



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza

Iñigo Gascón Royo 685215
Rubén Moreno Jimeno 680882
Guillermo Robles González 604409

Sistemas Legados

Práctica 3

Encapsular el acceso a una aplicación BASIC/MS-DOS

7º cuatrimestre
Ingeniería Informática
13 de diciembre de 2016,
Zaragoza

Contenido

1. Entender el funcionamiento de la aplicación legada.....	3
2. Análisis y diseño del sistema	3
3. Estructura del wrapper con el sistema legado.....	5
4. Puesta en marcha del sistema	5
5. API del sistema.....	6
5.1. Init application	6
5.2. Name Search	6
5.3. Tape Search.....	7
6. Problemas encontrados y organización del trabajo.....	7
7. Referencias.....	8

1. Entender el funcionamiento de la aplicación legada

Previamente a la implementación del wrapper para la aplicación legada, se ha ejecutado y explorado dicha aplicación para tener un conocimiento completo de esta.

En primer lugar, para ejecutar la aplicación, se ha empleado DOSBox, un emulador de MSDOS. En dicho emulador, primero se debe de montar un directorio de la máquina local simulando ser un disco del propio DOSBox. Una vez montado, se puede lanzar la aplicación legada ejecutando el fichero "gwbasic.bat".

A continuación, se ha explorado la aplicación para conocer a fondo todas las funcionalidades que esta ofrece, algunas de las cuales tendrán que emplearse para realizar la interfaz que se pide. Se ha realizado, además, un mapa de la aplicación para conocer todas las opciones que están disponibles y/o son de interés para realizar la interfaz, y que además podrá emplearse para conocer los pasos a seguir para realizar las acciones deseadas en la aplicación.

2. Análisis y diseño del sistema

Puesto que el problema al que hay que enfrentarse es un sistema legado que ha de ser dotado de una GUI web que actúe como wrapper del mismo, se ha decidido crear un diagrama de capas del sistema para poder tener una mejor visión del mismo.

Pese a que se había propuesto realizar la página web en lenguaje JSP, el equipo ha decidido utilizar otras tecnologías más potentes que dotan de un mayor dinamismo a la página web, además de la posibilidad de utilizar peticiones asíncronas mediante pulling utilizando tecnologías como REST/AJAX.

Una de las razones por las cuales se ha tomado la decisión de utilizar tecnologías como éstas, es porque al cargar la página se ha de hacer una petición de inicialización del sistema, consiguiendo así, el número de registros. Dicha petición tiene un tiempo de respuesta alto, la cual al poder hacerse de forma asíncrona, permite cargar toda la página en una primera iteración, haciendo posteriormente, cuando el servidor devuelva la respuesta, una segunda iteración en la que no ha de cargarse toda la página otra vez.

Para la correcta encapsulación del código y al tratarse de un wrapper a un sistema legado se ha tomado la decisión de crear una arquitectura de sistema de cuatro capas. La capa 1 encapsula la interfaz gráfica del cliente, asumiendo un papel de vista y de controlador gracias a la tecnología usada de AngularJS. La capa 2 asume el papel de controlador encapsulando el servidor web, comunicando el cliente con la capa de acceso al mainframe. La capa 3 asume el papel del modelo de datos, y encapsula el acceso y la comunicación con la aplicación legada "DATABASE" en el sistema legado MS-DOS/BASIC, ofreciendo para ello una interfaz java moderna. La capa 4 es el programa DOSBOX que controla la comunicación con la aplicación legada.

Las tecnologías usadas para las tres primeras capas han sido las siguientes:

- **Capa 1:** HTML5, CSS3, javascript, AngularJS y Bootstrap.

- **Capa 2:** Java Servlets.
- **Capa 3:** Java.

A continuación, se muestra el diagrama de capas y vista de módulos de la arquitectura del sistema:

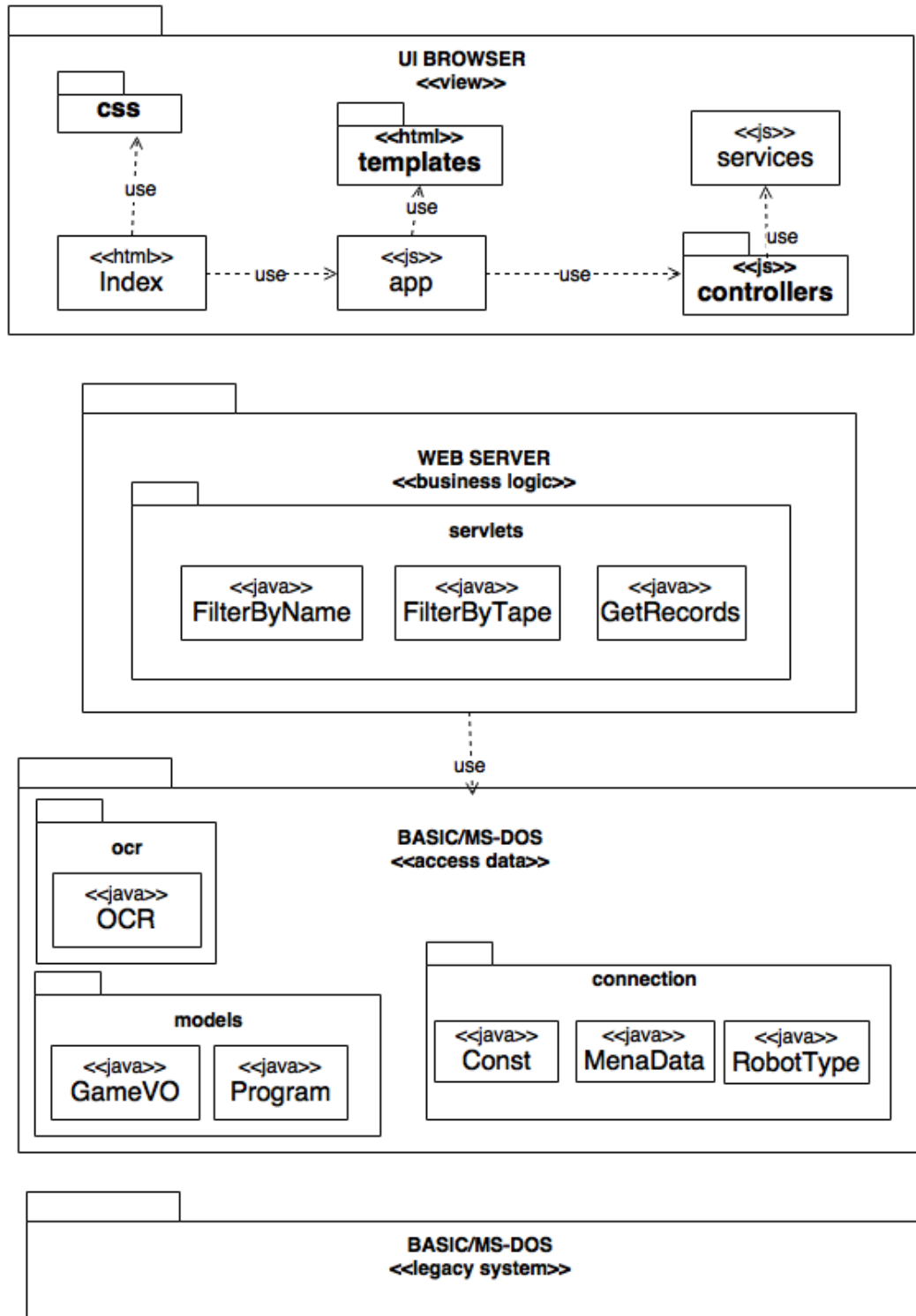


Figura 1. Diagrama de capas y vista de módulos de la arquitectura del sistema.

3. Estructura del wrapper con el sistema legado

Para la realización del trabajo es necesario un sistema OCR encargado de extraer la información presentada al usuario, de forma que se pueda procesar de manera programática. Para este objetivo se sugiere el motor Tesseract¹. Sin embargo, tras experimentar tanto con Tesseract como con otras soluciones, tanto libres (Caffe, servicios web varios) como privativas y de pago (ABBYY), se llegó a la conclusión de que no llegaban al nivel de calidad deseado, dado que tienen graves fallos de identificación, incluso tras un periodo de entrenamiento. Por ello, la solución que se ha adoptado es el desarrollo de un sencillo motor OCR especializado para la tarea realizada.

La capa 3 del proyecto es la encargada del funcionamiento de dicho motor OCR, con el objetivo de ofrecer una interfaz cómoda y simple para los servlets del servidor web.

El motor OCR creado utiliza comparaciones a nivel de pixel para establecer la igualdad con la fuente almacenada. Esto provoca que, con la arquitectura creada sólo se soporte una fuente, pero eso no es problema para el escenario del que se dispone.

El primer paso a realizar es la extracción de la fuente del programa. Para eso se recorrieron todas las pantallas del programa, extrayendo de cada una todos los caracteres presentes. Todo ello se realizó con ayuda de pequeños programas desarrollados ad-hoc. Finalmente, se almacenó la fuente preprocesada, la cual da servicio al motor OCR final.

Dado que el motor que se ha creado no utiliza ninguna técnica especialmente difícil, no posee de una arquitectura compleja. Se ha utilizado acceso indexado a una tabla hash que almacena todos los caracteres, la cual devuelve el carácter pasado en formato textual. Sobre esto se han construido un conjunto de funciones que ayudan a convertir de formas más convenientes (líneas o pantallas enteras).

4. Puesta en marcha del sistema

Para lanzar la aplicación legada y el wrapper implementado, junto a la página web mediante la que interactúa el usuario con dicha aplicación, hay que seguir una serie de pasos:

1. La aplicación se ejecuta sobre DOSBox, por lo que es necesario tenerlo instalado en la máquina donde se vaya a ejecutar.
2. Copiar el fichero "dosbox-0.74.conf" en la carpeta de configuración de DOSBox (por defecto se encontrará en \$HOME/.dosbox en los sistemas Linux), sustituyendo el fichero de configuración existente. Con esto, conseguiremos realizar una serie de modificaciones en el emulador para mejorar la ejecución del programa, además de realizar de forma automática ciertas configuraciones necesarias para su correcto funcionamiento (las mencionadas en el punto 1 relacionadas con el montaje de un directorio).
3. Copiar la aplicación legada en el subdirectorio "c" de la carpeta de configuración de DOSbox.

¹ <https://github.com/tesseract-ocr>

4. Ejecutar el fichero “startServer.sh” para lanzar DOSBox, con la configuración que se ha especificado anteriormente, lanzar la aplicación sobre el emulador y lanzar la aplicación web.
5. Ahora ya puede accederse a la aplicación web en “localhost:8080” para poder emplear la aplicación legada. Este último paso ha de realizarse en otra máquina distinta a la que está lanzada la aplicación. Esto se debe a que la salida de la aplicación se extrae mediante visión por computador, por lo que la pantalla ha de permanecer constantemente con la terminal del DOSBox abierta y visible.

5. API del sistema

Para la comunicación entre cliente y servidor web se ha creado la siguiente API que recoge las 3 funcionalidades básicas que tiene que tener el sistema.

5.1. Init application

GET /getRegisters

Request:

Response:

Status 200

```
{
  "registers":""
}
```

Status 404:

A String with a description message.

5.2. Name Search

POST /filterByName

Request:

```
{
  "name":""
}
```

Response:

Status 200

```
{
  "programs": [
    {
      "number":"","name":"","type":"","tape":"","register":""
    }
  ]
}
```

Status 404

A String with a description message.

Nota: Si no se encuentran resultados en la búsqueda se devuelve un BAD_REQUEST (http code: 400).

5.3. Tape Search

GET /filterByTape

Request:

```
{
  "tape":""
}
```

Response:

Status 200

```
{
  "programs": [
    {
      "number":"","name":"","
      "type":"","tape":"","register":""
    }
  ]
}
```

Status 404

A String with a description message.

Nota: Si no se encuentran resultados en la búsqueda se devuelve un BAD_REQUEST (http code: 400).

6. Problemas encontrados y organización del trabajo

Los principales problemas a los que se ha enfrentado el grupo a la hora de realizar la práctica han sido los siguientes:

- A la hora de probar algunos OCR existentes como por ejemplo Tesseract, Caffé, y ABBYY, se tuvo algunos problemas con el reconocimiento en la lectura de la pantalla, y es por eso que se tomó la decisión de implementar un OCR personalizado.
- Existe un problema con el OCR que se ha implementado, el cual no se ha considerado que se deba resolver debido al alcance de la práctica. Al utilizar reconocimiento pixel a pixel de la pantalla para poder interpretarla, hay que cambiar el código fuente dependiendo de la resolución de las pantallas en las que se ejecute.
- Respecto a la eficiencia del programa legado a la hora de la búsqueda de los registros de programas, se solucionó reordenando sobre la misma base de datos dichos registros, ya que el tiempo de ordenación de los mismos a la hora de devolver la query, era bastante elevado.

En lo referente al reparto de tareas y organización del trabajo, las partes de asegurar que la aplicación funciona correctamente, que se entienda el funcionamiento de la misma, el análisis y diseño, y la redacción de la memoria se ha realizado conjuntamente entre los 3 miembros del grupo de forma presencial o vía Skype participando y consensuando así las decisiones importantes de la práctica. Puesto que, como se ha comentado anteriormente, el equipo ya había trabajado con algunas tecnologías de las que se han usado y cada uno tenía un poco más de conocimiento con algunas de ellas, se ha dividido el trabajo de las tres capas para optimizar el proceso, encargándose Guillermo Robles de la capa 3, Iñigo Gascón de la capa 2, y Rubén Moreno de la capa 1.

7. Referencias

- Web de la asignatura [<http://webdiis.unizar.es/asignaturas/SL/>]
- DOSBox wiki [<https://www.dosbox.com/wiki/>]
- DOSBox by angryflo [<http://www.angryflo.de/dosbox.htm>]
- ABBYY [<http://es.abbyy.com/>]
- Caffe [<https://github.com/pannous/caffe-ocr>]
- Tesseract [<https://github.com/tesseract-ocr>]