

EJERCICIOS CAJA NEGRA

Técnica de la Partición Equivalente

Ejercicio 1

Un programa clasifica a individuos teniendo en cuenta las siguientes entradas:

- **Creation date:** cadena con formato “dd-mm-yyyy”, que representa una fecha válida.
- **Type of person:** “student” o “professor”

Los valores de salida esperados son:

- “S1”, si el mes es 08 y el tipo de persona es “student”.
- “S2”, si el mes no es 08 y el tipo de persona es “profesor”.
- “S3”, si el tipo de persona es “student” o “professor”.
- “S4”, en cualquier otro caso correcto.
- “S5”, en cualquier caso incorrecto.

Nota. El orden o prioridad de evaluación de cada salida es S1..S5.

Ejercicio 1

Input	Valid Classes	Invalid Classes	Heuristic
Creation date	(1) Fecha válida, formato válido*, con mes 08 (2) Fecha válida, formato válido con mes <> 08*	(3) Fecha inválida (4) <10 char (5) > 10 char (6) 3 ^{er} char <>'-' (7) 6 ^o char<>'-'	Boolean Valores finitos Conjunto de valores aceptados Conjunto de valores aceptados Clases menores: boolean
Type of person	(8) "student" (9) "professor"	(10) Otro valor	Conjunto de valores aceptados

* Los casos 4,5,6,7 corresponden a los casos de formato inválido

Ejercicio 1

Entrada	Clases válidas	Clases Inválidas
Creation date	(1) Fecha válida, formato válido*, con mes 08 (2) Fecha válida, formato válido con mes <> 08*	(3) Fecha inválida (4) <10 char (5) > 10 char (6) 3 ^{er} char <>'-' (7) 6 ^o char<>'-'
Type of person	(8) "student" (9) "professor"	(10) Otro valor

T.C. Valid C.	Clases Válidas	Entrada	Salida
	(1)(8)	Creation date ='10-08-2018'; Type of person = "student"	S1
	(2)(9)	Creation date ='10-09-2018'; Type of person = "professor"	S2
T.C. Invalid C.	Clases Inválidas	Entrada	Salida
	(3)(8)	Creation date ='33-08-2018'; Type of person = "student"	S5
	(4)(8)	Creation date ='10-08-18'; Type of person = "student"	S5
	(5)(8)	Creation date ='10-08-22018'; Type of person = "student"	S5
	(6)(8)	Creation date ='10/08-2018'; Type of person = "student"	S5
	(7)(8)	Creation date ='10-08/2018'; Type of person = "student"	S5
	(1)(10)	Creation date ='10-08-2018'; Type of person = "other"	S5

Ejercicio 2

Un módulo de software clasifica a los individuos teniendo en cuenta las siguientes entradas:

- **Code:** cadena de 3 dígitos que no empieza por “00”
- **Control:** un carácter dentro de ‘a’..‘z’ o el carácter ‘-’
- **Person Type:** ‘hired’ o ‘interim’

The expected output values are:

- “S1”, si code representa un número par y el tipo de persona es “hired”.
- “S2”, si code representa un número par y el tipo de persona es “interim”.
- “S3”, si code representa un número impar y el tipo de persona es “hired”.
 - “S4”, si code representa un número impar y el tipo de persona es “interim”.
- “S5”, si el carácter de control es “-”.
- “S6”, otherwise.

Nota: encaso de conflicto, la salida S5 tiene prioridad.

Ejercicio 2

Para numerar las clases válidas, combinamos el resultado obtenido de las diferentes heurísticas todo lo posible. Así, obtener los casos de test válidos es más sencillo

Entrada	Clases Válidas	Clases inválidas	Heurística
Code	(1) Es un número de 3 dígitos, >9 y par (2) Es un número de 3, >9 e impar * Ver nota	(3) No es un número (4) < 3 dígitos (5) > 3 dígitos (6) <=9 (Comienza por 00, como '001')	Boolean Valores finitos, Boolean clases menores Boolean
Control Character	(7) Value in ['a'..'z'] (8) '-'	(9) <'a' (bajo límite inferior, excepto -) (10) >'z' (sobre el límite superior)	Rango Valores
Person	(11) 'hired' (12) 'interim'	(13) 'noaccepted'	Conjunto de valores aceptados

Ejercicio 2

Entrada	Clases Válidas	Clases Inválidas
Code	(1) Es un número de 3 dígitos, >9 y par (2) Es un número de 3 dígitos, >0 e impar	(3) No es un número (4) < 3 dígitos (5) > 3 dígitos (6) <=9 (comienza por 00, como '001')
Control Character	(7) Vaue in ['a'..'z'] (8) '-'	(9) <'a' (bajo límite inferior, excepto -) (10) >'z' (sobre el límite superior)
Person	(11) 'hired' (12) 'interim'	(13) 'noaccepted'

T.C. Valid C.	Clases Válidas	Entrada	Salida
	(1)(7)(11)	Code='222'; Control = 'b'; Person= 'hired'	S1
	(2)(8)(12)	Code='111'; Control = '-'; Person= 'interim'	S5

T.C. Invalid C.	Clases Inválidas	Entrada	
	(3)(7)(11)	Code='cod'; Control = 'b'; Person= 'hired'	S6
	(4)(7)(11)	Code='90'; Control = '-'; Person= 'hired'	S6
	(5)(7)(11)	Code='1000'; Control = 'b';Person= 'hired'	S6
	(6)(7)(11)	Code='001'; Control = 'b'; Person= 'hired'	S6
	(1)(9)(11)	Code='222'; Control = 'Z'; Person= 'hired'	S6
	(1)(10)(11)	Code='111'; Control = ' '; Person= 'hired'	S6
	(1)(7)(13)	Code='222'; Control = 'b'; Person='noaccepted'	S6

Ejercicio 3

- Un programa actualiza el precio de un producto teniendo en cuenta las siguientes entradas:
 - Product-code: un campo de 5 caracteres, que puede ser cualquier combinación de letras o números.
 - Sale-percentage: un campo double positivo, que debe ser menor o igual a 100.
 - Responsable: el código del empleado que decide que el producto se ponga en oferta. Es un campo alfanumérico de 6 caracteres. El primer carácter puede ser M,G o A, teniendo en cuenta la categoría del empleado. Los siguientes 5 deben ser números.
- Las posibles salidas son
 1. Cuando el product-code es incorrecto
 2. Cuando el sale-percentage es incorrecto
 3. Cuando responsable es incorrecto
 4. Cuando el producto correspondiente al product-code indicado se ha podido modificar correctamente
 5. Si el producto correspondiente al product-code indicado se ha podido modificar correctamente pero el sale-percentage indicado es mayor a 50.'

Ejercicio 3

* Para numerar las clases válidas, combinamos el resultado que nos ofrece las heurísticas lo máximo posible. Así, es más fácil obtener el conjunto de casos de test

Entrada	Clase Válida	Clase Inválida	Heurística
Product-code	(1) Alfanumérico de 5 dígitos * Ver nota	(2) No alfanumérico (3) < 5 dígitos (4) > 5 dígitos	Boolean Conjunto Valores Finitos
Sale-percentage	(5) [0,50] (6)]50,100]	(7) <0 (8) > 100	Rango de valores, Clases menores
Responsable	(9) 6 caracteres, con el primer car= G y los siguientes forman un número (10) 6 caracteres, con el primer car= M y los siguientes forman un número (11) 6 caracteres, con el primer car= A y los siguientes forman un número * Ver nota	(12)<6 caracteres (13)>6 caracteres (14) 1 ^{er} carácter no en {M,A,G} (15) 2 ^o a 6 ^o caracteres no son un número	Conjunto Valores Finitos Conjunto de valores aceptados Boolean

Ejercicio 3

Entrada	Clase Válida	Clase Inválida	Heurística
Product-code	(1) Alfanumérico de 5 dígitos	(2) No alfanumérico (3) < 5 dígitos (4) > 5 dígitos	Boolean Conjunto Valores Finitos
Sale-percentage	(5) [0,50] (6)]50,100]	(7) <0 (8) > 100	Rango de valores, Clases menores boolean
Responsable	(9) 6 caracteres, con el primer car= G y los siguientes forman un número (10) 6 caracteres, con el primer car= M y los siguientes forman un número (11) 6 caracteres, con el primer car= A y los siguientes forman un número	(12)<6 caracteres (13)>6 caracteres (14) 1 ^{er} carácter no en {M,A,G} (15) 2 ^o to 6 ^o caracteres no son un número	Conjunto Valores Finitos Conjunto de valores aceptados Boolean

T.C. Casos Válidos	Clase Válida	Entrada	Salida
	(1)(5)(9)	Product-code="F0001"; sale-percentage=10; Responsable="G12345"	4
	(1)(6)(10)	Product-code="F0001"; sale-percentage=90; Responsable="M12345"	5
	(1)(5)(11)	Product-code="F0001"; sale-percentage=40; Responsable="A12345"	4

Ejercicio 3

Entrada	Clase Válida	Clase Inválida
Product-code	(1) Alfanumérico de 5 dígitos	(2) No alfanumérico (3) < 5 dígitos (4) > 5 dígitos
Sale-percentage	(5) [0,50] (6)]50,100]	(7) <0 (8) > 100
Responsible	(9) 6 caracteres, con el primer car= G y los siguientes forman un número (10) 6 caracteres, con el primer car= M y los siguientes forman un número (11) 6 caracteres, con el primer car= A M y los siguientes forman un número	(12)<6 caracteres (13)>6 caracteres (14) 1 ^{er} carácter no en {M,A,G} (15) 2 ^o to 6 ^o caracteres no son un número

T.C. Clases Inválidas	Clases Inválidas	Entrada	Salida
	(2)(5)(9)	Product-code="?1234"; sale-percentage=10; Responsible="G12345"	1
	(3)(5)(9)	Product-code="1234"; sale-percentage=10; Responsible="G12345"	1
	(4)(5)(9)	Product-code="123456"; sale-percentage=10; Responsible="G12345"	1
	(1)(7)(9)	Product-code="C0001"; sale-percentage=-10; Responsible="G12345"	2
	(1)(8)(9)	Product-code="C0001"; sale-percentage=110; Responsible="G12345"	2
	(1)(5)(12)	Product-code="C0001"; sale-percentage=10; Responsible="G1234"	3
	(1)(5)(13)	Product-code="C0001"; sale-percentage=10; Responsible="G123456"	3
	(1)(5)(14)	Product-code="C0001"; sale-percentage=10; Responsible="B12345"	3
	(1)(5)(15)	Product-code="C0001"; sale-percentage=10; Responsible="Gabcde"	3

Ejercicio 4

Un método que genera un listado recibe la siguiente información:

- Nombre del alumno, tiene que tener como mínimo un nombre y apellido.
- Grupo, son tres caracteres donde lo primero es una de las siguientes letras A, C o D y los otros dos caracteres son dos dígitos que pueden ir del 01 al 15.
- Nota teoría (NT), número sobre 10.
- Nota práctica (NP), número sobre 10.
- Nota trabajo (NTr), número sobre 8.

El método devuelve como salida la nota final en función de la fórmula $NT*0.60+NP*0.4+NTr*0.1$. En caso contrario debe devolver error.

Se pide:

Crear la tabla de clases de equivalencia (indicando la regla heurística aplicada).

Generar los casos de prueba.

Ejercicio 4

Entrada	Clases Válidas	Clases Inválidas	Heurísticas
Nombre	(1) Con al menos 2 palabras	(2) Con menos de 2 palabras	Boolean o Debe ser
Grupo	<p>(3) 3 caracteres, con el primer carácter A, y 2do y 3r carácter forman un número, dentro del intervalo [01-15]</p> <p>(6) 3 caracteres, con el primer carácter C, y 2do y 3r carácter forman un número, dentro del intervalo [01-15]</p> <p>(7) 3 caracteres, con el primer carácter D, y 2do y 3r carácter forman un número, dentro del intervalo [01-15]</p>	<p>(4) < 3 caracteres (5) > 3 caracteres</p> <p>(8) Otro valor</p> <p>(9) No es número (10) <01 (11) >15</p>	<p>Conjunto finito de valores</p> <p>Conjunto de valores</p> <p>Boolean</p> <p>Rango de valores</p>
Teoría	(12) [0-10]	<p>(13) <0 (14) >10</p>	Rango de valores
Práctica	(15) [0-10]	<p>(16) <0 (17) >10</p>	Rango de valores
Trabajo	(18) [0-10]	<p>(19) <0 (20) >8</p>	Rango de valores

Ejercicio 4

Entrada	Clases Válidas	Clases Inválidas
Nombre	(1) Con al menos 2 palabras	(2) Con menos de 2 palabras
Grupo	(3) 3 caracteres, con el primer carácter A, y 2do y 3r carácter forman un número, dentro del intervalo [01-15] (6) 3 caracteres, con el primer carácter C, y 2do y 3r carácter forman un número, dentro del intervalo [01-15] (7) 3 caracteres, con el primer carácter D, y 2do y 3r carácter forman un número, dentro del intervalo [01-15]	(4) < 3 caracteres (5) > 3 caracteres (8) Otro valor (9) No es número (10) <01 (11) >15
Teoría	(12) [0-10]	(13) <0 (14) >10
Práctica	(15) [0-10]	(16) <0 (17) >10
Trabajo	(18) [0-8]	(19) <0 (20) >8

T.C. Valid C.	Clases Válidas	Input	Salida
	(1)(3)(12)(15)(18)	Nombre='Joe Doe'; Grupo='A10'; Teoría=10, Lab=10, Trab=8	10.8
	(1)(6)(12)(15)(18)	Nombre='Joe Doe'; Grupo='C10'; Teoría=10, Lab=10, Trab=8	10.8
	(1)(7)(12)(15)(18)	Nombre='Joe Doe'; Grupo='D01'; Teoría=5, Lab=5, Tra=8	5.8

Ejercicio 4

Entrada	Clases Válidas	Clases Inválidas
Nombre	(1) Con al menos 2 palabras	(2) Con menos de 2 palabras
Grupo	(3) 3 caracteres, con el primer carácter A, y 2do y 3r carácter forman un número, dentro del intervalo [01-15] (6) 3 caracteres, con el primer carácter C, y 2do y 3r carácter forman un número, dentro del intervalo [01-15] (7) 3 caracteres, con el primer carácter D, y 2do y 3r carácter forman un número, dentro del intervalo [01-15]	(4) < 3 caracteres (5) > 3 caracteres (8) Otro valor (9) No es número (10) <01 (11) >15
Teoría	(12) [0-10]	(13) <0 (14) >10
Práctica	(15) [0-10]	(16) <0 (17) >10
Trabajo	(18) [0-8]	(19) <0 (20) >8

T.C. Clases Inválidas	Clases Inválidas	Entrada	Salida
	(2)(3)(12)(15)(18)	Name='Joe'; Group='A10'; Theory=10, Lab=10, Tra=8	--
	(1)(4)(12)(15)(18)	Name='Joe Doe'; Group='A1'; Theory=10, Lab=10, Tra=8	--
	(1)(5)(12)(15)(18)	Name='Joe Doe'; Group='A001'; Theory=10, Lab=10, Tra=8	--
	(1)(8)(12)(15)(18)	Name='Joe Doe'; Group='B01'; Theory=10, Lab=10, Tra=8	--
	(1)(9) (12)(15)(18)	Name='Joe Doe'; Group='Ccc'; Theory=10, Lab=10, Tra=8	--
	(1)(10) (12)(15)(18)	Name='Joe Doe'; Group='C00'; Theory=10, Lab=10, Tra=8	--
	(1)(11) (12)(15)(18)	Name='Joe Doe'; Group='C16'; Theory=10, Lab=10, Tra=8	
	(1)(3)(13)(15)(18)	Name='Joe Doe'; Group='C10'; Theory=-1, Lab=10, Tra=8	--
	(1)(3)(14)(15)(18)	Name='Joe Doe'; Group='C10'; Theory=11, Lab=10, Tra=8	--
	(1)(3)(12)(16)(18)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=-1, Tra=8	--
	(1)(3)(12)(17)(18)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=12, Tra=8	--
	(1)(3)(12)(15)(19)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=8, Tra=-1	
	(1)(3)(12)(15)(20)	Name='Joe Doe'; Group='C10'; Theory=5, Lab=8, Tra=9	--

Ejercicio 5

- Un programa recibe como entrada un fichero de texto con las siguientes columnas:
 - **Product-number:** Un campo entero positivo menor a 256, con 3 dígitos.
 - **Product-code:** un campo alfanumérico de 4 caracteres.
 - **Expiry-Month:** representa los meses que transcurren hasta que el producto caduca; es un valor positivo de dos dígitos (excepto 00).
 - **Sale:** un campo de solo un character, es “+” cuando el producto está en oferta o “-” en otro caso.

Ejercicio 5

* Para numerar las clases válidas, combinamos el resultado que nos ofrece las heurísticas lo máximo posible. Así, es más fácil obtener el conjunto de casos de test

Entrada	Clases Válidas	Clases Inválidas	Heurísticas
Product-number	(1) Número de 3 dígitos, cuyo valor en [0,255] * Ver nota	(2) No número (3) <3 dígitos (4) >3 dígitos (5) <0 (6) >255	Boolean Conjunto finito de valores Rango valores
Product-code	(7) Alfanumérico de 4 caracteres * Ver nota	(8) No alfanumérico (9) < 4 car (10) > 4 car	Boolean Conjunto finito de valores
Expiry-Month	(11) Número de 2 dígitos, cuyo valor es >0 • Ver nota	(12) No número (13) <2 dígitos (14) > 2 dígitos (15) <0 valor	Boolean Conjunto finito de valores Boolean
Sale	(16) '+' (17) '-'	(18) Otro valor('*')	Conjunto de valores aceptados

Ejercicio 5

Entrada	Clases Válidas	Clases Inválidas	Heurísticas
Product-number	(1) Número de 3 dígitos, cuyo valor en [0,255] * Ver nota	(2) No número (3) <3 dígitos (4) >3 dígitos (5) <0 (6) >255	Conjunto finito de valores Rango valores
Product-code	(7) Alfanumérico de 4 caracteres * Ver nota	(8) No alfanumérico (9) < 4 car (10) > 4 car	Boolean Conjunto finito de valores
Expiry-Month	(11) Número de 2 dígitos, cuyo valor es >0 • Ver nota	(12) No número (13) <2 dígitos (14) > 2 dígitos (15) <0 valor	Boolean Conjunto finito de valores Boolean
Sale	(16) '+' (17) '-'	(18) Otro valor('*')	Conjunto de valores aceptados

T.C. Clas. Válid	Clases válidas	Entradas
	(1)(7)(11)(16)	Product-number='001' ; Product-code='code'; Expiry-Month='24'; Sale ='+'
	(1)(7)(11)(17)	Product-number='001' ; Product-code='code'; Expiry-Month='24'; Sale ='-'

Ejercicio 5

Entrada	Clases Válidas	Clases Inválidas
Product-number	(1) Número de 3 dígitos, cuyo valor en [0,255] * Ver nota	(2) No número (3) <3 dígitos (4) >3 dígitos (5) <0 (6) >255
Product-code	(7) Alfanumérico de 4 caracteres * Ver nota	(8) No alfanumérico (9) < 4 car (10) > 4 car
Expiry-Month	(11) Número de 2 dígitos, cuyo valor es >0 • Ver nota	(12) No número (13) <2 dígitos (14) > 2 dígitos (15) <=0 valor
Sale	(16) '+' (17) '-'	(18) Otro valor('*')

	Clases Invalidas	Entrada
T.C. Invalid C.	(2)(7)(11)(16)	Product-number='aaa'; Product-code='code'; Expiry-Month='24'; Sale = '+'
	(3)(7)(11)(16)	Product-number='00'; Product-code='code'; Expiry-Month='24'; Sale = '+'
	(4)(7)(11)(16)	Product-number='0000'; Product-code='code'; Expiry-Month='24'; Sale = '+'
	(5)(7)(11)(16)	Product-number='-01'; Product-code='code'; Expiry-Month='24'; Sale = '+'
	(6)(7)(11)(16)	Product-number='257'; Product-code='code'; Expiry-Month='24'; Sale = '+'
	(1)(8)(11)(16)	Product-number='001' ; Product-code='-*!+'; Expiry-Month='24'; Sale = '+'
	(1)(9)(11)(16)	Product-number='001' ; Product-code='cod'; Expiry-Month='24'; Sale = '+'
	(1)(10)(11)(16)	Product-number='001' ; Product-code='codes'; Expiry-Month='24'; Sale = '+'
	(1)(7)(12)(16)	Product-number='001' ; Product-code='code'; Expiry-Month='mm'; Sale = '+'
	(1)(7)(13)(16)	Product-number='001' ; Product-code='code'; Expiry-Month='1'; Sale = '+'
	(1)(7)(14)(16)	Product-number='001' ; Product-code='code'; Expiry-Month='001'; Sale = '+'
	(1)(7)(15)(16)	Product-number='001' ; Product-code='code'; Expiry-Month='00'; Sale = '+'
	(1)(7)(11)(18)	Product-number='001' ; Product-code='code'; Expiry-Month='24'; Sale = '*'

Ejercicio 6

El Presidente de la Elección introducirá a través de un formulario:

- La fecha en la que se celebra la elección. La fecha será una cadena con el formato DD/MM/AAAA.
- El valor a partir del cual se considera que un miembro ha recibido un número significativo de votos. Será un valor entre 10 y 50.
- El método de recuento de votos. Podrá ser “Hondt” o “Sainte”.

El resultado de ejecutar el módulo será el siguiente:

1. Si la fecha no es válida (formato incorrecto)
2. Si el valor de votos significativos no es correcto.
3. Si el método de recuento de votos no es correcto.
4. Si los datos son correctos se mostrará una lista con los miembros ordenados de acuerdo al método de recuento utilizado.
5. Si los datos son correctos, pero no hay ningún miembro con bastantes votos significativos se mostrará un mensaje indicándolo.

Ejercicio 6

Entrada	Clases Válidas	Clases Inválidas	Heurísticas
Fecha	(1) Fecha válida, con formato DD/MM/AAAA.	(2) Fecha no válida (3) <10 char (4) > 10 char (5) 3 ^{er} char <>'/' (6) 6 ^o char<>'/'	Booleana Valores finitos Conjunto de valores aceptados Conjunto de valores aceptados
Valor Significativo	(7) [10-50]	(8) <10 (9) >50 (10) No es número	Rango Booleana
Método	(11) "Hondt" (12) "Sainte"	(13) Otro valor	Conjunto de valores
Lista	(14) Con elementos (15) Vacía	(16) No lista	Booleana Clases menores

CP C. Validos	Clases válidas	Entrada	Salida
	1 – 7 - 11 - 14	"15/12/2018", 15, "Hondt", [{ 'Pepe', 1 }, { 'Juan', 4 }]	4
	1 – 7 – 12-15	"15/12/2018", 15, "Sainte", []	5

Ejercicio 6

Entrada	Clases Válidas	Clases Inválidas
Fecha	(1) Fecha válida, con formato DD/MM/AAAA.	(2) Fecha no válida (3) <10 char (4) > 10 char (5) 3 ^{er} char <>'/' (6) 6º char<>'/'
Valor Significativo	(7) [10-50]	(8) <10 (9) >50 (10) No es número
Método	(11) "Hondt" (12) "Sainte"	(13) Otro valor
Lista	(14) Con elementos (15) Vacía	16 (No lista)

Casos de Prueba No Validos	c. no válidas	Entrada	Salida
	2 - 7 - 11-14	"AA", 15, "Hondt", [{'Pepe', 1}, {'Juan', 4}]	1
	3 - 7 - 11-14	"15/12/18", 15, "Hondt", [{'Pepe', 1}, {'Juan', 4}]	2
	4-7 - 11-14	"15/12/20018", 15, "Hondt", [{'Pepe', 1}, {'Juan', 4}]	2
	5 - 7 - 11-14	"15-12/2018", 15, "Hondt", [{'Pepe', 1}, {'Juan', 4}]	2
	6 - 7 - 11-14	"15/12-2018", 15, "Hondt", [{'Pepe', 1}, {'Juan', 4}]	3
	1-8 - 11-14	"15/12/2018", 9, "Hondt", [{'Pepe', 1}, {'Juan', 4}]	2
	1- 9 - 11-14	"15/12/2018", 59, "Hondt", [{'Pepe', 1}, {'Juan', 4}]	2
	1 - 10 - 11-14	"15/12/2018", "AA", "Hondt", [[{'Pepe', 1}, {'Juan', 4}]]	2
	1 - 7- 13 - 10	"15/12/2018", 15, "AAA", [{'Pepe', 1}, {'Juan', 4}]	3
	1 - 7- 11 - 16	"15/12/2018", 15, "Hondt", null	--