



El futuro digital
es de todos

MinTIC



CRUD

OPERADO POR:



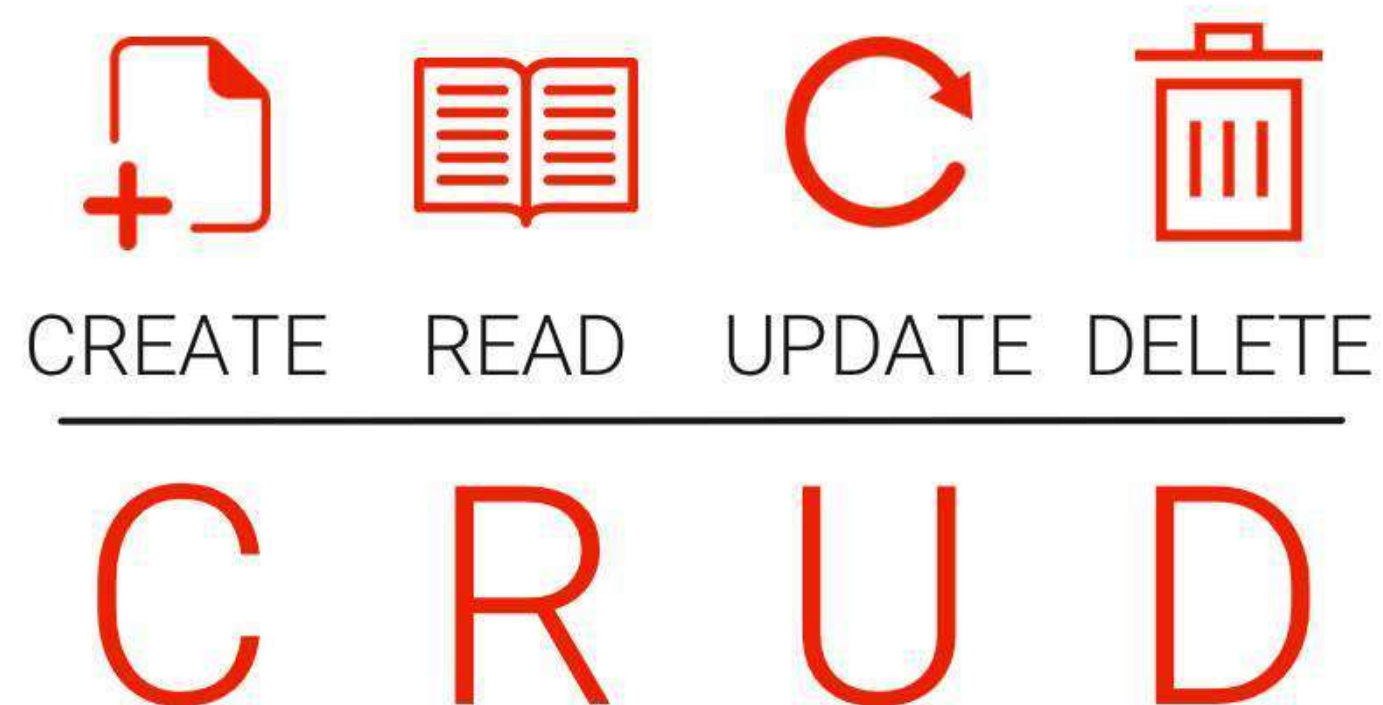
Mision
TIC 2022

ruta de aprendizaje 1



Que es CRUD

CRUD (Create, Read, Update, Delete) es un acrónimo para la manera en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos, pero también puede aplicarse a funciones de un nivel superior de una aplicación como **soft deletes** donde la información no es realmente eliminada, sino marcada como eliminada a través de un estatus.





Administración de categorías

localhost/tutorialesprogramacionyaejercicios/phpyotros/facturacion/categorias/administracion.html

Administracion de Categorías

Código	Descripción	Edición
1	Microprocesadores	Edita?
2	Discos Sólidos SSD	Edita?
3	Placas De Video	Edita?

Agregar

Finalizar

Bases de datos:

Se llama base de datos, o también banco de datos, a un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y/o transmisión. Existen actualmente muchas formas de bases de datos, que van desde una biblioteca hasta vastos conjuntos de datos de usuarios de una empresa de telecomunicaciones.



Construcción de una aplicación CRUD básica:

Normalmente para gestionar una base de datos usamos un gestor de bases de datos como **Mysql**, el cual nos brinda las herramientas necesarias para simplificar el manejo de una base de datos. Sin embargo este tema lo veremos mas adelante en el curso. Para la creación de nuestra aplicación **CRUD** básica con Python usaremos Excel como nuestra base de datos y crearemos nuestras propias funciones para gestionar la información.

Es necesario aclarar que Excel por si solo es una poderosa herramienta para la organización de datos sin embargo por motivos didácticos nos limitaremos a usarlo como una base de datos.

	A	B	C	D	E	F
1	Id	Titulo	Descripcion	Estado	Fecha inicio	Fecha finalizado
2	1	nueva tarea	una tarea crada	Finalizada	15/8/2021	15/8/2021
3	2	tarea actu	una tarea actualizada	Por aprobar	15/8/2021	hoy
4	3	probando	tarea de prueba	En espera	ayer	hoy
5	4	tarea actualizada	una tarea actualizada	Finalizada	15/8/2021	15/8/2021
6	5	probando	tarea de prueba	En espera	ayer	hoy
7	6	tarea borrada	una tarea borrada	En espera	15/8/2021	
8	7	tarea actualizada	tercera tarea actualizada	En ejecucion	15/8/2021	15/8/2021
9	8	nueva tarea	esta es una nueva tarea	Finalizada	15/8/2021	15/8/2021
10						



Nuestra aplicación será un organizador de tareas, el cual nos permitirá llevar una constancia de las tareas y el estado de las mismas, el estado de esta se refiere a si la tarea esta en espera, en ejecución, por aprobar o finalizada, a demás podremos observar cuando se creo la tarea y cuando se finalizo la tarea. Como toda aplicación CRUD nos permitirá Crear, Leer, Actualizar y borrar, por motivos de simplicidad usaremos comandos por consola.

	A	B	C	D	E	F
1	Id	Titulo	Descripcion	Estado	Fecha inicio	Fecha finalizado
2	1	nueva tarea	una tarea crada	Finalizada	15/8/2021	15/8/2021
3	2	tarea actu	una tarea actualizada	Por aprobar	15/8/2021	hoy
4	3	probando	tarea de prueba	En espera	ayer	hoy
5	4	tarea actualizada	una tarea actualizada	Finalizada	15/8/2021	15/8/2021
6	5	probando	tarea de prueba	En espera	ayer	hoy
7	6	tarea borrada	una tarea borrada	En espera	15/8/2021	
8	7	tarea actualizada	tercera tarea actualizada	En ejecucion	15/8/2021	15/8/2021
9	8	nueva tarea	esta es una nueva tarea	Finalizada	15/8/2021	15/8/2021
10						



Comenzaremos creando nuestro archivo Excel el cual usaremos como base de datos, la llamaremos “Base crud” y la hoja del archivo la llamaremos “Datos del crud”.

Podemos observar que Excel se organiza por filas y columnas, los índices de la filas son numeros mientras que los índices de las columnas son las letras de nuestro alfabeto. En nuestro archivo usaremos la fila 1 y las columnas desde la A hasta la F para agregar nuestros índices

The screenshot shows the Microsoft Excel interface. The title bar indicates the file is named 'Base crud'. The 'Inicio' ribbon is active, showing options for font (Calibri, size 11), alignment, and numbers. The spreadsheet area displays a table with the following structure:

	A	B	C	D	E	F
1	Id	Titulo	Descripcion	Estado	Fecha inicio	Fecha finalizado
2						
3						
4						
5						
6						

The bottom of the window shows the sheet tab 'Datos del crud' is selected.



Antes de comenzar a escribir nuestro código, necesitamos importar la librería que nos permitirán extraer la información desde nuestro archivo Excel y una librería que nos permitirá obtener la hora del sistema, es decir la fecha de nuestro computador

```
from datetime import datetime  
from openpyxl import load_workbook
```

Datetime nos permitirá obtener la hora del sistema y **openpyxl** nos permite manejar archivos Excel desde **python**





Ahora primero que todo definiremos una variable en la cual especificaremos la ruta del documento con el cual vamos a trabajar, el cual acabamos de crear.

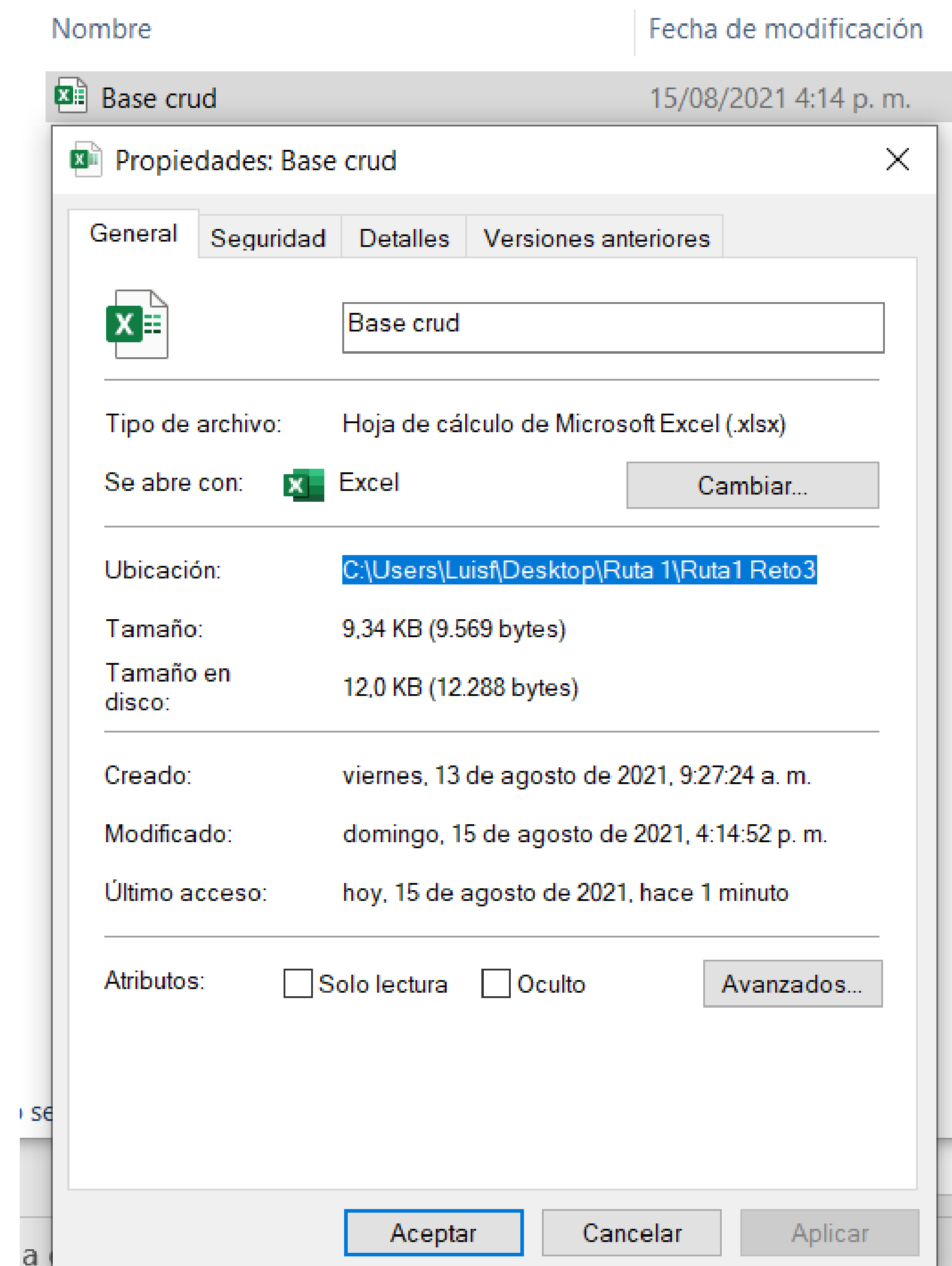
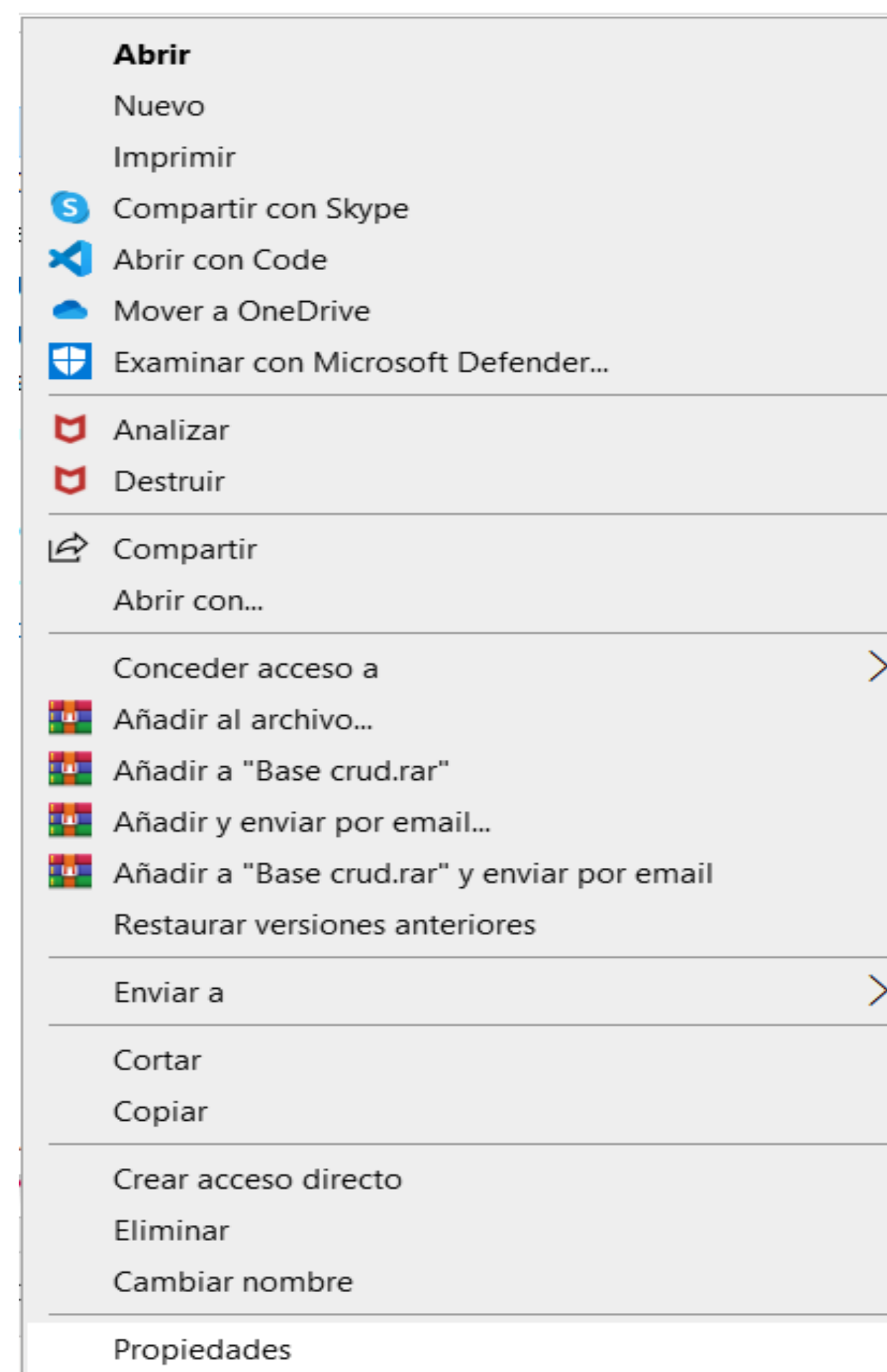
```
Rut="C:\\Users\\Usuarioprueba\\Desktop\\Ruta1\\Base crud.xlsx"
```

```
Rut=r"C:\Users\Usuarioprueba\Desktop\Ruta1\Base crud.xlsx"
```

Para que nuestro archivo pueda ser leído correctamente debemos utilizar dos estructuras diferentes, la letra r antes de la dirección o el doble backslas. Con esta configuración Python puede leer la ruta sin inconvenientes.



Para saber la dirección solo basta con darle clic derecho al archivo e ir a propiedades y en la pestaña de general podemos observar la dirección del archivo, solo copiamos y pegamos en nuestro documento de Python, debemos tener en cuenta que la dirección que copiamos no indica el nombre del archivo, debemos colocarlo nosotros y asegurarnos que el nombre del archivo termine con .xls también debemos colocar el doble slash en nuestra dirección





Primero crearemos nuestra función de leer, esta función lo que hará es recibir dos parámetros, la ruta de nuestro archivo y un segundo parámetro que usaremos para filtrar según el estado de nuestras tareas.

```
10 def leer(ruta:str, extraer:str):
11     Archivo_Excel = load_workbook(ruta)
12     Hoja_datos = Archivo_Excel['Datos del crud']
13     Hoja_datos=Hoja_datos['A2':'F'+str(Hoja_datos.max_row)]
14
15     info={}
16
17     for i in Hoja_datos:
18
19         if isinstance(i[0].value, int):
20             info.setdefault(i[0].value,{ 'tarea':i[1].value, 'descripcion':i[2].value,
21                                           'estado':i[3].value, 'fecha de inicio':i[4].value,
22                                           'fecha de finalizacion':i[5].value })
23
24     if not(extraer=='todo'):
25         info=filtrar(info, extraer)
26
27     for i in info:
28         print('***** Tarea *****')
29         print('Id: '+ str(i)+'\n'+ 'Titulo: ' +str(info[i]['tarea'])+'\n'+ 'Descripcion: '
30               +str(info[i]['descripcion']) + '\n'+ 'Estado: '+str(info[i]['estado'])
31               +'\n'+ 'Fecha Creacion: ' + str(info[i]['fecha de inicio'])
32               + '\n'+ 'Fecha de finalizacion: ' +str(info[i]['fecha de finalizacion']) )
33         print()
34     return
```




Explicaremos rápidamente la función.

Línea 11: **load_workbook(ruta)** esta instrucción nos permite importar el archivo Excel y lo guardamos en una variable

```
10 def leer(ruta:str, extraer:str):  
11     Archivo_Excel = load_workbook(ruta)
```

Línea 12: de la variable donde guardamos el archivo Excel seleccionamos la hoja “Datos del crud” que es el nombre que le dimos a la ja de nuestro archivo

```
10 def leer(ruta:str, extraer:str):  
11     Archivo_Excel = load_workbook(ruta)  
12     Hoja_datos = Archivo_Excel['Datos del crud']
```





Extraemos la información de la hoja desde A2 hasta la máxima fila de F, luego creamos un diccionario donde se organizaran los datos leídos desde el Excel.

```
Hoja_datos = Archivo_Excel['Datos del crud']  
Hoja_datos=Hoja_datos['A2':'F'+str(Hoja_datos.max_row)]  
info={}
```

Recorreremos la variable **Hoja_datos**, la cual es una **tupla**, la llaves de esta **tupla** son la filas del documento Excel y el valor de estas llaves son una tupla con los datos de las columnas de las filas.

Toma el valor de las tuplas, cuyos índices 1, 2, 3 etc, equivalen a la columna A, B, C etc en el documento Excel.





Vamos a usar el atributo **.value** para ir tomando el valor de la celda y guardarla en el diccionario, el diccionario **info** guarda toda la información del archivo Excel.

```
for i in Hoja_datos:
    if isinstance(i[0].value, int):
        info.setdefault(i[0].value, {'tarea':i[1].value, 'descripcion':i[2].value, 'estado':i[3].value, 'fecha de inicio':i[4].value,
                                     'fecha de finalizacion':i[5].value })
```

Comprobamos cual filtro queremos aplicar si es "todo" no llamamos a la función filtro, de lo contrario y si necesitamos todos llamamos a la función filtro. Una vez filtrado el diccionario imprimimos en consola el cada una de las tareas.



```
for i in info:
    print('***** Tarea *****')
    print('Id: ' + str(i) + '\n' + 'Titulo: ' + str(info[i]['tarea']) + '\n' + 'Descripcion: ' + str(info[i]['descripcion']) +
          '\n' + 'Estado: ' + str(info[i]['estado']) + '\n' + 'Fecha Creacion: ' + str(info[i]['fecha de inicio']) +
          '\n' + 'Fecha de finalizacion: ' + str(info[i]['fecha de finalizacion']))
    print()

return
```

Esta función recibe como parámetro el diccionario con toda la información del documento Excel y el estado por el cual queremos filtrar, luego creamos un diccionario auxiliar que guardara las tareas que cumplan con el filtro, recorremos el diccionario **info**, preguntamos si el estado de la tarea es igual al filtro que deseamos consultar y si cumple con la condición guardamos **i** como la llave y **info[i]** con el contenido, para que **aux** mantenga la misma estructura de diccionario **info**.



```
def filtrar(info:dict,filtro:str):  
    aux={}  
  
    for i in info:  
        if info[i]['estado']==filtro:  
            aux.setdefault(i,info[i])  
    return aux
```

Ahora escribimos la función que recibe como parámetros la ruta del archivo, el **id** de la tarea que editaremos y un diccionario con los datos que editaremos. Guardamos en archivo Excel especificado con la ruta en una variable, seleccionamos la hoja del archivo Excel que queremos leer y extraemos la información de la hoja desde A2 hasta la máxima fila de F.

Del archivo Excel seleccionamos la primera hora. A diferencia de las listas cuyos índices comienzan en 0, los archivos Excel comienzan en 1 por lo tanto la columna A equivale a índice 1 la B al índice 2 y así sucesivamente.



Al conocer la estructura del archivo Excel sabemos que la celda del título esta en la fila B que corresponde al índice 2, la celda de descripción esta en la columna C correspondiente al índice 3, y el estado se encuentra en la columna D correspondiente al índice 5.

Usamos una variable booleana para saber si se encontró el **id** que se específico. Este tipo de variables también son conocidas como **banderas**, recorreremos la hoja de datos, preguntamos si el id que estamos leyendo es igual al que estamos buscando. El atributo **.row** es similar al atributo **.value**, con la diferencia que este nos devuelve la fila en donde se encuentra. Colocamos la bandera de encontrado en **true**.

Una vez encontrado el **id** conocida, recorreremos el vector de datos actualizados, y vamos agregamos los nuevos datos, si la llave no es vacía entonces actualizamos. Ahora usamos el atributo **.cell** el cual recibe los parámetros fila y columna cuyos atributos ya conocemos y guardamos los cambios del archivo Excel, en caso que no se encuentre el imprime un mensaje de que no logro conseguir una tarea especifica con el **id**.



```
def actualizar(ruta:str, identificador:int,datos_actualizados:dict):
    Archivo_Excel = load_workbook(ruta)
    Hoja_datos = Archivo_Excel['Datos del crud']
    Hoja_datos=Hoja_datos['A2':'F'+str(Hoja_datos.max_row)]
    hoja=Archivo_Excel.active

    titulo=2
    descripcion=3
    estado=4
    fecha_inicio=5
    fecha_Finalizado=6
    encontro=False
    for i in Hoja_datos:
        if i[0].value==identificador:
            fila=i[0].row
            encontro=True
            for d in datos_actualizados:
                if d=='titulo' and not(datosActualizados[d]==''):
                    hoja.cell(row=fila, column=titulo).value=datosActualizados[d]
                elif d=='descripcion' and not(datosActualizados[d]==''):
                    hoja.cell(row=fila, column=descripcion).value=datosActualizados[d]
                elif d=='estado' and not(datosActualizados[d]==''):
                    hoja.cell(row=fila, column=estado).value=datosActualizados[d]
                elif d=='fecha inicio' and not(datosActualizados[d]==''):
                    hoja.cell(row=fila, column=fecha_inicio).value=datosActualizados[d]
                elif d=='fecha finalizacion' and not(datosActualizados[d]==''):
                    hoja.cell(row=fila, column=fecha_Finalizado).value=datosActualizados[d]
    Archivo_Excel.save(ruta)
    if encontro==False:
        print('Error: No existe una tarea con ese Id')
        print()
    return
```



Ahora vamos a escribir la función que recibe la dirección del archivo y un diccionario con los datos que deseamos agregar. Guardamos el archivo Excel, especificado con la ruta en una variable, seleccionamos la hoja del archivo Excel que queremos leer, extraemos la información de la hoja desde A2 hasta la máxima fila de F.

Del archivo Excel seleccionamos la primera hora, seleccionamos y guardamos los índices, recorremos el archivo y como recordamos que el **id** de las tareas es un entero, usamos **isinstance** para preguntar si es un valor entero, ya que si no es entero quiere decir que la fila esta libre y ahí agregaremos nuestra nueva tarea. Por ultimo usamos nuevamente el atributo **.row** para saber si la fila esta libre y procedemos agregar los datos.



```
def agregar(ruta:int, datos:dict):  
    Archivo_Excel = load_workbook(ruta)  
    Hoja_datos = Archivo_Excel['Datos del crud']  
    Hoja_datos=Hoja_datos['A2':'F'+str(Hoja_datos.max_row+1)]  
    hoja=Archivo_Excel.active  
  
    titulo=2  
    descripcion=3  
    estado=4  
    fecha_inicio=5  
    fecha_Finalizado=6  
    for i in Hoja_datos:  
  
        if not( isinstance(i[0].value, int)):  
            identificador=i[0].row  
            hoja.cell(row=identificador, column=1).value=identificador-1  
            hoja.cell(row=identificador, column=titulo).value=datos['titulo']  
            hoja.cell(row=identificador, column=descripcion).value=datos['descripcion']  
            hoja.cell(row=identificador, column=estado).value=datos['estado']  
            hoja.cell(row=identificador, column=fecha_inicio).value=datos['fecha inicio']  
            hoja.cell(row=identificador, column=fecha_Finalizado).value=datos['fecha finalizacion']  
            break  
    Archivo_Excel.save(ruta)  
    return
```



Después de tener lista la función anterior guardamos en el archivo Excel especificando con la ruta en una variable, seleccionamos la hoja del archivo Excel que queremos leer, extraemos la información

de la hoja desde A2 hasta la máxima fila de F del archivo Excel seleccionando la primera hora a diferencia de las listas cuyos índices comienzan en 0.

Usamos una variable booleana para saber si se encontró el **id** que especificamos, recorreremos la hoja de datos, revisando si el **id** que estamos leyendo es igual al que estamos buscando. Usamos el atributo **.row** nuevamente y colocamos la bandera de encontrado en **true**.

```
def borrar(ruta,identificador):
    Archivo_Exccel = load_workbook(ruta)
    Hoja_datos = Archivo_Exccel['Datos del crud']
    Hoja_datos=Hoja_datos['A2':'F'+str(Hoja_datos.max_row)]
    hoja=Archivo_Exccel.active

    titulo=2
    descripcion=3
    estado=4
    fecha_inicio=5
    fecha_Finalizado=6
    encontro=False
    for i in Hoja_datos:
        if i[0].value==identificador:
            fila=i[0].row
            encontro=True

            hoja.cell(row=fila, column=1).value=""
            hoja.cell(row=fila, column=titulo).value=""
            hoja.cell(row=fila, column=descripcion).value=""
            hoja.cell(row=fila, column=estado).value=""
            hoja.cell(row=fila, column=fecha_inicio).value=""
            hoja.cell(row=fila, column=fecha_Finalizado).value=""
    Archivo_Exccel.save(ruta)
    if encontro==False:
        print('Error: No existe una tarea con ese Id')
        print()
    return
```




```
rut="C:\\Users\\Usuarioprueba\\Desktop\\Ruta1\\Base crud.xlsx"

datosActualizados={'titulo':'', 'descripcion':'', 'estado':'', 'fecha inicio':'', 'fecha finalizacion':''}
while True:
    print('Indique la accion que desea realizar: ')
    print('Consultar: 1')
    print('Actualizar: 2')
    print('Crear nueva tarea: 3')
    print('Borrar: 4')
    accion = input('Escriba la opcion:')
```

Ahora solo vamos a realizar un ciclo infinito que nos servirá para consultar, modificar y actualizar la base de datos y con esto podemos hacer todo el procedimiento por consola.



```
if not(accion=='1') and not(accion=='2') and not(accion=='3') and not(accion=='4'):
    print('Comando invalido por favor eliga una opcion valida')
elif accion=='1':
    opc_consulta=''
    print('Indique la tarea que desea consultar: ')
    print('Todas las tareas: 1')
    print('En espera: 2')
    print('En ejecucion: 3')
    print('Por aprobar: 4')
    print('Finalizada: 5')
    opc_consulta = input('Escriba la tarea que dese consultar:')
    if opc_consulta=='1':
        print()
        print()
        print('** Consultando todas las tareas **')
        leer(rut,'todo')
    elif opc_consulta=='2':
        print()
        print()
        print('** Consultando tareas en espera **')
        leer(rut,'En espera')
    elif opc_consulta=='3':
        print()
        print()
        print('** Consultando tareas en ejecucion **')
        leer(rut,'En ejecucion')
    elif opc_consulta=='4':
        print()
        print()
        print('** Consultando tareas por aprobar **')
        leer(rut,'Por aprobar')

    elif opc_consulta=='5':
        print()
        print()
        print('** Consultando tareas finalizadas **')
        leer(rut,'Finalizada')
```




```
elif accion=='2':
    datosActualizados={'titulo':'', 'descripcion':'', 'estado':'', 'fecha inicio':'', 'fecha finalizacion':''}
    print(** Actualizar Tarea **)
    print()
    id_Actualizar=int(input('Indique el Id de la tarea que desea actualizar: '))
    print()
    print(** Nueva titulo **)
    print(**Nota: si no desea actualizar el titulo solo oprima ENTER**)
    datosActualizados['titulo']=input('Indique el nuevo titulo de la tarea : ')
    print()
    print(** Nueva descripcion **)
    print(**Nota: si no desea actualizar la descripcion solo oprima ENTER**)
    datosActualizados['descripcion']= input('Indique la nueva descripcion de la tarea : ')
    print()
    print(** Nueva estado **)
    print('En espera: 2')
    print('En ejecucion: 3')
    print('Por aprobar 4')
    print('Finalizada: 5')
```

Generamos la impresión en consola según las modificaciones que queramos hacer en nuestra base de datos, por esta razón debemos ser muy específicos, con la acción que queremos realizar y con cual id vamos a identificar cada tarea.



```
print('**Nota: si no desea actualizar el estado solo oprima ENTER')
estadoNuevo= input('Indique el nuevo estado de la tarea : ')
if estadoNuevo=='2':
    datosActualizados['estado']='En espera'
elif estadoNuevo=='3':
    datosActualizados['estado']='En ejecucion'
elif estadoNuevo=='4':
    datosActualizados['estado']='Por aprobar'
elif estadoNuevo=='5':
    now = datetime.now()
    datosActualizados['estado']='Finalizada'
    datosActualizados['fecha finalizacion']=str(now.day) + '/' + str(now.month) + '/' + str(now.year)

now = datetime.now()
datosActualizados['fecha inicio']=str(now.day) + '/' + str(now.month) + '/' + str(now.year)
actualizar(rut,id_Actualizar, datosActualizados)
print()
```




```
elif accion=='3':
    datosActualizados={'tarea':'', 'descripcion':'', 'estado':'', 'fecha inicio':'', 'fecha finalizacion':''}
    print('** Crear nueva Tarea **')

    print()
    print('** titulo **')
    print()
    datosActualizados['titulo']=input('Indique el titulo de la tarea : ')
    print()
    print('** descripcion **')
    datosActualizados['descripcion']= input('Indique la descripcion de la tarea : ')
    print()
    datosActualizados['estado']='En espera'
    now = datetime.now()
    datosActualizados['fecha inicio']=str(now.day) + '/' + str(now.month) + '/' + str(now.year)
    datosActualizados['fecha finalizacion']=''
    agregar(rut,datosActualizados)
elif accion=='4':
    print('')
    print('** Eliminar Tarea **')
    iden=int(input('Indique el Id de la tarea que desea eliminar: '))
    borrar(rut,iden)
```

En la próxima clase vamos a revisar paso por paso como realizar la impresión en consola según las **acciones** de nuestro código.





El futuro digital
es de todos

MinTIC

GRACIAS

OPERADO POR:

