**COBERTURA DE CONDICION:**

|  |  |  |
| --- | --- | --- |
| type(palabra) == type("") and len(palabra) > 0 | | |
| type(palabra) = int | len(palabra) = 0 | type(palabra) == type("") and len(palabra) > 0  (false) |
| type(palabra) = str | len(palabra) = 0 | type(palabra) == type("") and len(palabra) > 0  (false) |
| type(palabra) = str | len(palabra) = 1 | type(palabra) == type("") and len(palabra) > 0  (true) |
| type(palabra) = bool | len(palabra) = 0 | type(palabra) == type("") and len(palabra) > 0  (false) |
| type(palabra) = None | len(palabra) = 0 | type(palabra) == type("") and len(palabra) > 0  (false) |

|  |  |
| --- | --- |
| Iterador < len(palabra)  Palabra = salas | |
| len(palabra) = 5(true) | Iterador < len(palabra)  (true) |
| len(palabra) = 0(false) | Iterador < len(palabra)  (false) |

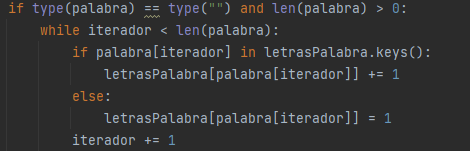
|  |  |  |
| --- | --- | --- |
| palabra[iterador] in letrasPalabra.keys() | | |
| palabra[iterador] = ‘s’ | letrasPalabra.keys() = [s, a, l] | palabra[iterador] in letrasPalabra.keys()  (true) |
| palabra[iterador] = ‘s’ | letrasPalabra.keys() = [] | palabra[iterador] in letrasPalabra.keys()  (false) |
| palabra[iterador] = ‘l’ | letrasPalabra.keys() = [s, a] | palabra[iterador] in letrasPalabra.keys()  (false) |

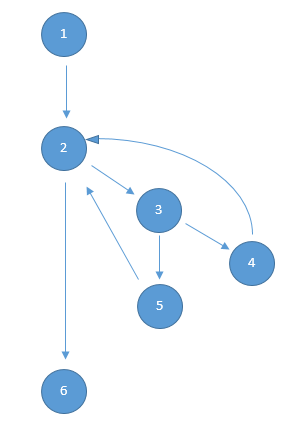
|  |  |  |
| --- | --- | --- |
| i in letrasPalabra1.keys() | | |
| i = ‘s’ | letrasPalabra1.keys() = [s, a, l] | i in letrasPalabra1.keys() (true) |

|  |  |  |
| --- | --- | --- |
| letrasPalabra1[i] != letrasPalabra2[i] | | |
| letrasPalabra1[i] = 1 | letrasPalabra2[i] = 1 | letrasPalabra1[i] != letrasPalabra2[i]  (true) |
| letrasPalabra1[i] = 2 | letrasPalabra2[i] = 1 | letrasPalabra1[i] != letrasPalabra2[i]  (false) |
| letrasPalabra1[i] = 1 | letrasPalabra2[i] = 2 | letrasPalabra1[i] != letrasPalabra2[i]  (false) |

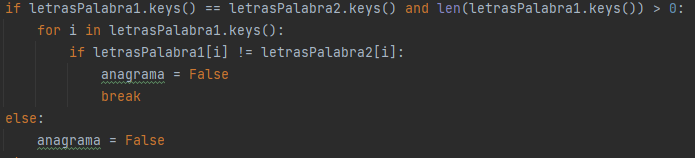
|  |  |  |  |
| --- | --- | --- | --- |
| letrasPalabra1.keys() == letrasPalabra2.keys() and len(letrasPalabra1.keys()) > 0 | | | |
| letrasPalabra1.keys() = [s, a, l] | letrasPalabra2.keys() = [s, a, l] | len(letrasPalabra1.keys()) = 3  (true) | letrasPalabra1.keys() == letrasPalabra2.keys() and len(letrasPalabra1.keys()) > 0 (true) |
| letrasPalabra1.keys() = [c, o, r, t, e] | letrasPalabra2.keys() = [t, e, r, c, o] | len(letrasPalabra1.keys()) = 3  (true) | letrasPalabra1.keys() == letrasPalabra2.keys() and len(letrasPalabra1.keys()) > 0 (true) |
| letrasPalabra1.keys() = [s, a, l] | letrasPalabra2.keys() = [c, o, t] | len(letrasPalabra1.keys()) = 3  (true) | letrasPalabra1.keys() == letrasPalabra2.keys() (false) |
| letrasPalabra1.keys() = [s, a] | letrasPalabra2.keys() = [s, a, l] | len(letrasPalabra1.keys()) = 2  (true) | letrasPalabra1.keys() == letrasPalabra2.keys()  (false) |
| letrasPalabra1.keys() = [s, a, l] | letrasPalabra2.keys() = [] | len(letrasPalabra1.keys()) = 3  (true) | letrasPalabra1.keys() == letrasPalabra2.keys()  (false) |
| letrasPalabra1.keys() = [] | letrasPalabra2.keys() = [] | len(letrasPalabra1.keys()) = 0  (false) | letrasPalabra1.keys() == letrasPalabra2.keys() and len(letrasPalabra1.keys()) > 0 (false) |

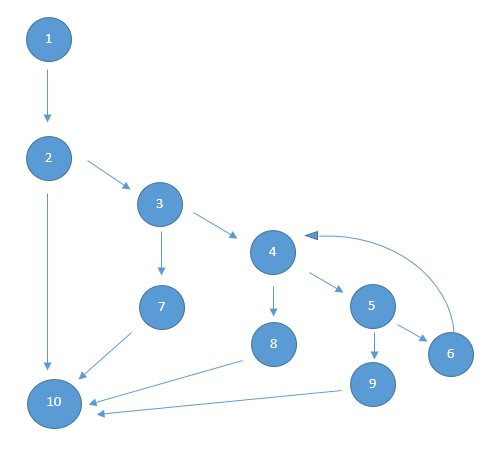
**Cobertura de sentencia:**





Separar palabras





Es anagrama

**Cobertura de decisión y camino:**

