



Universidad Carlos III
Curso Desarrollo de Software 2021-22
Práctica
Curso 2021-22

EG3 RF2 Y RF3

Fecha: **31/03/22**
GRUPO: **80** EQUIPO: **06**
Alumnos: **Jorge Álvarez Rodríguez / Alba Marco Ugarte**

ÍNDICE

| | |
|---------------------------------|----------|
| 1. Función RF2 | 3 |
| 1.1 Casos de prueba | 3 |
| 1.2 Gramática | 4 |
| 1.3 Árbol de derivación | 4 |
| 2. Función RF3 | 5 |
| 2.1 Gráfico de control de flujo | 5 |
| 2.2 Definición de rutas básicas | 8 |
| 3. Revisión grupo 7 | 9 |

1. Función RF2

1.1 Casos de prueba

En la función 2, encargada de obtener una cita para vacunarse, se han llevado a cabo múltiples pruebas que verifican el correcto funcionamiento de la misma. En total se han realizado 66 tests, en los cuales se comprueban todos los posibles errores que se generen en base a la gramática que previamente ha sido planteada.

Se distinguen dos tipos de casos de prueba:

- Test para un caso válido
- Test para un caso no válido

En el caso del código que hemos generado, únicamente se encuentra el test1_ok como un caso de prueba válido. En éste, el corchete inicial ha sido borrado y como resultado esperado se obtiene un hash válido para los valores introducidos.

En cuanto al resto del código, conformado por casos de pruebas no válidos, se diferencian cinco posibles tipo de error que serían mostrados al ocurrir una excepción. Los errores y tests que los muestran son los siguientes:

- “Error: el formato del JSON es incorrecto”: Los tests que muestran esto son del test2_not_ok al test30_not_ok; del test34_not_ok al test38_not_ok; del test41_not_ok al test44_not_ok; del test48_not_ok al test52_not_ok; del test55_not_ok al test59_not_ok; y del test61_not_ok al test66_not_ok. Por lo general, todos estos tests están relacionados con la duplicación, modificación y eliminación de símbolos, como la “,”,”:” o las comillas (“”).
- “Error: las keys del json no tienen el formato correcto”: Los tests que muestran esto son el test31_not_ok, test32_not_ok, test33_not_ok, test45_not_ok, test46_not_ok y test47_not_ok. Son aquellos relacionados con que PatientSystemID o ContactPhoneNumber ha sido modificado por otra palabra, duplicado o eliminado.
- “Error: el md5 no tiene 32 caracteres”: Los tests que muestran esto son el test39_not_ok y test40_not_ok. Son aquellos relacionados con que la ID hexadecimal no aparece o ha sido duplicada.
- “Error: el numero de telefono no tiene 9 digitos”: Los tests que muestran esto son el test53_not_ok y test54_not_ok. Son aquellos relacionados con que el número de teléfono del paciente no aparece o se encuentra duplicado.

- “Error: el md5 no es un hexadecimal”: El único test que muestra éste es el test60_not_ok. Salta el error debido a que la ID hexadecimal ha sido modificada y se le han añadido dos “z”.

1.2 Gramática

A continuación, se expone la gramática que hemos planteado para hacer posible la creación del árbol de derivación y evaluar correctamente todos los posibles casos de prueba.

$G = (\{ I, ID, AtribID, ValorID, HexID, PH, AtribU, ValorPH, Inicio, Final, Coma, Comillas, Puntos \}, \{ ^{([0-9][A-F])\{32\} \setminus Z}, \{ , , , , " , : , PatientSystemID, ContactPhoneNumber \}, P, I)$

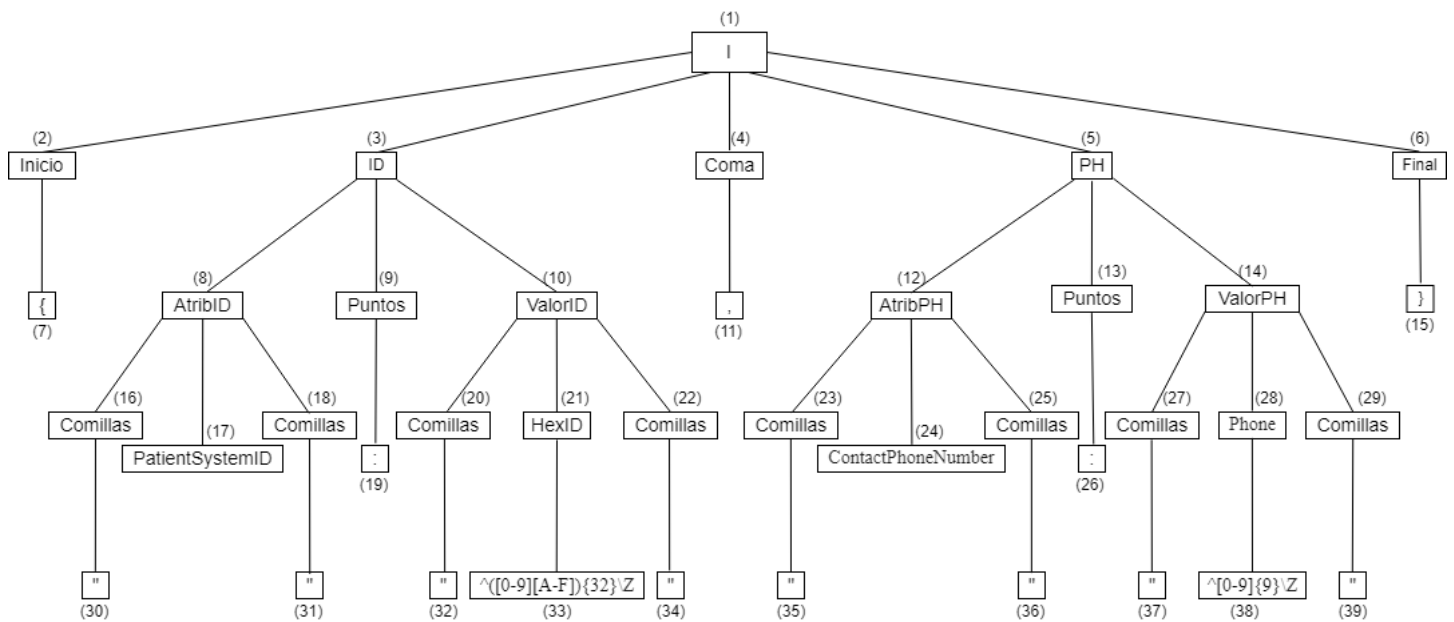
$P = \{$

- $I ::= Inicio ID Coma PH Final$
- $ID ::= AtribID Puntos ValorID$
- $AtribID ::= Comillas PatientSystemID Comillas$
- $ValorID ::= Comillas HexID Comillas$
- $HexID ::= ^{([0-9][A-F])\{32\} \setminus Z}$
- $PH ::= AtribPH Puntos ValorPH$
- $AtribPH ::= Comillas ContactPhoneNumber Comillas$
- $ValorPH ::= Comillas Phone Comillas$
- $Phone ::= ^{[0-9]\{9\} \setminus Z}$
- $Inicio ::= \{$
- $Final ::= \}$
- $Coma ::= ,$
- $Comillas ::= "$
- $Puntos ::= :$

$\}$

1.3 Árbol de derivación

A continuación, se muestra el árbol de derivación generado a partir de la gramática planteada anteriormente. Cada uno de los nodos que conforman el árbol han sido enumerados de arriba a abajo e izquierda a derecha, para así poder ofrecer un mejor entendimiento del mismo al generar el Excel explicativo de los casos de prueba.

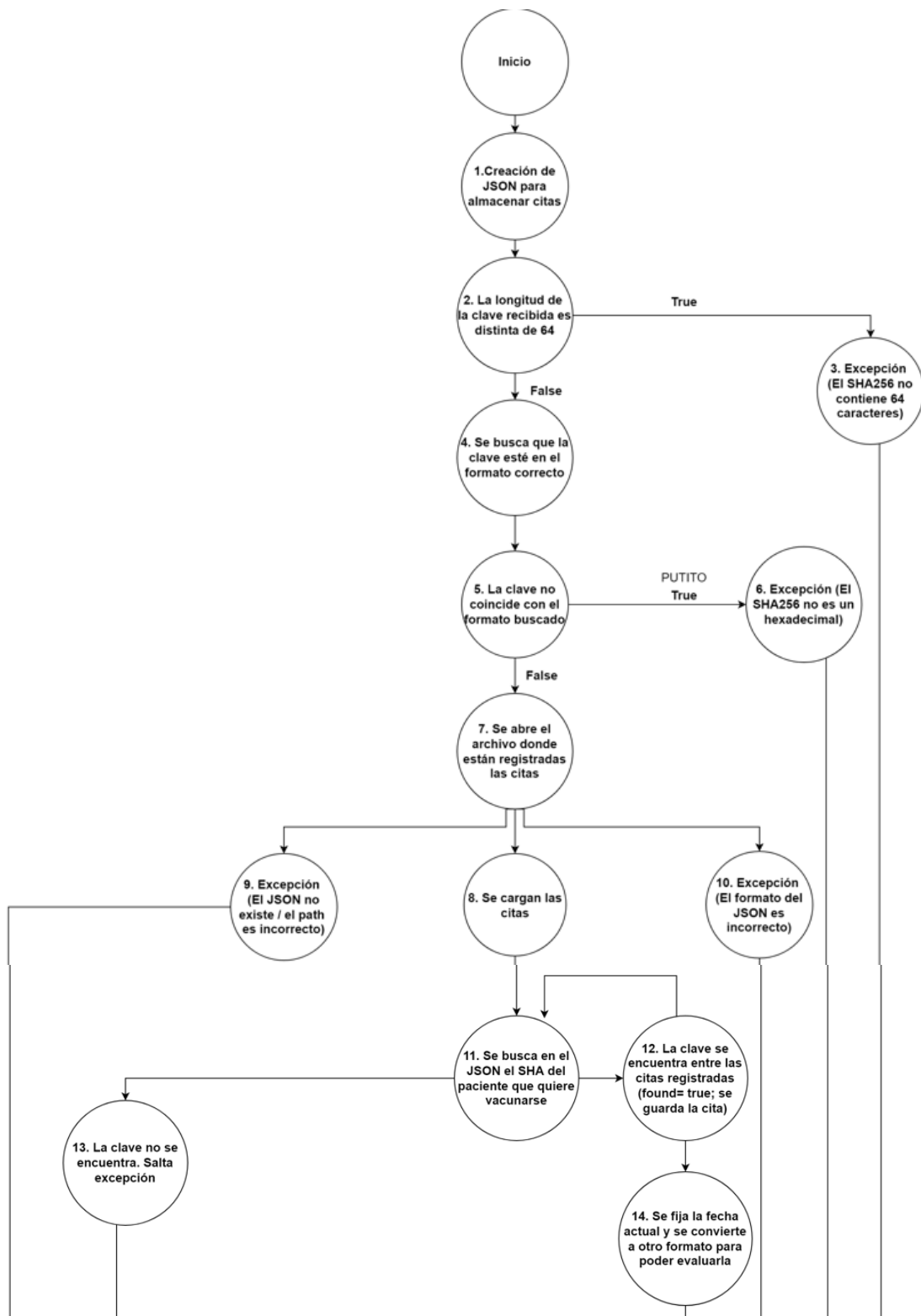


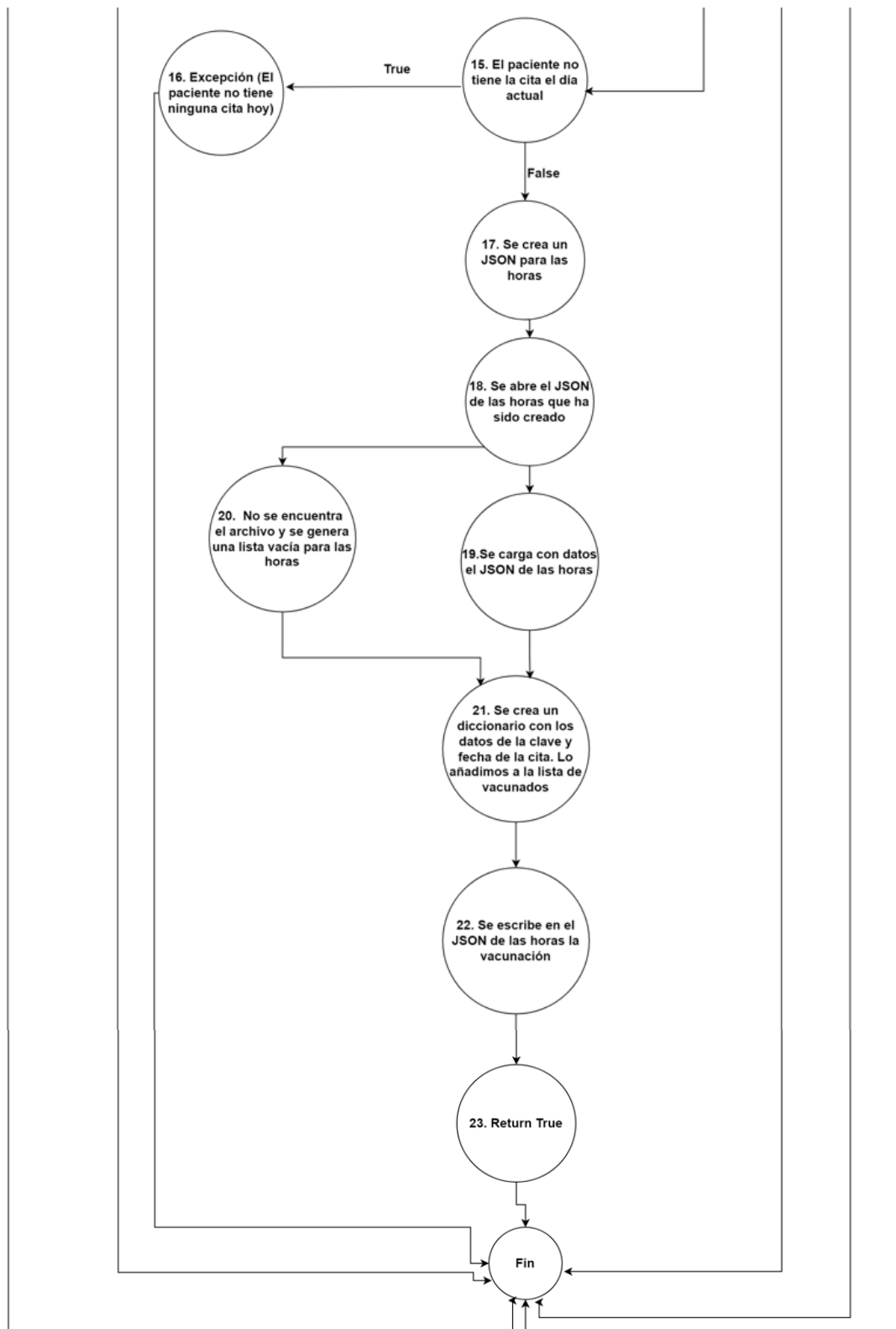
En el árbol se distinguen tanto nodos terminales (7, 11, 15...), como nodos no terminales (1, 2, 3...). En el documento Excel de la práctica, para los nodos no terminales hemos creado un test de duplicación y borrado. Sin embargo, los nodos terminales únicamente contienen tests de modificación. Esto se debe a que los nodos terminales son un subárbol que se repite y, por tanto, consideramos innecesario realizar de nuevo el mismo caso de prueba.

2. Función RF3

2.1 Gráfico de control de flujo

A continuación, se expone el diagrama de flujo que hemos diseñado para poder crear los tests / casos de prueba necesarios para la función tres (RF3), encargada de administrar la vacuna.





2.2 Definición de rutas básicas

En este apartado se explicará cada una de las rutas que se han llevado a cabo para la realización de los tests que verifican el correcto funcionamiento de la función tres (RF3).

Las rutas son nueve en total, y son las siguientes:

- 1) test1_ok: Recorre los nodos en el siguiente orden: 1, 2, 4, 5, 7, 8, 11, 12, 14, 15, 17, 18, 19, 22, 23, 24. Con esta ruta se consigue evaluar que la cita es correcta, el día registrado también lo es y se registra correctamente la administración de la vacuna.
- 2) test2_ok: Recorre los nodos en el siguiente orden: 1, 2, 4, 5, 7, 8, 11, 12, 14, 15, 17, 18, 20, 22, 23, 24. Con esta ruta se obtiene el mismo resultado que con el primer test, salvo que durante el proceso se recorre un nodo diferente que hace que genere una lista vacía para almacenar las horas, ya que no es capaz de encontrar el propio archivo JSON que corresponde a éstas.
- 3) test3_not_ok: Recorre los nodos en el siguiente orden: 1, 2, 3. Con esta ruta se obtiene una excepción durante la ejecución del proceso, que indica que el SHA256 recibido no cumple con la longitud esperada (64 caracteres).
- 4) test4_not_ok: Recorre los nodos en el siguiente orden: 1, 2, 4, 5, 6. Con esta ruta se obtiene una nueva excepción que salta al comprobar que el SHA256 recibido no cumple con el formato buscado (hexadecimal).
- 5) test5_not_ok: Recorre los nodos en el siguiente orden: 1, 2, 4, 5, 7, 9. Con esta ruta se obtiene una excepción que indica que el JSON no existe o el path es incorrecto, al abrir el archivo dónde están registradas las citas.
- 6) test6_not_ok: Recorre los nodos en el siguiente orden: 1, 2, 4, 5, 7, 10. Con esta ruta salta una nueva excepción que muestra un error de que el formato del JSON no es correcto. La excepción se lleva a cabo cuando se trata de abrir el archivo dónde están registradas las citas.
- 7) test7_not_ok: Recorre los nodos en el siguiente orden: 1, 2, 4, 5, 7, 8, 11, 13. Con esta ruta, tras buscar en el archivo JSON el SHA recibido del paciente que quiere vacunarse y no encontrar la clave, salta una excepción que muestra el mensaje “el paciente no tiene una cita registrada”.
- 8) test8_not_ok: Recorre los nodos en el siguiente orden: 1, 2, 4, 5, 7, 8, 11, 12, 14, 15, 16. Con esta ruta, tras haber encontrado el SHA del paciente que quiere vacunarse en el archivo JSON, se evalúa si la cita corresponde al día actual y cómo no se da el caso, salta una excepción.

3. Revisión grupo 7

Como le comentamos por correo, el grupo al que corregimos, formado por Ángel Casado y Jorge Francés, no nos ha llegado a hacer entrega de su Excel. Por tanto, no hemos podido realizar ningún tipo de corrección.