

# Aplicación CRUD

Utilización de Contenedores

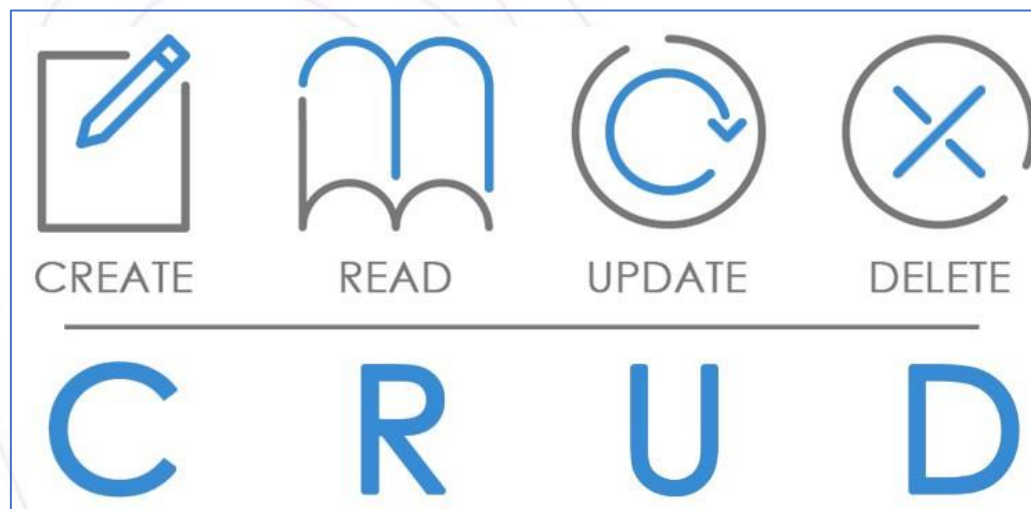




# Aplicación CRUD

- En informática, CRUD es el acrónimo de "Crear, Leer, Actualizar y Borrar" (del original en inglés: Create, Read, Update and Delete), que se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

<https://es.wikipedia.org/wiki/CRUD>





# Aplicación CRUD

- Se presenta una aplicación CRUD para manejar un listado de tareas, manejado en un diccionario de diccionarios como se presenta a continuación:

```
#Aplicación CRUD – Lista de Tareas Pendientes
#####

#Inicialización de tareas (desde el código)
tareas = {
    '01': {
        'descripcion': 'Ir a mercar',
        'estado': 'pendiente',
        'tiempo': 60
    },
    '02': {
        'descripcion': 'Estudiar',
        'estado': 'pendiente',
        'tiempo': 180
    },
    '03': {
        'descripcion': 'Hacer ejercicio',
        'estado': 'pendiente',
        'tiempo': 50
    }
}
```



# Aplicación CRUD

- Por consola se propone la siguiente interfaz:

```
-- Aplicación CRUD TareasPendientes ---  
1. Adicionar Tarea  
2. Consultar Tareas  
3. Actualizar Tarea  
4. Eliminar Tarea  
5. Salir  
Ingrese una opción: _
```





# CRUD: Create

- Adición de tareas al listado:

```
#Create
if opcion == 1:
    print()
    print("->Adicionando Tarea")

    identificador = str(input("Ingrese identificador de la Tarea: "))
    descripcion = str(input("Ingrese descripción de Tarea: "))
    estado = str(input("Ingrese el estado inicial de la Tarea: "))
    tiempo = int(input("Ingrese el tiempo de realización: "))

    tareaNueva = {
        'descripcion': descripcion,
        'estado' : estado,
        'tiempo' : tiempo
    }

    #Adicionar la tarea
    tareas = create(tareas, identificador, tareaNueva)
```



# CRUD: Create

- Especificación función create:

```
def create(tareas, identificador, tareaNueva):  
    tareas[identificador] = tareaNueva  
    return tareas
```



# CRUD: Create

- Interfaz:

```
-- Aplicación CRUD TareasPendientes ---  
1. Adicionar Tarea  
2. Consultar Tareas  
3. Actualizar Tarea  
4. Eliminar Tarea  
5. Salir  
Ingrese una opción: 1  
  
->Adicionando Tarea  
Ingrese identificador de la Tarea: 88  
Ingrese descripción de Tarea: Estudiar para MisiónTIC2022  
Ingrese el estado inicial de la Tarea: En proceso  
Ingrese el tiempo de realización: 24
```



# CRUD: Read

- Consulta de tareas del listado:

```
#Read
elif opcion == 2:
    print()
    print("->Listado de Tareas")
    print()

#Lectura de tareas
read(tareas)
```





# CRUD: Read

- Especificación función **read**:

```
def read(tareas):  
    for identificador, tarea in tareas.items():  
        print(identificador, '-', end='')  
        for nombre_atributo, atributo in tarea.items():  
            print(atributo, ', ', end='')  
        print()
```



# CRUD: Read

- Interfaz:

```
-- Aplicación CRUD TareasPendientes ---  
1. Adicionar Tarea  
2. Consultar Tareas  
3. Actualizar Tarea  
4. Eliminar Tarea  
5. Salir  
Ingrese una opción: 2  
  
->Listado de Tareas  
  
01  - Ir a merchar , pendiente , 60 ,  
02  - Estudiar , pendiente , 180 ,  
03  - Hacer ejercicio , pendiente , 50 ,  
88  - Estudiar para MisiónTIC2022 , En proceso , 24 ,
```



# CRUD: Update

- Actualización de tarea:

```
#Update
elif opcion == 3:
    print()
    print("-->Actualizar Tarea")
    print()

    #Solicitar al usuario el identificador
    identificador = str(input("Ingrese identificador de la Tarea para modificar: "))

    #Revisar si se encuentra el elemento solicitado
    if estaElemento(identificador, tareas):

        #Proceder a actualizar
        #-----

        #Modificar los campos de interés
        nuevaDescripcion = str(input('Nueva descripción: '))
        if nuevaDescripcion:
            tareas[identificador]['descripcion'] = nuevaDescripcion
        nuevoEstado = str(input('Nuevo estado: '))
        if nuevoEstado:
            tareas[identificador]['estado'] = nuevoEstado
        nuevoTiempo = input('Nuevo tiempo: ')
        if nuevoTiempo:
            nuevoTiempo = int(nuevoTiempo)
            tareas[identificador]['tiempo'] = nuevoTiempo

    else:
        print("No ha sido encontrada la Tarea!")
```



# CRUD: Update

- Especificación función **estaElemento**:

```
def estaElemento(identificador, tareas):  
    #Extraer de la base de datos (contenedor) los identificadores  
    conjuntoIdentificadores = set(tareas.keys())  
  
    #Revisar si se encuentra el elemento solicitado  
    if identificador in conjuntoIdentificadores:  
        return True  
    else:  
        return False
```





# CRUD: Update

- Interfaz:

```
-- Aplicación CRUD TareasPendientes ---  
1. Adicionar Tarea  
2. Consultar Tareas  
3. Actualizar Tarea  
4. Eliminar Tarea  
5. Salir  
Ingrese una opción: 3  
  
->Actualizar Tarea  
  
Ingrese identificador de la Tarea para modificar: 88  
Nueva descripción: Estudiar mucho Programación  
Nuevo estado: Procesando  
Nuevo tiempo: 100
```



# CRUD: Delete

- Eliminación de tarea:

```
#Delete
elif opcion == 4:
    print()
    print("->Eliminar Tarea")
    print()

    #Solicitar al usuario el identificador
    identificador = str(input("Ingrese identificador de la Tarea para eliminar: "))

    #Extraer de la base de datos (contenedor) los identificadores
    conjuntoIdentificadores = set(tareas.keys())

    #Revisar si se encuentra el elemento solicitado
    if identificador in conjuntoIdentificadores:

        #Proceder a eliminar
        #-----
        tareas.pop(identificador)

    else:
        print("No ha sido encontrada la Tarea para eliminar!")
```



# CRUD: Delete

- Interfaz:

```
-- Aplicación CRUD TareasPendientes ---  
1. Adicionar Tarea  
2. Consultar Tareas  
3. Actualizar Tarea  
4. Eliminar Tarea  
5. Salir  
Ingrese una opción: 4  
  
->Eliminar Tarea  
  
Ingrese identificador de la Tarea para eliminar: 01
```



# Aplicación CRUD

- En el siguiente repositorio encontrarás el código completo de la aplicación CRUD presentada:

[https://github.com/luismescobarf/clasesCiclo1/blob/master/AppCRUD\\_ParadigmaProceduralEstructurado.py](https://github.com/luismescobarf/clasesCiclo1/blob/master/AppCRUD_ParadigmaProceduralEstructurado.py)

