

Plan de tesis

Villanueva Portella, Jhon Gesell

25/06/2018

0.1 Introducción

Un software hidro-sedimentario ayudar a que las personas puedan tomar mejores decisiones en la gestión de los recursos hídricos y así a la vez evitar que en épocas de máximas avenidas ocurran desastres humanos, con esta herramienta lo que se busca contribuir a la comunidad científica haciendo posible su acceso a todo el mundo.

En el pasado ya se ha contado con herramientas privativas llevando a malas practicas como lo es la piratería informática o en otros casos con programas no muy intuitivos para los usuarios finales.

El método utilizado es de sobre posicin pasicin de valores en una matrz raz que representa los valores de posicin en las coordenadas x, y y una tercera componente que es de de la velocidad.

El software que se propone esta a la altura de otras herramientas con posibilidades abiertas para más adelante, será alojada en una plataforma web y de esta manera hacer computación en la nube con sistemas embebidos, por ahora buscamos hacer computación en un localhost que brinde posibilidades de acceso desde centros de estudio, centros de investigación o incluso para el sector privado; las herramientas que muchas veces podemos encontrar en el mercado tienen un costo que no permite acceder a ellas y por ser de consideración a los países en vías de desarrollo nuestra propuesta pretende beneficiar a ellos bajo la licencia GPL (General Public License).

0.1.1 Ojetivo General

- Crear un software hidrosedimentario con interfaz gráfica de usuario.
- Brindar un producto que ayude a los científicos e ingenieros que trabajan con fluídos geofísicos.
- Entregar un producto open-source para la comunidad científica internacional.

0.1.2 Objetivo Específico

- Leer los archivos ASCII que nos brinda el equipo ADCP (Acoustic Doppler Current Profiler).
- Crear un formulario para insertar los datos de las muestras se sedimentos.
- Crear una base de datos.
- Graficar la seccin del río para visualizar las cotas de fondo y los valores de sedimentos en suspensión.
- Crear

0.1.3 Hipótesis

El software debe ser calibrado con el modelo HidroMESAD y ofrecer valores cercanos al 5% de este.

0.2 Materiales y métodos

1. Materiales

- (a) Archivos de cota de fondo y velocidad:
Los datos son importados desde el ordenador del usuario.
- (b) Datos ingresados por el usuario:
Es insertar en el software la tabla de muestras procesadas en laboratorio.
- (c) Framework de desarrollo de aplicaciones con interfaz gráfica de usuario:
Utilizamos PyQt5, entorno que nos permite trabajar con el lenguaje Python haciendo uso de múltiples clases.
- (d) Lenguaje de programación Python 2.x y 3.x:
Es un lenguaje open-source multiplataforma y multipropósito que viene creciendo en diferentes campos, hoy muy usado en el campo científico, tiene una gran comunidad en todo el mundo, tiene una sintaxis fácil y limpia.
- (e) SQLite:
Es un facilitador de base de datos liviano y robusto multiplataforma teniendo alcance hasta para aplicaciones móviles.
- (f) Scrum:
Es un marco metodológico ágil hoy muy usado en el desarrollo de software y también va calando en otros rubros, se trata de hacer iteraciones (pivotar), trabajar a partir de un prototipo funcional y a partir de esto seguir avanzando, en caso pare el proyecto siempre haya algo que mostrar.
- (g) LaTeX:
Es una elaboración de textos orientados a la creación de documentos muy usando en el ámbito científico, para el caso de la presente tesis fue usado para la redacción de la misma y para el manual de usuario.
- (h) VIM:
Vim es un editor de texto plano mejorado que nos permite navegar desde la terminal del sistema operativo GNU/Linux y Windows.

2. Métodos

- (a) Lectura de archivos brindados por el usuario.
- (b) Insertar datos por el usuario.
- (c) Almacenamiento en la base de datos.
- (d) Generación de una matriz.
- (e) Interpolación de puntos.
- (f) Gráfica de la sección del río.
- (g) Gráfica de las celdas de aforo.
- (h) Cálculo del caudal.

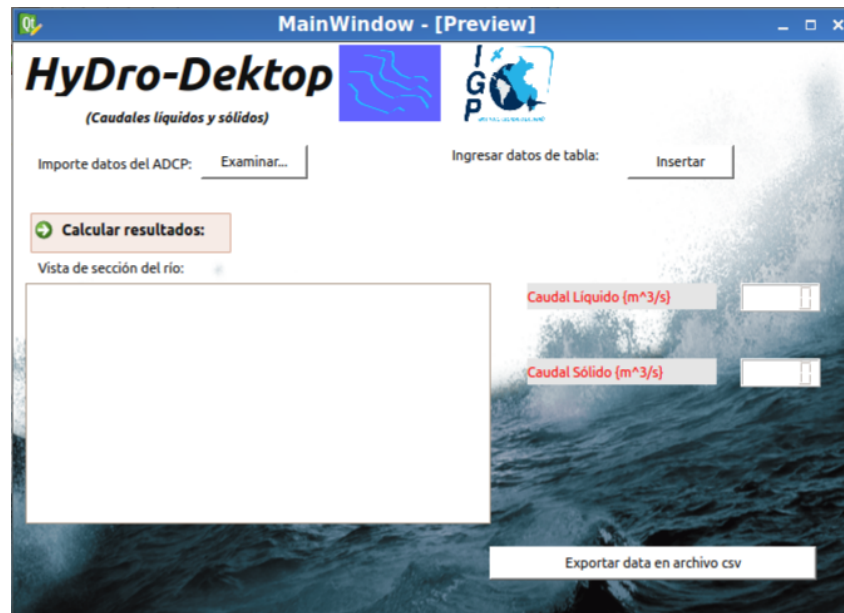


Figure 1: Vista construída con Qt Designer

0.3 Resultados esperados

1. Vista preliminar de la interfaz gráfica de usuario
Para ello vea a la Figura 1
2. Base de datos Se ha elaborado unos scripts que permiten crear la tabla; llenar, actualizar y eliminar los registros que constituyen la base de datos. Todo esto se logr desde el lenguaje Python2.x, los códigos se visualizan en los Anexos 01, 02 y 03 respectivamente.

Para ello vea a la Figura 2

0.4 Cronograma

Meses	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
R.I.	x	-	-	-	-	-	-
L.B.	x	x	-	-	-	-	-
C.B.D.	x	-	-	-	-	-	-
C.F.	x	-	-	-	-	-	-
I.F.B.D.	-	x	-	-	-	-	-
I.R.	-	-	x	x	-	-	-
A.M.	-	-	-	-	x	-	-
C.M.	-	-	-	-	-	x	-
M.U.	-	-	-	-	-	-	x

- Leyenda:

– R.I.: Recolección de información.

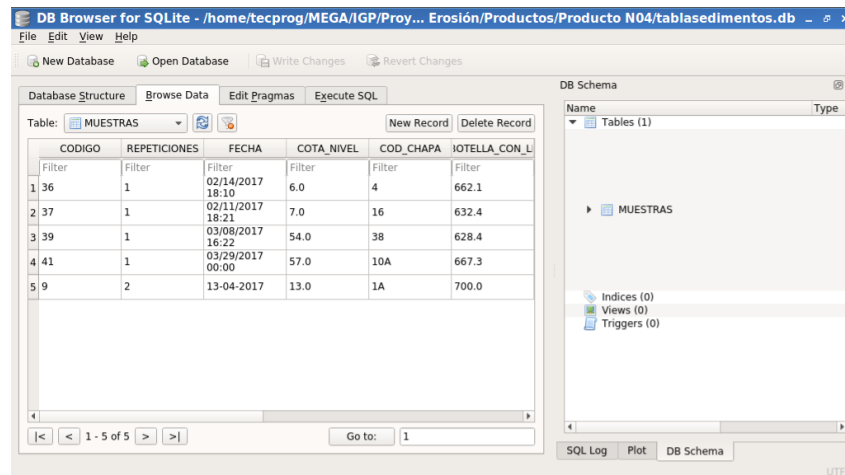


Figure 2: Tabla ingresada por el usuario desde la terminal

- L.B.: Lectura de la bibliografía.
- C.B.D.: Creación de la base de datos.
- C.F.: Creación del formulario.
- I.F.B.D.: Implementación funcional de la base de datos.
- I.R.: Impresión de resultados en la GUI.
- A.M.: Ajustes del modelo.
- C.M.: Corrección del modelo.
- M.U.: Manual de usuario.

0.5 Discusiones

La calibración del modelo se logra a través de evaluar resultados que nos brinda el software HydroMesad con nuestro producto, el caudal conseguido líquido y sólido deben ser cercanos al 5%.

0.6 Conclusiones

- La librería Tkinter que viene para el estandar de Python tiene una curva de aprendizaje muy suave.
- Al trabajar con el Qt Designer logramos obtener una GUI no responsive, para que el software se acomode a cualquier dimensión de trabajo se sugiere hacer toda la construcción del programa desde el módulo Qt Creator.
- Se necesita convertir los archivos *.ui que nos brinda el Qt Designer a archivos *.py para que se puedan ejecutar desde la terminal.

0.7 Bibliografía

- Gonzáles, R. (s.f.). *Python para todos*. Recuperado de: <http://mundogeek.net/tutorial-python>
- Coutinho, N. (2016). *Introducción a la programación con Python: Algoritmos y lógica de programación para principiantes*, Brasil: Novatec Editora Ltda.
- Harwani, B. (2012). *Introduction to Python Programming and Developing GUI Applications with PyQt*, Estados Unidos: Course Technology PTR .

0.8 Anexos

0.8.1 Base de datos

1. Creación de la tabla.

CrearTablaSedimentos.py

```
#!/usr/bin/python
# -*- coding: cp1251 -*-

# Crear base de datos y tabla con sqlite3
# 20/06/2018

__autor__ = u"Jhon Gesell"

import sqlite3

conexion = sqlite3.connect('tablasedimentos.db')
cursor = conexion.cursor()

#Crear tabla
cursor.execute('''CREATE TABLE MUESTRAS
(CODIGO TEXT NOT NULL,
REPETICIONES TEXT NOT NULL,
FECHA TEXT NOT NULL,
COTA_NIVEL REAL NOT NULL,
COD_CHAPA TEXT NOT NULL,
PESO_BOTELLA_CON_LIQUIDO REAL NOT NULL,
PESO_BOTELLA REAL NOT NULL,
VOLUMEN REAL NOT NULL,
FINOS_PESO_INICIAL_FILTRO REAL NOT NULL,
FINOS_PESO_FINAL_FILTRO REAL NOT NULL,
GRUESOS_PESO_INICIAL_FILTRO REAL NOT NULL,
GRUESOS_PESO_FINAL_FILTRO REAL NOT NULL,
ESTACION TEXT NOT NULL)''')

conexion.close()
```

2. Inserción de registros en los campos.

datossedimetos.py

```
#!/usr/bin/python
```

```
# -*- coding: cp1252 -*-
```

```
# Insertar datos de una tabla con sqlite3
```

```
# 21/06/2018
```

```
__autor__ = u"Jhon Gesell"
```

```
import sqlite3
```

```
codigo = raw_input("Codigo: ")
```

```
repeticiones = raw_input("Repeticiones: ")
```

```
fecha = raw_input("Fecha: ")
```

```
cota_nivel = raw_input("Cota nivel: ")
```

```
cod_chapa = raw_input("Codigo de botella: ")
```

```
peso_botella_con_liquido = raw_input("Peso botella con liquido: ")
```

```
peso_botella = raw_input("Peso botella: ")
```

```
volumen = raw_input("Volumen: ")
```

```
finos_peso_inicial_filtro = raw_input("Peso inicial filtro fino: ")
```

```
finos_peso_final_filtro = raw_input("Peso final filtro fino: ")
```

```
gruesos_peso_inicial_filtro = raw_input("Peso inicial filtro grueso: ")
```

```
gruesos_peso_final_filtro = raw_input("Peso final filtro grueso: ")
```

```
estacion = raw_input("Estacion: ")
```

```
conexion = sqlite3.connect('tablasedimentos.db')
```

```
cursor = conexion.cursor()
```

```
#Insertar datos en la tabla
```

```
cursor.execute('''INSERT INTO MUESTRAS(CODIGO, REPETICIONES, FECHA, COTA_NIVEL, COD_CHAPA, PESO_BOTELLA_CON_LIQUIDO, PESO_BOTELLA, VOLUMEN, FINOS_PESO_INICIAL_FILTRO, FINOS_PESO_FINAL_FILTRO, GRUESOS_PESO_INICIAL_FILTRO, GRUESOS_PESO_FINAL_FILTRO, ESTACION)
```

```
VALUES ('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')
''' %(codigo, repeticiones, fecha, cota_nivel, cod_chapa, peso_botella_con_liquido, peso_botella,
```

```

        volumen, finos_peso_inicial_filtro ,
        finos_peso_final_filtro ,
        gruesos_peso_inicial_filtro ,
        gruesos_peso_final_filtro , estacion))

conexion.commit()
conexion.close()

```

3. Actualización de registros en los campos.

ActualizarSedimentos.py

```

#!/usr/bin/python
#-*- coding: cp1252 -*-

# Actualizar datos en una tabla con sqlite3
# 23/06/2018

__autor__ = u"Jhon Gesell"

import sqlite3

#codigo = raw_input("Codigo: ")
#repeticiones = raw_input("Repeticiones: ")
#fecha = raw_input("Fecha: ")
#cota_nivel = raw_input("Cota nivel: ")
#cod_chapa = raw_input("Codigo de botella: ")
#peso_botella_con_liquido = raw_input("Peso botella con liquido: ")
#peso_botella = raw_input("Peso botella: ")
#volumen = raw_input("Volumen: ")
#finos_peso_inicial_filtro = raw_input("Peso inicial filtro fino: ")
#finos_peso_final_filtro = raw_input("Peso final filtro fino: ")
#gruesos_peso_inicial_filtro = raw_input("Peso inicial filtro grueso: ")
#gruesos_peso_final_filtro = raw_input("Peso final filtro grueso: ")
#estacion = raw_input("Estacion: ")

conexion = sqlite3.connect('tablasedimentos.db')
cursor = conexion.cursor()

#Actualizar datos en la tabla
cursor.execute("UPDATE MUESTRAS SET VOLUMEN=:volumen
WHERE FECHA=:fecha and COD_CHAPA=:cod_chapa",
              {"volumen": volumen, "fecha": fecha,

```



```
        "cod_chapa": cod_chapa}))
```

```
conexion.commit()
```

4. Eliminación de registros por campos.

vaciartablasedimentos.py

```
#!/usr/bin/python
```

```
#-*- coding: cp1252 -*-
```

```
# Borrar datos en una tabla con sqlite3
```

```
--autor-- = u"Jhon Gesell"
```

```
import sqlite3
```

```
codigo = raw_input("Codigo: ")
```

```
#repeticiones = raw_input("Repeticiones: ")
```

```
#fecha = raw_input("Fecha: ")
```

```
#cota_nivel = raw_input("Cota nivel: ")
```

```
#cod_chapa = raw_input("Codigo de botella: ")
```

```
#peso_botella_con_liquido = raw_input("Peso botella con liquido: ")
```

```
#peso_botella = raw_input("Peso botella: ")
```

```
#volumen = raw_input("Volumen: ")
```

```
#finos_peso_inicial_filtro = raw_input("Peso inicial filtro fino: ")
```

```
#finos_peso_final_filtro = raw_input("Peso final filtro fino: ")
```

```
#gruesos_peso_inicial_filtro = raw_input("Peso inicial filtro grueso: ")
```

```
#gruesos_peso_final_filtro = raw_input("Peso final filtro grueso: ")
```

```
#estacion = raw_input("Estacion: ")
```

```
conexion = sqlite3.connect('tablasedimentos.db')
```

```
cursor = conexion.cursor()
```

```
#Eliminar datos en la tabla
```

```
cursor.execute("DELETE FROM MUESTRAS WHERE CODIGO = %s" %codigo)
```

```
conexion.commit()
```

```
conexion.close()
```