**Actividad:**

**Gestión de datos de estudiantes universitarios**

|  |  |
| --- | --- |
| Asignatura: | Big Data |
| Integrantes: | Ignacio Arancibia  Eduardo Silva |
| Profesor: | Edio Mardones |
|  |  |

**Índice General**

[1. Creación de colecciones 2](#_Toc176493961)

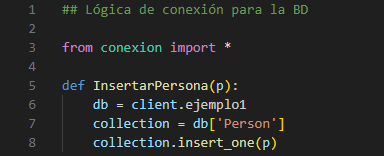
[2. Creación de funciones 7](#_Toc176493962)

[3. Menú o “main” 17](#_Toc176493963)

[4. Conexión a mongodb 20](#_Toc176493964)

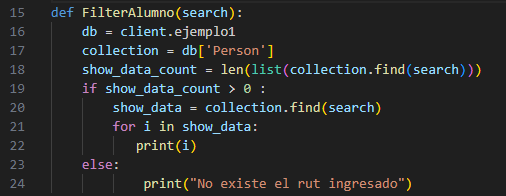
# Creación de colecciones

Comenzamos con el archivo “models.py” en el cual definimos la base de datos más sus colecciones, así definiendo los metadatos de nuestro proyecto.



Se define la primera función la cual cumple la finalidad de crear la base de datos si es que esta no existe y queda almacenada en el cluster. Para este proyecto “ejemplo1” es el nombre la base de datos.

Luego está definida la creación de la colección “Person” la cual está insertada en un documento.

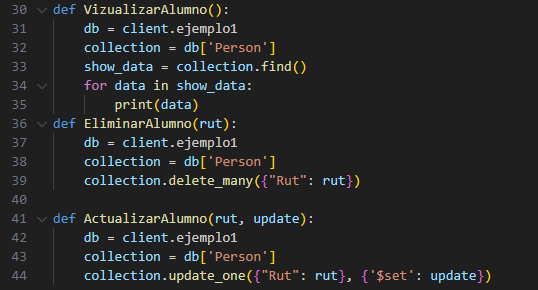


Se crea la función para filtrar alumnos dependiendo de su rut.

Para esto comenzamos conectando con la base de datos y luego entrando a la colección “Person”.

Luego “find” realiza una búsqueda en la colección utilizando el criterio “search”.

Se agrega la condición para verificar la existencia de los resultados y se itera sobre los resultados para imprimir cada documento.



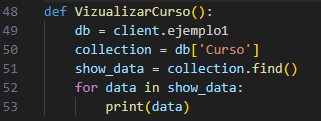
Finalmente, para la colección de los alumnos tenemos las funciones para visualizar, eliminar y actualizar los datos de sus alumnos.

Cada función sigue la estructura de, primero, conectarse a la base de datos, luego a la colección “person” y finalmente cumplir sus funciones asignadas a su respectivo nombre.

“find”, Encuentra al alumno solicitado.

“delete\_many”, Elimina alumnos.

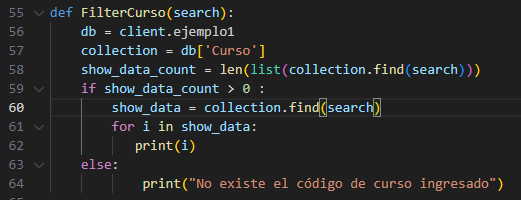
“update\_one”, Actualiza datos de alumnos.



Para comenzar con la colección de los cursos tenemos la primera función que nos permite visualizar cada curso.

La primera variable se define para conectarse a la base de datos llamada “ejemplo1” que luego entra a la colección que tiene por nombre “Curso”

Luego tenemos definida la variable para encontrar la colección e imprimir los datos de la misma.

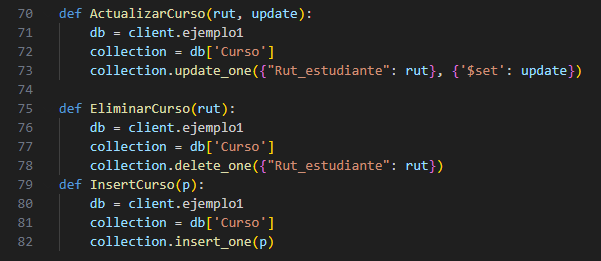


Seguidamente tenemos la función para filtrar cursos dependiendo de los requisitos por parte del usuario.

Como en cada función, se define primero la conexión a la base de datos.

Luego existe un “find” que realiza una búsqueda en la colección usando el criterio “search”. Se agrega la condición para verificar la existencia de los resultados solicitados.

Finalmente sobre los resultados para imprimir cada documento.



Para finalizar con la colección “Curso”, existen las funciones para actualizar, eliminar e insertar cursos.

Las tres funciones siguen la misma lógica de: Conectarse a la base de datos, entrar a la colección y cumplir la función indicada respecto a su nombre.

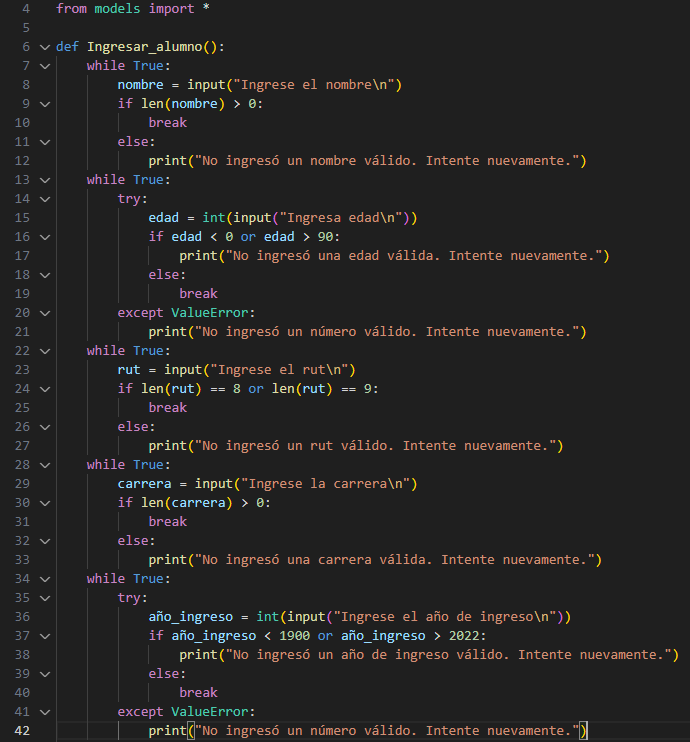
“update\_one”, Actualiza los cursos.

“delete\_one”, Eliminar cursos.

“insert\_one”, Crea cursos.

# Creación de funciones

Continuamos con el archivo “funciones.py”



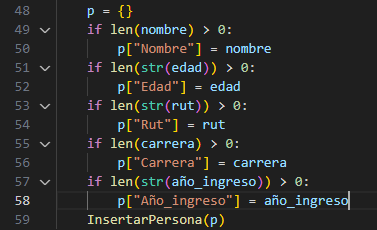
Se define la función para ingresar alumnos la cual solicita los datos de las personas.

Lo primero que se solicita es el nombre del alumno indicado con su propio “else” para evitar valores erróneos o inválidos.

Seguimos con la edad del alumno la cual no debe ser menor a 0 y mayor a 90, también acompañada de sus propias clausulas para indicar el error en el ingreso de datos.

Luego tenemos el ingreso del rut que está definido para ser de mínimo 8 números y máximo de 9, si se ingresa un valor más corto o más largo pasa al mensaje de error.

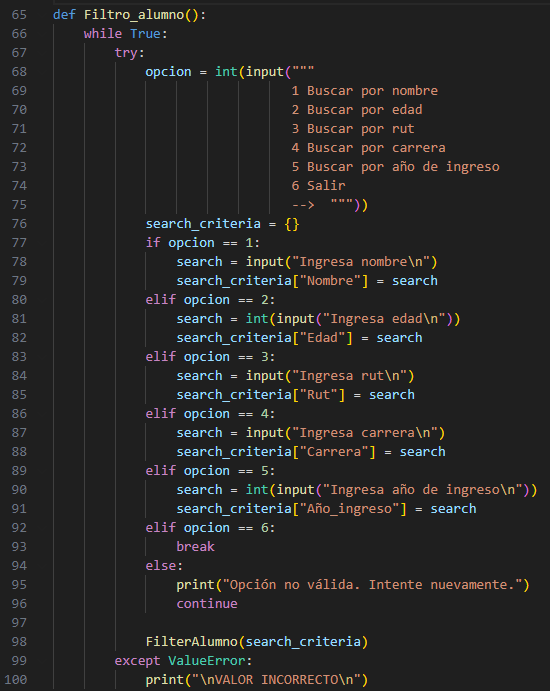
Finalmente tenemos el ingreso del año de admisión del alumno, permitiendo valores desde el 1900 hasta máximo el año 2022, al ingresar un dato fuera de este rango, salta el error determinado.



Para finalizar con el ingreso de alumnos, se crea un “json” para enviar el método de inserción.



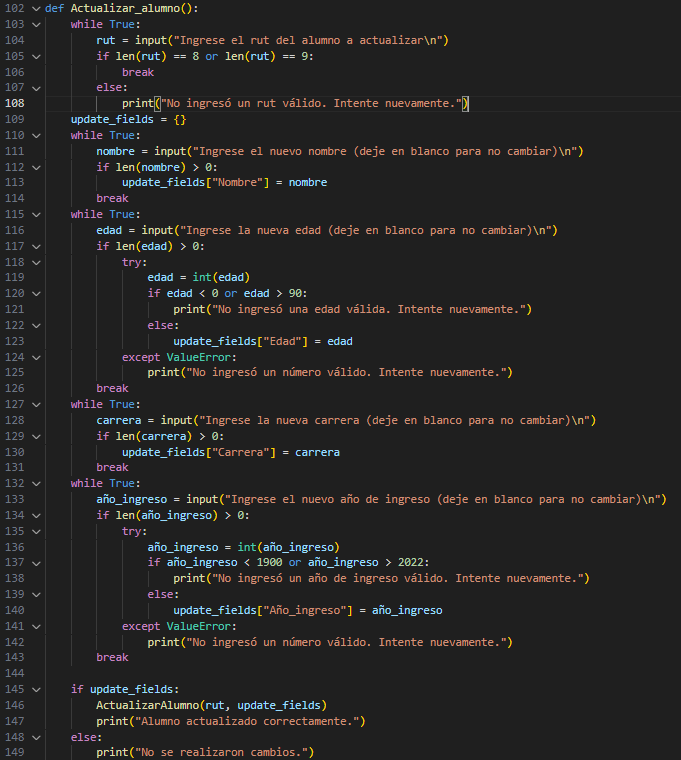
Luego se llama a la función para visualizar alumnos importando desde el archivo “models” explicado anteriormente.



Continuando con la colección de alumnos, existe la función para filtrar alumnos dependiendo de lo que requiera el usuario, en este caso tenemos las opciones para buscar por: El nombre del alumno, la edad el alumno, el rut del alumno, la carrera del alumno o el año de ingreso del alumno.

Cada opción sigue el mismo método de, luego que el usuario ingresa el dato que requiere, busca en la colección de alumnos para entregar los valores que coincidan con los solicitado.

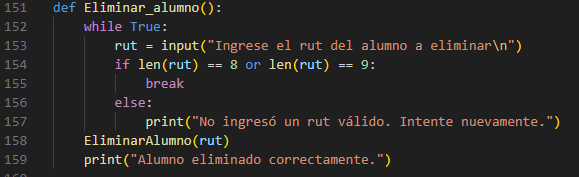
Al ingresar una opción fuera de este menú se imprime el debido error.



Seguimos con la función que nos permite actualizar los datos de los alumnos.

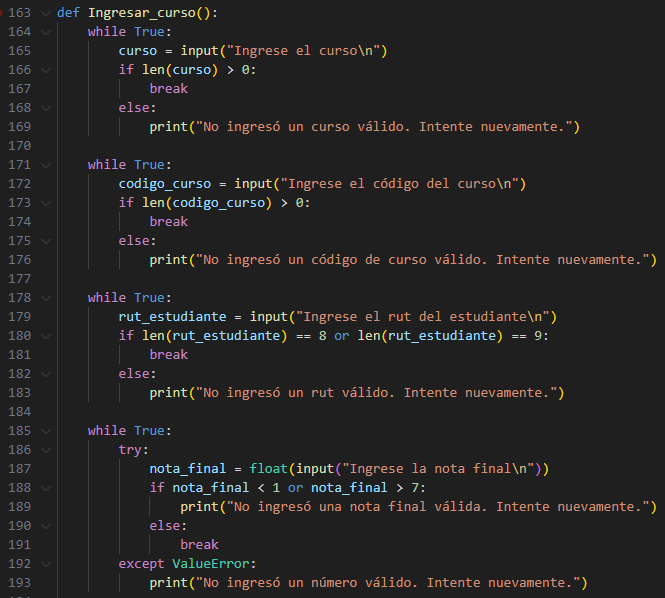
En el cual se inicia solicitando el rut del alumno que se desea modificar, para luego solicitar el nuevo nombre del alumno, su dato de edad modificado, su carrera y año de ingreso.

Cada opción tiene su apartado especificado para hacer entender al usuario que, al dejar ese dato en blanco, no se actualizará el mismo.



Para finalizar con las funciones enfocadas en la colección de los alumnos existe la opción para eliminar alumnos.

En esta se nos solicita el rut del alumno que se desea eliminar, para luego proceder a eliminar el alumno con éxito.



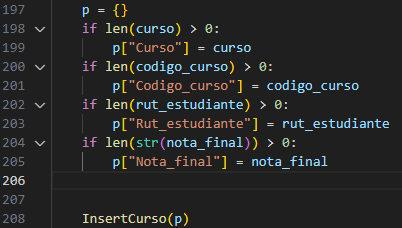
Iniciando con la colección de los cursos, existe la función para ingresar un curso nuevo.

Para comenzar se solicita al usuario el nombre del curso, seguido de su código, luego se solicita ingresar el rut del estudiante que desea agregar al curso.

Finalmente se solicita la nota final del alumno determinado en un rango desde 1 hasta 7.

Cada dato solicitado tiene su respectivo mensaje de error que indica al usuario su falla.

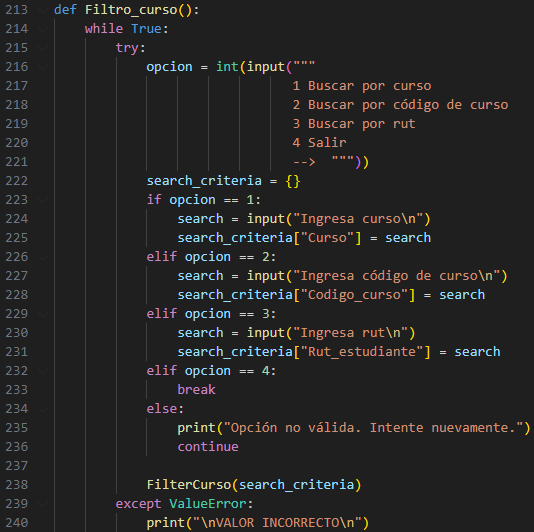
Para finalizar el ingreso de datos al curso, tenemos un “json” creado para enviar el método de inserción.



Continuamos con la visualización de los cursos.



Importado desde el archivo “models.py” la cual sus funciones están determinadas con anterioridad.

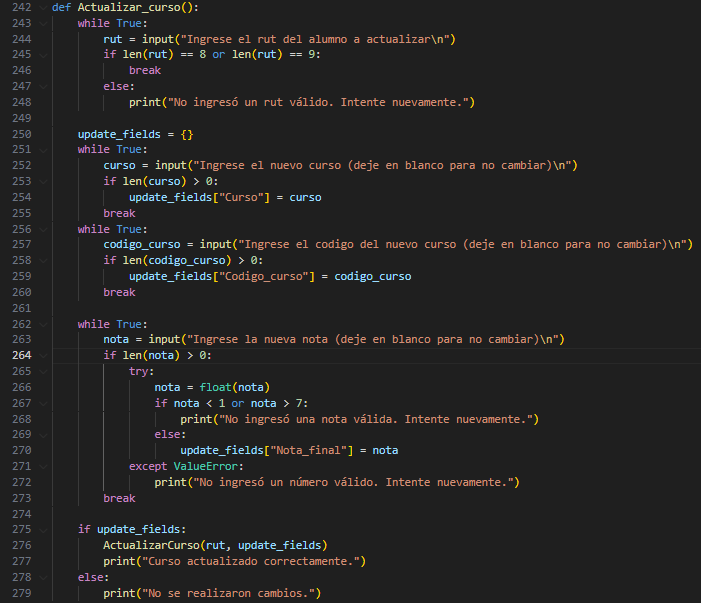


Continuamos con el filtrado por curso.

En esta función se pide al usuario ingresar la opción por la cual desea buscar datos, buscar por curso, código de curso o rut.

Cada opción cumple la regla de, primero buscar los datos solicitados por el usuario, luego, de buscar en la colección para encontrar los datos que coincidan y finalmente imprimir dichos valores.

Al ingresar una opción no válida, saltará el error específico.

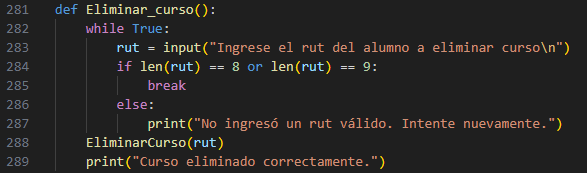


Llegando a las últimas funciones para la colección de cursos, tenemos la actualización de los mismos.

En el cual se solicita al usuario el rut del alumno que se desea actualizar.

Luego se solicita el nuevo curso en el cual desea ingresar al alumno, después se solicita ingresar el código del nuevo curso para el alumno. Finalmente se solicita la nueva nota del alumno. Cada opción tiene su especificación dentro de los requisitos para indicar al usuario que, dejando en blanco cualquier campo, no se modificará el valor de este.

Por último se define que si todos los valores son correctos, se actualizan los datos.



Para finalizar la colección de cursos, existe la función que permite eliminar alumnos de determinado curso.

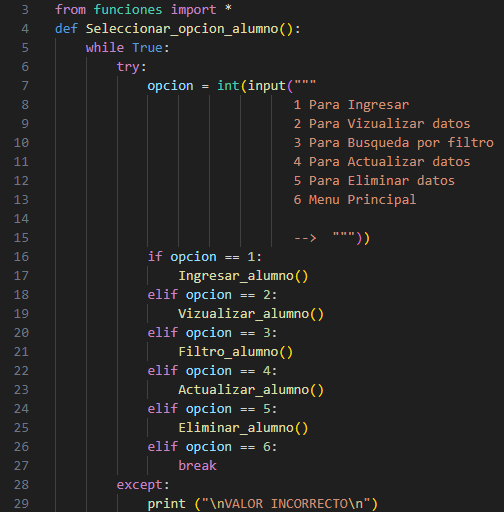
Se solicita al usuario el rut del alumno que se desea eliminar de un curso específico.

Si el rut ingresado es válido, procederá la eliminación del curso del alumno.

En caso contrario, saltará el mensaje de error específico.

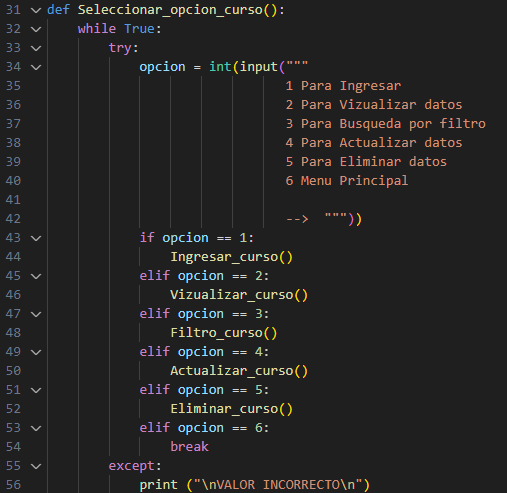
# Menú o “main”

Llegando casi al fin de los archivos, tenemos el archivo “main.py” que lanza la aplicación y funciona como interfaz para el usuario.



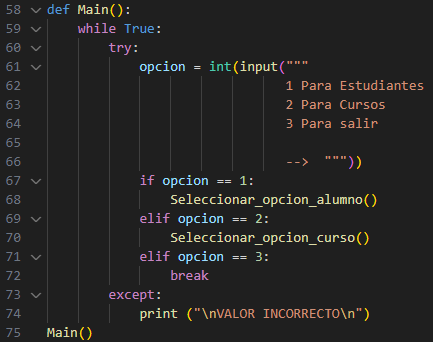
Iniciado con las funciones, primero se crea el menú para solicitar al usuario que ingrese que opción desea ingrear.

Para ingresar alumnos, para visualizar datos de los alumnos, para buscar por filtros determinados, para actualizar datos de alumnos o para volver al menú principal.



Luego tenemos un menú similar para las opciones de cursos.

Se solicita al usuario ingresar por teclado la opción que solicita, en este caso tenemos las opciones para ingresar curso, visualizar datos de los cursos, buscar por filtros, actualizar datos, eliminar datos o para volver al menú principal.



Finalmente tenemos el menú principal que se divide en dos opciones principales, una para ingresar al menú de estudiantes y otra para ingresar al menú de los cursos.

También se tiene una opción para salir por totalidad.

# Conexión a mongodb

Finalizamos los archivos del programa con el archivo llamado “conexión.py” que nos permite establecer conexión con mongodb.

