

Lorenzano Sebastian

1)

Clases de equivalencia

Matrícula:

- 1) 5 caracteres alfanuméricos. El primero debe ser obligatoriamente una letra. (Matrícula válida).
- 2) 5 caracteres alfanuméricos. primer carácter no es una letra (Error, matrícula invalida).
- 3) Matrícula > 5 caracteres (Error, matrícula invalida).
- 4) Matrícula < 5 caracteres (Error, matrícula invalida).

Nota:

- 1) Nota ≥ 0 y Nota ≤ 10 (Nota válida).
- 2) Nota < 0 (Error, nota invalida).
- 3) Nota > 10 (Error, nota invalida).
- 4) Nota ≥ 0 y Nota ≤ 10 y número no entero (Error, nota invalida).

Matrícula y Nota:

- 1) 5 caracteres alfanuméricos. El primero debe ser obligatoriamente una letra y Nota ≥ 0 y Nota ≤ 10 (Datos introducidos correctamente).

Valores límite:

Matrícula: El valor límite es de 5 caracteres y el primero debe ser una letra.

Nota: El valor límite de la nota es 0 y 10.

Casos de prueba:

idPrueba	Nota	Matricula	Salida
1	8	A1234	TRUE
2	-	A1234	FALSE
3	-	12345	FALSE
4	-	123456	FALSE
5	-	1234	FALSE
6	-	-	FALSE
7	-	NULL	FALSE
8	-1	A1234	FALSE
9	-1	12345	FALSE
10	-1	123456	FALSE
11	-1	1234	FALSE
12	-1	-	FALSE
13	-1	NULL	FALSE
14	11	A1234	FALSE
15	11	12345	FALSE
16	11	123456	FALSE
17	11	1234	FALSE
18	11	-	FALSE
19	11	NULL	FALSE
20	NULL	NULL	FALSE

NIF:

- 1) NIF con 8 caracteres numéricas y una letra al final.
- 2) NIF > 8 caracteres numéricos. (Error NIF)
- 3) NIF < 8 caracteres numéricos. (Error NIF)
- 4) NIF con 8 caracteres numéricos, pero sin letra. (Error NIF)
- 5) No se introduce valor. (Error NIF)

Número de plazas:

- 1) $50 \geq$ Número de plazas ≥ 1 (Valido)
- 2) Número de plazas > 50 O Número de plazas < 1 (Error en Número de plazas)
- 3) No se introduce (Error en Número de plazas)

NIF y Número de Plazas:

NIF 8 caracteres numéricos. El ultimo debe ser obligatoriamente una letra y Número de plazas ≥ 1 y ≤ 50 (Valido).

Valores límites:

NIF: El valor límite es de 8 caracteres numéricos. El ultimo debe ser obligatoriamente una letra.

Nota: 1 y 50

Casos de prueba:

idPrueba	NIF	Número de Plazas	Salida
1	12345678Z	20	Valido
2	12345678Z	51	Error Número de plazas
3	12345678Z	0	Error Número de plazas
4	12345678Z	NULL	Error Número de plazas
5	123Z	20	Error NIF
6	123Z	51	Error Número de plazas y NIF
7	123Z	0	Error Número de plazas y NIF
8	123Z	NULL	Error Número de plazas y NIF
9	123456789Z	20	Error NIF
10	123456789Z	51	Error Número de plazas y NIF
11	123456789Z	0	Error Número de plazas y NIF
12	123456789Z	NULL	Error Número de plazas y NIF
13	12345678	20	Error NIF
14	12345678	51	Error Número de plazas y NIF
15	12345678	0	Error Número de plazas y NIF
16	12345678	NULL	Error Número de plazas y NIF
17	NULL	20	Error NIF
18	NULL	51	Error Número de plazas y NIF
19	NULL	0	Error Número de plazas y NIF
20	NULL	-	Error Número de plazas y NIF

3)

idPrueba	Num1	Num2	Resultado
1	15	10	10, 11, 12, 13, 14, 15
2	10	15	10, 11, 12, 13, 14, 15
3	""	15	Error: System.FormatException: Input string was not in a correct format.
4	15	""	Error: System.FormatException: Input string was not in a correct format.

Dado a que nos encontramos con un programa sencillo, en el que solo se reciben 2 inputs por parte del usuario, y que además solo presentamos un IF y un ELSE, la cobertura de ramas se obtiene al 100% con tan solo realizar las primeras 2 pruebas
Aun así, las otras dos pruebas no son redundantes, dado que con esto demostramos que el programa no se encuentra preparado para manejar inputs que no sean específicamente enteros.

idPrueba	num1	num2	Resultado
4	-3	3	-3, -2, -1, 0, 1, 2, 3
5	-6	-8	-8, -7, -6
6	3	4	3, 4
7	10	7	7, 8, 9, 10
8	-8	-5	-8, -7, -6, -5

Esto es un ejemplo de casos validos.

4)

Antes de elaborar las pruebas, realizaremos un cuadro intentando ver todas las condiciones y posibilidades que existen en nuestro programa. Si bien esto es tedioso y complicado, será importante para obtener la mayor cobertura posible. Para esto comenzaremos haciendo un análisis de la Cobertura de ramas:

1er IF: *if (numMayus > 0)*

2do IF: *if (numMinus > 0)*

3er IF: *if (numNumeros > 0)*

4to IF: *if (password.Length > 7)*

5to ELSE corresponde al 4to IF y contiene 2 posibilidades:

a) if (password.Length > 0)

b) Else

Por último, tenemos un SWITCH que presenta 4 casos posibles en base a la seguridad obtenida. Este presenta tan solo 4 posibilidades, que serán consecuencia única y exclusivamente de las condiciones anteriores.

Ahora realizaremos un planteo teórico para verificar todos los casos posibles.

Gracias a que nos encontramos con un programa sencillo, esto es posible. Dado a que es finito, y no utiliza bucles. Si no lo fuera, no necesitaríamos pasar por todas las posibilidades, sino intentar que todas las ramas ocurran una vez. Con las ramas hacemos referencia tanto a los IF y Else, como a los CASE.

1er IF	2do IF	3er IF	4to IF	5to ELSE	Punto A	Punto B	Seguridad
TRUE	TRUE	TRUE	TRUE	-	-	-	4
TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	-	1
TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	-	1
TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	-	1
FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	-	1
TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	-	1
FALSE	TRUE	FALSE	FALSE	TRUE	TRUE		1
FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	-	1
FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	0
TRUE	TRUE	FALSE	TRUE	-	-	-	3
TRUE	FALSE	TRUE	TRUE	-	-	-	3
FALSE	TRUE	TRUE	TRUE	-	-	-	3
TRUE	FALSE	FALSE	TRUE	-	-	-	2
FALSE	TRUE	FALSE	TRUE	-	-	-	2
FALSE	FALSE	TRUE	TRUE	-	-	-	2

Lo último sería pasar al diseño de las pruebas, en el cual no pasaremos por todas las posibilidades, pero si por todas las ramas.

idPrueba	numMayus	numMinus	numNumeros	seguridad	password	Salida
1	2	2	4	4	AaBb1234	SEGURIDAD MUY ALTA
2	4	4	0	3	AAAAaaaa	SEGURIDAD ALTA
3	0	0	8	2	12345678	SEGURIDAD MEDIA
4	3	1	2	1	AB12ab	Contraseña demasiado corta SEGURIDAD BAJA

5	0	0	0	0	" "	La contraseña está vacía SEGURIDAD NULA
6	0	0	0	1	#!"#\$%&'()*+,-./:;<=>?@[\]^_`{ }~	SEGURIDAD BAJA
7	0	0	0	1	#!"	Contraseña demasiado corta SEGURIDAD BAJA

Con estas 7 pruebas tendríamos cubiertas todas las ramas de código. Sin dejar ninguna parte al azar, y con el agregado de 2 pruebas, para verificar cómo reacciona nuestro programa al agregado de símbolos, a los cuales no reconoce ni como letras mayúsculas, minúsculas o números, pero que admite de igual manera y que pueden utilizarse como contraseña.

