



**Universidad  
Católica  
de Cuenca**

UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE LA  
COMPUTACIÓN, E INNOVACIÓN TECNOLÓGICA CARRERA DE SOFTWARE

AVANCE DEL PROYECTO FINAL

**ESTUDIANTES:**

ANGEL CEDILLO  
ROBINSON BARBECHO  
DANNY SOLANO  
EMILIO OCHOA

**MATERIA:**

INVESTIGACIÓN DE OPERACIONES

**DOCENTE:**

PABLO BUESTAN

**CICLO:**

CUARTO CICLO

**FECHA:**

07/07/2023

**Pyqt5:**

Es un binding de la biblioteca gráfica Qt para el lenguaje de programación Python.

Se usa para crear interfaces gráficas con Python de una forma más sencilla ya que tiene una buena legibilidad en el código Python.

**SQLite:**

Es un software libre que en dispositivos empujados nos permite almacenar información de forma rápida y eficaz.

Es fácil de usar y utiliza sintaxis SQL, sigue el estándar pero con pequeñas modificaciones.

algunas de las características de SQLite son:

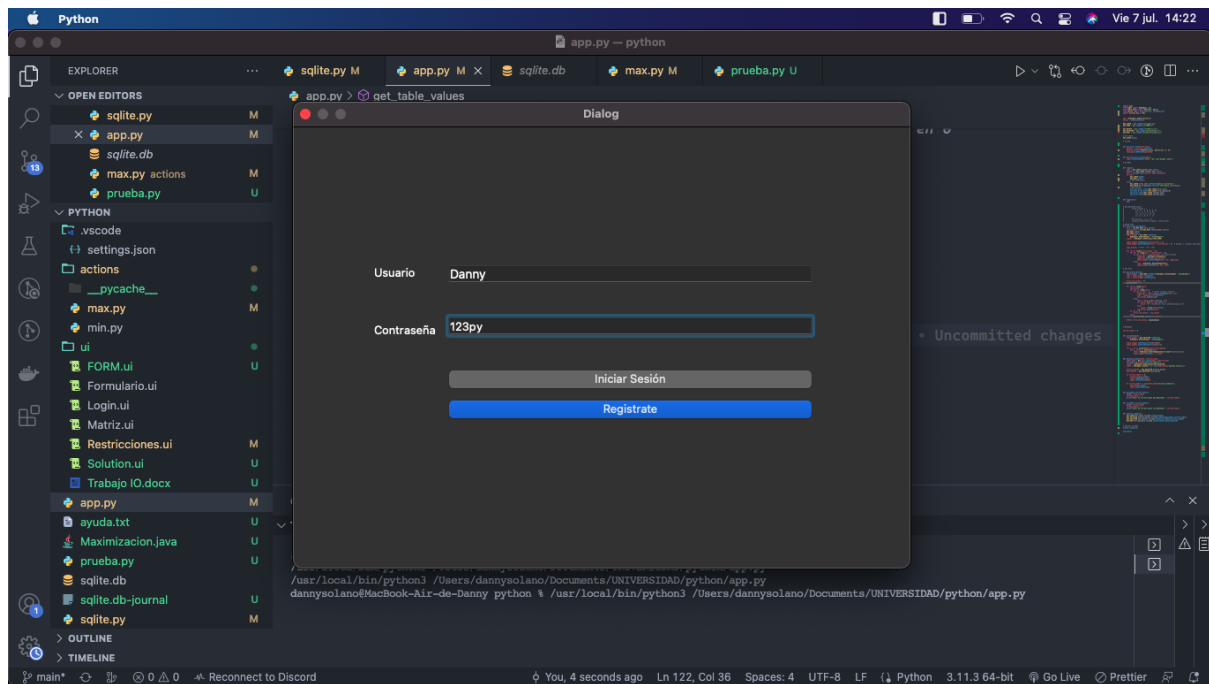
- \* Gestor DB “de fábrica” de Android.
- \* Basado en archivos y fácil de configurar y usar.
- \* Portátil
- \* Fácil de usar.

**Qt designer:**

Es un programa para realizar interfaces gráficas de un usuario. Qt designer utiliza el lenguaje C ++ y es utilizado en un entorno de escritorio.

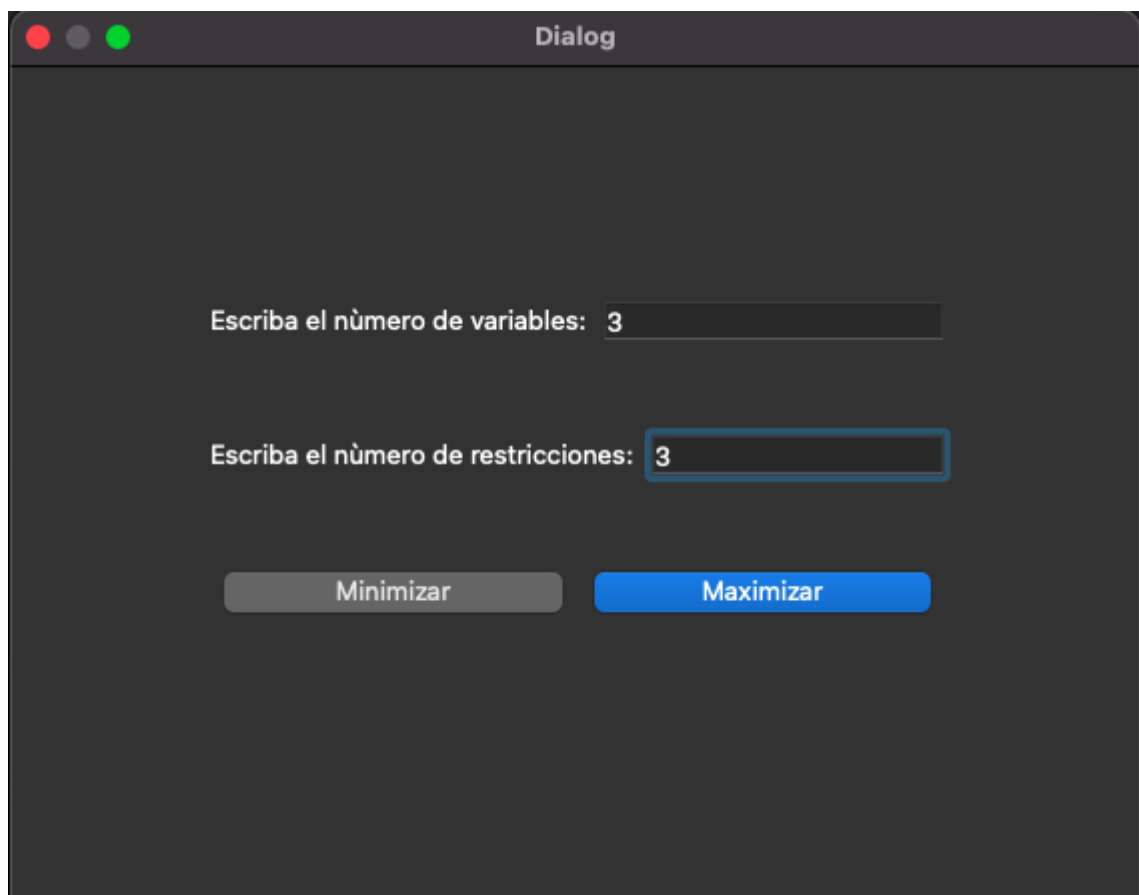
La ventaja de qt designer es que tiene la facilidad de utilizar el mismo código para diferentes plataformas integradas ya que QT utiliza un modelo de licencia dual.

**Características del Programa:**



El programa cuenta con conexión a una base de datos relacional llamada SQLITE, que permite hacer operaciones como cualquier base de datos que use sql, con la diferencia de que la base de datos se guarda en un archivo.

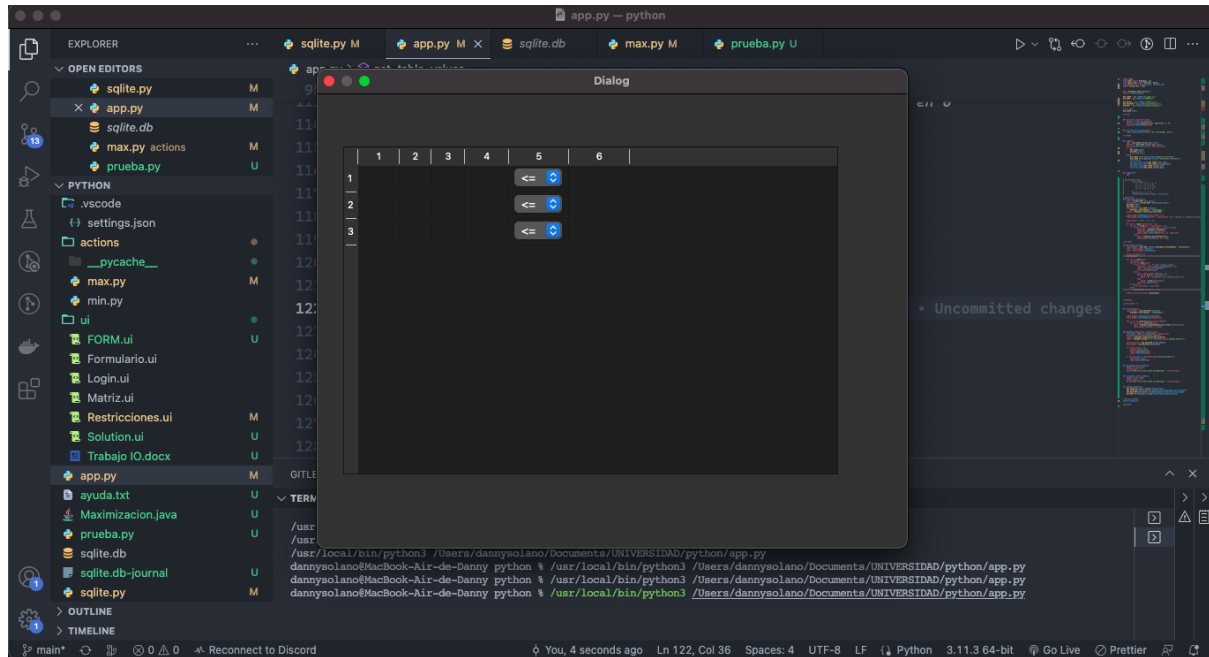
El programa puede registrar y validar a los usuarios que intenten acceder al programa



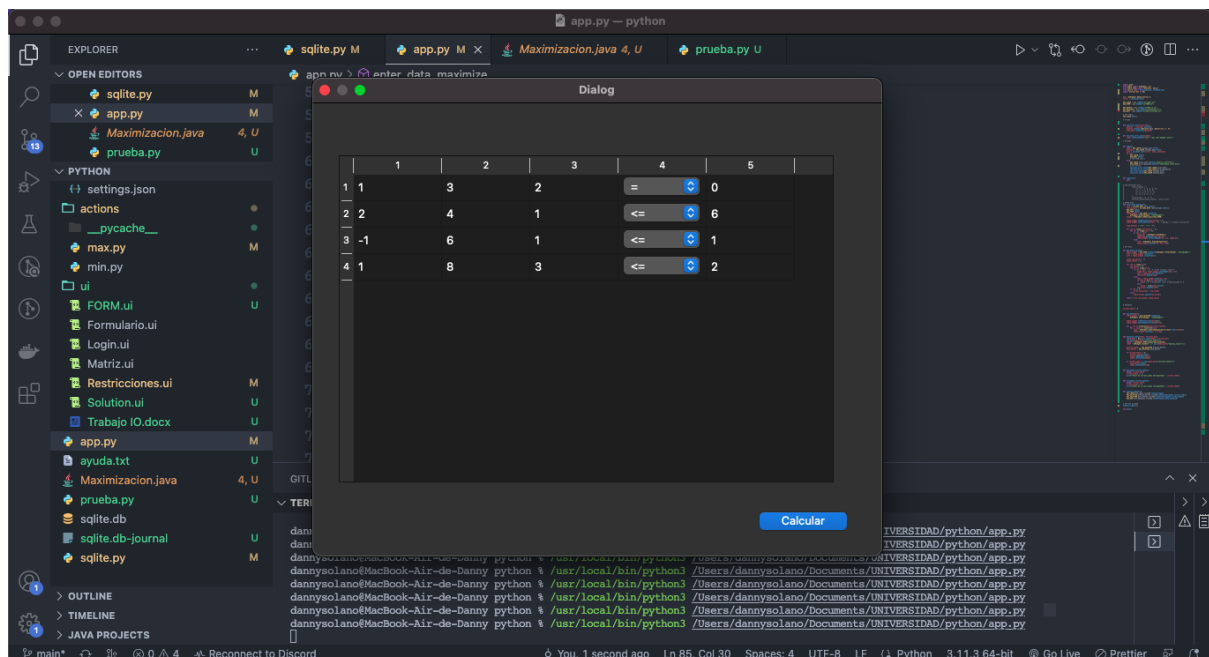
Una vez iniciado sesión te pide que ingreses el número de variables y restricciones que tiene el ejercicio y también qué tipo de optimización queremos realizar (maximización o minimización)

Por el momento solo está habilitada la opción de maximización

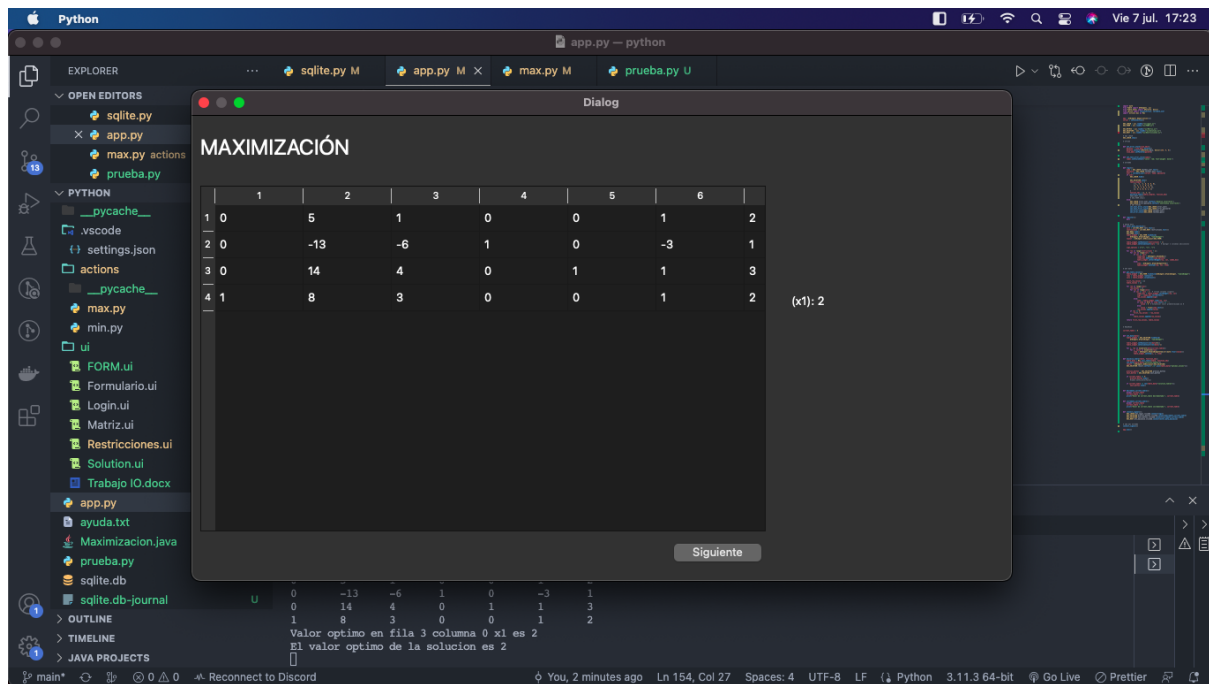
Una vez de click en maximizar



Aparecerá esta ventana en donde podremos ingresar la función objetivo y las restricciones de la forma  $(x_1 + x_2 + x_3 = R)$



Una vez da click a calcular realiza el cálculo del resultado óptimo (por el momento solo muestra la tabla final y su resultado)



El cálculo de la minimización y maximización con tablas Simplex del programa fue hecho en primer lugar en JAVA, dado al conocimiento de este lenguaje, es una alternativa sencilla para empezar a crear el algoritmo que será luego traducido a Python. En esta etapa del trabajo, el algoritmo es capaz de realizar problemas de minimización o maximización y de mostrar en el compilador las tablas una por una hasta el resultado final y el resultado de la función objetivo.

Se usó esta tabla como referencia para ir creando y asegurándose el correcto funcionamiento del resto de algoritmos:

```
public static void main(String[] args) {
    //Tabla que debe contener los datos ingresados
    float tablaSimplex[][] = {
        {-1, -3, -2, 0, 0, 0, 0},
        {2, 4, 1, 1, 0, 0, 6},
        {-1, 6, 1, 0, 1, 0, 1},
        {1, 8, 3, 0, 0, 1, 2}
    };
}
```

En el algoritmo, se tienen los procesos que se realizan normalmente en el método Simplex, como encontrar la fila y columna pivote, el elemento pivote, la simplificación de la fila, la resta con otras filas y la verificación de números negativos.

Además, se tiene ya la visualización en el compilador de las iteraciones de la tabla, junto con valores que nos ayudan a ver el correcto funcionamiento de esta:

Iteracion #0

-1.0	-3.0	-2.0	0.0	0.0	0.0	0.0
2.0	4.0	1.0	1.0	0.0	0.0	6.0
-1.0	6.0	1.0	0.0	1.0	0.0	1.0
1.0	8.0	3.0	0.0	0.0	1.0	2.0

Aun hay negativos

La columna pivote tiene el numero -3.0 posici n 2

La fila pivote tiene el numero 6.0 posicion 3

El numero para dividir es: 6.0

-0.16666667 1.0 0.16666667 0.0 0.16666667 0.0 0.16666667

Iteracion #3

0.6000001	0.0	0.0	0.0	-0.7	0.9	1.0999999
2.2	0.0	0.0	1.0	-0.40000015	-0.19999993	5.2000003
-0.4	1.0	0.0	0.0	0.3	-0.099999994	0.100000024
1.4000001	0.0	1.0	0.0	-0.8	0.6	0.39999995

Iteracion #4

-0.3333332	2.3333333	0.0	0.0	0.0	0.6666666	1.3333333
1.6666665	1.3333338	0.0	1.0	0.0	-0.3333333	5.3333335
-1.3333333	3.3333333	0.0	0.0	1.0	-0.3333333	0.3333334
0.3333335	2.6666667	1.0	0.0	0.0	0.33333337	0.6666667

Iteracion #5

0.0	4.9999976	0.9999991	0.0	0.0	0.9999997	1.9999993
0.0	-11.999992	-4.999997	1.0	0.0	-1.9999993	2.000002
0.0	13.999994	3.9999979	0.0	1.0	0.99999946	2.9999988
1.0	7.9999967	2.9999986	0.0	0.0	0.99999964	1.9999992

Valor optimo en fila 3 columna 0 x1 es 1.9999992

El valor optimo de la solucion es 1.9999992