



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Gº en Ingeniería en Informática



**Trabajo final del Gº Ing. Informática:**

**App para el diagnóstico de fallos en  
fresadoras**



Presentado por Juan Francisco Benito Cuesta  
en junio de 2018  
Tutor Raúl Marticorena Sánchez





**I -**

**II -**

**III -**

**IV -**

**V -**

**VI -**

**VII -**

**VIII -**

**IX -**

**X -**





## Índice de contenido

Índice de ilustraciones.....	1	2.1.Librería externa para parsear JSON. 5
Índice de tablas.....	1	XV -Aspectos relevantes del desarrollo del
XI -Introducción.....	3	proyecto .....5
XII -Objetivos del proyecto.....	3	XVI -Trabajos relacionados .....6
XIII -Conceptos teóricos.....	3	XVII -Conclusiones y líneas de trabajo futuras
1.JSON.....	3	.....6
2.API REST.....	4	XVIII -referencias.....7
XIV -Técnicas y herramientas.....	4	Bibliografía.....7
1.Metodología usada.....	5	Índice alfabético.....7
2.Alternativas estudiadas.....	5	

## Índice de ilustraciones

Ilustración 1: Ejemplo de la alarma 713.....	4	Ilustración 2: Modelo Scrum.....	6
--	---	----------------------------------	---

## Índice de tablas

Tabla 1: Alternativas estudiadas.....	5
---------------------------------------	---





## Resumen

Aplicación en Java para Android que diagnostica los fallos de fresadoras a partir de las alarmas que producen estas. Realiza una serie de preguntas, dependiendo de la alarma que se haya activado, y el usuario debe ir eligiendo las respuestas según el estado de la fresadora.

Una vez finalizadas las preguntas, el problema se habrá resuelto o la aplicación te dirá una posible solución.

Estas alarmas con sus respectivas preguntas y respuestas están definidas en ficheros JSON. La aplicación obtiene de un API REST la alarma o alarmas que se han activado y parsea los ficheros JSON correspondientes para obtener las preguntas y respuestas de dichas alarmas.

## Abstract

*Application in Java for Android that diagnoses the failures of milling machines from the alarms that produce these. It asks a series of questions, depending on the alarm that has been activated, and the user must choose the answers according to the state of the milling machine.*

*Once the questions are finished, the problem will have been solved or the application will tell you a possible solution.*

*These alarms with their respective questions and answers are defined in JSON files. The application obtains from an REST API the alarm or alarms that have been activated and parses the corresponding JSON files to obtain the questions and answers of those alarms.*



## **XI - INTRODUCCIÓN**

El trabajo consta de un proyecto de Android Studio, varios ficheros JSON y una serie de imágenes.

- El proyecto es la aplicación en sí, que consta de la funcionalidad: los activity de cada pantalla y el modelo de clases de las alarmas, y de la interfaz gráfica: los layouts.
- Los ficheros JSON representan las diferentes alarmas que pueden saltar en las fresadoras. Cada fichero es una alarma y consta de su número, título, descripción, preguntas, respuestas y sus imágenes asociadas.
- Las imágenes son ayudas visuales que se aportan para resolver la situación. Cada alarma puede tener una o varias imágenes.

Esta memoria se divide en unos objetivos que se persiguen con la realización del proyecto, unos conceptos teóricos para explicar algunos detalles del trabajo, las técnicas y herramientas utilizadas en la elaboración del proyecto, aspectos relevantes a mencionar, trabajos que pueden tener relación con este, una serie de conclusiones y la bibliografía.

## **XII - OBJETIVOS DEL PROYECTO**

Los objetivos principales del proyecto son:

- Desarrollar una aplicación para que los empleados de una empresa puedan diagnosticar fácilmente los fallos de fresadoras.
- Utilizar Java para Android para desarrollar dicha aplicación.
- Realizar una interfaz gráfica usable y amigable para el usuario.
- Que dicha interfaz tenga los mismos colores, la misma fuente y logo de la empresa.
- Que la aplicación conecte con una API REST para la obtención de las alarmas activadas.
- Representar el título, descripción, preguntas y respuestas de las alarmas en ficheros JSON.
- Que la aplicación parsee dichos ficheros.

## **XIII - CONCEPTOS TEÓRICOS**

A continuación se explican brevemente algunos conceptos que se usan en el trabajo y sirve de ayuda entenderlos.

### **1. JSON**

JSON (JavaScript Object Notation) es un formato de texto ligero que representa datos estructurados y se usa para su intercambio.

Puede ser utilizado independientemente de JavaScript, ya que muchos contextos de programación tienen la capacidad de parsearlo y generarlo. Es una alternativa al lenguaje XML, siendo JSON más fácil de usar.



Los tipos de datos que puede representar son: números, cadenas (string), booleanos, null,



arrays de cualquiera de los tipos anteriores y propios objetos JSON.

```
{
  "Número": 713,
  "Título": "AUTOMÁTICO BOMBA ASPIRACIÓN SALTADO",
  "Descripción": "La entrada E461 ha cambiado a 0, indicando que el térmico G11-Q21 se ha saltado.",
  "Preguntas": [
    {
      "Id": "1.0",
      "Texto": "¿Físicamente, el térmico G11-Q21 se ha saltado por consumo?",
      "Imagen": "",
      "Respuestas": [
        {
          "Id": "A",
          "Texto": "Sí, está saltado.",
          "Camino": "1.1",
          "Mensaje": ""
        },
        {
          "Id": "B",
          "Texto": "No, no está saltado.",
          "Camino": "0.0",
          "Mensaje": "Comprobar cableado circuito auxiliar del térmico y la entrada del autómata."
        }
      ]
    },
    {
      "Id": "1.1",
      "Texto": "Si se rearma a mano, ¿se vuelve a saltar de nuevo al poco tiempo?",
      "Imagen": "",
      "Respuestas": [
        {
          "Id": "A",
          "Texto": "Sí, se vuelve a saltar.",
          "Camino": "0.0",
          "Mensaje": "Comprobar consumo de bomba y/o cambiarla."
        },
        {
          "Id": "B",
          "Texto": "No, no se vuelve a saltar.",
          "Camino": "0.0",
          "Mensaje": "Dejar bomba en observación."
        }
      ]
    }
  ],
  "Imágenes": [
    "a713",
    "a713_2"
  ]
}
```

**Ilustración 1: Ejemplo de la alarma 713**

## 2. API REST

API REST (Application Programming Interface Representational State Transfer) es una interfaz entre sistemas que utiliza el protocolo HTTP para obtener datos o realizar operaciones sobre esos datos en cualquier formato posible, como los mencionados anteriormente JSON y XML.

Las operaciones más importantes que podemos hacer son: POST (crear), GET (leer y obtener), PUT (modificar) y DELETE (eliminar). Para acceder o manipular esos datos u objetos de la REST se utiliza la URI, que es su identificador único.

## XIV - TÉCNICAS Y HERRAMIENTAS

La herramienta de desarrollo principal que se ha usado para llevar a cabo el proyecto ha sido el programa Android Studio. Se trata de un entorno de desarrollo integrado creado por Google, y por tanto es el IDE oficial para la creación de aplicaciones para Android. Con ella se ha programado la funcionalidad de las diferentes pantallas y su diseño gráfico.

También se ha utilizado Eclipse para el desarrollo de la aplicación en modo texto, y StarUML para su diagrama de clases correspondiente.

Para la creación de los ficheros JSON se ha usado los editores de código fuente Notepad++ y Sublime Text.

Para la realización del prototipo de la aplicación se ha utilizado el programa Pencil.



## 1. Metodología usada

La metodología usada para el desarrollo del software ha sido Scrum, planificando y realizando tareas cada dos semanas. Este control de las tareas se ha llevado a cabo con un repositorio en GitHub, realizando un commit por cada tarea.

Además, en este repositorio se encuentran todos los archivos que se han ido utilizando durante el trabajo.

Para almacenar y estructurar la información de las alarmas se ha utilizado los ficheros JSON mencionados en los otros apartados.

Para representar una alarma en el proyecto de Android Studio se ha empleado un modelo de clases Answer, Question y Alarm, con sus diferentes atributos y métodos.

Y para guardar las alarmas que se van parseando se ha creado una clase singleton, para que de esta forma se tenga un acceso global a la misma. Dicha clase utiliza un LinkedHashMap, siendo las claves los números de las alarmas y los valores los objetos alarmas. Se ha decidido que sea esta estructura de datos para que las alarmas se guarden en el mismo orden en el que se parsean.

## 2. Alternativas estudiadas

Alternativa	
Librería externa para parsear JSON	

Tabla 1: Alternativas estudiadas

### 2.1. Librería externa para parsear JSON

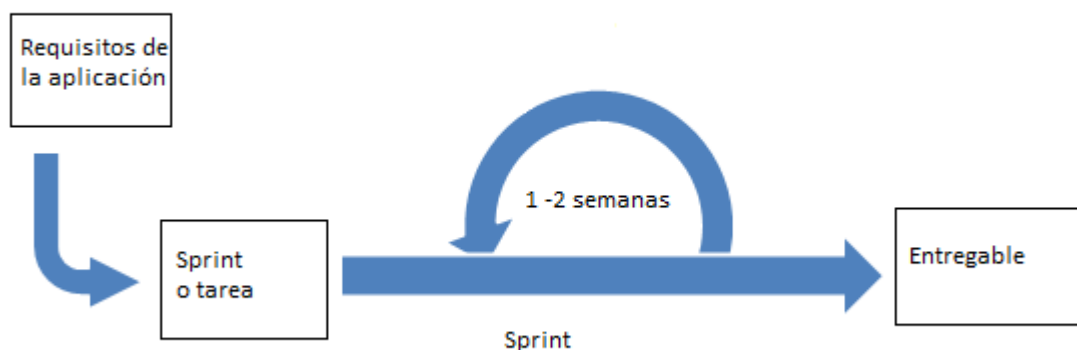
Al principio me descargué de Internet una librería para parsear ficheros JSON, que la utilicé en la aplicación en modo texto desarrollada en Eclipse. Cuando empecé con Android Studio mi intención era también usarla, pero descarté esta opción porque Android ya tiene librerías propias para parsear JSON.

## XV - ASPECTOS RELEVANTES DEL DESARROLLO DEL PROYECTO

El ciclo de vida utilizado para el desarrollo del proyecto ha sido Scrum, que es un tipo de método ágil. Este ha consistido en que a partir de unos requisitos para el avance de la aplicación, se han ido realizando sprints o tareas cada dos semanas para el cumplimiento de dichos requisitos. De esta forma se obtiene un “entregable” después de ese tiempo y se va avanzando en el desarrollo de la aplicación. Después de los dos primeros meses el tiempo entre sprints pasó a ser de una semana, para llevar un mayor control del proyecto.







**Ilustración 2: Modelo Scrum**

Durante este desarrollo se ha seguido una serie de fases:

- Fase de análisis: en esta primera fase se toman las decisiones de cómo hacer el proyecto y qué pasos se han de seguir. Como por ejemplo de qué forma representar y estructurar las alarmas, cómo debe ser la interfaz de la aplicación, etc.
- Fase de diseño: en esta segunda fase se realiza el prototipo de la aplicación con la herramienta Pencil y se elabora un primer proyecto en Eclipse que representa la aplicación en modo texto.
- Fase de implementación: en esta última fase se realiza el desarrollo de la aplicación en la herramienta Android Studio, teniendo en cuenta todo lo que se ha hecho anteriormente.

Algún aspecto del proyecto a comentar es la representación de las alarmas en los ficheros JSON.

Mencionar que las preguntas están identificadas con un id de tipo double, y que gracias a esto podemos diferenciar dos tipos de preguntas: las principales que serían las 1.0, 2.0, 3.0, etc., y las preguntas que llevan en el enunciado una posible solución y derivan de las principales, que serían las 1.1, 1.2, 2.1, etc.

De las respuestas comentar que se identifican por letras: A, B, C, etc. y que cada una de ellas tiene un camino a seguir. Este camino es un valor de tipo double que puede ser 0.0 si la respuesta ya no tiene continuación, en ese caso se mostraría un mensaje final, o puede ser el id de una pregunta, que en este otro caso se mostraría dicha pregunta.

También los ficheros JSON contienen un array con el nombre de los archivos de las imágenes que están asociadas a la alarma. Una pregunta también puede tener asociada una imagen, y está representada en el fichero de la misma forma.

## **XVI - TRABAJOS RELACIONADOS**

## **XVII -CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS**

Las conclusiones sobre las herramientas utilizadas durante el proyecto son:



- El lenguaje JSON es de mucha utilidad a la hora de estructurar datos. Es más sencillo, más fácil de comprender y de parsear que el XML.
- El IDE Android Studio me parece una potente herramienta para la creación y desarrollo de aplicaciones. Te permite una gran variedad de opciones y funcionalidades para construir tu proyecto.

Las conclusiones sobre los resultados del proyecto son:

- Me parece buena idea “traducir” la colección de preguntas y respuestas que tiene la empresa sobre cada alarma, en una aplicación sencilla que te permita ir respondiendo y navegando sobre esas preguntas hasta llegar a una posible solución final. Me parece útil para los empleados de la empresa que dispongan de esta aplicación y puedan usarla cada vez que falle una fresadora.

El proyecto se puede mejorar o continuar en los siguientes aspectos:

- Que un empleado pueda iniciar sesión en la aplicación y pueda ver un registro de su actividad: las veces que ha utilizado la aplicación, cuando lo ha hecho, las alarmas que ha consultado, etc.
- Que se puede buscar por palabras clave la alarma correspondiente. Un usuario introduciría varias palabras que tengan que ver con el fallo de la fresadora, y la aplicación te mostraría la alarma o alarmas que más se aproximen a lo que haya escrito el usuario.

## **XVIII - REFERENCIAS**

### **Bibliografía**

[García et al99] G<sup>a</sup>Peñalvo,FJ; Maudes Raedo,JM; Piattini,MG; G<sup>a</sup>-Bermejo,JR; Moreno,MN: «Proyecto fin de carrera en la ingeniería técnica en informática» 1999

### **Índice alfabético**

R

Referencia.....7



