



|  |
| --- |
|  |

**Manual Técnico**

**Proyecto:**

**Administrador de Alumnos – González Dojo**

**Materia**

**María Sánchez de Thompson (ETN3)**

**Proyecto I**

**Contenido**

**PRESENTACIÓN 1**

**CONTENIDO 2**

**1** **Objetivos 3**

**2** **Alcance 3**

**3** **Requerimientos técnicos 3**

**4** **Herramientas utilizadas 4**

**5** **Desarrollo 4**

**6** **Resumen 6**

**1 Objetivo**

Esta aplicación fue creada por el equipo de desarrollo de **Gonzalez Dojo Capital** con la finalidad de facilitar a los instructores de la institución **Karate Do Miyazato** el control y seguimiento de los alumnos pertenecientes a sus Dojos.

El objetivo principal de este desarrollo es aportar una herramienta tecnológica e innovadora para utilizar en la organización y la estructuración de los *Dojos pertenecientes* a la institución **Miyazato.**

El aborde de esta aplicación a la estructura organizativa de los Dojos generará una optimización de la tarea administrativa…

**2 Alcance**

La aplicación Administrador de Alumnos (Miyazato) es dirigida exclusivamente a los Dojos pertenecientes a la Institución **Miyazato**

**Todos los derechos reservados**

**3 Requerimientos técnicos**

**Software**

* Sistema operativo Windows +7 / Linux Ubuntu
* Python 3.+

**Hardware**

* Una computadora completa, esto incluye mouse, teclado, cpu, monitor, etc
* Conexión a internet

**4 Herramientas utilizadas**

* Se utilizó Python 3.9 en toda la elaboración del proyecto
* Base de datos en la nube (host db4free)
* Photoshop creación y edición de imágenes utilizadas

**5 Desarrollo**

La aplicación consta de cuadro módulos:

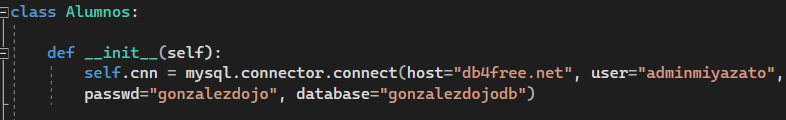
* Alumno.py
* Login.py
* CRUD.py
* **Main.py (execute)**

Dentro de estos módulos, se desarrollan todas las funcionalidades del producto.

**Alumno**

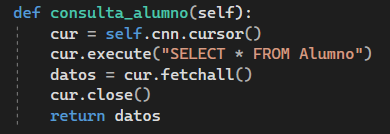
El modulo alumno(class) contiene todas las funcionalidades relacionadas con la conexión con la base de datos.

**\_\_init\_\_**



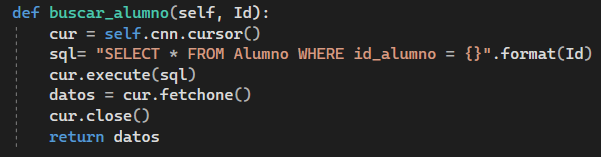
Se crea la conexión a la base de datos “gonzalezdojodb” que está alojada en los servidores de **db4free.**

**consulta\_alumno**



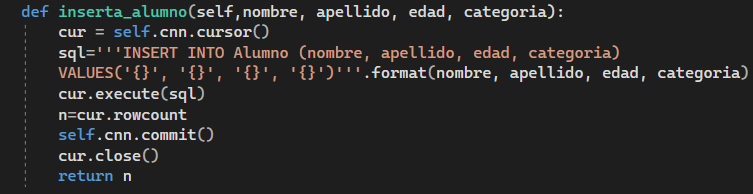
La función consulta alumno se encarga de seleccionar todos los datos de la tabla Alumno para mostrarlos en pantalla para luego poder realizar las demás funcionalidades.

**buscar\_alumno**



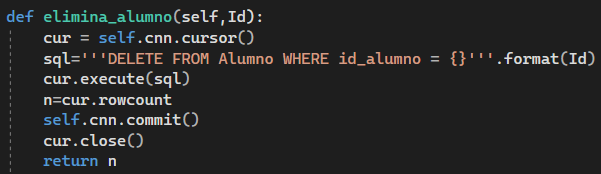
Se llama a esta función cuando se requiere traer un alumno concreto

**Inserta\_alumno**



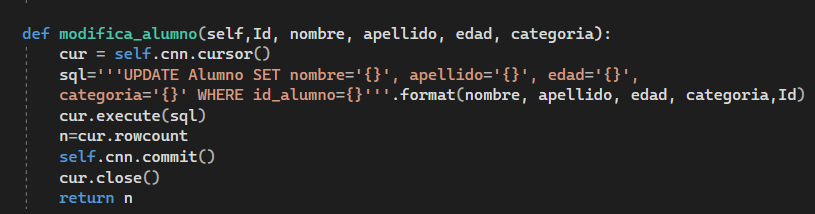
Inserta\_alumno inserta los datos escritos por el usuario a la tabla Alumno

**elimina\_alumno**



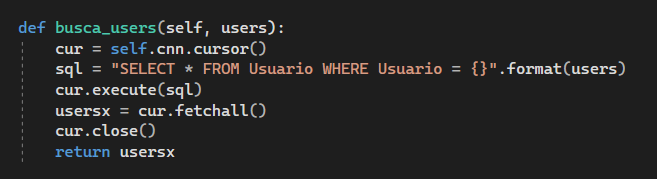
Esta función se encarga de eliminar un registro de la base de datos que el usuario desee.

**modifica\_alumno**



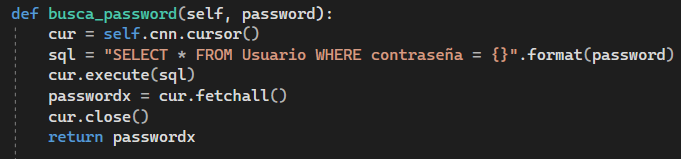
Permite modificar los campos de un registro previamente existente

**busca\_users**



Busca en la tabla Usuario el usuario escrito en el panel Login

**busca\_password**



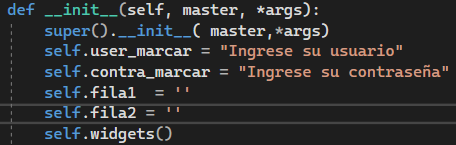
Busca en la tabla Usuario el password escrito en el panel Login

Estas son todas las funcionalidades que posee el modulo Alumno.py

**Login**

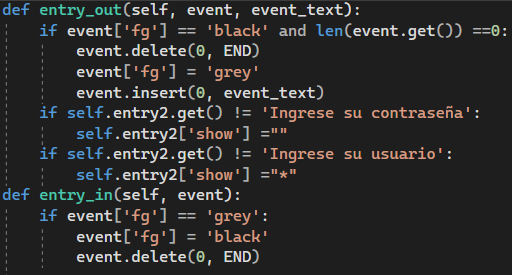
En el módulo(class) Login.py encontramos la primera interfaz de la aplicación, y las funcionalidades relacionadas con el ingreso a la misma.

**\_\_init\_\_**



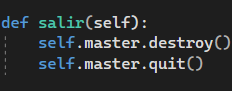
Se inicializan los atributos y las funciones con las que arranca la aplicación

**entry\_out / entry\_in**



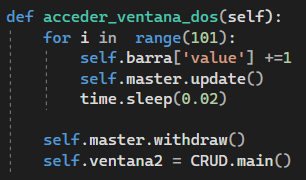
Se administra el text que tienen los campos de texto antes de ser escritos por el usuario

**salir**



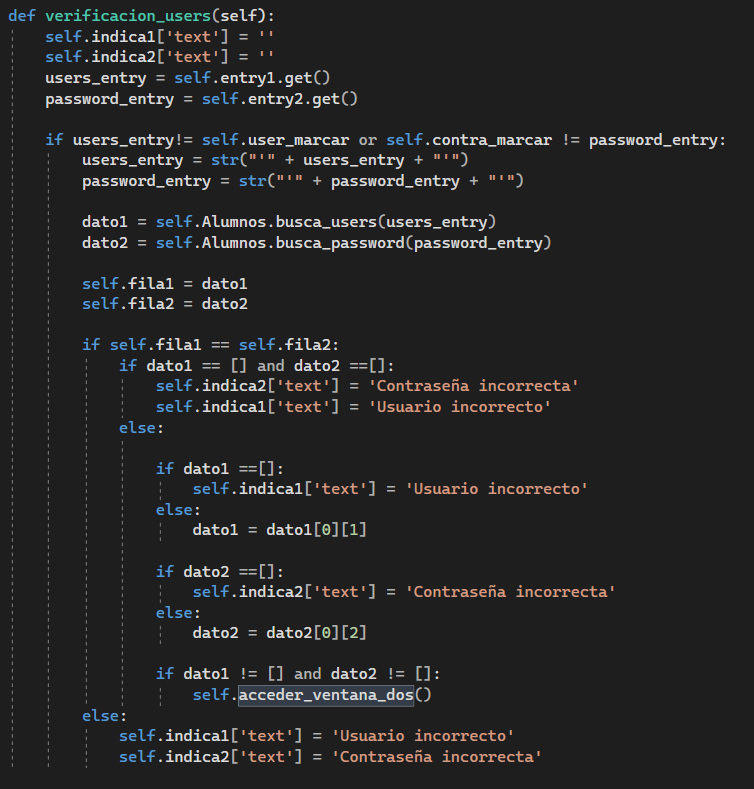
Se destruye la ventana

**acceder\_ventana\_dos**



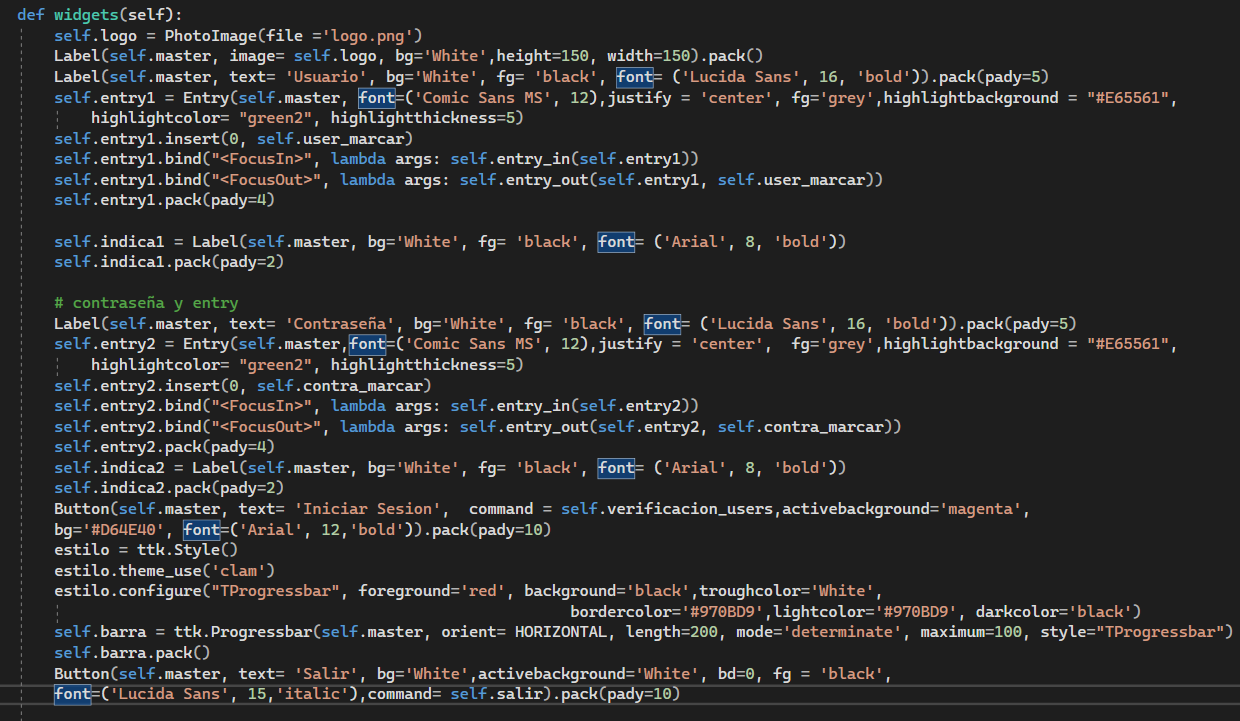
Se crea la animación de la barra de espera y se accede a la segunda ventana

**Verificación\_users**



Función encargada de dar acceso a la aplicación, llama a las funciones del módulo Alumno.py para realizar las consultas a la base de datos

**Widgets**



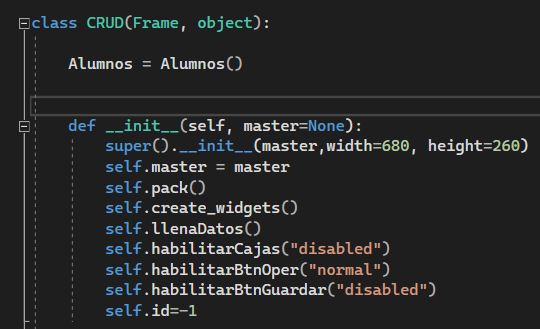
Creación de la interfaz gráfica utilizando tkinter (frame)

Estas son todas las funcionalidades que posee el modulo Login.py

**CRUD**

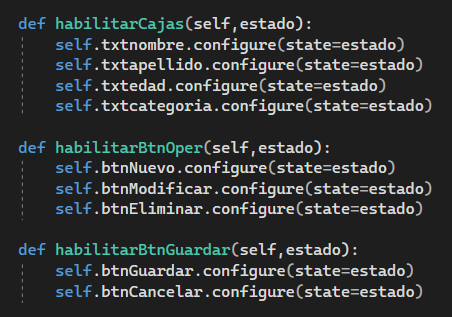
En el módulo(class) CRUD.py se realizan todas las funcionalidades primarias del desarrollo (Eliminar, Insertar, etc), utilizando las funciones creadas en el módulo Alumno.py para conectar con la base de datos

**\_\_init\_\_**



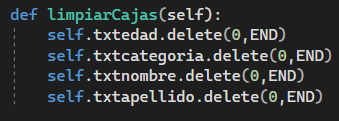
Se inicializan los atributos y las funciones que utilizará el modulo

**Funciones habilitadoras**



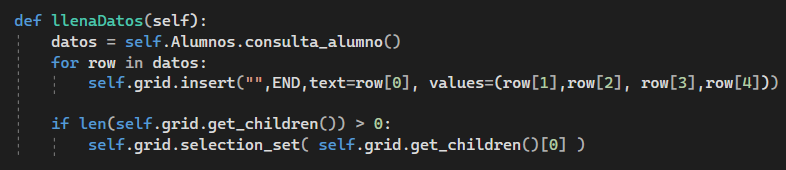
Estas funciones establecen cuando un widget está apto para ser usado, esto es parte del control de errores de la aplicación.

**LimpiarCajas**



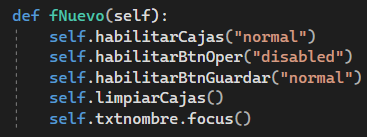
Limpia las cajas de texto

**LlenaDatos**



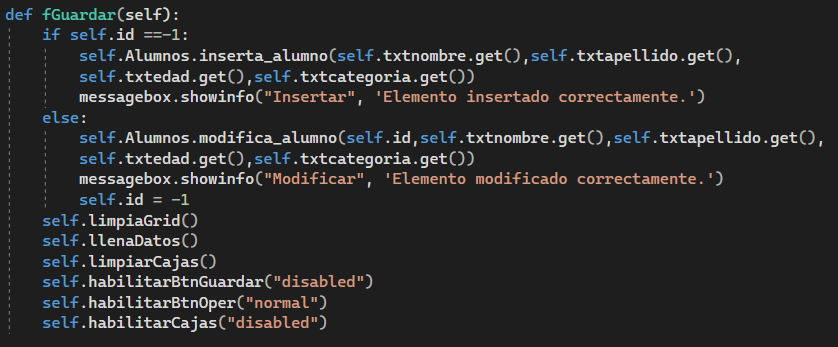
Extrae de una fila de la tabla Alumno los campos para su posterior uso

**fNuevo**



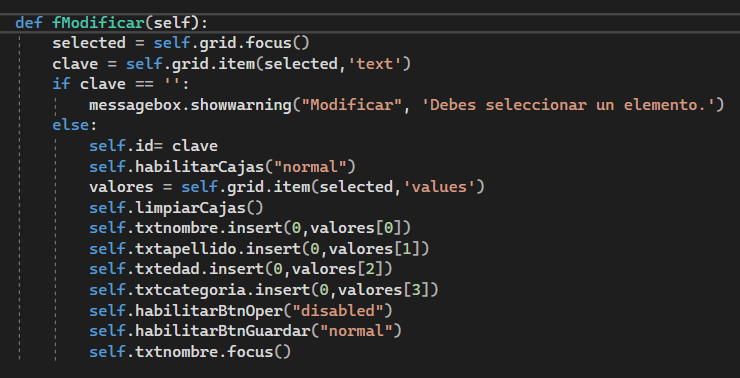
Prepara los campos y los widgets para un nuevo entry en la DDB

**fGuardar**



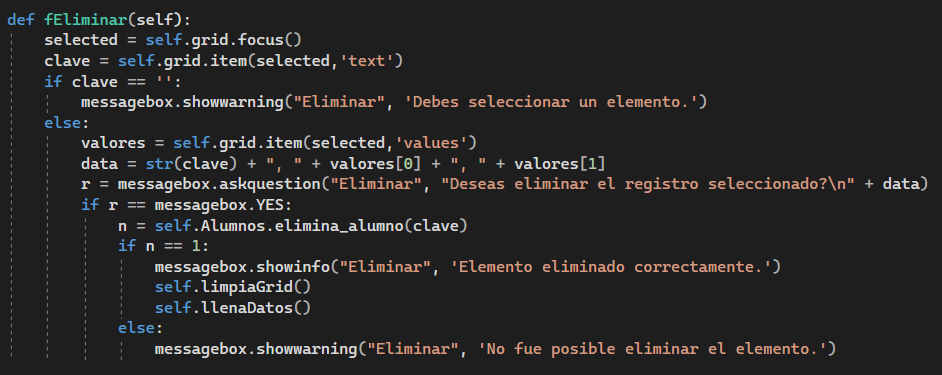
Función encargada de guardar los datos en la tabla Alumno

**fModificar**



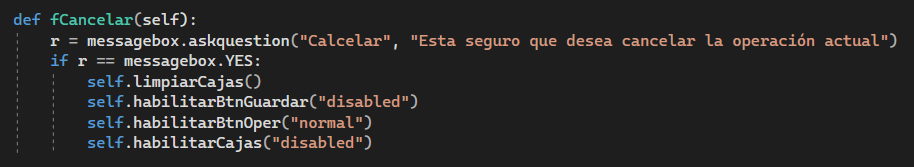
Función encarga de realizar las modificaciones a entry previamente realizados en la base de datos

**fEliminar**



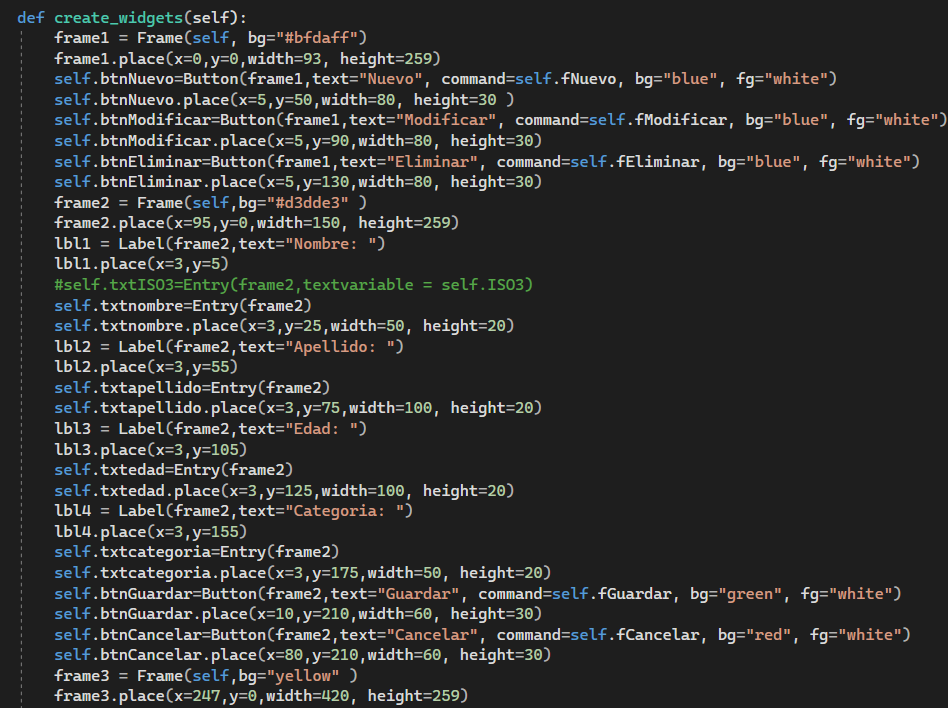
Elimina un entry en la tabla Alumnos

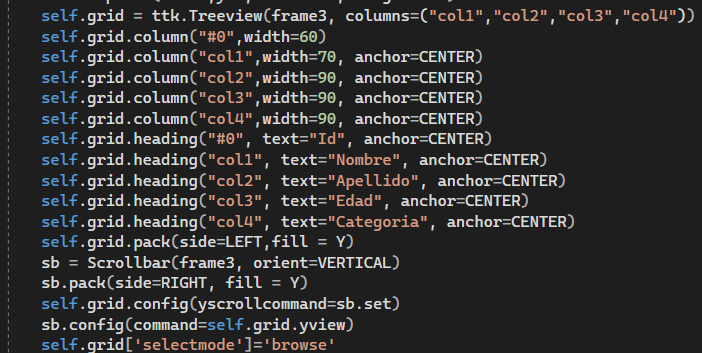
**fCancelar**



Cancela cualquier proceso para volver al estado original

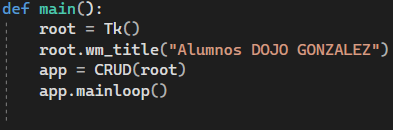
**create\_widgets**





Crea y da forma a todos los elementos gráficos de la aplicación (tkinter)

**main**

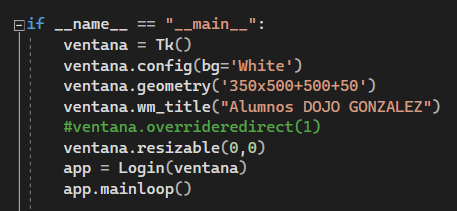


Inicializa la ventana

Estas son todas las funcionalidades que posee el modulo Alumno.py

**MAIN**

**El modulo main.py inicia el programa**

****

Crea la ventana y inicia el modulo Login.py

**6 Resumen**

En este documento se desarrollaron los objetivos, requerimientos técnicos, el alcance y el mismo código sobre la aplicación **Administrador de Alumnos MIYAZATO**

La aplicación posee varias funcionalidades divididas en cuatro módulos que construyen un sistema de administración de alumnos enlazado con base de datos en la nube.



**CONTACTO**

TELÉFONO: 11 2688-8209

CORREO: karatedojogonzalez@gmail.com

REDES SOCIALES: instagram.com/gonzalezdojocapital/