CONSIDERACIONES AL ESCRIBIR SENTENCIAS if

El principal problema planteado fue la validación de cabeceras de la sentencia if en python, dicha problemática fue modificada a la estructura en sí que maneja esta sentencia en este lenguaje de programación tan conocido e implementado. Dicho problema quedando de la siguiente manera:

Se establece una aplicación la cual a través de autómatas pueda validar una sentencia if sencilla, entiendiéndose como sencilla, sentencias que no requieran más de 3 funciones pasando la cabecera de la sentencia, así como una cabecera que maneje valores comunes como enteros, flotantes, booleanos y variables y comparaciones sencillas para estas. De igual manera, al tratarse de una sentencia if dentro de python, debe validar procesos como la correcta escritura del elif y else, siendo este primero las misma reglas en esencia del if principal.

Una vez establecida la problemática y la aplicación a realizar es necesario dejar y delimitar ciertas acciones dentro de estas validaciones de sentencia, no se trata de validar cada posible estado dentro de las posibilidades del if en este tipo de tecnología (que se podría hacer en caso de pensar en una aplicación escalable, que más adelante veremos que puede ser), se trata de validar las partes iniciales de una correcta estructura if dentro de Python. A continuación se enumerarán las consideraciones.

1. De la estructura del documento a ingresar. Puesto que valida las buenas prácticas en sentencias simples de if en python, el documento que nosotros ingresemos necesita tener ciertas consideraciones para poder ser ingresado y validado correctamente

1.1. Las sentencias if a considerar y que el programa extrae del documento, que por cierto este mismo en formato .py, se encuentran fuera de alguna función y son entendidas como la dentro de una función principal.

1.2. El programa identifica cuales son las funciones dentro de la cabecera de if, pero esta misma debe tener ciertas consideraciones, las cuales se mencionarán posteriormente, pero deben tener un TAB, el cual corresponde e indica que está dentro de la función if y será extraído para su posterior validación dentro de los parámetros establecidos.

1.3. El programa como se mencionó debe ser subido bajo la condición y sufijo .py lo cual indica que se trata de un programa nativo de python, el programa puede leer otro tipo de documentos de formato de texto, pero lo ideal es el antes mencionado.

2. De la cabecera if. Este punto podría considerarse el más relevante dentro de las consideraciones, puesto que de este mismo dependen todos los demás, la sentencia if es por muchas cosas de las sentencias de cabecera más dificiles de validar, ya que su versatilidad dentro del mundo del software lo hace una herramienta tan amplia que limitarla parece necesario al momento de intentar validar sentencias dentro de esta, a continuación se enlistarán las validaciones implementadas en este proyecto.

2.1. La principal cosa a considerar es que la sentencia, por lógica misma, debe empezar con la palabra reservada “if”, puesto que se trata no solo de la palabra necesaria para la extracción del bloque de código, sino del inicio crucial de la validación.

2.2. La siguiente consideración consiste en la validación de un primer estado, no considera operaciones complejas como la palabra reservada “not”, para ello el autómata podrá validar y considerar como correctamente implementadas los siguientes casos.

a) Entidades booleanas de True o False, para este caso el programa entiende por terminada la cabecera puesto que consiste de una entidad definida.
b) Estados de inicialización. Python cuenta, no solo con if’s convencionales en los cuales se ingrese una condición aleatoria según sea el caso, también cuenta con la posibilidad de inicializar el programa, dicha acción puede ser validada por el programa.
c) Números, ya sea enteros o flotantes, son validados por el programa.
d) Variables, estos son validados con condiciones especiales como que el primer carácter debe ser una letra del alfabeto americano en minúscula, este estado puede ser evaluado con alguna comparación, pero de igual manera, si el usuario finaliza la cabecera aquí, no habría ningún problema, ya que puede tratarse de alguna variable con valores booleanos.
e) Funciones, con la misma condición inicial de la variable, una letra en minúscula, dentro de sus parámetros, puesto que pueden ser distintas opciones, existe el caso de no contar con ningún parámetro dentro de asignación, pero para casos de práctica, se estableció que validara exclusivamente funciones con parámetros. Estos parámetros son variables (con las condiciones previas), enteros, flotantes, booleanos y cadenas, cada una, además de la cadena, con las consideraciones previas.

2.3. la siguiente validación posible es dos casos particulares, puesto que hay algunos estados previos que directamente compararlos no es posible o resulta innecesario en el momento de validar sentencias SENCILLAS, algunos estados finalizan directamente, lo que nos deja hablar de la forma de culminación, las cuales son dos posibles, un “:” pegado al estado anterior o un salto de linea, delimitado por la expresión \n

Si se trata de una comparación, el estado anterior debe contener un espacio, las comparaciones son implementados por medio de expresiones algebraicas conocidas como =, <,>, etc.

2.4. El segundo estado comparado debe empezar con un espacio de las expresiones algebraicas, y estos estados pueden ser, cadenas (ya sea implementadas con comillas dobles o simples), enteros, flotantes y variables (iniciando con una letra minúscula). En este caso las funciones quedan excluidas puesto que no representa una lógica el utilizar una función pasado el primer estado de comparación.

2.5. La forma de finalizar fue descrita anteriormente, pero es claro recalcar que en cada estado debe existir un espacio, efectuando los dos puntos “:” finales. También dentro de los parámetros de las funciones, el paréntesis final no cuenta con una separación del último parámetros puesto.

3. Del cuerpo dentro del if. Por motivos de delimitar, puesto que dentro de un if pueden vivir millones de posibles combinaciones de funciones, se implementaron 3 principales:

3.1. El programa valida funciones dentro del if, bajo los mismos parámetros establecidos dentro del punto 2 de esta sección.

3.2. El programa valida la función de impresión dentro de consola, conocido como print, este puede tener variables con su respectiva implementación de la función str, y al igual que los parámetros de las funciones, solamente el último parámetro colocado no cuenta ni con una coma ni un espacio y se encuentra pegado al cierre del programa.

3.3. Probablemente la palabra reservada más utilizada en python, el programa valida la acción pass, y entiende por finalizada dentro de su cuerpo.

3.4. Como consideración la sentencia if, como sus opciones extra else y elif, necesitan mínimo una función dentro.

4. Respecto a la función else. La única consideración relevante es la finalización de los dos puntos sin el espacio con el estado anterior y la finalización con un salto de línea \n.

5. Respecto a la función elif. Las consideraciones son las mismas mencionadas para la cabecera if.