Práctica

P

PRINCIPIOS DE DESARROLLO DE SOFTWARE

2019-2020

---

Universidad Carlos III de Madrid

# Práctica Final 2019/2020

DESARROLLO DE UN COMPONENTE SOFTWARE

# Práctica Final

---

Universidad Carlos III de Madrid. Escuela Politécnica Superior

*Javier García Guzmán*

Sección 1

## Objetivos

El propósito de la practica final consiste en evaluar el nivel de aprendizaje que los estudiantes han conseguido en relación con las técnicas tratadas en la asignatura en relación con el Desarrollo Dirigido por Pruebas (TDD), refactoring y diseño simple con patrones.

## Caso Práctico

Transport4Future es una empresa que está especializada en el desarrollo de tecnologías para vehículos autónomos.

A fin de garantizar que solamente sensores y actuadores debidamente autorizados y registrados pueden formar parte de la red de un vehículo autónomo, se ha decidido evolucionar el componente de software para la gestión de tokens para los sensores y actuadores autorizados para conectarse al vehículo.

Este componente se desarrollará en Java, plataforma J2EE, y se entregará en forma de un JAR que proporcione acceso a los métodos de la interfaz de programación que permita su integración en los servicios de información gestionados por el vehículo.

En primer lugar, será necesario abordar una reorganización de código, mediante la cual, en el método VerifyToken se deberá decodificar el Token que se recibe mediante una cadena URL 64. Por tanto, habrá que eliminar el campo Token Value de la clase Token, de tal manera que ese valor no se pueda buscar directamente en el almacén Token Store.

A) Los nuevos requisitos funcionales establecidos para el componente y que se deben considerar para la práctica final son los siguientes:

- Se podrá desactivar un token emitido antes de que finalice su periodo de validez. La solicitud de revocación se realizará mediante un el fichero de entrada que se proporcionará, en formato JSON, los siguientes datos:

```
{
   "Token Value": "<Cadena de 64 caracteres hexadecimales>",
   "Type of revocation":"<Temporal | Final >",
    "Reason":"<Cadena de 100 caracteres como máximo >"
}
```

La funcionalidad proporcionará como resultado la dirección de e-mail (Notification e-mail) correspondiente con el token que se acaba de revocar.

A través de una excepción, se proporcionará información de los mensajes de error a proporcionar relativos a:

- El fichero de entrada tiene algún problema de formato o de acceso.
- El token que se quiere revocar no existe.
- El token que se quiere revocar ya ha caducado.
- El token que se quiere revocar ya está revocado en la misma modalidad.

La definición del método de la interfaz del componente que proporcionará esta funcionalidad es la siguiente:

```
String RevokeToken (String FilePath) throws TokenManagementException;
// String represents the notification e-mail obtenido como resultado del método
// String FilePath represents the path to the file including the input required for the functionality
// TokenManagementException represents the possible error situations
```

NOTA: La inclusión de esta funcionalidad podrá suponer la actualización de funcionalidades previamente desarrolladas.

- El componente de gestión permitirá utilizar el token para realizar operaciones en la red del vehículo.
  - Ruta del fichero con los datos necesarios del dispositivo para la cual se quiere ejecutar la operación. El fichero de entrada debe cumplir el siguiente formato en JSON:

```
{
   "Token Value": "<Cadena de 64 caracteres hexadecimales>",
   "Type of operation":"<Send Information from Sensor| Send Request to Actuator | Check State >"
}
```

  - El componente verificará que el token existe y que actualmente es válido

- En este momento, el componente generará como salida un booleano que indicará si la acción puede ser ejecutada correctamente. Si la operación es relativa a un token correspondiente con un sensor, solo podrá ser ejecutado por un token correspondiente a un sensor, si la operación es relativa a un actuador, solo podrá ser ejecutada mediante un token de actuador y la operación Check State estará autorizada a sensores y actuadores.
- Un mensaje de error ante las siguientes situaciones:
    - El fichero de entrada tiene algún problema de formato o de acceso.
    - El token que se quiere utilizar no existe o no es válido actualmente.
    - La operación solicitada no puede ser realizada con el token adjuntado a la solicitud.

La definición del método de la interfaz del componente que proporcionará esta funcionalidad es la siguiente:

```
boolean ExecuteAction (String FilePath) throws TokenManagementException;
// Boolean represents the success in the action execution
// String FilePath represents the path to the file including the input required for the functionality
// TokenManagementException represents represents the possible error situations
```

Sección 2

## Reglas y Procedimientos

La práctica final se completará en parejas.


### Documentación a entregar

You must register the code in the Git repository indicated by the teacher of practices for this final practice.

La solución de la práctica final debe estar compuesta por la siguiente información:

A. Se deben aplicar las técnicas de Desarrollo Dirigido por pruebas mediante la definición y automatización de un plan de pruebas para las nuevas funcionalidades públicas que tiene que proporcionar la nueva versión del componente. Los casos de prueba propuestos se realizarán combinando las técnicas de prueba que el usuario considere más apropiado en cada caso. Esta cuestión se tendrá en cuenta en la evaluación de la práctica final. La información a entregar en este punto se compondrá de:
   - Un documento PDF que indique las clases de equivalencia, valores límite, gramáticas o árboles de derivación que se utilicen para la identificación de los casos de prueba para cada uno de los métodos de la interfaz pública de programación definidos para el componente desarrollado.
   - El código fuente que permite automatizar los casos de prueba definidos en el anterior documento, debidamente comentados. Los métodos de los casos de prueba deben estar comentados de la siguiente manera:

```
/* Caso de Prueba: <Id - Nombre>
* Clase de Equivalencia, Valor Límite o Nodo del Árbol de Derivación Asociado: <Valor>
* Técnica de prueba: <Clases de Equivalencia | Valor Límite | Análisis Sintáctico>
* Resultado Esperado: <Descripción>
*/
```

   - Un registro, generado por JUnit, que incluya los resultados de cada caso de prueba incluidos en el conjunto de pruebas.

B. Código fuente y ejecutable con la implementación de las funcionalidades solicitadas. Es imprescindible que el código implementado esté correctamente reorganizado, satisfaga el patrón singleton para las clases TokenManager y TokenStore y TokenRequestStore, así como el patrón Strategy para las clases que lean distintos tipos de ficheros JSON y las que generan un Hash utilizando distintos tipos de algoritmos.

Se debe registrar en el repositorio de Git que el profesor de prácticas indique para la práctica final.

**La fecha límite para la entrega del ejercicio es 28/05/2020 antes de 23:55.**

De acuerdo con las normas de evaluación en esta asignatura, si un estudiante no envía la solución del ejercicio antes de la fecha límite, el ejercicio será evaluado con 0 puntos.

## Criterios de Evaluación

Los criterios de evaluación serán los mismos que los criterios establecidos para cada una de las partes solicitadas y que están especificados en los correspondientes ejercicios guiados.

## Sugerencias

Cada estudiante debe guardar una copia de la solución entregada hasta la publicación de las calificaciones finales de la asignatura.