



# Corrección Cruzada

Desarrollo de Software

Raúl Aguilar Arroyo, 100472050

Carlos Seguí Cabrera, 100472060

Grupo 3 → Grupo 4

29 mar 2023

# ÍNDICE

<b>ÍNDICE</b>	<b>1</b>
<b>Corrección clases de prueba del grupo 4</b>	<b>2</b>
<b>Introducción</b>	<b>2</b>
Definición de los casos de prueba	2
<b>Entradas</b>	<b>3</b>
Product_ID	3
Order_type	4
Address	4
Phone_number	5
Zip_Code	5
Salidas y conclusión de la revisión	6

# Corrección clases de prueba del grupo 4

## Introducción

En este documento pretendemos corregir los casos de prueba del grupo 4, para ello utilizaremos como base la teoría de las clases de equivalencia y los valores límite, ya que nosotros también hemos tenido que implementar y diseñar dichas pruebas tenemos una visión de las mismas que nos puede facilitar la detección de cualquier caso erróneo.

Aunque por otro lado vamos a obviar lo máximo posible nuestras pruebas para intentar tener la máxima imparcialidad posible.

En última instancia la organización del documento se centrará primero en una breve explicación del método de selección de las pruebas y en segunda instancia las diversas pruebas realizadas por el grupo 4, separadas por cada entrada que se precisa comprobar en cada una de ellas.

## Definición de los casos de prueba

Para la correcta selección de los casos de prueba debemos obtener una clase de equivalencia para cada valor válido y una clase inválida que represente el resto de los valores, teniendo en cuenta si los tiene los valores límite, haciendo los casos con 1 número menor, superior e igual al valor límite.

Tanto las entradas como las salidas las dividimos en estas clases de equivalencia, siendo las salidas esperadas probadas con las entradas que elegimos y del mismo modo las excepciones siendo manejadas cada una de ellas como una clase de equivalencia.

## Entradas

Tenemos las siguientes 5 entradas.

### Product\_ID

Es un código EAN13 que identifica el producto que debe ser servido.

Según hemos visto en la definición de los casos de prueba en este caso, tenemos de valor límite 13 con lo cual tenemos que comprobar con 12,13,14, un caso válido y otra clase de equivalencia para los valores inválidos.

Como 13 es el caso válido, nos quedan otras tres pruebas con 12, 14 y el caso inválido.

Esto es lo que tiene el grupo 4:

TEC HNI QUE	FIELD	VALID /INVA LID	ID TEST	DESCRIPTION	PRODUCT_ID
EC	PRODUC T_ID	VALID	CE_V_1, CE_V_6, CE_V_9, CE_V_13, CE_V_16	ALL CORRECT (REGULAR)	8421691423220
EC	PRODUC T_ID	NOT VALID	CE_NV_2	PRODUCT ID NOT NUMBER	842169142322A
EC	PRODUC T_ID	NOT VALID	CE_NV_3	PRODUCT ID NOT CHECK SUM	8421691423225
EC	PRODUC T_ID	NOT VALID	CE_NV_4	PRODUCT ID 12	842169142322
EC	PRODUC T_ID	NOT VALID	CE_NV_5	PRODUCT ID 14	84216914232200

Podemos observar que los casos de prueba 1,4 y 5 son correctos (Válido y 13 ,12,14)

,sin embargo, en el método que hemos expresado las dos pruebas 2 y 3 son redundantes por ello con una de las dos que comprobase que es inválido sería suficiente

## Order\_type

En este caso order\_type puede ser regular o premium.  
Como no tiene valores límite tenemos 3 pruebas que sea regular, premium y otra inválida.

TECHNIQUE	FIELD	VALID/INVALID	ID TEST	DESCRIPTION	ORDER_TYPE
EC	ORDER_TYPE	VALID	CE_V_7	ALL CORRECT (PREMIUM)	"PREMIUM"
EC	ORDER_TYPE	NOT VALID	CE_NV_8	ORDER_TYPE WRONG	"PRE"

Sumado al primer caso que se comprueba el regular, se obtienen las tres pruebas así que en esta entradas está definido perfectamente.

## Address

Dirección a la que se envía el producto. (entre 20 y 100 caracteres con al menos 2 cadenas separadas por un espacio blanco).

Definimos por ello los valores límite en 20 y 100(caracteres) como 2 cadenas y por ello un blanco. La pruebas inválidas que debemos definir son por ello la de aquella cadena con 19 caracteres, 101, una única cadena, dos cadenas y un caso inválido, ya que los demás casos se comprueban como válidos en nuestras pruebas anteriores.

TECHNIQUE	FIELD	VALID/INVALID	ID TEST	DESCRIPTION	ADDRESS
EC	ADDRESS	NOT VALID	CE_NV_10	ADDRESS 19	"C/LISBOA, 4, MADRID,"
EC	ADDRESS	NOT VALID	CE_NV_11	ADDRESS 101	"C/LISBOA,4, MADRID, SPAIN SPAINSPAINSPAINSPAINSPAINSPAI

Como el caso anterior al comprobar la clase válida en el primer test(9) se cubren todos los casos en esta menos un caso inválido de la clase de equivalencia(ya que se trata de caracteres este podría ser un caso de espacios en blanco) y el caso del valor límite de 2 cadenas.

En este caso es más simple siendo solo 9 el valor límite, necesitamos un caso de prueba de 8, 9, 10 y un inválido.

Se cumplen las pruebas especificadas con lo cual es correcto el planteamiento en este caso( la válida de 9 se comprueba en el primer caso).

Como última entrada tenemos el Zip\_Code, este debe de tener 5 dígitos, ya que se trata de un código postal. Por ello a valor límite 5 las pruebas serian con 4,5, 6 y una clase inválida.

TECHNIQUE	FIELD	VALID/INVALID	ID TEST	DESCRIPTION	ZIP_CODE
EC	ZIP_CODE	NOT VALID	CE_V_17	ZIP_CODE 4	"2800"
EC	ZIP_CODE	NOT VALID	CE_V_18	ZIP_CODE 6	"280055"
EC	ZIP_CODE	NOT VALID	CE_V_21	ZIP_CODE STRING	"280A5"
EC	ZIP_CODE	NOT VALID	CE_V_22	ZIP_CODE NOT VALID	"67005"

Como en el primer caso (product\_id) se cubre una prueba de más, aunque en aquí se encuentra más exagerado ya que tanto el test que valida el 18 como el 22 comprueban el mismo error no válido (6 dígitos). Siendo este último innecesario.

## Salidas y conclusión de la revisión

RESULT
6f651f23a6af8d79e5a30276d136b1ee y se almacenan los datos en el fichero JSON
Exception: Invalid EAN13 code string y no se almacenan los datos en el fichero JSON
Exception: Invalid EAN13 code sum y no se almacenan los datos en el fichero JSON
Exception: Invalid EAN13 code len <13 y no se almacenan los datos en el fichero JSON
Exception: Invalid EAN13 code len >13 y no se almacenan los datos en el fichero JSON
ecc7f631f98930aca6e183c6e505dFFF se almacenan los datos en el fichero JSON
Exception : order type wrong y no se almacenan los datos en el fichero JSON
Exception : address too short y no se almacenan los datos en el fichero JSON
Exception : address too long y no se almacenan los datos en el fichero JSON
Exception : address wrong y no se almacenan los datos en el fichero JSON
Exception : phone_number too long y no se almacenan los datos en el fichero JSON
Exception : phone_number too short y no se almacenan los datos en el fichero JSON
Exception : phone_number string y no se almacenan los datos en el fichero JSON
Exception : zip_code too short y no se almacenan los datos en el fichero JSON
Exception : zip_code too long y no se almacenan los datos en el fichero JSON
Exception : zip_code string y no se almacenan los datos en el fichero JSON

Exception : zip\_code is not valid y no se almacenan los datos en el fichero JSON

### Todas las salidas de los diferentes casos de prueba

Conforme se ha comentado en la sección de definición de los casos de prueba, las salidas se comprueban con las entradas en este caso se cubren las excepciones y las diversas salidas (se detallan incluso más específicamente).

Las excepciones correspondientes a cada una de las salidas en función de cada entrada inválida se cumple, como la cadena MD5 Y el fichero(caso válido).

Como conclusión del documento excel de los casos de prueba el grupo 4 ha planteado unos casos en su mayoría correctos, exceptuando algunos casos de redundancia y la falta de un único caso de prueba. Siendo estos:

#	FIELD	VALID/INVALID	ID TEST	DESCRIPTION	Corrección
1	PRODUCT_ID	VALID	CE_V_1, CE_V_6, CE_V_9, CE_V_13, CE_V_16	ALL CORRECT (REGULAR)	✓
2	PRODUCT_ID	NOT VALID	CE_NV_2	PRODUCT ID NOT NUMBER	✓
3	PRODUCT_ID	NOT VALID	CE_NV_3	PRODUCT ID NOT CHECK SUM	✗
4	PRODUCT_ID	NOT VALID	CE_NV_4	PRODUCT ID 12	✓
5	PRODUCT_ID	NOT VALID	CE_NV_5	PRODUCT ID 14	✓
6	ORDER_TYPE	VALID	CE_V_7	ALL CORRECT (PREMIUM)	✓
7	ORDER_TYPE	NOT VALID	CE_NV_8	ORDER_TYPE WRONG	✓
8	ADDRESS	NOT VALID	CE_NV_10	ADDRESS 19	✓
9	ADDRESS	NOT VALID	CE_NV_11	ADDRESS 101	✓
10	ADDRESS	NOT VALID	CE_NV_12	ADDRESS CON UNA SOLA CADENA (SIN ESPACIOS)	✓
					✗ falta caso inválido y caso de valor límite de 2 cadenas
11	PHONE_NUMBER	NOT VALID	CE_NV_14	PHONE_NUMBER 10	✓
12	PHONE_NUMBER	NOT VALID	CE_NV_15	PHONE_NUMBER 8	✓
13	PHONE_NUMBER	NOT VALID	CE_V_19	PHONE_NUMBER STRING	✓
14	ZIP_CODE	NOT VALID	CE_V_17	ZIP_CODE 4	✓
15	ZIP_CODE	NOT VALID	CE_V_18	ZIP_CODE 6	✓
16	ZIP_CODE	NOT VALID	CE_V_21	ZIP_CODE STRING	✓
17	ZIP_CODE	NOT VALID	CE_V_22	ZIP_CODE NOT VALID	✗