Documentación Técnica del Software SpotyUN

Esteban Rodríguez Muñoz, Paula Valentina Barrero, Santiago Pereira Plata y Andrés Yesid Bejarano. Estudiantes de la facultad de ingeniería de la UNAL.

**Resumen**

SpotyUN Interfaz es un programa que simula una interfaz de la base de datos para mostrar el contenido de la tabla canciones.

1. **LENGUAJE Y LIBRERÍAS USADAS**

Este proyecto fue realizado en su totalidad basado en Python, y el uso del entorno grafico QtDesigner.

importamos SQLite3 con el fin de tener un mejor manejo de las bases de datos en el proyecto por su portabilidad.

Importamos Q tableWidgetItem

from PyQt5.QtWidgets import QTableWidgetItem que provee la vista de tablas basadas en ítems en los cuales vamos a colocar los datos extraídos de la base de datos

QtDesigner ayudo a crear una plantilla de interfaz grafica con la cual podemos relacionar a las funciones del programa en Python y esta se pantallaproyectofinal.py

**Importante:** Para el correcto funcionamiento del programa se debe tener instalado PyQt5

Se puede instalar por medio del siguiente link:

https://www.riverbankcomputing.com/software/pyqt/download5

por medio del siguiente comando se instala por consola cmd: “pip install PyQt5”

1. **CAMBIOS Y FUNCIONES DE CADA VERSIÓN**
   1. (FECHA): SpotyUN cuenta con las funciones de agregar los siguientes datos de canciones: código, nombre, genero, álbum e intérprete y almacenarlas en una base de datos.

Para esta versión en particular del documento ya se usaron conocimientos más propios de la POO tales como el uso de las clases para separar y agrupar por categorías las funciones como mostraremos a continuación.

Como el requerimiento principal fue mostrar la tabla canciones, se suprimieron las demás tablas en esta versión.

También es importantísimo considerar que esta modificación cambia los requerimientos del código y agrega requisitos para no entorpecer su funcionalidad tales como:

Para instanciar la clase se necesitas dos elementos fundamentales como el “\_\_init\_\_” y el “self”, los cuales cumplen la función de adicionar el argumento a la lista de atributos. Ya en la parte práctica, se debe colocar self.atributo antes de cambiar/asignar el nombre de dicho atributo en particular.

Todos los anteriores datos que fueron ingresados pueden ser también modificados o eliminados a decisión del usuario o administrador.

1. **ANOTACIONES DENTRO DEL CÓDIGO**

El Código cuenta con anotaciones dentro del mismo que facilitarán la navegación e identificación de diferentes funciones a cualquier desarrollador que deba llevar a cabo alguna función de actualización, mantenimiento o mejora del programa lo que facilitará el desarrollo del mismo a los diferentes responsables posteriormente.

1. **CLASES Y FUNCIONES USADAS**

# Pantallaproyectofinal

# Se utilizo un QTabWidget para generar la pantalla principal de la interfaz grafica, se eligió por que tiene la facilidad de añadir pestañas que serán usadas para cada función de la tabla canción.

# En la primera pestaña llamada canciones se añadió una tabla con QTableWidget la cual usaremos para visualizar la tabla canciones.

# Tiene un botón de recargar que consultaría de nuevo los datos para actualizar los posibles cambios.

# En la segunda pestaña llamada agregar canciones tenemos un label y un QlineEdit para ingresar cada uno de los campos de una nueva canción a la base de datos.

# Con un botón de agregar se llevan todos los campos a la base de datos.

# En la tercera pestaña llamada ModificarCanciones tenemos un QLineEdit para ingresar el código de la canción y también los campos que vamos a editarle.

# En la Cuarta pestaña podemos consultar la base de datos ingresando el código de la canción.

# Main:

* **Class main:** En esta clase están todos los métodos usados para manipular las tablas generales de la interfaz grafica y que se use correctamente las funciones sql.

Se setean comandos ui que sirven para poder usar las tablas generadas en la interfaz grafica y se cargan los datos de canciones en datos total.

Se conectan los botones “refrescar,actualizar,buscar y agregar” con las correspondientes funciones sql..

Se definen los tamaños de las tablas: tabla\_canciones y tabla\_buscar.

* MostrarTabla: Esta función obtiene los datos de datos\_total y con un for va llenando la tabla canciones con las canciones registradas actualmente.

se usa el comando setSortingEnabled para poder ordenar las tablas de mayor a menor o viceversa dándole click a la parte superior del campo

* InsertarCanciones: Función que carga los valores agregados en los **QLineEdit,** los envia a la función AñadirCanciones y luego limpia los campos usados.
* ModificadorCanciones: obtiene el id de la canción ingresada en el QLineEdit y los busca en la base de datos, si existe entonces actualiza los campos con los ingresados, muestra varios mensajes de error si llega a fallar la actualización o si no existe el campo.
* ConsultorCanciones: Función que recibe el nombre de la canción y consulta en la base de datos, muestra los campos asociados a ese nombre de canción.

1. **ARCHIVOS DEL PROYECTO**

**FuncionesSQL:** En este archivo se encuentran todas las funciones relacionadas con el almacenamiento de los datos ingresados por el usuario y el administrador a la base de datos, para esto se importa sqlite3.

**Main.py:** Este es el archivo principal del programa, en el llama a cada una de las funciones que controlan la interfaz grafica

**labasededatospooparcial.db:** Este archivo contiene la base de datos estructurada y organizada en tablas, basado en SQLite3

**Pantallaproyectofinal.py**: Este archivo es la versión py del archivo generado por el designer y que se ejecuta cuando es llamada por el main. Contiene toda la información relevante para generar la interfaz grafica.