



*Práctica Final*  
*Desarrollo de Software*  
*Ingeniería informática - Grupo 81*

|         |   |
|---------|---|
| Fecha   | 23/05/2022  |
| Autores | Ángel del Pozo Manglano - 100442123<br>angel.pozo@alumnos.uc3m.es           |
|         | Francisco Antonio Gallardo Fuentes - 100451146<br>100451146@alumnos.uc3m.es |

# ÍNDICE

|   |          |
|---|----------|
| <b>INTRODUCCIÓN</b>                             | <b>2</b> |
| <b>CLASES DE EQUIVALENCIA Y VALORES LÍMITES</b> | <b>3</b> |
| <b>GRAMÁTICA Y ÁRBOL DE DERIVACIÓN</b>          | <b>4</b> |
| <b>DIAGRAMA DE FLUJO</b>                        | <b>5</b> |
| <b>CONCLUSIONES</b>                             | <b>6</b> |

# INTRODUCCIÓN

En esta práctica tenemos que aplicar todos los conocimientos adquiridos a lo largo del curso: Clases de Equivalencia y Valores Límites: hallar el rango de valores en los cuales se mueven nuestros parámetros de entrada y sus límites superiores e inferiores.

Pruebas Estructurales: realizar el Diagrama de Flujo para encontrar el número de caminos y por ende, encontrar los casos de prueba.

Pruebas Funcionales: realizar la Gramática y su correspondiente Árbol de Derivación para obtener más casos de prueba.

Proceso de Refactoring: limpieza de código, entendimiento sencillo para aumentar la eficiencia del programa.

Patrones de Diseño .

Todas ellas para desarrollar una nueva función en un código base dado por los profesores, en la cual tenemos que implementar una forma de que un paciente pueda cancelar su cita para la vacuna.

Además de incorporar un nuevo parámetro en la función `get_vaccine_date` para, en vez de poner una vacuna por defecto para dentro de 10 días, el paciente pueda incluir la fecha de vacunación que le gustaría.

## CLASES DE EQUIVALENCIA Y VALORES LÍMITES

En esta sección evaluaremos las clases de equivalencia y sus correspondientes valores límites de cada parámetro de la función `cancel_appointment`:

Nota: LS = Límite superior.

LI = Límite inferior.

| Campo            | Tipo de Entrada                       | Clases Válidas  | Clases Inválidas   | Valores Límites               |
|------------------|---------------------------------------|---|--|-------------------------------|
| date_signature   | String finito de 64 caracteres        | String exactamente igual de 64 caracteres                                     | String diferente a 64 caracteres.<br>Variable de tipo distinto a string  | LS = LI = 64 caracteres       |
| cancelation_type | Conjunto de valores {Temporal, Final} | Valores: Temporal o Final   | Valores distintos a temporal y final.<br>Variable de tipo distinto a string                                      | Temporal o Final              |
| reason           | String entre 2 y 100 caracteres       | String mayor o igual a 2 caracteres.<br>String menor o igual a 100 caracteres | Valores inferiores a 2 caracteres.<br>Valores superiores a 100 caracteres.<br>Variable de tipo distinto a string | LS = 100<br>LI = 2 caracteres |

Los casos de prueba se podrán consultar en el archivo `xlsx` de la carpeta `docs`.

# GRAMÁTICA Y ÁRBOL DE DERIVACIÓN

En esta sección aplicaremos la técnica Pruebas Funcionales para nuestra función:

cancel\_appointment:

En ella tenemos 3 campos: date\_signature, que acepta un string hexadecimal de 64 caracteres; cancelation\_type, que acepta los valores Temporal o Final y reason, que acepta un string entre 2 y 100 caracteres.

Su forma esquemática sería la siguiente:

```
{
  "date_signature": "<String having 64 hexadecimal characters>",
  "cancelation_type": "Temporal | Final",
  "reason": "<String with 2 - 100 characters>",
}
```

Si desarrollamos la gramática quedaría:

Fichero → CorcheteAbierto Datos CorcheteCerrado  
CorcheteAbierto = {  
CorcheteCerrado = }  
Datos = Campo1 Separador Campo2 Separador Campo3  
Separador = ,  
Campo1 = Etiqueta1 Igualdad ValorDato1  
Etiqueta1 = Comillas ValorEtiqueta1 Comillas  
ValorEtiqueta1 = date\_signature  
ValorDato1 = Comillas Valor1 Comillas  
Valor1 = a|b|c|e|f|0|1...|9| {64}  
Campo2 = Etiqueta2 Igualdad ValorDato2  
Etiqueta2 = Comillas ValorEtiqueta2 Comillas  
ValorEtiqueta2 = cancelation\_type  
ValorDato2 = Comillas Valor2 Comillas  
Valor2 = Temporal | Final  
Campo3 = Etiqueta3 Igualdad ValorDato3  
Etiqueta3 = Comillas ValorEtiqueta3 Comillas  
ValorEtiqueta3 = reason  
ValorDato3 = Comillas Valor3 Comillas  
Valor3 = a|b|c...|z {2-100}  
Comillas = “”  
Igualdad = :

Finalmente, su forma esquemática es (archivo svg en carpeta docs):

[Árbol de derivación](#)

## DIAGRAMA DE FLUJO

En esta sección aplicaremos la técnica Pruebas Estructurales para la función `cancel_appointment`.

Para ello hemos de conocer cuál es el Diagrama de Flujo correspondiente, el cual está expuesto a continuación:

## CONCLUSIONES