: 'Titulo:' y 'Fecha:'. Debajo de ellos hay un botón 'Enviar'. En la parte inferior de la ventana, se repite el botón 'Crear organigrama' y el menú desplegable 'UNA'.Una ventana emergente titulada 'PERSONA'. Contiene varios campos de texto para ingresar datos de una persona: 'Ingrese al jefe', 'Ci', 'Apellido', 'Nombre', 'Telefono', 'Direccion', 'Salario'. Hay un botón 'Enviar' al final.

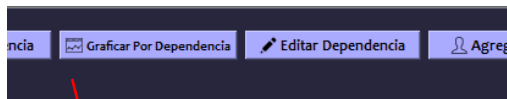
### CREACIÓN Y EDICIÓN DE DEPENDENCIA

Una ventana emergente titulada 'Agregar Dependencia' con el subtítulo 'Crear dependencia'. Contiene un campo de texto 'Ingrese el nombre de la dependencia:', un menú desplegable 'Elegir lider:' con la opción 'DECANO' seleccionada, y un botón 'Enviar'.

**Crear dependencia:** abre un formulario para ingresar el nombre y permite elegir el nombre del líder de esa dependencia.

Una ventana emergente titulada 'Agregar Dependencia' con el subtítulo 'Editar dependencia'. Contiene un campo de texto 'Ingrese el nuevo nombre:', un menú desplegable 'Elegir dependencia:' con la opción 'FACULTADES' seleccionada, y un botón 'Enviar'.

**Editar dependencia:** permite elegir el nombre de la dependencia y modificarlo.

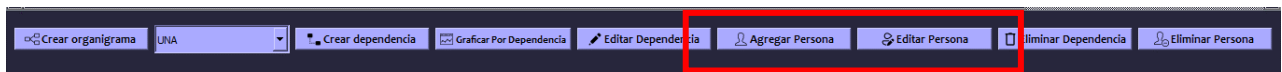


Dependencia a Graficar

Enviar

**Graficar por dependencia:** permite elegir la dependencia a graficar.

## CREACIÓN Y EDICIÓN DE PERSONA



**Agregar persona:** abre un formulario para rellenar con los datos de las personas y elegir su dependencia.

PERSONA

Añadir Persona

Ci:

Apellidos:

Nombres:

Telefono:

Dirección:

Dependencia:

Salario:

Enviar

**Editar persona:** muestra los datos de una persona y permite modificarlos.

Editar Persona

Dependencia:

Ci:

Apellidos:

Nombres:

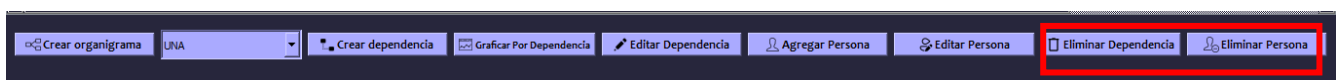
Telefono:

Dirección:

Salario:

Editar

## ELIMINACIÓN DE PERSONAS Y DEPENDENCIA



**Eliminar Dependencia:** Despliega un menú para elegir y eliminar la dependencia deseada.

Form

Dependencia a eliminar:

Eliminar

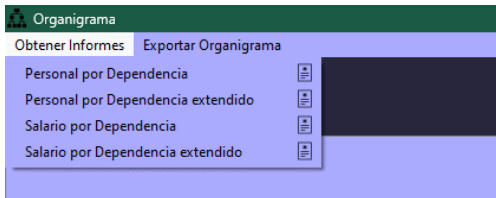
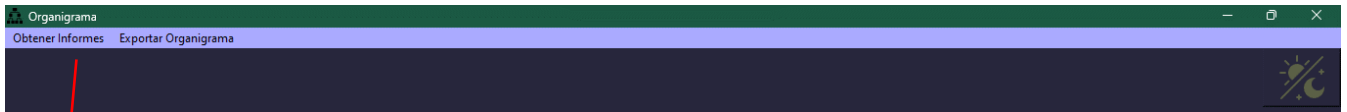
**Eliminar Persona:** Despliega un menú para elegir y eliminar la persona deseada.

Form

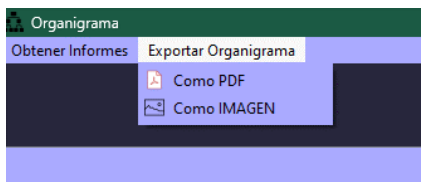
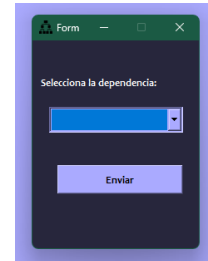
Persona a eliminar:

Eliminar

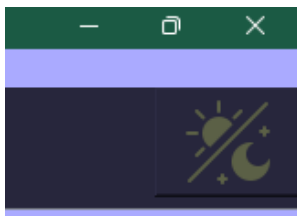
# MENÚ SUPERIOR



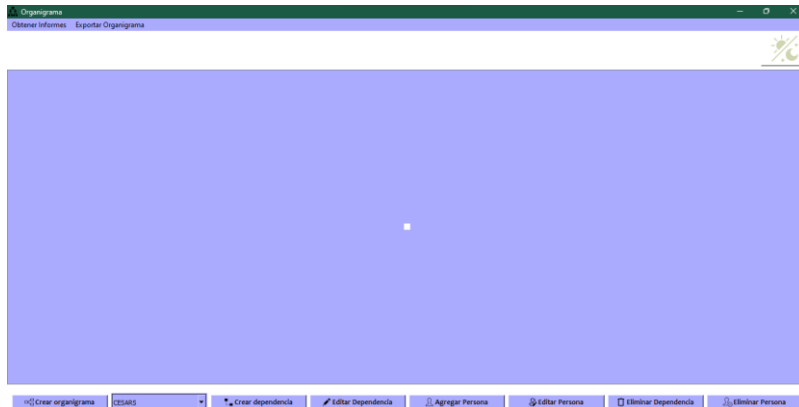
**Obtener informes:** despliega un menú para elegir el tipo de informe y luego de la dependencia deseada.



**Exportar Organigrama:** permite elegir en qué tipo de archivo exportar el gráfico del organigrama.



**SWITCH:** permite cambiar el espacio de trabajo en modo oscuro/claro.



## FUENTES BIBLIOGRAFICAS

- <https://pypi.org/project/PyQt5/>
- <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
- <https://doc.qt.io>
- <https://graphviz.org>
- <https://openai.com/blog/chatgpt>
- <https://iconos8.es>
- <https://www.youtube.com/@codefirstwithhala>
- <https://www.youtube.com/@SpinnTV>
- <https://www.youtube.com/playlist?list=PL2PZw96yQChwe5ZrLoRif8kjTYd3KKzKZ>
- <https://python.readthedocs.io/en/latest/library/sqlite3.html>