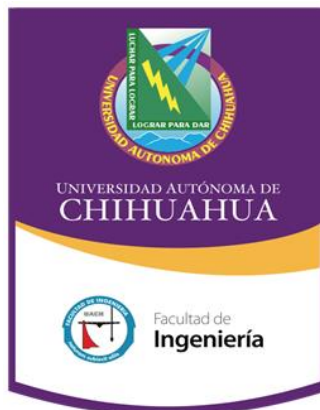


**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
FACULTAD DE INGENIERÍA
INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN**



Bases de Datos Avanzadas - 7CC2

**Proyecto
Evidencia**

Docente: José Saúl De Lira Miramontes

Alumnos:

José Carlos Chaparro Morales	329613
Juan Luis Del Valle Sotelo	338912
Omar Alonso Escápita Chacón	338886

Chihuahua, Chihuahua. A 5 de octubre de 2022

Código:

```
"""Proyecto_Parcial_1

Este programa contiene 9 fuciones, las cuales se muestran en una interface.
Estas funciones se utilizan para llamar procedimientos y funciones
almacenadas en una base de datos Scott
esta instalada de manera local, con nombre de srevicio xepdb1.
Estas 9 funciones incluyen añadir, acutualizar, borrar, buscar y
eliminar un elemento de cualquiera de sus dos tablas: emp y depto

"""

from cProfile import label
from functools import partial
import tkinter as tk
from tkinter import messagebox
from PIL import ImageTk, Image

import cx_Oracle
cx_Oracle.init_oracle_client(lib_dir=r"C:\oracle\instantclient_21_6")

# Variables globales
global counter_windows
counter_windows = 0

def callback(ventana):
    """Se asegura de mantener el control del numero de ventanas abiertas

    Parameters
    -----
    ventana : Tk()
        Ventana a cerrar

    Return
    -----
    None
    """
    global counter_windows
    counter_windows = 0
    ventana.destroy()

# Colores
color_1 = "#5690e6"
```

```

color_2 = "gray95"
color_3 = "black"
color_4 = "white"
font_1 = "times new roman"
font_2 = "helvetica"

# Ventana principal
window = tk.Tk()
window.title("Proyecto 1")
window.geometry("780x600")
window.minsize(780,600)

# Marco izquierdo
frame_1 = tk.Frame(window, bg = color_1)
frame_1.place(x=0, y=0, width=100, height = 100)
frame_1.pack(fill =tk.BOTH,side=tk.LEFT,expand=True, padx=5, pady=5)
img = ImageTk.PhotoImage(Image.open("uach.png"))

# Marco derecho
frame_2 = tk.Frame(window, bg = color_2)
frame_2.place(x=0, y=0, width=200, height = 100)
frame_2.pack(fill =tk.BOTH,side=tk.LEFT,expand=False, padx=5, pady=5)

# Datos del equipo
titulo_consulta = tk.Label(frame_1, text="Integrantes", bg = color_1, fg =
color_4, font='Helvetica 18 bold')
titulo_consulta.pack()
integrantes = tk.Label(frame_1, text="José Carlos Chaparro Morales -
329613\nJuan Luis Del Valle Sotelo - 338912\nOmar Alonso Escápita Chacón -
338912", bg = color_1, fg = color_4, font='Helvetica 16')
integrantes.pack()
resultado_sulta = tk.Label(frame_1, text="Docente", bg = color_1, fg =
color_4, font='Helvetica 18 bold')
resultado_sulta.pack()
profe = tk.Label(frame_1, text="M. A. José Saúl De Lira Miramontes", bg =
color_1, fg = color_4, font='Helvetica 16')
profe.pack()
label = tk.Label(frame_1, image = img)
label.pack(pady=40)

def add_depto(dept_no ,dept_name ,dept_loc ):
    """Add department to database with parameters.

    Parameters

```

```

-----
dept_no: int
    Llave única de la tabla de departamentos
dept_name string
    Nombre del departamento a agregar
dept_loc: string
    Ubicación del departamento a agregar
Returns
-----
None
"""
try:
    with cx_Oracle.connect(user="SCOTT", password="scott",
dsn="//localhost:1521/xepdb1") as connection:
        with connection.cursor() as cursor:
            cursor.callproc('add_depto',[dept_no ,dept_name ,dept_loc])
            messagebox.showinfo(message='Departamento añadido con éxito')

except cx_Oracle.Error as error:
    messagebox.showerror(message=error,title="Error")
    print(error)

def update_emp(pemp_no ,pename ,pjob ,pmgr ,phiredate ,psal ,pcomm ,pdeptno
):
    """Actualizar los valores del empleado dado por el numero de empleado.
Parameters
-----
pemp_no: int
    hace referencia al numero de empleado del empleado que se busca
actualizar
pename: string
    Nuevo nombre del empleado a actualizar
pjob: string
    Nuevo puesto del empleado
pmgr: int
    Numero de empleado del jefe del empleado a actualziar
phiredate: string
    Fecha de contratación del empleado a actualizar
psal: float
    Nuevo sueldo del empleado a actualizar
pcomm: float
    Nueva comisión del empleado a actualizar
pdepto: int
    Nuevo numero de departamento del empleado a actualizar

```

```

Returns
-----
None
"""
try:
    with cx_Oracle.connect(user="SCOTT", password="scott",
dsn="//localhost:1521/xepdb1") as connection:
        with connection.cursor() as cursor:
            cursor.callproc('update_emp',[pemp_no ,pename ,pjob ,pmgr
,phiredate ,psal ,pcomm ,pdeptno])
            messagebox.showinfo(message='Empleado actualizado con exito')

except cx_Oracle.Error as error:
    messagebox.showerror(message=error,title="Error")
    print(error)

def update_depto(dept_no ,dept_name ,dept_loc ):
    """Actualizar los valores del departamento dado por el numero de
departamento.

Parameters
-----
dept_no: int
    Numero de departamento del departamento a cambiar datos
dept_name: string
    Nuevo nombre del departamento
dept_loc: string
    Nueva localización del departamento

Returns
-----
None
"""
try:
    with cx_Oracle.connect(user="SCOTT", password="scott",
dsn="//localhost:1521/xepdb1") as connection:
        with connection.cursor() as cursor:
            cursor.callproc('update_depto',[dept_no ,dept_name
,dept_loc])
            messagebox.showinfo(message='Departamento actualizado con exito')

except cx_Oracle.Error as error:
    messagebox.showerror(message=error,title="Error")

```

```

        print(error)

def delete_emp(pemp_no):
    """Elimina un empleado dado por su numero de empleado.

    Parameters
    -----

    pemp_no: int
        Numero de empleado del empleado a borrar

    Returns
    -----
    None
    """

    try:
        with cx_Oracle.connect(user="SCOTT", password="scott",
dsn="//localhost:1521/xepdb1") as connection:
            with connection.cursor() as cursor:
                cursor.callproc('delete_emp',[pemp_no ])
                messagebox.showinfo(message='Empleado eliminado con exito')

    except cx_Oracle.Error as error:
        messagebox.showerror(message=error,title="Error")
        print(error)

def delete_depto(dept_no):
    """Elimina un departamento dado por su numero de departamento.

    Parameters
    -----

    dept_no: int
        Llave única de la tabla de departamentos

    Returns
    -----
    None
    """

    try:
        with cx_Oracle.connect(user="SCOTT", password="scott",
dsn="//localhost:1521/xepdb1") as connection:
            with connection.cursor() as cursor:
                cursor.callproc('delete_depto',[dept_no ])

```

```

        messagebox.showinfo(message='Departamento eliminado con exito')

except cx_Oracle.Error as error:
    messagebox.showerror(message=error,title="Error")
    print(error)

def add_emp(pemp_no ,pename ,pjob ,pmgr ,phiredate ,psal ,pcomm ,pdeptno ):
    """Añade un empleado cuyos valores son los parametros definidos a
    continuación

    Parameters
    -----

    timestamp: int
        Hace referencia al numero de empleado del empleado que se busca
    añadir
    pename: string
        Nombre del empleado nuevo
    pjob: string
        Puesto del empleado nuevo
    pmgr: int
        Numero de empleado del jefe del empleado nuevo
    phiredate: string
        Fecha de contratación del empleado
    psal: float
        Sueldo del nuevo empleado
    pcomm: float
        Comisión del nuevo empleado
    pdepto: int
        Numero de departamento del empleado nuevo

    Return
    -----
    None
    """

    try:
        with cx_Oracle.connect(user="SCOTT", password="scott",
            dsn="//localhost:1521/xepdb1") as connection:
            with connection.cursor() as cursor:
                cursor.callproc('add_emp',[pemp_no ,pename ,pjob ,pmgr
                    ,phiredate ,psal ,pcomm ,pdeptno])
                messagebox.showinfo(message='Empleado añadido con exito')

```

```

except cx_Oracle.Error as error:
    messagebox.showerror(message=error,title="Error")
    print(error)

def noemp_depto(dept_no):
    """Encuentra el numero de empleados del departamento dado el número de
    departamento.

    Parameters
    -----

    dept_no: int
        Numero de departamento del departamento que se busca conocer el
        numero de empleados

    Return
    -----
    None
    """

    try:
        with cx_Oracle.connect(user="SCOTT", password="scott",
            dsn="//localhost:1521/xepdb1") as connection:
            with connection.cursor() as cursor:
                noEmps = cursor.callfunc('noemp_depto',int,[dept_no ])
                messagebox.showinfo(message=f'El departamento con id :{dept_no}
                tiene:{noEmps} empleados')

    except cx_Oracle.Error as error:
        messagebox.showerror(message=error,title="Error")
        print(error)

def read_emp(emp_no):
    """Funcion que obtiene y muestra los datos de un empleado, dado su
    número de empleado.

    Parameters
    -----

    emp_no: int
        Numero de empleado del empleado a buscar

    Return
    -----
    None

```



```

"""

    try:
        with cx_Oracle.connect(user="SCOTT", password="scott",
dsn="//localhost:1521/xepdb1") as connection:
            with connection.cursor() as cursor:
                registro = cursor.callfunc('read_emp',str,[emp_no ])
                print(registro)
            messagebox.showinfo(message=f'\n{registro}',title='Resultado')

    except cx_Oracle.Error as error:
        messagebox.showerror(message=error,title="Error")
        print(error)

"""

:param depto_no:
"""

def read_dept(emp_no):
    """Funcion que obtiene y muestra los datos de un departamento, dado su
número de departamento.

    Parameters
    -----
    dept_no: int
        Numero de departamento del departamento a buscar
    Returns
    -----
    None
    """
    try:
        with cx_Oracle.connect(user="SCOTT", password="scott",
dsn="//localhost:1521/xepdb1") as connection:
            with connection.cursor() as cursor:
                registro = cursor.callfunc('read_dept',str,[emp_no ])
                x = registro.split(',')
            messagebox.showinfo(message=f'El departamento con id :{emp_no}
es:\nnno. departamento: {x[0]}\nnombre: {x[1]}\nlocalizacion:
{x[2]}',title='Resultado')

    except cx_Oracle.Error as error:
        messagebox.showerror(message=error,title="Error")
        print(error)

def handle_focus_in(event):
    """Funcion que borra el texto de ejemplo al hacer clic en un entry

```

```

Parameters
-----
event: tkinter.Event
    Es el evento que se usa para obtener el entry al cual se le va a
    borrar el texto de ejemplo
Returns
-----
None
"""
event.widget.delete(0, tk.END)
event.widget.config(fg='black')

def add_depto_window():
    """Funcion que crea la ventana que se encarga de obtener los datos de un
    departamento para añadirlo, llamando a la funcion add_depto.

    Parameters
    -----
    None

    Return
    -----
    None
    """

    check_window_open()
    global counter_windows

    if counter_windows == 0:
        counter_windows += 1
        new_dept_window = tk.Toplevel(window)
        new_dept_window.protocol("WM_DELETE_WINDOW",
partial(callback,new_dept_window))
        new_dept_window.minsize(300,500)
        new_dept_window.title("Añadir departamento")

        deptno_entry = tk.Entry(new_dept_window, font=('Helvetica 12'),
fg="grey" )
        deptno_label = tk.Label(new_dept_window, text="Numero del
departamento:")
        deptno_label.pack()

```

```

deptno_entry.insert(tk.END, "Ejemplo: 1234")
deptno_entry.pack(pady=5)

deptno_entry.bind("<FocusIn>", handle_focus_in)

deptno_nombre_label = tk.Label(new_dept_window, text="Nombre del
departameto nuevo:")
deptno_nombre_label.pack()
deptno_nombre_entry = tk.Entry(new_dept_window, font=('Helvetica
12'), fg="grey")
deptno_nombre_entry.insert(tk.END, "Ejemplo: Jose")
deptno_nombre_entry.pack(pady=5)

deptno_nombre_entry.bind("<FocusIn>", handle_focus_in)

deptno_loc_label = tk.Label(new_dept_window, text="ubicación del
departamento nuevo:")
deptno_loc_label.pack()
deptno_loc_entry = tk.Entry(new_dept_window, font=('Helvetica 12'),
fg="grey")
deptno_loc_entry.insert(tk.END, "Ejemplo: CLERK")
deptno_loc_entry.pack(pady=5)

deptno_loc_entry.bind("<FocusIn>", handle_focus_in)

def ok():
    depno = int(deptno_entry.get())
    depname = str(deptno_nombre_entry.get())
    deploc = str(deptno_loc_entry.get())

    add_depto(depno, depname ,deploc )

    boton = tk.Button(new_dept_window, text="Añadir departamento",
command=ok )

    boton.pack()

def delete_dept_window():

```

```
"""Funcion que crea la ventana que se encarga de obtener los datos de un
departamento para eliminarlo, llamando a la funcion delete_depto.
```

```
Parameters
```

```
-----
```

```
None
```

```
Return
```

```
-----
```

```
None
```

```
"""
```

```
check_window_open()
```

```
global counter_windows
```

```
if counter_windows == 0:
```

```
    counter_windows += 1
```

```
    delete_dept_window = tk.Toplevel(window)
```

```
    delete_dept_window.protocol("WM_DELETE_WINDOW",
partial(callback,delete_dept_window))
```

```
    delete_dept_window.minsize(300,500)
```

```
    delete_dept_window.title("Eliminar departamento")
```

```
        deptno_entry = tk.Entry(delete_dept_window, font=('Helvetica 12'),
fg="grey" )
```

```
        deptno_label = tk.Label(delete_dept_window, text="Numero del
departamento a eliminar:")
```

```
        deptno_label.pack()
```

```
        deptno_entry.insert(tk.END, "Ejemplo: 1234")
```

```
        deptno_entry.pack(pady=5)
```

```
        deptno_entry.bind("<FocusIn>", handle_focus_in)
```

```
def ok():
```

```
    deptno = int(deptno_entry.get())
```

```
    delete_depto(deptno)
```

```

        boton = tk.Button(delete_dept_window, text="Eliminar departamento",
command=ok )

        boton.pack()

def update_dept_window():
    """Funcion que crea la ventana que se encarga de obtener los datos de un
departamento para actualizar sus datos, llamando a la funcion update_depto.

Parameters
-----
None

Return
-----
None

"""

    check_window_open()
    global counter_windows

    if counter_windows == 0:
        counter_windows += 1
        new_dept_window = tk.Toplevel(window)
        new_dept_window.protocol("WM_DELETE_WINDOW",
partial(callback,new_dept_window))
        new_dept_window.minsize(300,500)
        new_dept_window.title("Modificar departamento")

        deptno_entry = tk.Entry(new_dept_window, font=('Helvetica 12'),
fg="grey" )
        deptno_label = tk.Label(new_dept_window, text="Numero del
departamento:")
        deptno_label.pack()

        deptno_entry.insert(tk.END, "Ejemplo: 1234")
        deptno_entry.pack(pady=5)

        deptno_entry.bind("<FocusIn>", handle_focus_in)

```

```

        deptno_nombre_label = tk.Label(new_dept_window, text="Nuevo nombre
del departamento a modificar:")
        deptno_nombre_label.pack()
        deptno_nombre_entry = tk.Entry(new_dept_window, font=('Helvetica
12'), fg="grey")
        deptno_nombre_entry.insert(tk.END, "Ejemplo: Jose")
        deptno_nombre_entry.pack(pady=5)

        deptno_nombre_entry.bind("<FocusIn>", handle_focus_in)

        deptno_loc_label = tk.Label(new_dept_window, text="Nueva
localización del departamento a modificar:")
        deptno_loc_label.pack()
        deptno_loc_entry = tk.Entry(new_dept_window, font=('Helvetica 12'),
fg="grey")
        deptno_loc_entry.insert(tk.END, "Ejemplo: CLERK")
        deptno_loc_entry.pack(pady=5)

        deptno_loc_entry.bind("<FocusIn>", handle_focus_in)

def ok():
    depno = int(deptno_entry.get())
    depname = str(deptno_nombre_entry.get())
    deploc = str(deptno_loc_entry.get())

    update_depto(depno, depname ,deploc )

    boton = tk.Button(new_dept_window, text="Actualizar departamento",
command=ok )

    boton.pack()

def add_emp_window():
    """Funcion que crea la ventana que se encarga de obtener los datos de un
empleado para añadirlo, llamando a la funcion add_emp.

Parameters
-----
None

```

```

Return
-----
None

"""

check_window_open()
global counter_windows

if counter_windows == 0:
    counter_windows += 1
    new_emp_window = tk.Toplevel(window)
    new_emp_window.protocol("WM_DELETE_WINDOW",
partial(callback,new_emp_window))
    new_emp_window.minsize(300,500)
    new_emp_window.title("Añadir empleado")


    empno_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey" )
    empno_label = tk.Label(new_emp_window, text="Numero del empleado:")
    empno_label.pack()

    empno_entry.insert(tk.END, "Ejemplo: 1234")
    empno_entry.pack(pady=5)

    empno_entry.bind("<FocusIn>", handle_focus_in)


    nombre_label = tk.Label(new_emp_window, text="Nombre del empleado
nuevo:")
    nombre_label.pack()
    nombre_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey")
    nombre_entry.insert(tk.END, "Ejemplo: Jose")
    nombre_entry.pack(pady=5)

    nombre_entry.bind("<FocusIn>", handle_focus_in)


    job_label = tk.Label(new_emp_window, text="Puesto del empleado
nuevo:")

```

```

        job_label.pack()
        job_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey")
        job_entry.insert(tk.END, "Ejemplo: CLERK")
        job_entry.pack(pady=5)

        job_entry.bind("<FocusIn>", handle_focus_in)


        manager_label = tk.Label(new_emp_window, text="Numero de empleado
del jefe del empleado nuevo:")
        manager_label.pack()
        manager_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey")
        manager_entry.insert(tk.END, "Ejemplo: 4567")
        manager_entry.pack(pady=5)

        manager_entry.bind("<FocusIn>", handle_focus_in)


        date_label = tk.Label(new_emp_window, text="Fecha de contratacion
del empleado nuevo:")
        date_label.pack()
        date_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey")
        date_entry.insert(tk.END, "Ejemplo: 04/10/2022")
        date_entry.pack(pady=5)

        date_entry.bind("<FocusIn>", handle_focus_in)


        salary_label = tk.Label(new_emp_window, text="Salario del empleado
nuevo:")
        salary_label.pack()
        salary_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey")
        salary_entry.insert(tk.END, "Ejemplo: 1600")
        salary_entry.pack(pady=5)

        salary_entry.bind("<FocusIn>", handle_focus_in)

```



```

        comision_label = tk.Label(new_emp_window, text="Comision del
empleado nuevo:")
        comision_label.pack()
        comision_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey")
        comision_entry.insert(tk.END, "Ejemplo: 200")
        comision_entry.pack(pady=5)

        comision_entry.bind("<FocusIn>", handle_focus_in)

        depto_label = tk.Label(new_emp_window, text="Numero del departamento
del empleado nuevo:")
        depto_label.pack()
        depto_entry = tk.Entry(new_emp_window, font=('Helvetica 12'),
fg="grey")
        depto_entry.insert(tk.END, "Ejemplo: 10")
        depto_entry.pack(pady=5)

        depto_entry.bind("<FocusIn>", handle_focus_in)

def ok():
    empno = int(empno_entry.get())
    nombre = str(nombre_entry.get())
    job = str(job_entry.get())
    manager_id = int(manager_entry.get())
    date = str(date_entry.get())

    salary = int(salary_entry.get())
    comision = int(comision_entry.get())
    depto = int(depto_entry.get())

    add_emp(empno, nombre ,job ,manager_id ,date ,salary ,comision
,depto )

    boton = tk.Button(new_emp_window, text="Añadir empleado", command=ok
)

    boton.pack()

```

```

def update_emp_window():
    """Funcion que crea la ventana que se encarga de obtener los datos de un
    empleado para actualizar sus datos, llamando a la funcion update_emp.

    Parameters
    -----
    None

    Return
    -----
    None
    """

    check_window_open()
    global counter_windows

    if counter_windows == 0:
        counter_windows += 1
        edit_emp_window = tk.Toplevel(window)
        edit_emp_window.protocol("WM_DELETE_WINDOW",
partial(callback,edit_emp_window))
        edit_emp_window.minsize(300,500)
        edit_emp_window.title("Modificar empleado")

        empno_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey" )
        empno_label = tk.Label(edit_emp_window, text="Numero del empleado a
editar:")
        empno_label.pack()

        empno_entry.insert(tk.END, "Ejemplo: 1234")
        empno_entry.pack(pady=5)

        empno_entry.bind("<FocusIn>", handle_focus_in)

        nombre_label = tk.Label(edit_emp_window, text="Nuevo nombre del
empleado a editar:")
        nombre_label.pack()
        nombre_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey")
        nombre_entry.insert(tk.END, "Ejemplo: Jose")
        nombre_entry.pack(pady=5)

        nombre_entry.bind("<FocusIn>", handle_focus_in)

```

```

        job_label = tk.Label(edit_emp_window, text="Nuevo puesto del
empleado a editar:")
        job_label.pack()
        job_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey")
        job_entry.insert(tk.END, "Ejemplo: CLERK")
        job_entry.pack(pady=5)

        job_entry.bind("<FocusIn>", handle_focus_in)

        manager_label = tk.Label(edit_emp_window, text="Nuevo numero de
empleado del jefe del empleado a editar:")
        manager_label.pack()
        manager_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey")
        manager_entry.insert(tk.END, "Ejemplo: 4567")
        manager_entry.pack(pady=5)

        manager_entry.bind("<FocusIn>", handle_focus_in)

        date_label = tk.Label(edit_emp_window, text="Nueva fecha de
contratacion del empleado a editar:")
        date_label.pack()
        date_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey")
        date_entry.insert(tk.END, "Ejemplo: 04/10/2022")
        date_entry.pack(pady=5)

        date_entry.bind("<FocusIn>", handle_focus_in)

        salary_label = tk.Label(edit_emp_window, text="Nuevo salario del
empleado a editar:")
        salary_label.pack()
        salary_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey")
        salary_entry.insert(tk.END, "Ejemplo: 1600")
        salary_entry.pack(pady=5)

        salary_entry.bind("<FocusIn>", handle_focus_in)

```

```

        comision_label = tk.Label(edit_emp_window, text="Nueva comision del
empleado a editar:")
        comision_label.pack()
        comision_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey")
        comision_entry.insert(tk.END, "Ejemplo: 200")
        comision_entry.pack(pady=5)

        comision_entry.bind("<FocusIn>", handle_focus_in)

        depto_label = tk.Label(edit_emp_window, text="Nuevo numero del
departamento del empleado a editar:")
        depto_label.pack()
        depto_entry = tk.Entry(edit_emp_window, font=('Helvetica 12'),
fg="grey")
        depto_entry.insert(tk.END, "Ejemplo: 10")
        depto_entry.pack(pady=5)

        depto_entry.bind("<FocusIn>", handle_focus_in)
    def ok():
        empno = int(empno_entry.get())
        nombre = str(nombre_entry.get())
        job = str(job_entry.get())
        manager_id = int(manager_entry.get())
        date = str(date_entry.get())
        salary = int(salary_entry.get())
        comision = int(comision_entry.get())
        depto = int(depto_entry.get())

        update_emp(empno, nombre ,job ,manager_id ,date ,salary
,comision ,depto )

        boton = tk.Button(edit_emp_window, text="Modificar empleado",
command=ok )

        boton.pack()

def delete_emp_window():
    """Funcion que crea la ventana que se encarga de obtener los datos de un
empleado para eliminarlo, llamando a la funcion delete_emp.

    Parameters
    -----

```

```

None

Return
-----
None

"""

check_window_open()
global counter_windows

if counter_windows == 0:
    counter_windows += 1
    delete_emp_window = tk.Toplevel(window)
    delete_emp_window.protocol("WM_DELETE_WINDOW",
partial(callback,delete_emp_window))
    delete_emp_window.minsize(300,500)
    delete_emp_window.title("Eliminar empleado")

    empno_entry = tk.Entry(delete_emp_window, font=('Helvetica 12'),
fg="grey" )
    empno_label = tk.Label(delete_emp_window, text="Numero del empleado
a eliminar:")
    empno_label.pack()

    empno_entry.insert(tk.END, "Ejemplo: 1234")
    empno_entry.pack(pady=5)

    empno_entry.bind("<FocusIn>", handle_focus_in)

    def ok():
        empno = int(empno_entry.get())
        delete_emp(empno)

    boton = tk.Button(delete_emp_window, text="Eliminar empleado",
command=ok )

    boton.pack()

def noemp_depto_window():

```

```
"""Funcion que crea la ventana que se encarga de obtener el numero de
departamento para calcular el numero de empleados registrados en el,
llamando a la funcion noemp_depto.
```

```
Parameters
```

```
-----
```

```
None
```

```
Return
```

```
-----
```

```
None
```

```
"""
```

```
check_window_open()
```

```
global counter_windows
```

```
if counter_windows == 0:
```

```
    counter_windows += 1
```

```
    empleados_windows = tk.Toplevel(window)
```

```
    empleados_windows.protocol("WM_DELETE_WINDOW",
partial(callback,empleados_windows))
```

```
    empleados_windows.minsize(300,500)
```

```
    empleados_windows.title("Obtener cantidad de empleados de depto")
```

```
    empno_entry = tk.Entry(empleados_windows, font=('Helvetica 12'),
fg="grey" )
```

```
    empno_label = tk.Label(empleados_windows, text="Número de
departamento:")
```

```
    empno_label.pack()
```

```
    empno_entry.insert(tk.END, "Ejemplo: 1234")
```

```
    empno_entry.pack(pady=5)
```

```
    empno_entry.bind("<FocusIn>", handle_focus_in)
```

```
def ok():
```

```
    empno = int(empno_entry.get())
```

```
    noemp_depto(empno)
```

```
    boton = tk.Button(empleados_windows, text="Obtener cantidad de
empleados", command=ok )
```

```

        boton.pack()

def search_dept_window():
    """Funcion que crea la ventana que se encarga de obtener el numero de
    departamento para obtener y mostrar sus datos llamando a la función
    read_dept.

    Parameters
    -----
    None

    Return
    -----
    None
    """

    check_window_open()
    global counter_windows

    if counter_windows == 0:
        counter_windows += 1
        empleados_windows = tk.Toplevel(window)
        empleados_windows.protocol("WM_DELETE_WINDOW",
partial(callback,empleados_windows))
        empleados_windows.minsize(300,500)
        empleados_windows.title("Buscar departamento")

        empno_entry = tk.Entry(empleados_windows, font=('Helvetica 12'),
fg="grey" )
        empno_label = tk.Label(empleados_windows, text="Número de
departamento:")
        empno_label.pack()

        empno_entry.insert(tk.END, "Ejemplo: 1234")
        empno_entry.pack(pady=5)

        empno_entry.bind("<FocusIn>", handle_focus_in)

    def ok():
        dept_no = int(empno_entry.get())
        read_dept(dept_no)

```

```

        boton = tk.Button(empleados_windows, text="Buscar departamento",
command=ok )

        boton.pack()

def search_emp_window():
    """Funcion que crea la ventana que se encarga de obtener el numero de
    empleado para obtener y mostrar sus datos llamando a la función read_emp.

    Parameters
    -----
    None

    Return
    -----
    None
    """
    check_window_open()
    global counter_windows

    if counter_windows == 0:
        counter_windows += 1
        empleados_windows = tk.Toplevel(window)
        empleados_windows.protocol("WM_DELETE_WINDOW",
partial(callback,empleados_windows))
        empleados_windows.minsize(300,500)
        empleados_windows.title("Buscar empleado")

        empno_entry = tk.Entry(empleados_windows, font=('Helvetica 12'),
fg="grey" )
        empno_label = tk.Label(empleados_windows, text="Número de
empleado:")
        empno_label.pack()

        empno_entry.insert(tk.END, "Ejemplo: 1234")
        empno_entry.pack(pady=5)

        empno_entry.bind("<FocusIn>", handle_focus_in)

    def ok():
        emp_no = int(empno_entry.get())

        read_emp(emp_no)

```



```

        boton = tk.Button(empleados_windows, text="Buscar empleado",
command=ok )

        boton.pack()

def check_window_open():
    """Funcion que verifica el número de ventanas abiertas, para evitar que
se abra mas de una y saturar el sistema, mostrando un mensaje de alerta si
hay mas de una ventana de petición abierta.

Parameters
-----
None

Return
-----
None
"""
    global counter_windows
    if counter_windows > 0:
        messagebox.showwarning(message = "Sólo puedes hacer una acción a la
vez")

def Exit():
    """Funcion que destruye la ventana principal para terminar la ejecución
del programa.

Parameters
-----
None

Return
-----
None
"""
    window.destroy()

"""Definición y creación de los objetos de tipo Button, los cuales hacen las
llamadas a las funciones correspondientes, abriendo una ventana para cada
caso específico de consulta o procedimiento

```

```

"""
add_emp_bt = tk.Button(frame_2, text='Añadir empleado', font=(font_1, 12),
bd=2, command=add_emp_window , cursor="hand2",
bg=color_2,fg=color_3,width=25).pack( padx=5, pady=10)

view_bt = tk.Button(frame_2, text='Modificar empleado', font=(font_1, 12),
bd=2, command=update_emp_window, cursor="hand2",
bg=color_2,fg=color_3,width=25).pack( padx=5, pady=10)

delete_empn_bt = tk.Button(frame_2, text='Borrar empleado', font=(font_1,
12), bd=2, command=delete_emp_window,cursor="hand2",
bg=color_2,fg=color_3,width=25).pack(padx=5, pady=10)

add_depto_bt = tk.Button(frame_2, text='Añadir departament', font=(font_1,
12), bd=2, command=add_depto_window, cursor="hand2",
bg=color_2,fg=color_3,width=25).pack(padx=5, pady=10)

update_dept_bt = tk.Button(frame_2, text='Modificar departamento',
font=(font_1, 12), bd=2, command=update_dept_window, cursor="hand2",
bg=color_2,fg=color_3,width=25).pack(padx=5, pady=10)

delete_dept_bt = tk.Button(frame_2, text='Borrar departamento',
font=(font_1, 12), bd=2, command=delete_dept_window, cursor="hand2",
bg=color_2,fg=color_3,width=25).pack(padx=5, pady=10)

get_no_emp_bt = tk.Button(frame_2, text='No. Empleados', font=(font_1, 12),
bd=2, command=noemp_depto_window, cursor="hand2",
bg=color_2,fg=color_3,anchor=tk.CENTER,width=25).pack(padx=5, pady=10)

buscar_emp_bt = tk.Button(frame_2, text='Buscar un empleado', font=(font_1,
12), bd=2, command=search_emp_window, cursor="hand2",
bg=color_2,fg=color_3,anchor=tk.CENTER,width=25).pack(padx=5, pady=10)

buscar_depto = tk.Button(frame_2, text='Buscar un departamento',
font=(font_1, 12), bd=2, command=search_dept_window, cursor="hand2",
bg=color_2,fg=color_3,anchor=tk.CENTER,width=25).pack(padx=5, pady=10)

exit_bt = tk.Button(frame_2, text='Exit', font=(font_1, 12), bd=2,
command=Exit, cursor="hand2",
bg=color_2,fg=color_3,anchor=tk.CENTER,width=15).pack(padx=5, pady=10)

window.mainloop()

```

Ventana principal:

Proyecto 1

Integrantes

José Carlos Chaparro Morales - 329613
Juan Luis Del Valle Sotelo - 338912
Omar Alonso Escápita Chacón - 338912

Docente

M. A. José Saúl De Lira Miramontes



Añadir empleado

Modificar empleado

Borrar empleado

Añadir departamento

Modificar departamento

Borrar departamento

No. Empleados

Buscar un empleado

Buscar un departamento

Exit

Añadir empleado:

Añadir empleado

Numero del empleado:

Nombre del empleado nuevo:

Puesto del empleado nuevo:

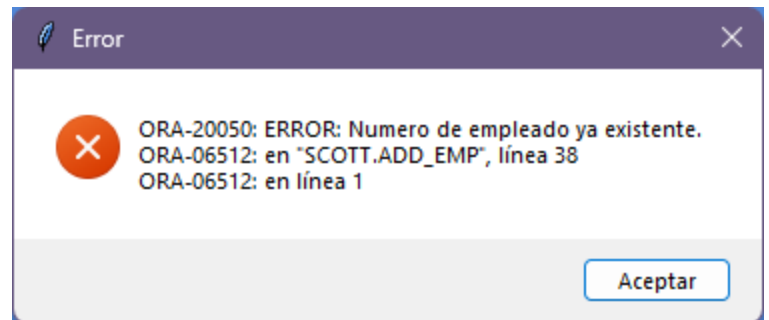
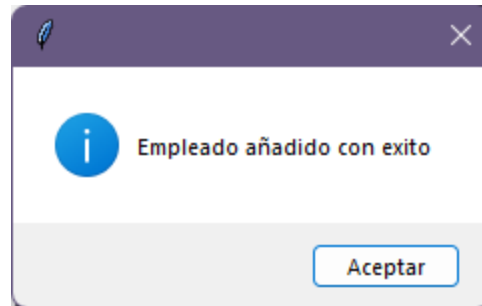
Numero de empleado del jefe del empleado nuevo:

Fecha de contratacion del empleado nuevo:

Salario del empleado nuevo:

Comision del empleado nuevo:

Numero del departamento del empleado nuevo:



Modificar empleado:

Modificar empleado

Numero del empleado a editar:

Nuevo nombre del empleado a editar:

Nuevo puesto del empleado a editar:

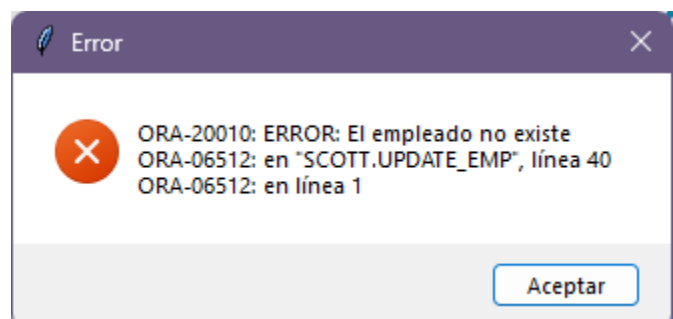
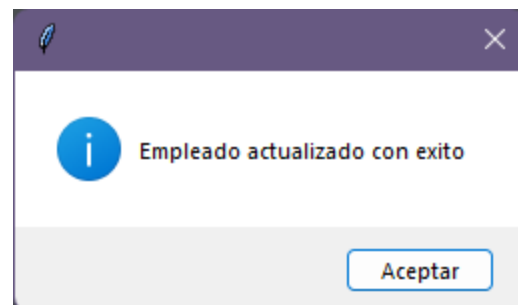
Nuevo numero de empleado del jefe del empleado a editar:

Nueva fecha de contratacion del empleado a editar:

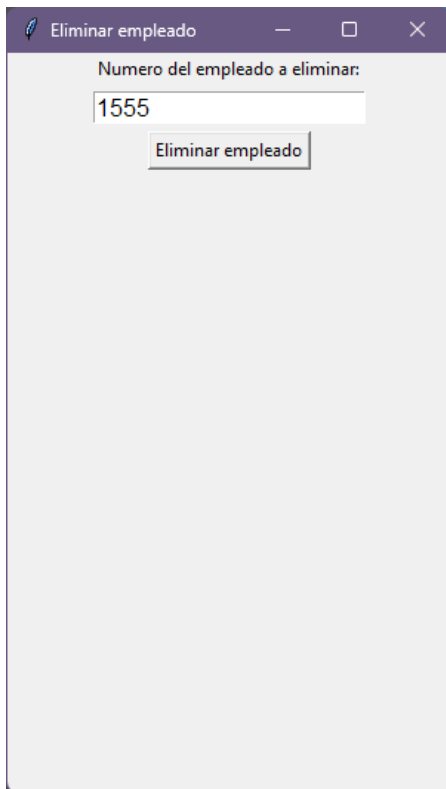
Nuevo salario del empleado a editar:

Nueva comision del empleado a editar:

Nuevo numero del departamento del empleado a editar:



Eliminar empleado:

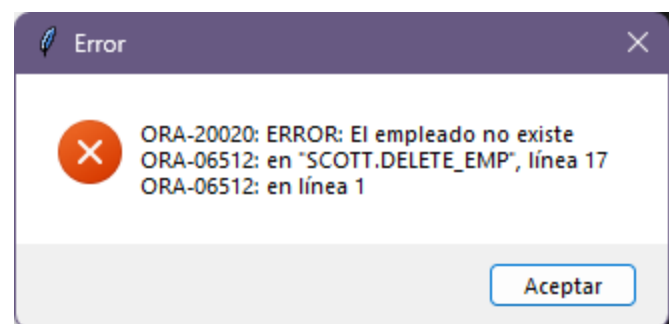
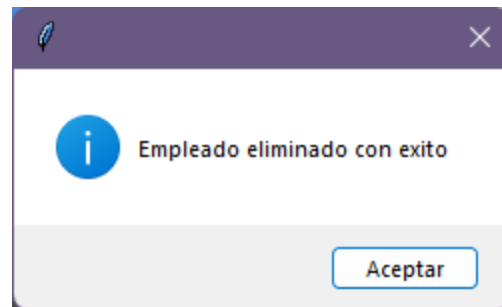


Eliminar empleado

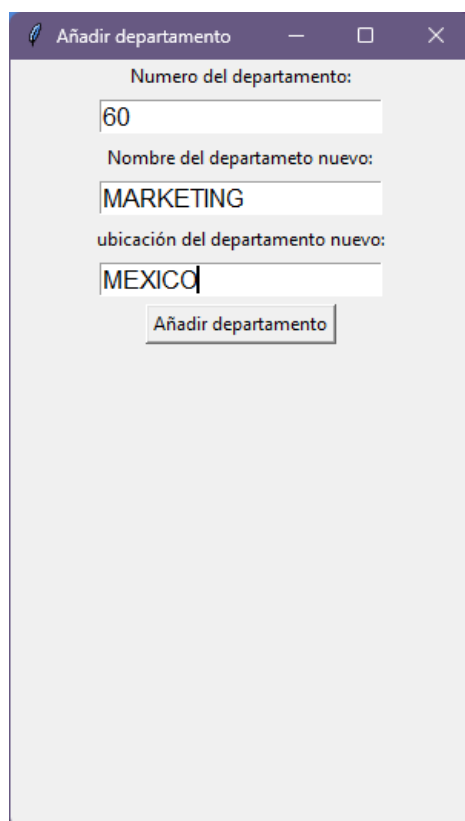
Numero del empleado a eliminar:

1555

Eliminar empleado



Añadir departamento:



Añadir departamento

Numero del departamento:

60

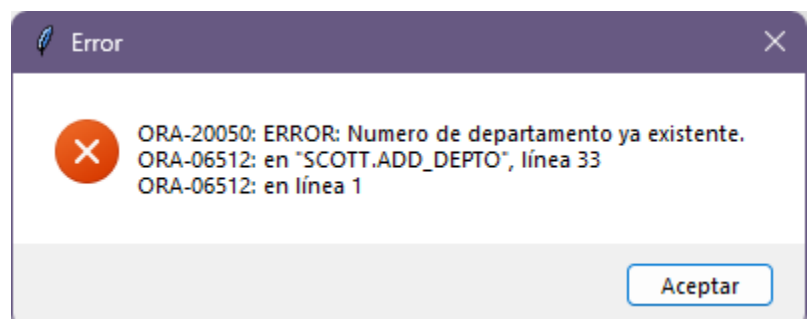
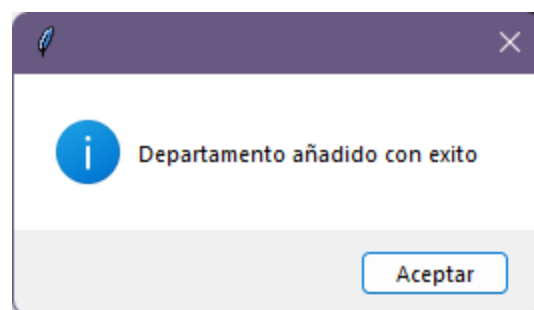
Nombre del departameto nuevo:

MARKETING

ubicación del departamento nuevo:

MEXICO

Añadir departamento



Modificar departamento:

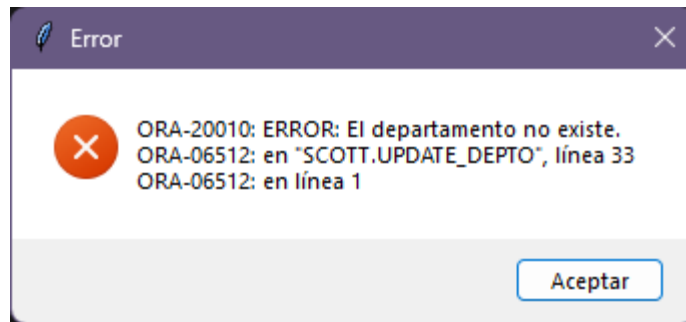
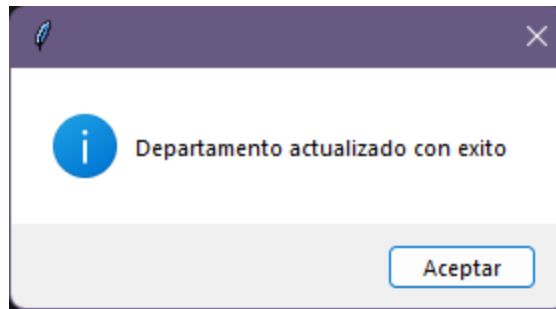
Modificar departamento

Numero del departamento:
70

Nuevo nombre del departamento a modificar:
ESPORTS

Nueva localización del departamento a modificar:
CHIHUAHUA

Actualizar departamento

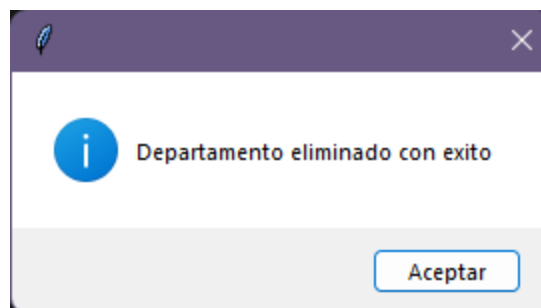


Eliminar departamento:

Eliminar departamento

Numero del departamento a eliminar:
60

Eliminar departamento



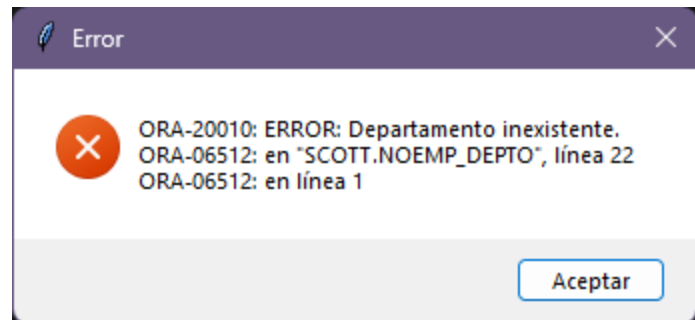
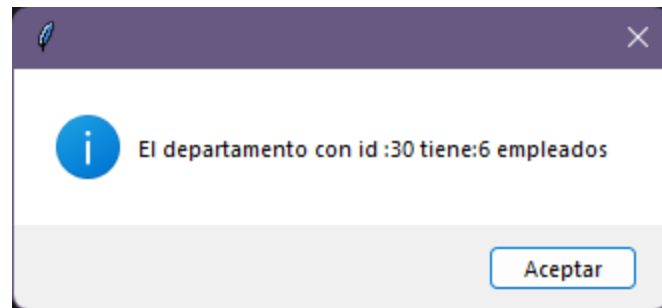
Número de empleados:

Obtener cantidad de e...

Número de departamento:

30

Obtener cantidad de empleados



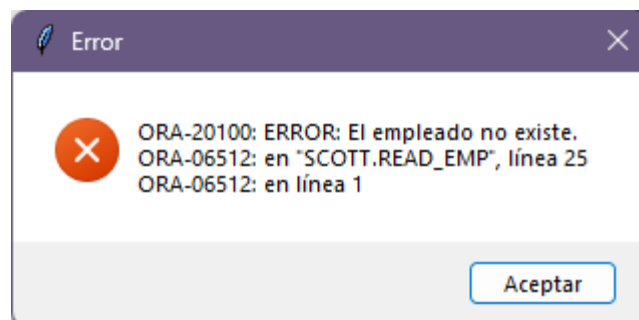
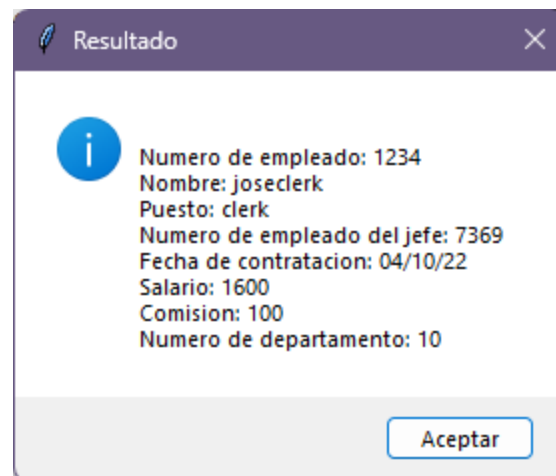
Buscar empleado:

Buscar empleado

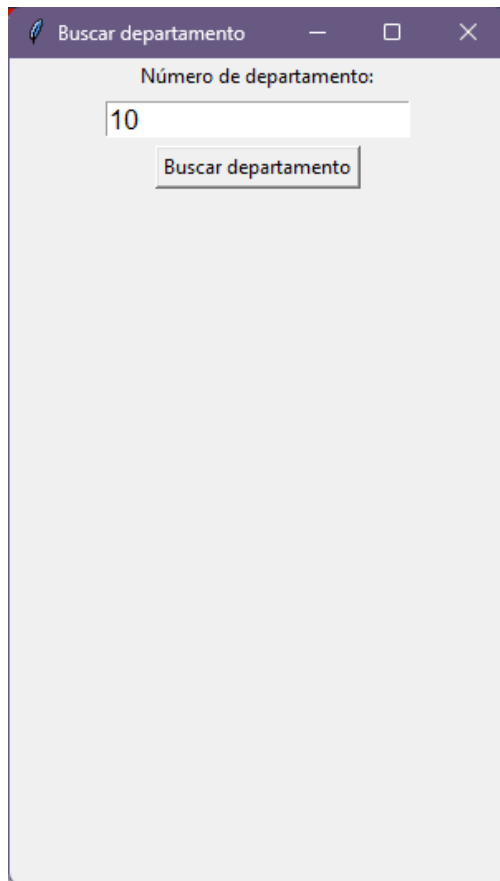
Número de empleado:

1234

Buscar empleado



Buscar departamento:

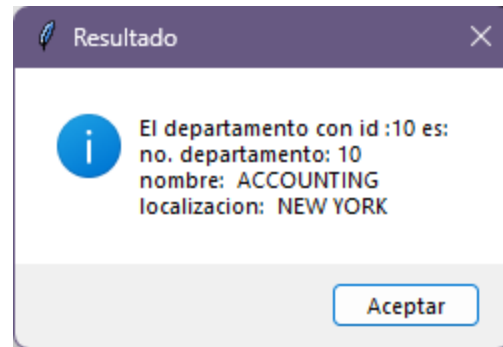


Buscar departamento

Número de departamento:

10

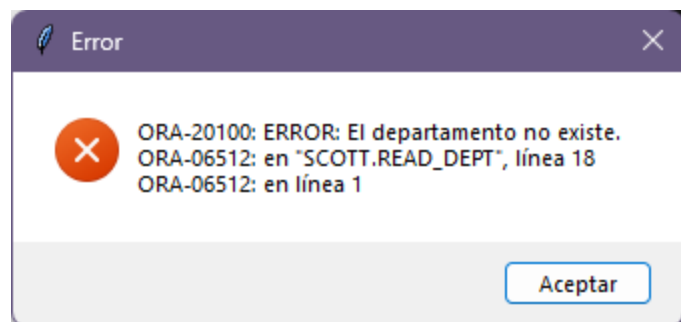
Buscar departamento



Resultado

El departamento con id :10 es:
no. departamento: 10
nombre: ACCOUNTING
localizacion: NEW YORK

Aceptar



Error

ORA-20100: ERROR: El departamento no existe.
ORA-06512: en "SCOTT.READ_DEPT", línea 18
ORA-06512: en línea 1

Aceptar