

EG2



Adrián García Ochando 100526397@alumnos.uc3m.es

Fran Jurado Baste 100522213@alumnos.uc3m.es

ÍNDICE:

Primera función: 3

 Clases de equivalencias: 3

 Valores límite: 4

Segunda Función: 5

 Gramática: 5

 Árbol con Nodos ordenados: 6

Tercera Función: 7

 Gráfico de Control de Flujo: 7

 Camino básico: 1_3_5_7_9_10_11 8

Primera función:

Clases de equivalencias:

<i>Clases de Equivalencia válidas</i>	<i>Clases de Equivalencia NO válidas</i>
CEV1 - string	CENV1 - distinto de string
CEV2 - iban válido	CENV2 - iban no cumple algoritmo de validación
CEV3 - Dos primeros char son ES	CENV3 - 2 primeros caracteres no son ES
CEV4 - string	CENV4 - Distinto de string
CEV5 - iban válido	CENV5 - iban no cumple algoritmo de validación
CEV6 - Dos primeros char son ES	CENV6 - primeros caracteres no son ES
CEV7 - string	CENV7 - distinto de string
CEV8 - Char contenidos en [a-z A-Z]	CENV8 - Char no contenido en [a-z A-Z]
CEV9 - string value = "ORDINARY" o "URGENT" o "IMMEDIATE"	CENV9 - string value != "ORDINARY" o "URGENT" o "IMMEDIATE"
CEV10 - integer	CENV10 - distinto de integer
CEV11 - fecha <= fecha_posterior	CENV11 - fecha no válida
CEV12 - float	CENV12 - distinto de float

Valores límite:

Valor límite válido	Valor límite NO válido
VLV1 - valor límite 24	VLVN1 - 23 char
VLV2 - valor límite 24	VLVN2 - 25 char
VLV3 - 10 char	VLVN3 - 9 char
VLV4 - 10 char	VLVN4 - 25 char
VLV5 - mínimo de ≥ 2 cadenas separadas por espacio en blanco	VLVN5 - 1 cadena que no cumpla
VLV6 - primera cadena = 30	VLVN6 - primera cadena = 0
VLV7 - primera cadena = 31	VLVN7 - primera cadena = 32
VLV8 - segunda cadena = 12	VLVN8 - segunda cadena = 0
VLV9 - segunda cadena = 11	VLVN9 - segunda cadena = 13
VLV10 - tercera cadena = 2025	VLVN10 - tercera cadena = 2024
VLV11 - tercera cadena = 2050	VLVN11 - tercera cadena = 2052
VLV12 - longitud decimales ≤ 2	VLVN12 - longitud decimales = 3
VLV13 - amount = 10.00	VLVN13 - amount = 9.99
VLV14 - amount = 10000.00	VLVN14 - amount = 10000.01
VLV15 - amount = 10.01	
VLV16 - amount = 99999.99	

Segunda Función:

Datos:

```
{"IBAN": "ES9421000418450200051332",  
  "AMOUNT": "EUR 123.45"}
```

Gramática:

Datos:= <Campos1> <separador> <Campos2>

Campos1:= <Etiqueta_dato1> <Igualdad> <Valor_dato1>

Campos2:= <Etiqueta_dato2> <Igualdad> <Valor_dato2>

Etiqueta_dato:= <Comillas> <Valor1> <Comillas>

Valor_d1:= IBAN

Valor_dato:= <Comillas> <Valor1> <Comillas>

Valor1:= <códigoPaís> <numero>

CódigoPaís:= ES

Número:= 0-9 {23}

Etiqueta_dato:= <Comillas> <Valor2> <Comillas>

Valor_d2:= Amount

Valor_dato:= <Comillas> <Valor2> <Comillas>

Valor2:= <moneda> <cantidad> <punto> <cantidad_dec>

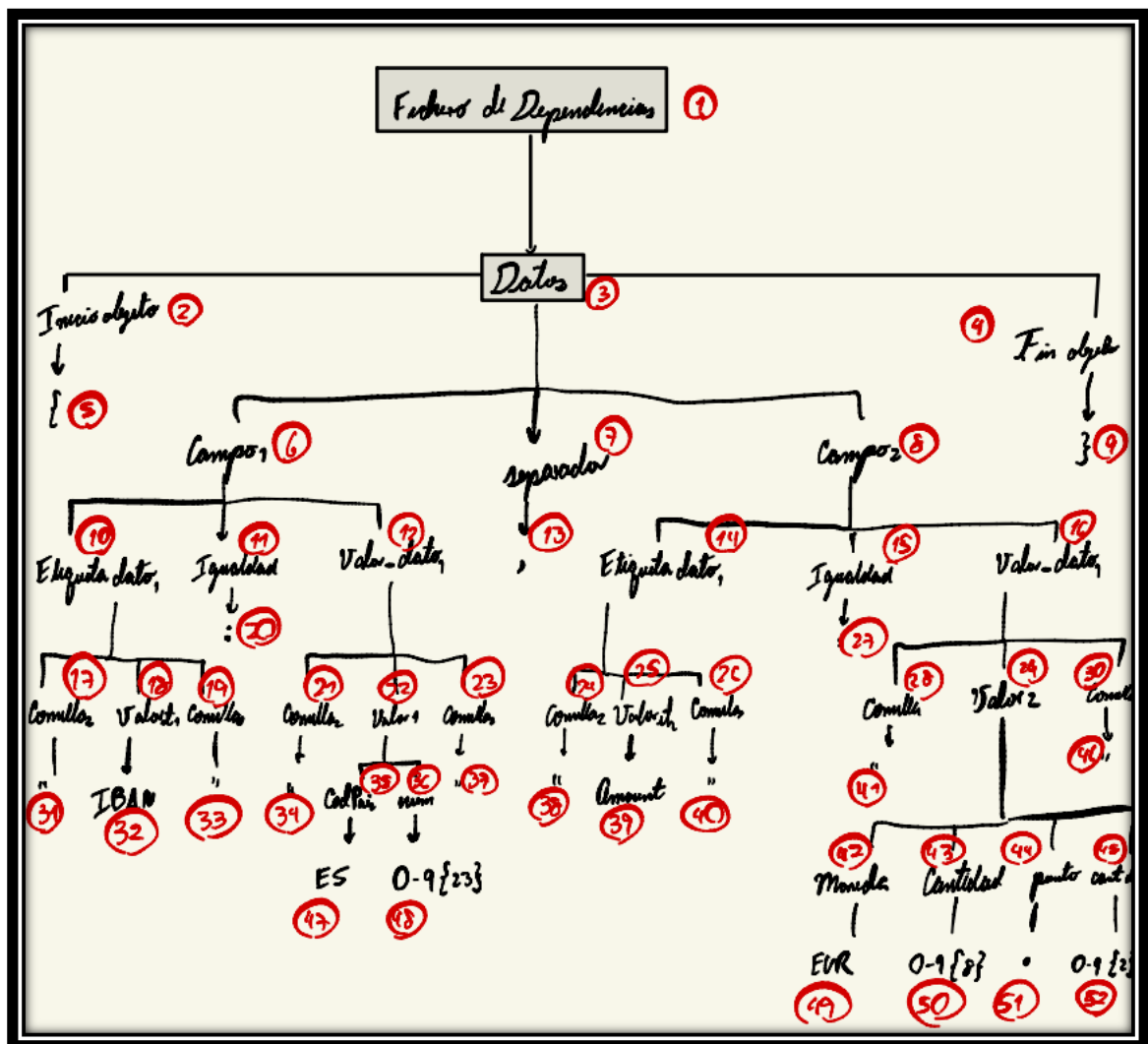
Moneda:= EUR

Cantidad:= 0-9 {3}

Cantidad_dec:= 0-9 {2}

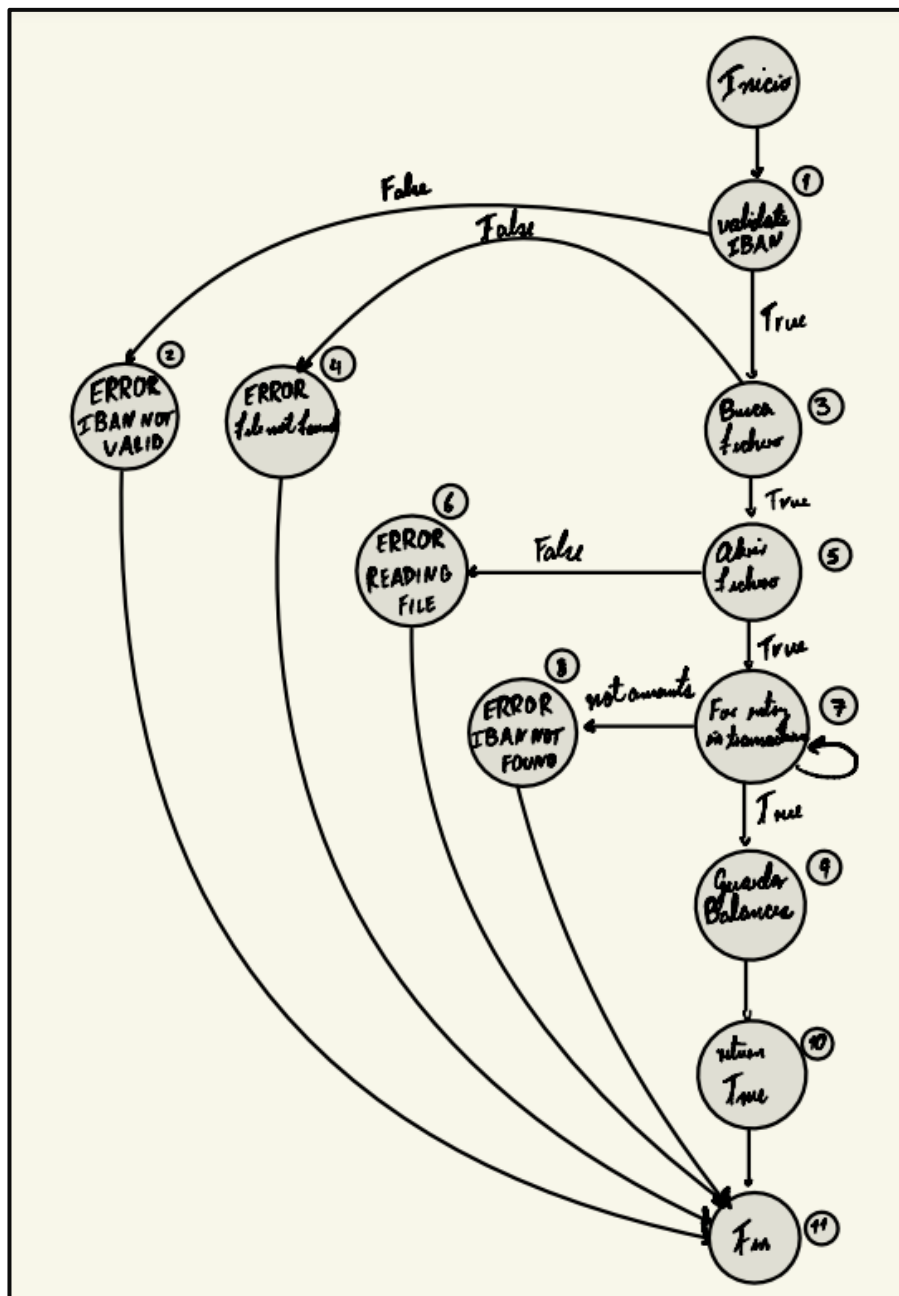
Punto:= .

Árbol con Nodos ordenados:



Tercera Función:

Gráfico de Control de Flujo:



Rutas básicas:

Camino básico: 1_3_5_7_9_10_11

Prueba básica: el IBAN aparece una vez, suma correcta.

Datos introducidos:

```
data = [  
  {  
    "IBAN": "ES9121000418450200051332",  
    "amount": "+123.45"  
  }  
]
```

Caminobucle1: 1_3_5_7_7_7_9_10_11

Varias entradas para el mismo IBAN, bucle ejecutado varias veces.

Datos introducidos:

```
iban = "ES9121000418450200051332"  
transactions = [  
  {"IBAN": iban, "amount": "+100.00"},  
  {"IBAN": iban, "amount": "+200.50"},  
  {"IBAN": iban, "amount": "-50.00"}  
]
```

CaminoERROR1: 1_2_11

El IBAN tiene un formato no válido

Datos introducidos:

```
data = [  
  {"IBAN": "ES999999999999999999999999", "amount": "+100.00"}  
]
```


CaminoERROR2: 1_3_4_11

Error file no encontrado el archivo de transacciones no existe:

CaminoERROR3: 1_3_5_6_11

Error formato JSON no válido, falta el cierre.

Datos introducidos

```
'{"IBAN": "ES9121000418450200051332", "amount": "+123.45"}'
```

CaminoERROR4: 1_3_5_7_8_11

El IBAN esperado no se encontró

Datos introducidos:

```
data = [  
  {"IBAN": "ES999999999999999999999999", "amount": "+50.00"},  
  {"IBAN": "ES888888888888888888888888", "amount": "-25.00"}  
]
```

Caminobucle2: 1_3_5_7_7_7_9_10_11

Una entrada tiene amount inválido, se ignora

Datos introducidos:

```
data = [  
  {"IBAN": "ES9121000418450200051332", "amount": "+100.00"},  
  {"IBAN": "ES9121000418450200051332", "amount": "NO_ES_NUMERO"},  
  {"IBAN": "ES9121000418450200051332", "amount": "+50.00"}  
]
```