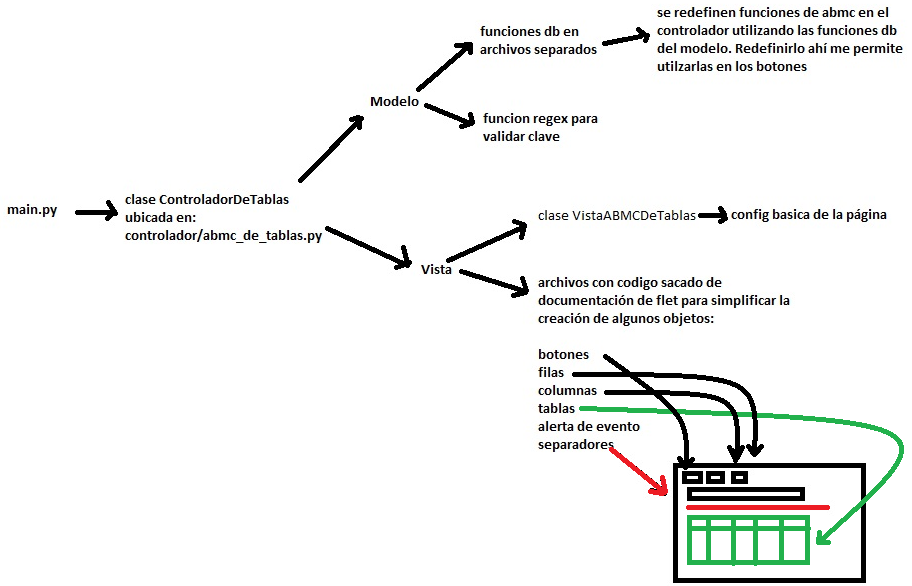
TP Python Nivel Inicial - Federico dos Santos

# Esquema Modelo/Vista/Controlador



## Controlador

En el programa [main.py](http://main.py) se instancia un controlador ABMC de tablas.

El controlador tiene como atributos, la conexión a la base de datos, que se crea al momento de instanciar el objeto, nombre por defecto, pero también parametrizable, al igual que la tabla que pretende controlar el controlador que se puede configurar tanto el nombre como los campos.

También tiene un objeto en sus atributos que es el alerta\_evento, que es una instancia de una clase de Flet para poder generar popups.

Además puse como atributo el registro modificable que inicializa vacío pero después en el método inicializar() le instancio la cantidad de celdas modificables que necesito según los campos que tengo en la tabla.

Tanto **alerta\_evento** como **registro** los puse por la facilidad de modificarlos desde casi cualquier lado con un self.atributo. Sobre todo porque en alta,baja, modificacion cuando las redefini en el método inicializar() tuve que acceder a ambos atributos y se me hizo más fácil así.

Luego en el método inicializar redefino las funciones del modelo de la base de datos para poder pasarlas como atributos a los botones y que manejen los eventos onclick. En estas nuevas funciones les paso los parámetros que necesitan las funciones del modelo.

Además en el modelo en el método inicializar() instancio los objetos de la vista, de flet.

## Modelo

El controlador interactua con funciones de bases de datos de SQLITE3. Utilicé las que vimos en clase, con unas pequeñas modificaciones para que funcionen de manera genérica, ya que mi objetivo es que luego se pueda interactuar con otras tablas, no solo con una.

En un principio había utilizado manejo de errores en todas las funciones de base de datos, pero luego me di cuenta que las alertas de evento debía manejarlas en el controlador para poder hacer uso de la interfaz gráfica para notificar si hubo un problema. Entonces de algunas funciones de base de datos tuve que quitar el manejo de errores y hacerlo desde la función alta/baja/modificación definidas en el controlador.

Luego vi que si intentaba agregar un registro con id que ya existe, el manejo de excepciones me permite informar que no se puede, pero con baja y modificación tuve que implementar una validación con una consulta a la base de datos para saber si el id a modificar o borrar no existe y poder informarlo.

La función regex que creé es simplemente para validar la clave (id) de cada tabla. Usé las siguiente expresión regular: [0-9]{1,10}

La misma es para validar que sea un número entero de al menos 1 dígito y hasta menos de 10 dígitos porque googleando vi que el máximo de un integer es 2.147.483.647 que tiene 10 dígitos pero no los aprovecha hasta el 9.999.999.999.

## Vista

Para la vista utilicé una clase VistaABMCDeTablas donde defino la configuración básica de la página: título y la posibilidad de scrollear.

Luego viendo documentación de flet creé algunas funciones para abstraer las clases de flet en funciones definidas por mi que me permitan crear algunos objetos:

ejemplo:

* **generar\_boton()** abstrae la creación de un objeto con la clase flet.OutlinedButton()
* **generar\_columnas()** abstrae la creación de un objeto con la clase flet.Column()
* **generar\_filas()** abstrae la creación de un objeto con la clase flet.Row()
* generar\_tabla() copié código de [*https://github.com/flet-dev/examples/blob/main/python/apps/controls-gallery/examples/layout/datatable/01\_basic\_datatable.py*](https://github.com/flet-dev/examples/blob/main/python/apps/controls-gallery/examples/layout/datatable/01_basic_datatable.py)pero lo hice más genérico y parametrizable.
* **generar\_separador()** copia un código largo que está en la documentación [*https://github.com/flet-dev/examples/blob/main/python/apps/controls-gallery/examples/layout/divider/02\_draggable\_divider.py*](https://github.com/flet-dev/examples/blob/main/python/apps/controls-gallery/examples/layout/divider/02_draggable_divider.py)con el agregado de que yo le paso por parámetro unos botones (parte superior) y una tabla (parte inferior) y me arma un objeto con una separación entre parte superior y parte inferior.
* **generar\_alerta()** copia código de acá [*https://github.com/flet-dev/examples/blob/main/python/apps/controls-gallery/examples/dialogs/alertdialog/01\_basic\_and\_modal\_dialogs.py*](https://github.com/flet-dev/examples/blob/main/python/apps/controls-gallery/examples/dialogs/alertdialog/01_basic_and_modal_dialogs.py)para generar un objeto alerta como popup

Luego parte de la vista está en el controlador que crea los objetos con las abstracciones que hice en la carpeta vista.