

Soluciones Ejercicios prácticos

Ejercicio 1

```
notas=[]

opcion=0

def guardarnota(nota):

    notas.append(nota)

def calcularaprobados():

    aprobados=0

    for n in notas:

        if n>5:

            aprobados+=1

    return aprobados

def mayor():

    return max(notas)

def menor():

    return min(notas)

while opcion!=4:

    print("1. Introducir nota. ")

    print("2. Total aprobados. ")

    print("3. Nota mayor. ")

    print("4. Nota inferior. ")

    print("5. Salir. ")

    opcion=int(input("Elige opción: "))

    if opcion==1:

        nota=float(input("Introduce nota: "))
```

```
    guardarnota(nota)

elif opcion==2:

    print("Total de aprobados: ",calcularaprobados())

elif opcion==3:

    print("Nota mayor: ",mayor())

elif opcion==4:

    print("Nota menor: ",menor())
```

Ejercicio 2

```
def agregarContacto(nombre,email,telefono):

    #crea un diccionario con los datos del nuevo contacto

    #y lo añade a la lista

    contacto={}

    contacto["nombre"]=nombre

    contacto["email"]=email

    contacto["telefono"]=telefono

    agenda.append(contacto)
```

```
def eliminarContacto(email):

    #recorre la lista de contactos y si encuentra uno

    #con ese email lo elimina

    pos=0

    for cont in agenda:

        if cont["email"]==email:

            del(agenda[pos])

            pos+=1
```

```
def buscarContacto(email):
```

```

#si encuentra el contacto lo devuelve

for cont in agenda:

    if cont["email"]==email:

        return cont

return None

def mostrarContactos():

    for cont in agenda:

        print(cont["nombre"]," - ",cont["email"], " - ",cont["telefono"])

agenda=[]

opcion=0

while opcion!=5:

    print("1. Nuevo contacto. ")

    print("2. Buscar contacto. ")

    print("3. Eliminar contacto. ")

    print("4. Mostrar contactos. ")

    print("5. Salir. ")

    opcion=int(input("Elige opción: "))

    if opcion==1:

        nombre=input("Nombre del contacto: ")

        email=input("Email del contacto: ")

        telefono=int(input("Teléfono del contacto: "))

        #comprueba si el contacto existe antes de agregarlo

        if buscarContacto(email)==None:

            agregarContacto(nombre,email,telefono)

            print("Contacto agregado ")

        else:

            print("El contacto ya existe!")

```

```

elif opcion==2:

    email=input("Introduce email: ")

    contacto=buscarContacto(email)

    #si encuentra el contacto lo muestra

    if contacto!=None:

        print(contacto["nombre"]," - ",contacto["email"], " - ",contacto["telefono"])

    else:

        print("Ese contacto no existe ")

elif opcion==3:

    email=input("Introduce email: ")

    eliminarContacto(email)

elif opcion==4:

    mostrarContactos()

```

Ejercicio 3

```

def nuevocontacto(nombre, email, telefono):

    f=open("contactos.txt","a")

    f.write(nombre+"|" +email+"|" +str(telefono)+"\n")

    f.close()

def buscarcontacto(email):

    try:

        f=open("contactos.txt","r")

        contacto=f.readline()

        while contacto!="":

            datos=contacto.split("|")

            if datos[1]==email:

                resultado={}

```

```
        resultado["nombre"]=datos[0]

        resultado["email"]=datos[1]

        resultado["telefono"]=int(datos[2])

        f.close()

        return resultado

    contacto=f.readline()

    f.close()

    return None

except FileNotFoundError:

    return None
```

```
def eliminarcontacto(email):
```

```
    try:

        f=open("contactos.txt","r")

        lista=f.readlines()

        pos=0

        for item in lista:

            if item.split(" ")[1]==email:

                del(lista[pos])

                volcarlista(lista)

                f.close()

                return True

            pos+=1

        f.close()

        return False

    except FileNotFoundError:

        return False
```

```

def volcarlista(lista):

    f=open("contactos.txt","w")

    for elm in lista:

        f.write(elm)

    f.close()


def imprimircontactos():

    try:

        f=open("contactos.txt","r")

        contacto=f.readline()

        while contacto!="":

            datos=contacto.split("|")

            print("nombre: ",datos[0], " email:",datos[1]," telefono:",datos[2])

            contacto=f.readline()

        f.close()

    except FileNotFoundError:

        print("No existen contactos ")

opcion=0

while opcion!=5:

    print("1. Nuevo contacto. ")

    print("2. Buscar contacto. ")

    print("3. Eliminar contacto. ")

    print("4. Mostrar contactos. ")

    print("5. Salir. ")

    opcion=int(input("Elige opción: "))

    if opcion==1:

        nombre=input("Introduce nombre: ")

```

```
email=input("Introduce email: ")

telefono=input("Introduce teléfono: ")

#si ese email no existe, se añade el nuevo contacto

if buscarcontacto(email)==None:

    nuevocontacto(nombre,email,telefono)

    print("contacto añadido")

else:

    print("ese email ya está siendo utilizado!!")

elif opcion==2:

    email=input("Introduce email: ")

    contacto=buscarcontacto(email)

    if contacto!=None:

        print("Nombre: ",contacto["nombre"])

    else:

        print("No existe ese contacto ")

elif opcion==3:

    email=input("Introduce email: ")

    res=eliminarcontacto(email)

    if not res:

        print("Contacto no existente, no se pudo eliminar")

    else:

        print("Contacto eliminado")

elif opcion==4:

    imprimircontactos()
```

Ejercicio 4

```
import json

import datetime

def contagiospais(nombrepais):

    f=open("contagios.json","r",encoding="UTF-8")

    #devuelve una lista de diccionarios

    datos=json.load(f)

    contagios=0

    #recorre cada objeto JSON de la lista y suma los

    #contagios del pais recibido como parámetro

    for item in datos:

        if item["pais"]==nombrepais:

            contagios+=item["contagios"]

    f.close()

    return contagios

def contagiosposterioresfecha(fecha):

    f=open("contagios.json","r",encoding="UTF-8")

    #devuelve una lista de diccionarios

    datos=json.load(f)

    contagios=0

    #recorre cada objeto JSON de la lista y suma los

    #contagios cuyas fechas sean posteriores a la recibida

    for item in datos:

        if transformarafecha(item["fecha"])>transformarafecha(fecha):

            contagios+=item["contagios"]

    f.close()

    return contagios

def paismascontagios():
```



```

f=open("contagios.json","r",encoding="UTF-8")

#devuelve una lista de diccionarios

datos=json.load(f)

#inicializa el resultado al nombre del primer pais

nombrepais=datos[0]["pais"]

#recorre cada objeto JSON de la lista y si encuentra un pais

#cuyos contagios sean superiores a los del pais registrado, actualiza

#el resultado con el nombre de ese pais

for item in datos:

    if contagiospais(item["pais"])>contagiospais(nombrepais):

        nombrepais=item["pais"]

f.close()

return nombrepais

def transformarafecha(texto):

    formato="%Y-%m-%d"

    return datetime.datetime.strptime(texto,formato)

#def agruparporpais():

# grupo={}

# f=open("contagios.json","r")

# #devuelve una lista de diccionarios

# datos=json.load(f)

# for item in datos:

#     nombrepais=item["pais"]

#     if nombrepais not in grupo.keys():

#         grupo[nombrepais]=item["contagios"]

#     else:

#         grupo[nombrepais]+=item["contagios"]

```

```

# f.close()

# return grupo

opcion=0

while opcion!=4:

    print("1. Contagios de pais")

    print("2. Contagios posteriores a fecha")

    print("3. País con más contagios")

    print("4.- Salir")

    opcion=int(input("Introduce opción: "))

    if opcion==1:

        nombre=input("Introduce nombre de país: ")

        print("Total de contagios en ",nombre," : ",contagiospais(nombre))

    elif opcion==2:

        fecha=input("Introduce fecha (año-mes-día): ")

        print("Total de contagios posteriores a esa fecha: ",contagiosposterioresfecha(fecha))

    elif opcion==3:

        print("País con más contagios ",paismascontagios())

```

Ejercicio 5

Suponemos que hemos creado una base de datos llamada “contagios” y que la colección donde vamos a guardar los datos será “registros”:

```

import pymongo, json

uri="mongodb://localhost:27017"

sr=pymongo.MongoClient(uri)

db=sr.contagios

db.drop_collection("registros")

col=db.registros

```

```
#abre el fichero y vuelca los datos en la BD
```

```
f=open("contagios.json","r",encoding="UTF-8")
```

```
datos=json.load(f)
```

```
for doc in datos:
```

```
    col.insert_one(doc)
```

```
f.close()
```