

Manual de usuario del sistema Trivial-4a

Arquitectura del Software



3 de marzo de 2015 v1

Escuela Ingeniería Informática Oviedo



ÍNDICE

[DESCRIPCIÓN 2](#_Toc414223254)

[IMPLEMENTACIÓN DEL SISTEMA 2](#_Toc414223255)

[a) Requisitos de hardware 2](#_Toc414223256)

[b) Requisitos de software 2](#_Toc414223257)

[INGRESANDO AL SISTEMA 2](#_Toc414223258)

[INSTALACIÓN DE MongoDB 2](#_Toc414223259)

[FUNCIONAMIENTO DEL PARSER 3](#_Toc414223260)

[EJECUCIÓN EN LÍNEA DE COMANDOS 3](#_Toc414223261)

[EJECUCIÓN CON INTERFAZ GRÁFICA 4](#_Toc414223262)

[INTRODUCCIÓN DE NUEVAS PREGUNTAS 5](#_Toc414223263)

El propósito de este Manual es facilitar al usuario la navegación de las diferentes pantallas de la aplicación para poder usar correctamente el Trivial.

# DESCRIPCIÓN

Para la v1 de la entrega se ha implementado un Parser que nos permitirá transformar las preguntas en formato GIFT en preguntas con formato JSON, lo que permitirá su almacenamiento en una base de datos no relacional más fácilmente.

Para esta primera versión, las preguntas se almacenarán en ficheros agrupadas por categorías que mediante el parser se convertirán a JSON y se incluirán en la base de datos

# IMPLEMENTACIÓN DEL SISTEMA

### Requisitos de hardware

* Ordenador personal: Cualquier tipo de ordenador con unos requisitos mínimos para poder soportar la aplicación y base de datos es válido

### Requisitos de software

* **GitHub:** Herramienta para el control de versiones en el desarrollo de este proyecto. Se ha desarrollado la aplicación mediante cuatro ramas: tres ramas correspondientes cada una de las capas del MVC y una cuarta capa (master) donde se enlazaban los cambios realizados. Aunque no siempre se han mantenido activas todas estas ramas.
* **Sistema Operativo:** En el desarrollo del proyecto ha sido utilizado el sistema operativo Windows 7 y 8.1, pero cualquier sistema operativo compatible con los demás requisitos es también válido
* **Java:** Es el lenguaje en el que está desarrollado este proyecto, versión Java jdk 1.6 o superior, el cual utilizamos con las siguientes herramientas:
  + **Maven:** es una herramienta de software para la gestión y construcción de proyectos Java
  + **JSon:** La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON (Como nosotros hacemos en nuestro proyecto).
* **MongoDB:** es un sistema de base de datos NoSQL orientado a documentos. Utilizamos la versión 3.0.0

# INGRESANDO AL SISTEMA

Dentro de su navegador Web ingrese a la siguiente dirección: <https://www.java.com/es/download/> para proceder a la descarga de Java, necesario para el funcionamiento de la aplicación.

A continuación, desde una consola de comandos (cmd) utilizaremos el comando **“java –version”** para comprobar que java está correctamente instalado.

# INSTALACIÓN DE MongoDB

Nos descargamos el software de la dirección: <http://www.mongodb.org/downloads?_ga=1.54361128.1252863717.1426339425>.

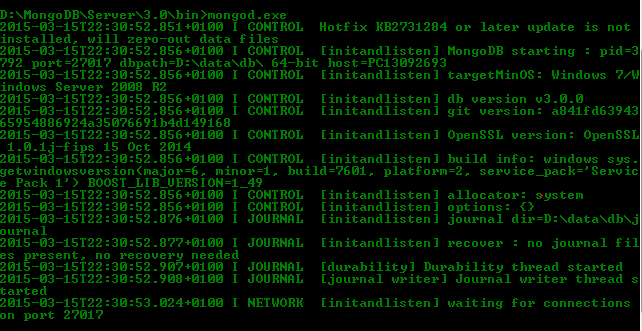
Descomprimimos el zip en la ubicación que queramos de nuestro sistema y creamos una carpeta data/db dentro del mismo directorio, aquí será donde se almacene la base de datos.

Para ejecutar MongoDB debemos usar un terminal de sistema, en él nos colocaremos en la carpeta /bin de la carpeta MongoDB donde descomprimimos el zip y ejecutamos el comando **mongod.exe**, con el que abriremos la base de datos y podremos trabajar con ella. A continuación se describe el proceso:

1.- Abrimos un terminal y nos colocamos en la carpeta donde se encuentra mongod.exeD:\ANGEL\Escritorio\Captura.PNG

2.-Ejecutamos mongod.exe para abrir la base de datos

D:\ANGEL\Escritorio\Captura.PNG  
3.- Deberíamos obtener algo así



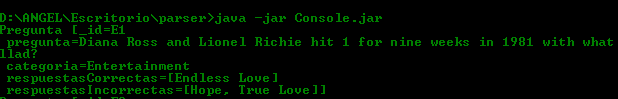
Una vez hecho esto ya podemos empezar a trabajar con el parser

# FUNCIONAMIENTO DEL PARSER

Se proporciona una carpeta en la que se incluye el parser en dos versiones diferentes, una versión para ser ejecutada en línea de comandos y otra que proporciona una pequeña interfaz gráfica.

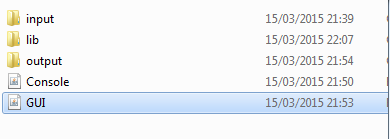
## EJECUCIÓN EN LÍNEA DE COMANDOS

En un terminal nos situamos en la carpeta del parser y ejecutamos **java -jar Console.jar**

****

Esto nos convierte las preguntas en formato .gift que debemos tener en una carpeta llamada input junto al .jar del parser en preguntas .json, y las almacena en la base de datos, es **muy importante tener abierta la base de datos.**

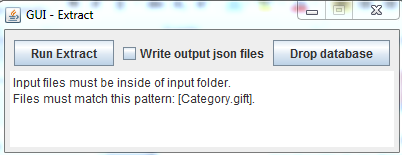
La estructura de directorios debería ser así



A continuación se detalla el contenido de cada carpeta:

* ***Input***: Contiene los ficheros con las preguntas en formato .gift, si queremos añadir nuevas preguntas debemos incluirlas en esta carpeta
* ***Lib:*** Bibliotecas necesarias para la utilización del parser
* ***Output***: Contiene los ficheros con las preguntas .gift convertidas a formato .json

## EJECUCIÓN CON INTERFAZ GRÁFICA

Para ejecutar la interfaz gráfica del parser basta con hacer doble click sobre GUI.jar, lo podemos encontrar junto al otro parser, esto nos mostrará una pequeña ventana con dos botones que nos permitirá realizar la conversión de las preguntas y almacenarlas en la base de datos o borrar la base de datos. 

Al igual que la opción anterior **es muy importante tener abierta la base de datos**, podemos marcar la opción de escribir los ficheros JSON para que se almacenen en la carpeta output para su posterior revisión.

# INTRODUCCIÓN DE NUEVAS PREGUNTAS

Para la introducción de nuevas preguntas se usará el formato GIFT, para ello se creará un nuevo fichero de texto con las preguntas en este formato y se incorporará a los demás bancos de preguntas dentro de la carpeta input que se encuentra junto a los .jar del parser, o bien se modificará un banco ya existente y se incluirán las nuevas preguntas en este formato.