



Universidad Carlos III
Curso Desarrollo del Software 2021

Definición casos de prueba

GRUPO:80 EQUIPO:08

Alumnos: **Diego Robes Lozano (100429029)** / **Gonzalo Llosá Cea (100429091)**

Función 1. Solicitar el código de acceso	3
Casos DNI	3
Casos tipo de acceso	4
Casos nombre y apellidos	4
Casos email	5
Casos validez	6
Función 2. Acceso al edificio	7
Gramática del archivo JSON	7
Árbol de generación sintáctica	8
Función 3. Apertura de puertas	8
LLave existente y validez correcta	9
LLave existente y expiración 0	10
Llave existente y expiración no válida	10
Longitud de la llave incorrecta	10
La llave no es un string	10
La llave no existe en el almacén	10

Función 1. Solicitar el código de acceso

En esta funcionalidad dividiremos las distintas clases de equivalencia según el parámetro con el que esté relacionado, de este modo tendremos 5 campos diferentes: dni, tipo de acceso, nombre y apellidos, email y validez. Para la realización de todas estas pruebas será necesario comentar los campos justnow y time_stamp de la clase AccessRequest, ya que estos nos llevan a obtener diferentes hashes dependiendo del momento de la ejecución. Además el archivo json en el que se guardan estas peticiones debe estar vacío, ya que si se intenta repetir una petición que ya existe en el json el sistema lanzará una excepción tras comprobar que ya fue realizada.

- Casos DNI

- DNI correcto.

Se trata de probar la función cuando el DNI introducido es correcto, es decir, 8 números y a continuación una letra que sigue las reglas establecidas por el gobierno español.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.es	<hash válido>

- Solo números en el DNI.

Se trata de probar la función cuando el DNI está conformado solo por números, es decir, carece de la letra final.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.es	AccessManagerException: "DNI no válido"

- DNI con formato correcto pero la letra no sigue el algoritmo.

Se trata de probar la función cuando el DNI está en un formato correcto (8 números seguidos de una letra) pero esta letra no sigue el algoritmo establecido por el gobierno español.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463D	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.es	AccessManagerException: "DNI no válido"

- DNI conformado solo con letras.

Se trata de probar la función cuando el DNI está compuesto únicamente de letras, es decir, carece de números.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
FEUYDYD WYV	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.e s	AccessManagerException: "DNI no válido"

- DNI correcto pero la letra está en minúscula.

Se trata de probar el DNI cuando tiene un formato válido y su letra, siguiendo el algoritmo establecido por el gobierno español, está en minúsculas.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
FEUYYD WYV	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.es	AccessManagerException: "DNI no válido"

- Casos tipo de acceso

- Tipo de acceso válido. Valor Resident.

Se trata de probar el tipo de acceso cuando el valor es Resident.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	RESIDENT	JOSE LOPEZ	5	jllopez@inf.uc3m.es	<hash válido>

- Tipo de acceso válido. Valor Guest

Se trata de probar el tipo de acceso cuando el valor es Guets.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.es	<hash válido>

- Tipo de acceso inválido. Distinto de Guest o Resident

Se trata de probar el tipo de acceso cuando el valor es cualquier otro distinto a Resident o Guest.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	OWNER	JOSE LOPEZ	5	jllopez@inf.uc3m.es	AccessManagerException: "AccesType no válido"

- Casos nombre y apellidos

- Nombre y apellidos correctos

Se trata de comprobar el correcto funcionamiento cuando el nombre y apellido o apellidos introducidos son correctos.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.es	<hash válido>

- El nombre es un string vacío

Se trata de probar la función cuando se recibe un string vacío. En este caso deberá devolver un error.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	""	5	jllopez@inf.uc3m.es	AccessManagerException: "Nombre no válido"

- String sin ningún espacio en blanco

Se trata de probar la función cuando se recibe un string vacío. En este caso deberá devolver un error.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	""	5	jllopez@inf.uc3m.es	AccessManagerException: "Nombre no válido"

- Casos email

- Email válido

Se trata de probar el correcto funcionamiento cuando se introduce un email con el formato esperado, es decir: caracteres@caracteres.caracteres.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	jllopez@inf.uc3m.es	<hash válido>

- Email sin la "@"

Se trata de probar la función cuando se introduce un email al que le falta la "@"

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	jllopezinf.uc3m.es	AccessManagerException: "EMAIL NO VÁLIDO"

- Email sin dominio

Se trata de probar la función cuando se introduce un email sin el dominio

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	jllopez@infuc3mes	AccessManagerException: "EMAIL NO VÁLIDO"

- Email sin nada entre la "@" y el dominio

Se trata de probar la función cuando se introduce un email sin nada entre la "@" y el dominio

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	jllopez@.es	AccessManagerException: "EMAIL NO VÁLIDO"

- Email sin nada antes de la "@"

Se trata de probar la función cuando no se escribe nada antes de la "@"

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	5	@inf.uc3m.es	AccessManagerException: "EMAIL NO VÁLIDO"

- Casos validez

- Número dentro del rango. Límite superior

La función debe aceptar un número entre el 2 y el 15. En esta ocasión se prueba con el límite superior.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	15	jllopez@inf.uc3m.es	<Hash válido>

- Número dentro del rango. Límite inferior

La función debe aceptar un número entre el 2 y el 15. En esta ocasión se prueba con el límite superior.

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	2	jllopez@inf.uc3m.es	<Hash válido>

- Número específico para cuando el valor del tipo de acceso es Resident

La función debe aceptar el valor 0 en validity cuando el tipo de acceso es Resident

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	RESIDENT	JOSE LOPEZ	0	jllopezinf.uc3m.es	<Hash válido>

- Número fuera del rango. Por debajo del límite inferior

Se prueba un valor por debajo del límite inferior

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	1	jllopezinf.uc3m.es	AccessManagerException: "Número de días inválido"

- Número fuera del rango. Por encima del límite superior

Se prueba un valor por encima del límite superior

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	16	jllopezinf.uc3m.es	AccessManagerException: "Número de días inválido"

- No es un número

Se prueba cuando en lugar de un número se recibe cualquier otro tipo

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	GUEST	JOSE LOPEZ	SIEMPRE	jllopezinf.uc3m.es	AccessManagerException: "Número de días inválido"

- Distinto de 0 cuando el tipo de acceso es Resident

Se prueba cuando el valor de validity es distinto de 0 cuando el tipo de acceso es Resident

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	RESIDENT	JOSE LOPEZ	12	jllopezinf.uc3m.es	AccessManagerException: "Número de días inválido"

- Número dentro del rango. Límite superior menos 1

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	RESIDENT	JOSE LOPEZ	3	jllopezinf.uc3m.es	<Hash válido>

- Número dentro del rango. Límite inferior más 1

DNI	ACCESS TYPE	NAME	VALIDITY	EMAIL	RESULTADO ESPERADO
41694463V	RESIDENT	JOSE LOPEZ	14	jllopezinf.uc3m.es	<Hash válido>

Función 2. Acceso al edificio

- Gramática del archivo JSON

El símbolo inicial es “Fichero”. Los símbolos en cursiva y negrita son terminales.

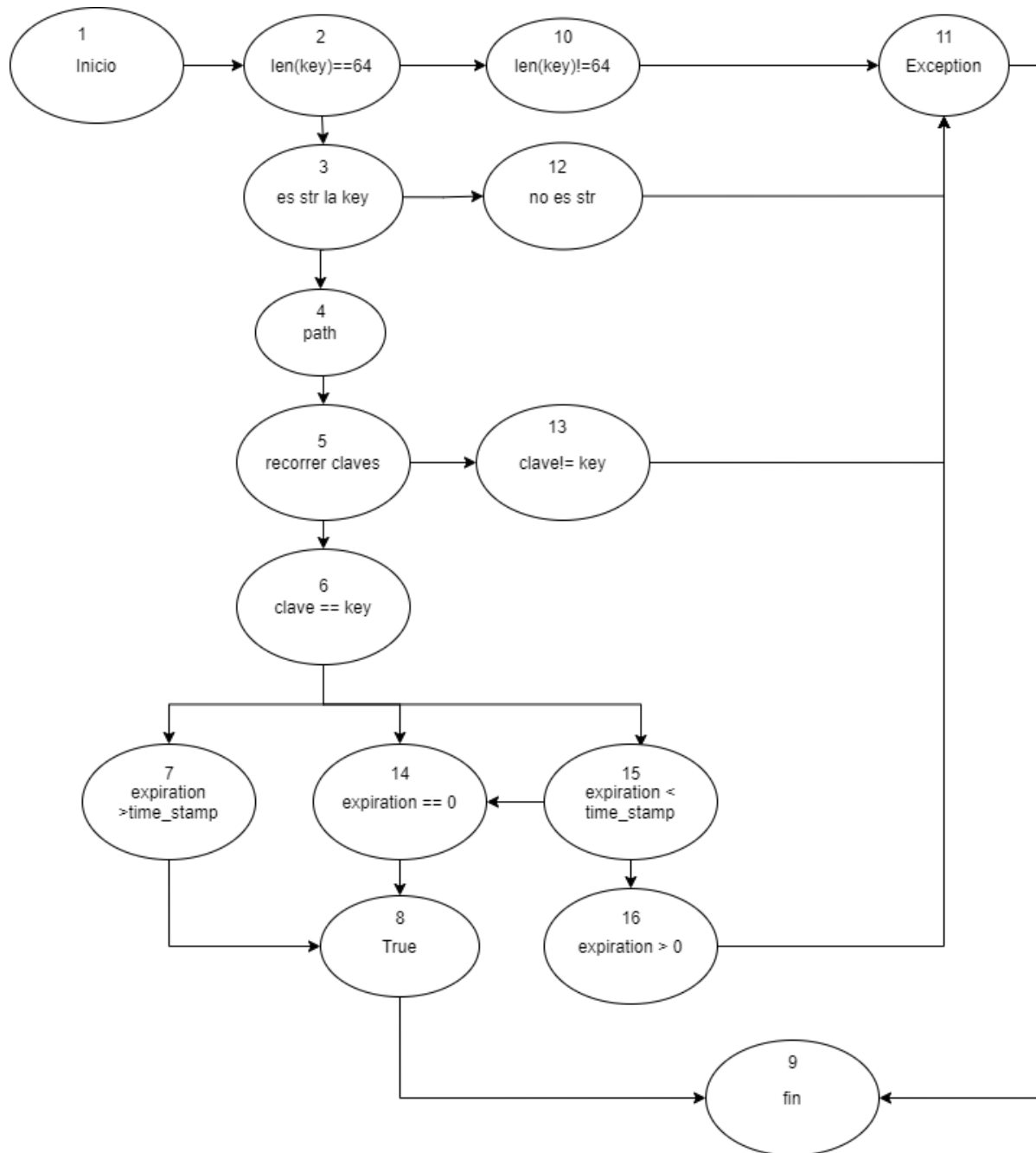
Diferentes símbolos separados por una coma seguida de un espacio.

- Fichero ::= Inicio_fichero, Contenido, Fin_fichero
- Inicio_fichero ::= { (terminal)
- Contenido ::= Dato1, Coma, Dato2, Coma, Dato3
- Fin_fichero ::= }
- Dato1 ::= Nombre1, Igual, Valor1
- Coma ::= ,
- Dato2 ::= Nombre2, Igual, Valor2
- Dato3 ::= Nombre3, Igual, Valor3
- Nombre1 ::= Comilla, String Nombre 1, Comilla
- Igual ::= :
- Valor1 ::= Comilla, String Valor 1, Comilla

- Árbol de generación sintáctica

[illegible]

El grafo de control de flujo que hemos realizado para la realización de las pruebas estructurales de este método es el siguiente:



Este grafo tiene 21 aristas y 16 nodos, por lo tanto el número de caminos distintos es 7. Aún así hemos detectado que hay dos caminos que llevan al mismo caso, estos son el 1-2-3-4-5-6-14-8-9 y el 1-2-3-4-5-6-15-14-8-9, por lo que hemos decidido implementar solo el test de uno de estos dos casos.

Los casos son los siguientes:

- **LLave existente y validez correcta**

Camino: 1-2-3-4-5-6-7-8-9

Se espera que el método devuelva True

- LLave existente y expiración 0

Camino: 1-2-3-4-5-6-14-8-9

Se espera que el método devuelva True

- Llave existente y expiración no válida

Camino: 1-2-3-4-5-6-15-16-11-9

Se espera que el método lance una AccessManagementException

- Longitud de la llave incorrecta

Camino: 1-2-10-11-9

Se espera que el método lance una AccessManagementException

- La llave no es un string

Camino: 1-2-3-12-11-9

Se espera que el método lance una AccessManagementException

- La llave no existe en el almacén

Camino: 1-2-3-4-5-13-11-9

Se espera que el método lance una AccessManagementException