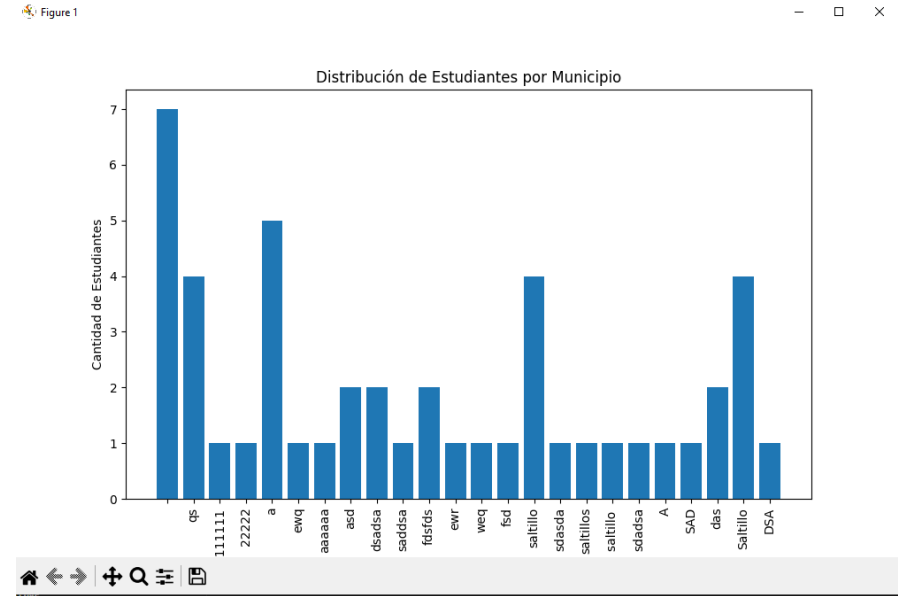
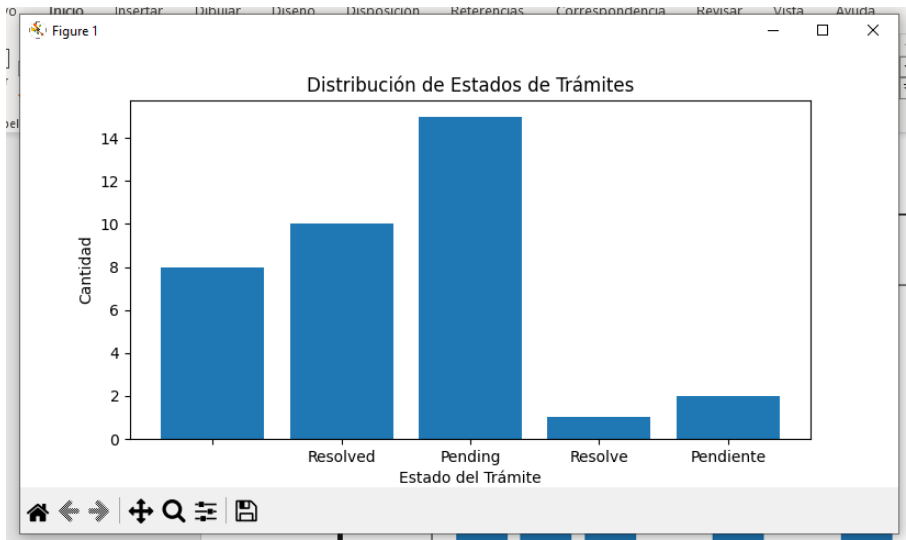
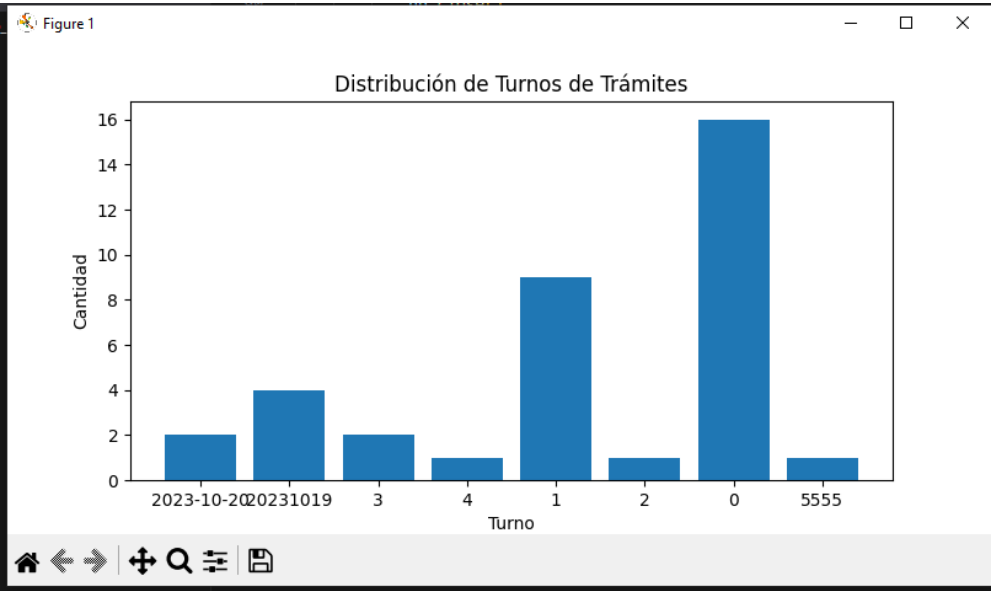


Dashboard



Panel de Control									
Estudiantes	Trámites	Admins							
	CURP	Nombre	Paterno	Materno	Municipio	Escuela	Trámite	Teléfono	Correo
	00000o	jos	ads	dsa	qs	dsa	qw	2132	a@dsa
	11111111111	111111	1111111	11111111	11111111	1111	111111	111111	1111111
	2222222	2222222	222	222222	22222	2222222	22222	2222	2222
	23133ddsad	asd	as	as	a	a	a	1234322343	aa
	323434eqwc	weq	weq	qwe	ewq	weq	wqe	12332	qwe
	a	a	a	a	a	a	a	123	a
	aaaaaaa	aaaaa	aaaa	aaaaaaa	aaaaaa	aaaaa	aaaaaaa	1232132	aaaaaaa
	ads	asd	dsa	sda	asd	dsa	dsa	213232323	23
	adsdsa	sad	ads	dsa	dsadsa	sda	dsa	232131	dsadsa

Panel de Control						
Estudiantes	Trámites	Admins				
	CURP	Trámite	Turno	Fecha	Hora	
	OAMJ9207HNLNRS0	solar	3	2023-10-20	1:20:27	
	oooooooooooooooo	ads	4	2023-10-20	1:21:43	
	yyyyyyyyyyyyyy	dsa	1	2023-10-20	1:38:34	
	323434eqwq	wqe	2	2023-10-20	1:39:19	
	dsadads	asdad	3	2023-10-20	1:39:34	
	oimj920716hnlmrs12	dsadsa	0	2023-10-20	3:47:40	
	kkkkkkkkkk	dsadsa	0	2023-10-20	3:48:36	
	dsffdsf	fdsfds	1	2023-10-20	3:51:46	
	1111111111111	111111	1	2023-10-20	3:52:39	
	2222222	22222	1	2023-10-20	3:53:03	

Panel de Control						
Estudiantes	Trámites	Admins				
	ID	Usuario	Nombre	Paterno	Materno	Teléfono
1						
2		jose	123456	JOSE	ADRIAN	ONTIVEROS
3		a	1	asasd	ads	ads
4		qw	wq	qw	qw	qw
5		jose	asd	jose	asd	asd
6		jose	as	asa	as	as
7		wqe	eqw	qew	qwe	eqw
8		a	a	a	a	a
9		i	i	i	i	i
10		j	j	j	j	j

Codigo con tkinter y matplotlib

```
import tkinter as tk
from tkinter import ttk, messagebox
import mysql.connector
import matplotlib.pyplot as plt

# Configuración de la base de datos
DB_CONFIG = {
    'host': 'localhost',
```

```

        'user': 'root',
        'password': '123456',
        'database': 'e'
    }

class DashboardApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Panel de Control")

        # Crear pestañas
        self.tabControl = ttk.Notebook(root)
        self.tab_estudiantes = ttk.Frame(self.tabControl)
        self.tab_tramites = ttk.Frame(self.tabControl)
        self.tab_admins = ttk.Frame(self.tabControl)
        self.tabControl.add(self.tab_estudiantes, text="Estudiantes")
        self.tabControl.add(self.tab_tramites, text="Trámites")
        self.tabControl.add(self.tab_admins, text="Admins")
        self.tabControl.pack(expand=1, fill="both")

        # Estudiantes Tab
        self.estudiantes_tree = ttk.Treeview(self.tab_estudiantes,
columns=("curp", "nombre", "paterno", "materno", "municipio", "escuela",
"tramite", "telefono", "correo"))
        self.estudiantes_tree.heading("#1", text="CURP")
        self.estudiantes_tree.heading("#2", text="Nombre")
        self.estudiantes_tree.heading("#3", text="Paterno")
        self.estudiantes_tree.heading("#4", text="Materno")
        self.estudiantes_tree.heading("#5", text="Municipio")
        self.estudiantes_tree.heading("#6", text="Escuela")
        self.estudiantes_tree.heading("#7", text="Trámite")
        self.estudiantes_tree.heading("#8", text="Teléfono")
        self.estudiantes_tree.heading("#9", text="Correo")
        self.estudiantes_tree.pack()

        # Trámites Tab
        self.tramites_tree = ttk.Treeview(self.tab_tramites,
columns=("curp", "tramite", "turno", "fecha", "hora"))
        self.tramites_tree.heading("#1", text="CURP")
        self.tramites_tree.heading("#2", text="Trámite")
        self.tramites_tree.heading("#3", text="Turno")
        self.tramites_tree.heading("#4", text="Fecha")
        self.tramites_tree.heading("#5", text="Hora")
        self.tramites_tree.pack()

```

```

        # Admins Tab
        self.admins_tree = ttk.Treeview(self.tab_admins, columns=("id",
"usuario", "nombre", "paterno", "materno", "telefono"))
        self.admins_tree.heading("#1", text="ID")
        self.admins_tree.heading("#2", text="Usuario")
        self.admins_tree.heading("#3", text="Nombre")
        self.admins_tree.heading("#4", text="Paterno")
        self.admins_tree.heading("#5", text="Materno")
        self.admins_tree.heading("#6", text="Teléfono")
        self.admins_tree.pack()

        # Cargar datos desde la base de datos
        self.load_data()

def load_data(self):
    try:
        db = mysql.connector.connect(**DB_CONFIG)
        cursor = db.cursor()

        # Cargar datos de estudiantes
        cursor.execute("SELECT * FROM estudiante")
        estudiantes_data = cursor.fetchall()
        for estudiante in estudiantes_data:
            self.estudiantes_tree.insert("", "end", values=estudiante)

        # Cargar datos de trámites
        cursor.execute("SELECT * FROM tramite")
        tramites_data = cursor.fetchall()
        for tramite in tramites_data:
            self.tramites_tree.insert("", "end", values=tramite)

        # Cargar datos de admins
        cursor.execute("SELECT * FROM admin")
        admins_data = cursor.fetchall()
        for admin in admins_data:
            self.admins_tree.insert("", "end", values=admin)

        # Generar y mostrar un gráfico de barras de turnos y estados de
trámites
        self.plot_tramite_data(tramites_data)

        db.close()
    except mysql.connector.Error as err:
        print("Error:", err)

```

```

def plot_tramite_data(self, tramites_data):
    turnos = {}
    estados = {}

    for tramite in tramites_data:
        turno = tramite[2]
        estado = tramite[5]

        if turno in turnos:
            turnos[turno] += 1
        else:
            turnos[turno] = 1

        if estado in estados:
            estados[estado] += 1
        else:
            estados[estado] = 1

    # Crear un gráfico de barras para los turnos
    plt.figure(figsize=(8, 4))
    plt.bar(turnos.keys(), turnos.values())
    plt.xlabel("Turno")
    plt.ylabel("Cantidad")
    plt.title("Distribución de Turnos de Trámites")
    plt.show()

    # Crear un gráfico de barras para los estados
    plt.figure(figsize=(8, 4))
    plt.bar(estados.keys(), estados.values())
    plt.xlabel("Estado del Trámite")
    plt.ylabel("Cantidad")
    plt.title("Distribución de Estados de Trámites")
    plt.show()

if __name__ == "__main__":
    root = tk.Tk()
    app = DashboardApp(root)
    root.mainloop()

```

La ultima grafica

```

import tkinter as tk
import mysql.connector
from tkinter import ttk
import matplotlib.pyplot as plt

```

```

# Configuración de la base de datos
DB_CONFIG = {
    'host': 'localhost',
    'user': 'root',
    'password': '123456',
    'database': 'e'
}

class DashboardApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Panel de Control")

        # Crear pestañas
        self.tabControl = ttk.Notebook(root)
        self.tab_estudiantes = ttk.Frame(self.tabControl)
        self.tab_tramites = ttk.Frame(self.tabControl)
        self.tab_admins = ttk.Frame(self.tabControl)
        self.tabControl.add(self.tab_estudiantes, text="Estudiantes")
        self.tabControl.add(self.tab_tramites, text="Trámites")
        self.tabControl.add(self.tab_admins, text="Admins")
        self.tabControl.pack(expand=1, fill="both")

        # Estudiantes Tab
        self.estudiantes_tree = ttk.Treeview(self.tab_estudiantes,
columns=("curp", "nombre", "paterno", "materno", "municipio", "escuela",
"tramite", "telefono", "correo"))
        self.estudiantes_tree.heading("#1", text="CURP")
        self.estudiantes_tree.heading("#2", text="Nombre")
        self.estudiantes_tree.heading("#3", text="Paterno")
        self.estudiantes_tree.heading("#4", text="Materno")
        self.estudiantes_tree.heading("#5", text="Municipio")
        self.estudiantes_tree.heading("#6", text="Escuela")
        self.estudiantes_tree.heading("#7", text="Trámite")
        self.estudiantes_tree.heading("#8", text="Teléfono")
        self.estudiantes_tree.heading("#9", text="Correo")
        self.estudiantes_tree.pack()

        # Trámites Tab
        self.tramites_tree = ttk.Treeview(self.tab_tramites,
columns=("curp", "tramite", "turno", "fecha", "hora"))
        self.tramites_tree.heading("#1", text="CURP")
        self.tramites_tree.heading("#2", text="Trámite")
        self.tramites_tree.heading("#3", text="Turno")
        self.tramites_tree.heading("#4", text="Fecha")

```

```

        self.tramites_tree.heading("#5", text="Hora")
        self.tramites_tree.pack()

        # Admins Tab
        self.admins_tree = ttk.Treeview(self.tab_admins, columns=("id",
"usuario", "nombre", "paterno", "materno", "telefono"))
        self.admins_tree.heading("#1", text="ID")
        self.admins_tree.heading("#2", text="Usuario")
        self.admins_tree.heading("#3", text="Nombre")
        self.admins_tree.heading("#4", text="Paterno")
        self.admins_tree.heading("#5", text="Materno")
        self.admins_tree.heading("#6", text="Teléfono")
        self.admins_tree.pack()

        # Botón para mostrar gráfica de municipios
        self.show_municipios_button = tk.Button(self.tab_estudiantes,
text="Mostrar Gráfica de Municipios", command=self.show_municipios_chart)
        self.show_municipios_button.pack()

        # Cargar datos desde la base de datos
        self.load_data()

def load_data(self):
    try:
        db = mysql.connector.connect(**DB_CONFIG)
        cursor = db.cursor()

        # Cargar datos de estudiantes
        cursor.execute("SELECT * FROM estudiante")
        estudiantes_data = cursor.fetchall()
        for estudiante in estudiantes_data:
            self.estudiantes_tree.insert("", "end", values=estudiante)

        # Cargar datos de trámites
        cursor.execute("SELECT * FROM tramite")
        tramites_data = cursor.fetchall()
        for tramite in tramites_data:
            self.tramites_tree.insert("", "end", values=tramite)

        # Cargar datos de admins
        cursor.execute("SELECT * FROM admin")
        admins_data = cursor.fetchall()
        for admin in admins_data:
            self.admins_tree.insert("", "end", values=admin)

```

```

        db.close()
    except mysql.connector.Error as err:
        print("Error:", err)

def show_municipios_chart(self):
    try:
        db = mysql.connector.connect(**DB_CONFIG)
        cursor = db.cursor()

        cursor.execute("SELECT municipio FROM estudiante")
        municipios_data = cursor.fetchall()

        municipios = {}

        for municipio in municipios_data:
            municipio = municipio[0]
            if municipio in municipios:
                municipios[municipio] += 1
            else:
                municipios[municipio] = 1

        # Crear un gráfico de barras para los municipios
        plt.figure(figsize=(10, 6))
        plt.bar(municipios.keys(), municipios.values())
        plt.xlabel("Municipio")
        plt.ylabel("Cantidad de Estudiantes")
        plt.title("Distribución de Estudiantes por Municipio")
        plt.xticks(rotation=90)
        plt.show()

        db.close()
    except mysql.connector.Error as err:
        print("Error:", err)

if __name__ == "__main__":
    root = tk.Tk()
    app = DashboardApp(root)
    root.mainloop()

```