



Universidad Carlos III
Curso 2023-24
Desarrollo de Software

Curso 2023-24

Definición de los casos de prueba

Alicia Sáenz de Zaítigui, 100495718
Luca Petidier Iglesias, 100496633
Marcos Romo Poveda, 100496072

Índice:

Introducción.....	2
Primera Función:.....	3
Segunda Función:.....	4
Tercera Función:.....	5

Introducción

En este documento se incluyen las diferentes definiciones de los casos de prueba para cada función de la práctica.

Para empezar, se aborda la primera función, donde se identifican las diferentes clases de equivalencia y se establecen los valores límite pertinentes para una evaluación completa y precisa. Seguidamente, se dedica atención a la segunda función, en la que se presenta la gramática asociada, junto con el árbol de derivación correspondiente. Finalmente, se aborda la tercera función, donde se profundiza en el diseño del flujo de control mediante la presentación de un grafo. Además, se definen las rutas básicas necesarias para una cobertura exhaustiva del código, y se discuten los casos adicionales requeridos para poner a prueba eficazmente los bucles y otros aspectos críticos de la función.

Primera Función, roomReservation:

Para la primera función hemos identificado un total de 7 valores límites válidos y 8 no válidos:

- ❖ VLV1: Tarjeta de crédito de 16 dígitos
- ❖ VLV2: Id_card de 9 cifras
- ❖ VLV3: Longitud de name_and_surname = 10
- ❖ VLV4: Longitud de name_and_surname = 50
- ❖ VLV5: Longitud de name_and_surname = 11
- ❖ VLV6: Longitud de name_and_surname = 49
- ❖ VLV7: Número de telefono de 9 dígitos
- ❖ VLNV1: Tarjeta de crédito de 17 dígitos
- ❖ VLNV2: Tarjeta de crédito de 15 dígitos
- ❖ VLNV3: Id_card de 10 cifras
- ❖ VLNV4: Id_card de 8 cifras
- ❖ VLNV5: Longitud de name_and_surname = 9
- ❖ VLNV6: Longitud de name_and_surname = 51
- ❖ VLNV7: Número de telefono de 8 dígitos
- ❖ VLNV8: Número de telefono de 10 dígitos

Para la primera función hemos identificado un total de 13 clases de equivalencias válidas:

- ❖ CEV1: Tarjeta de credito valida, formato str y de 16 dígitos
- ❖ CEV2: DNI válido, formato str y con 8 número y una letra
- ❖ CEV3: Nombre y Apellido válido, formato str y entre 10 y 50 caracteres
- ❖ CEV4: Un nombre y dos apellidos
- ❖ CEV5: Número de teléfono válido, formato str y de 9 dígitos
- ❖ CEV6: Tipo de habitacion single y formato str
- ❖ CEV7: Tipo de habitacion double
- ❖ CEV8: Tipo de habitacion suite
- ❖ CEV9: Fecha de llegada válida y formato str
- ❖ CEV10: 1 valor, formato str y entre 1 y 10
- ❖ CEV11: num_days = 10
- ❖ CEV12: num_days = 2
- ❖ CEV13: num_days = 9

Para la primera función hemos identificado un total de 28 clases de equivalencias no válidas:

- ❖ CENV1: Tarjeta de credito de 16 digitos pero uno no es valido
- ❖ CENV2: Tarjeta de credito en formato int
- ❖ CENV3: Tarjeta de credito de 17 dígitos
- ❖ CENV4: Tarjeta de credito de 15 dígitos
- ❖ CENV5: La letra del DNI no es válida
- ❖ CENV6: DNI con formato incorrecto
- ❖ CENV7: DNI con caracteres especiales
- ❖ CENV8: DNI en formato int
- ❖ CENV9: DNI con más de 9 cifras
- ❖ CENV10: DNI con menos de 9 cifras
- ❖ CENV11: Entre el nombre y el apellido no hay un espacio
- ❖ CENV12: Hay elementos no alfabéticos en el nombre o en los apellidos
- ❖ CENV13: Hay dos espacios seguidos entre el nombre o los apellidos
- ❖ CENV14: Nombre y apellidos en formato int
- ❖ CENV15: Longitud del nombre y apellidos mayor a 50
- ❖ CENV16: Longitud del nombre y apellidos menos a 10
- ❖ CENV17: Número de teléfono con elementos no numéricos
- ❖ CENV18: Número de teléfono en formato int
- ❖ CENV19: Número de teléfono con menos de 9 dígitos
- ❖ CENV20: Número de teléfono con más de 9 dígitos
- ❖ CENV21: Tipo de habitación triple
- ❖ CENV22: Tipo de habitación en formato int
- ❖ CENV23: Fecha de llegada no cumple el formato adecuado
- ❖ CENV24: Fecha de llegada en formato int
- ❖ CENV25: Número de días no es un número
- ❖ CENV26: Número de días en formato int
- ❖ CENV27: Número de días igual a 0
- ❖ CENV28: Número de días igual a 11

Segunda Función, guestArrival:

DEFINICIÓN GRAMÁTICA:

fichero -> <llave ini> <datos> <llave fin>

llave ini -> "{"

llave fin -> "}"

datos -> <dato 1><coma><dato 2>

coma -> ","

dato 1 -> <label 1><dos puntos><datos label 1>

dos puntos -> ":"

label 1 -> <comilla><label 1 contenido><comilla>

comilla -> " " "

label 1 contenido -> "Localizer"

datos label 1 -> <comilla><datos label 1 contenido><comilla>

datos label 1 contenido -> (String having 32 hexadecimal characters)

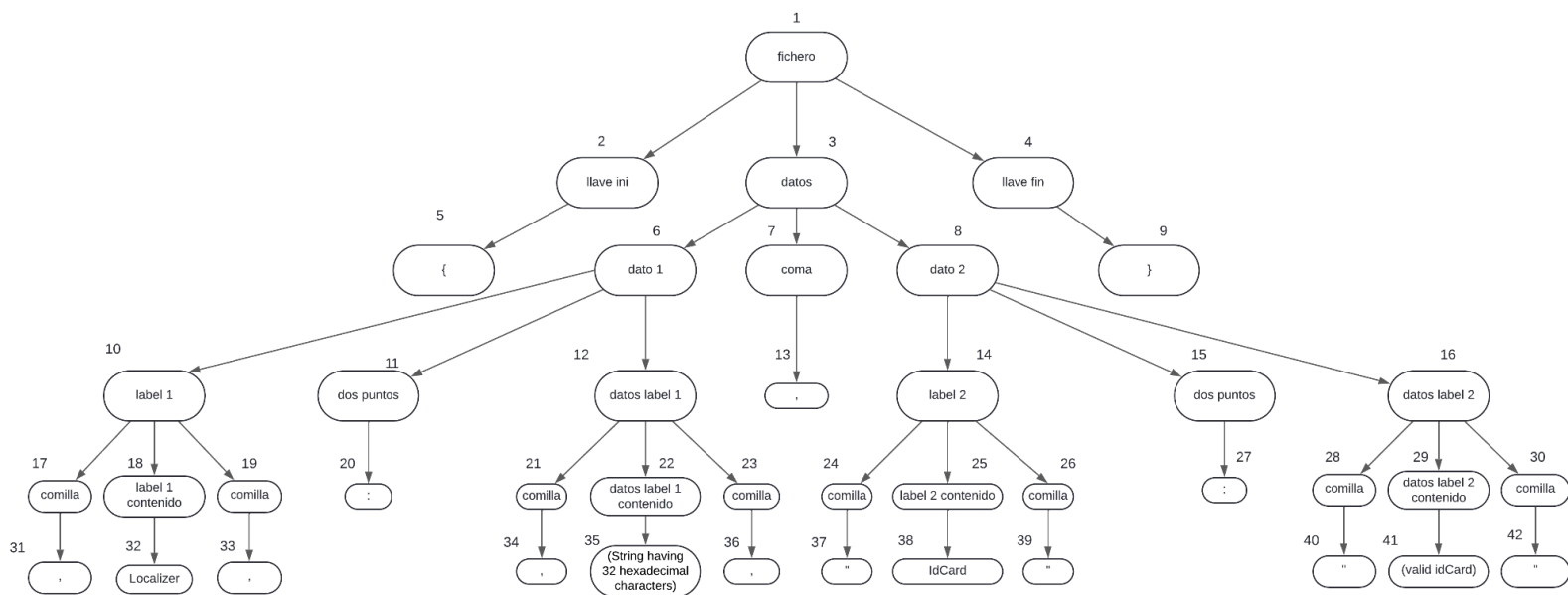
dato 2 -> <label 2><dos puntos><datos label 2>

label 2 -> <comilla><label 2 contenido><comilla>

label 2 contenido -> "IdCard"

datos label 2 -> <comilla><datos label 2 contenido><comilla>

datos label 2 contenido -> (valid idCard)



test_guest_arrival_valido	Nodos T válidos
test_guest_arrival_dup_1	Duplicamos nodo 1
test_guest_arrival_del_1	Eliminamos nodo 1
test_guest_arrival_dup_2	Duplicamos nodo 2
test_guest_arrival_del_2	Eliminamos nodo 2
test_guest_arrival_dup_3	Duplicamos nodo 3
test_guest_arrival_del_3	Eliminamos nodo 3
test_guest_arrival_dup_4	Duplicamos nodo 4
test_guest_arrival_del_4	Eliminamos nodo 4
test_guest_arrival_mod_5	Modificamos el nodo 5
test_guest_arrival_dup_6	Duplicamos nodo 6
test_guest_arrival_del_6	Eliminamos nodo 6
test_guest_arrival_dup_7	Duplicamos nodo 7
test_guest_arrival_del_7	Eliminamos nodo 7
test_guest_arrival_dup_8	Duplicamos nodo 8
test_guest_arrival_del_8	Eliminamos nodo 8
test_guest_arrival_dup_9	Modificamos el nodo 9
test_guest_arrival_dup_10	Duplicamos nodo 10
test_guest_arrival_del_10	Eliminamos nodo 10
test_guest_arrival_dup_11	Duplicamos nodo 11
test_guest_arrival_del_11	Eliminamos nodo 11
test_guest_arrival_dup_12	Duplicamos nodo 12
test_guest_arrival_del_12	Eliminamos nodo 12
test_guest_arrival_mod_13	Modificamos el nodo 13
test_guest_arrival_dup_14	Duplicamos nodo 14
test_guest_arrival_del_14	Eliminamos nodo 14
test_guest_arrival_dup_15	Duplicamos nodo 15
test_guest_arrival_del_15	Eliminamos nodo 15
test_guest_arrival_dup_16	Duplicamos nodo 16
test_guest_arrival_del_16	Eliminamos nodo 16
test_guest_arrival_dup_17	Duplicamos nodo 17
test_guest_arrival_del_17	Eliminamos nodo 17
test_guest_arrival_dup_18	Duplicamos nodo 18
test_guest_arrival_del_18	Eliminamos nodo 18
test_guest_arrival_dup_19	Duplicamos nodo 19
test_guest_arrival_del_19	Eliminamos nodo 19
test_guest_arrival_mod_20	Modificamos nodo 20
test_guest_arrival_dup_21	Duplicamos nodo 21
test_guest_arrival_del_21	Eliminamos nodo 21
test_guest_arrival_dup_22	Duplicamos nodo 22

test_guest_arrival_del_22	Eliminamos nodo 22
test_guest_arrival_dup_23	Duplicamos nodo 23
test_guest_arrival_del_23	Eliminamos nodo 23
test_guest_arrival_dup_24	Duplicamos nodo 24
test_guest_arrival_del_24	Eliminamos nodo 24
test_guest_arrival_dup_25	Duplicamos nodo 25
test_guest_arrival_del_25	Eliminamos nodo 25
test_guest_arrival_dup_26	Duplicamos nodo 26
test_guest_arrival_del_26	Eliminamos nodo 26
test_guest_arrival_mod_27	Modificamos nodo 27
test_guest_arrival_dup_28	Duplicamos nodo 28
test_guest_arrival_del_28	Eliminamos nodo 28
test_guest_arrival_dup_29	Duplicamos nodo 29
test_guest_arrival_del_29	Eliminamos nodo 29
test_guest_arrival_dup_30	Duplicamos nodo 30
test_guest_arrival_del_30	Eliminamos nodo 30
test_guest_arrival_mod_31	Modificamos nodo 31
test_guest_arrival_mod_32	Modificamos nodo 32
test_guest_arrival_mod_33	Modificamos nodo 33
test_guest_arrival_mod_34	Modificamos nodo 34
test_guest_arrival_mod_35	Modificamos nodo 35
test_guest_arrival_mod_36	Modificamos nodo 36
test_guest_arrival_mod_37	Modificamos nodo 37
test_guest_arrival_mod_38	Modificamos nodo 38
test_guest_arrival_mod_39	Modificamos nodo 39
test_guest_arrival_mod_40	Modificamos nodo 40
test_guest_arrival_mod_41	Modificamos nodo 41
test_guest_arrival_mod_42	Modificamos nodo 42

Tercera Función, guestCheckout:

```
def guestCheckout(self, room_key):  
    """Método que comprueba la salida del cliente"""
```

ASIGNACIÓN DE NODOS DEL GRAFO

```
FilePath = str(Path.home()) +  
"/PycharmProjects/G81.2024.T01.EG2/src/JsonFiles/check_in.json" 1  
  
# Apertura del fichero en modo lectura  
try:  
with open(FilePath, "r", encoding="utf-8", newline="") as File: 2  
KeyList = json.load(File) 3  
except FileNotFoundError: 4  
KeyList = []  
  
except json.JSONDecodeError as ex: 5  
raise HotelManagementException("JSON Decode Error - Formato JSON no  
válido") from ex  
  
# Búsqueda de la roomkey en el fichero para saber si es válida  
for Key in KeyList: 6  
Found = False 7  
if room_key == Key["_HotelStay__roomkey"]: 8  
Found = True 9  
Salida = Key 9  
if not Found: 10  
raise HotelManagementException("Código de habitación no registrado.") 11  
  
# Sólo se puede dejar la habitación en la fecha prevista.  
# Si la comparación no se cumple, la fecha no es válida.  
if Salida["HotelStay departure"] != self.fechaHoy(): 12  
raise HotelManagementException("La fecha de salida no es válida") 13  
  
# Se llama a la ruta del fichero donde se almacenan las salidas  
FileStore = str(Path.home()) +  
"/PycharmProjects/G81.2024.T01.EG2/src/JsonFiles/check_out.json" 14  
  
#Comprobamos que existe el fichero para las salidas  
try:  
with open(FileStore, "r", encoding="utf-8", newline="") as File: 15  
CheckOutList = json.load(File) 16  
except FileNotFoundError: 17  
CheckOutList = []  
  
except json.JSONDecodeError as ex: 18  
raise HotelManagementException("JSON Decode Error - Formato JSON no  
válido") from ex
```

#Guardamos los datos en el fichero para las salidas

```
CheckOutData = {"Room_key": room_key, "Time": self.fechaHoy()}  
CheckOutList.append(CheckOutData)
```

19

```
try:
```

20

```
with open(FileStore, "w", encoding="utf-8", newline="") as File:
```

```
    json.dump(CheckOutList, File, indent=2)
```

21

```
except FileNotFoundError as ex:
```

```
    raise HotelManagementException("Error en el path o archivo") from ex
```

22

```
return True
```

23

RUTAS POSIBLES

Nº	SECUENCIA DE NODOS
1	1-2-4
2	1-2-5
3	1-2-3-6-7-8-6-10-11
4	1-2-3-6-7-8-9-6-10-12-13
5	1-2-3-6-7-8-9-6-10-12-14-15-17
6	1-2-3-6-7-8-9-6-10-12-14-15-18
7	1-2-3-6-7-8-9-6-10-12-14-15-16-19-20-22
8	1-2-3-6-7-8-9-6-10-12-14-15-16-19-20-21-23

La primera ruta a seguir se daría en el caso de que hubiese un error al intentar abrir el archivo en el que se almacenan los check in.

La segunda ruta, se daría en el caso de que el formato del archivo de los check in no sea válido.

La tercera ruta, se daría en el caso de que el código de habitación no estuviese almacenado.

La cuarta ruta, se daría en el caso de que la salida no se haga en una fecha correcta.

La quinta ruta, se daría en el caso de que hubiese un error al intentar abrir el archivo en el que se almacenarán los check out.

La sexta ruta, se daría en el caso de que el formato del archivo para los check out no sea válido.

La séptima ruta, se daría en el caso de que ocurriese un error al abrir el archivo de los check out en modo escritura.

La última ruta se trata de la única que se debería de seguir para conseguir el objetivo de esta función sin ningún tipo de error.

