



Unidad de Computación  
Ingeniería en Computación  
Msc. Leonardo Víquez Acuña

Generación automática de CRUD's

Jeremy Live González 2015068507  
Ricardo Soto Araya 2020035358  
Junior Segura Ballesteros 2019390109

Sede San Carlos

09/26/2021

# Índice

Introducción	2
Análisis del problema	2
solución del problema	3
análisis de resultados	5
conclusiones	6
recomendaciones	7
referencias	7
anexos	7

# Introducción

El porqué se realiza el proyecto enfocado a la generación automática de CRUD's es por el motivo de hacer eficiente, flexible y económico en el tiempo que se invierte en hacer los procedimientos uno por uno. Se demuestra efectivamente que al elaborar procedimientos que se vayan generando automáticamente en donde estos van ligados a una tabla específica de una base de datos previamente seleccionada. Da una herramienta potente para lograr el alcance deseado de todo administrador de base de datos, que es disminuir el tiempo en que invierte para resolver un problema en específico. Esta misma solución brinda una ventaja para poder manipular la base de datos de forma más rápida con respecto a la generación del propio CRUD de las tablas.

## Análisis del problema

El proyecto desarrollado trata de darle el poder al usuario administrador de tener a la mano una herramienta donde puede conectarse a dos bases de datos, estas son SQL server y PostGreSQL. Seguidamente elige qué tabla va hacer con la que trabaje para generar el CRUD, en esta parte el administrador elige qué acción hacer. Puede generar procesos almacenados de crear, leer, actualizar o eliminar eficientemente con respecto a la tabla seleccionada anteriormente. Una situación planteada donde puede dar solución esta herramienta es en la agilidad del puesto de trabajo del administrador de base de datos, para consigo desarrollar procedimiento a tablas de alguna base de datos.

# Solución del problema

## *Etapas de la solución*

- Interfaz
- Conexiones a los motores de bases de datos
- Código que autogenera los CRUD's
- Conexión de python a los motores de bases de datos

Se lleva acabo la generación de los procesos en el schema por defecto. Esto para efectos prácticos.

En PostgreSQL se utiliza como schema "public" y en SQL Server se utiliza "customer\_services".

Se muestra el código donde está la generación de comandos ha ejecutarse para la generación de los debidos procesos del CRUD:

```

"""
Accion del segundo boton para obtener que eligio el usuario exactamente del CRUD y tabla.
"""
def submitact2():
    valor_combox2 = c2.get()
    combobox1 = c1.get()

    # Mostrar privilegios de la tabla en el schema especifico de su dicha db
    if combobox1 == 'POSTGRESQL':
        showPriTable()
    elif combobox1 == 'SQL SERVER':
        showPriTableSQL()

    # Obtengo arreglo con los comandos.
    if var1.get() == 1:
        if combobox1 == 'POSTGRESQL':
            command1 = "select public.genInsProc('public',"
            command1 += valor_combox2
            command1 += "');"
            comandos.append(command1)
        elif combobox1 == 'SQL SERVER':
            command1 = "select dbo.insertCrud ('customer_services',"
            command1 += valor_combox2
            command1 += "');"
            comandos.append(command1)

    if var2.get() == 1:
        if combobox1 == 'POSTGRESQL':
            command2 = "select public.genSelect('public',"
            command2 += valor_combox2
            command2 += "');"
            comandos.append(command2)
        elif combobox1 == 'SQL SERVER':
            command1 = "select dbo.selectCrud ('customer_services',"
            command1 += valor_combox2
            command1 += "');"
            comandos.append(command1)

```

```

if var3 (variable) combobox1: str
    if combobox1 == 'POSTGRESQL':
        command3 = "select public.genUpdate('public','"
        command3 += valor_combox2
        command3 += "');"
        comandos.append(command3)
    elif combobox1 == 'SQL SERVER':
        command1 = "select dbo.updateCrud ('customer_services','"
        command1 += valor_combox2
        command1 += "');"
        comandos.append(command1)

if var4.get() == 1:
    if combobox1 == 'POSTGRESQL':
        command4 = "select public.genDelete('public','"
        command4 += valor_combox2
        command4 += "');"
        comandos.append(command4)
    elif combobox1 == 'SQL SERVER':
        command1 = "select dbo.deleteCrud ('customer_services','"
        command1 += valor_combox2
        command1 += "');"
        comandos.append(command1)

# Boton para procesar mostrar comandos y los ejecuta
submitbtn = tk.Button(root, text="Run",
                        bg='green', command = getComandos)
submitbtn.place(x = 280, y = 170, width = 55)

# Boton para procesar mostrar comandos
submitbtn = tk.Button(root, text="Gen",
                        bg='orange', command = commandGenShow)
submitbtn.place(x = 345, y = 170, width = 55)

```

# Análisis de resultados

<i>Tarea/Requerimiento</i>	<i>Estado</i>	<i>Observaciones</i>
<i>Conexión a servidores</i>	<i>Completo</i>	<i>El algoritmo realiza la adecuada conexión</i>
<i>Listado y selección de componentes</i>	<i>Completo</i>	<i>La estructura está bien planteada para llegar a una solventura agradable.</i>
<i>Generación de código</i>	<i>Completo</i>	<i>Fue exitosa, comentada y bien gestionado hasta el final.</i>
<i>Interfaz</i>	<i>Completo</i>	<i>Flexible al manejo en la desenvoltura de la navegación.</i>

## Conclusiones

En el presente proyecto se pudo aprender acerca de las conexiones a sql y postgres a una aplicación externa mediante uso de librerías. Escogimos python por su sencillez a la hora de crear interfaz y también por la basta información para realizar conexiones a diferentes interfaces.

Se contribuye de gran manera en la solución de una herramienta capaz de facilitar la vida de los administradores de base de datos. Esto da confort en que lo programado tiene mucha utilizada para un trabajo profesional específico.

## Recomendaciones

Esperamos que este proyecto sea de gran ayuda a los programadores para que puedan generar los CRUD's automaticos y asi automaticen los procesos de creación de CRUD's en bases de datos PostgreSQL y SQL.

El proyecto se puede complementar con investigaciones en bases de datos para la generación de CRUD's, lo cuales permiten al desarrollador agilizar procesos en las bases de datos.

## Referencias

S. (s. f.). *Microsoft SQL documentation - SQL Server*. Microsoft Docs.

Recuperado 27 de septiembre de 2021, de

<https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>

*PostgreSQL: Documentation*. (s. f.). The PostgreSQL Global Development Group. Recuperado 27 de septiembre de 2021, de <https://www.postgresql.org/docs/>



3.9.7 *Documentation*. (s. f.). python. Recuperado 27 de septiembre de 2021, de <https://docs.python.org/3/>

## Anexos

Video en español

<https://youtu.be/Beza5Lgf2SM>

Video en inglés

<https://youtu.be/ocZ0f7CXSaY>