

Práctica 3.

Encapsulación de la aplicación legada BASIC/MS-DOS

Sistemas Legados
Grado en Ingeniería informática
28 de enero de 2022

Subías Rodríguez, Rubén (759406)
Chen Zhou, Binhui (779799)
Rea Avila, Salomé (720162)

Índice

Análisis de la aplicación legada	3
Frontend	7
Backend	8
Creación del ejecutable	9
Anexo. Esfuerzos dedicados	10

1. Análisis de la aplicación legada

La aplicación legada se trata de un programa en BASIC que almacena en ficheros información de distintos programas para el ordenador ZX Spectrum. En la Figura 1, puede ver el menú de inicio de la aplicación; como la práctica consiste en construir una página Web únicamente para consultar la información ya almacenada, se utilizarán las operaciones 3 (ordenación según distintos campos), 4 (Información del sistema en general), 6 (listado de los programas ordenado según el campo indicado con la operación 3) y 7 (búsqueda de un programa concreto por su nombre).

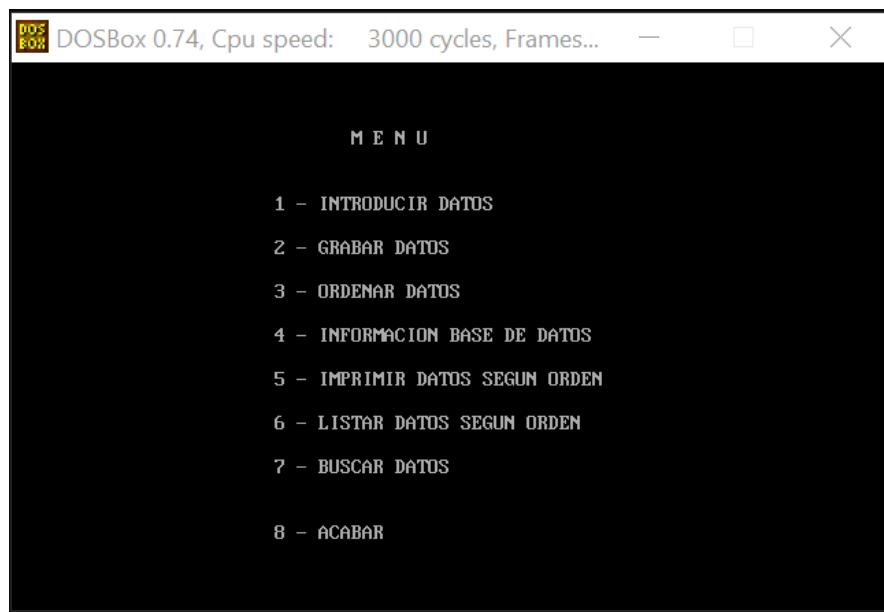


Figura 1. Menú inicial

Para interactuar con la aplicación legada se ha utilizado el emulador DOSBox. Se puede lanzar ejecutando en la consola:

```
"Database-MSDOS/DOSBox-0.74\\DOSBox.exe  
Database-MSDOS/Database\\gwbasic.bat -noconsole"
```

La primera funcionalidad a implementar consiste en la visualización del número total de registros de la aplicación legada (ver Figura 2), este dato se obtiene directamente accediendo a la opción 4 (ver Figura 3).

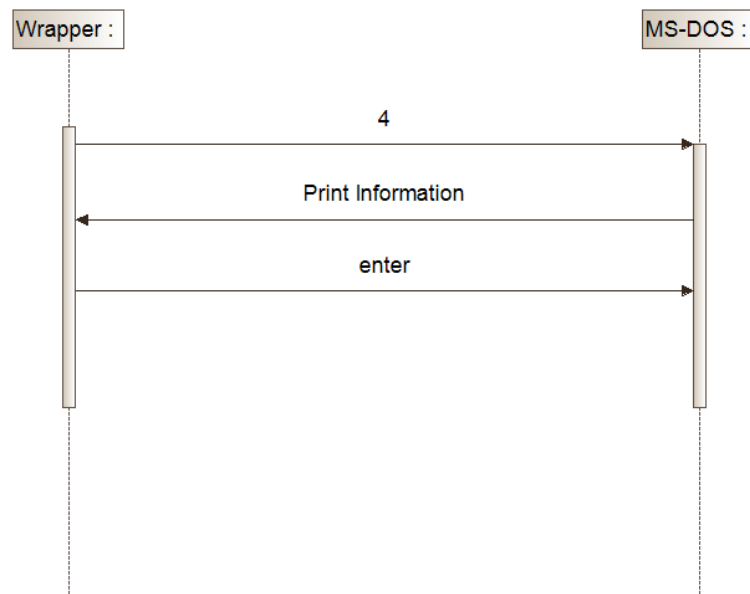


Figura 2. Diagrama “Visualización del número de registro”

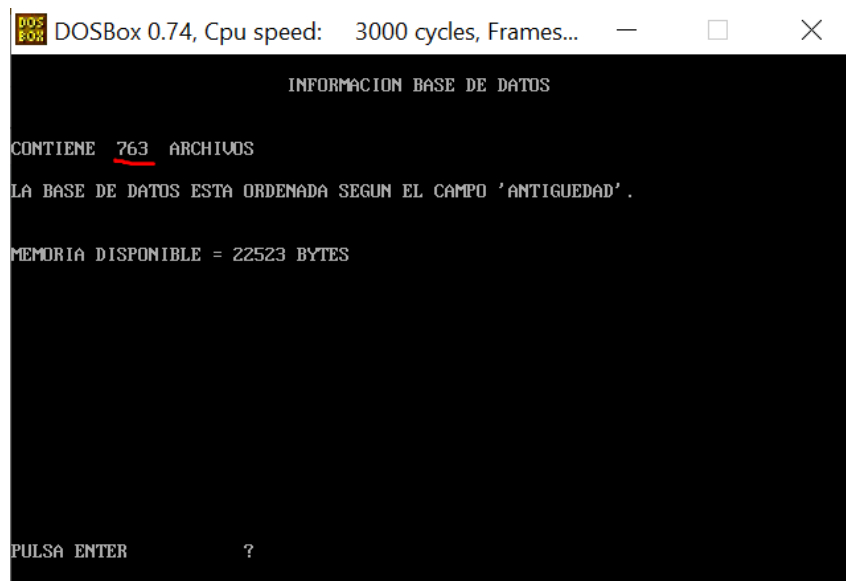


Figura 3. Pantalla con la información general

La siguiente funcionalidad de la búsqueda de un programa por su nombre también está disponible en la aplicación legado, por tanto, simplemente interactuamos (ver Figura 4) y capturamos la información mostrada en la opción 7 (ver Figura 5), que lista los programas de una en una.

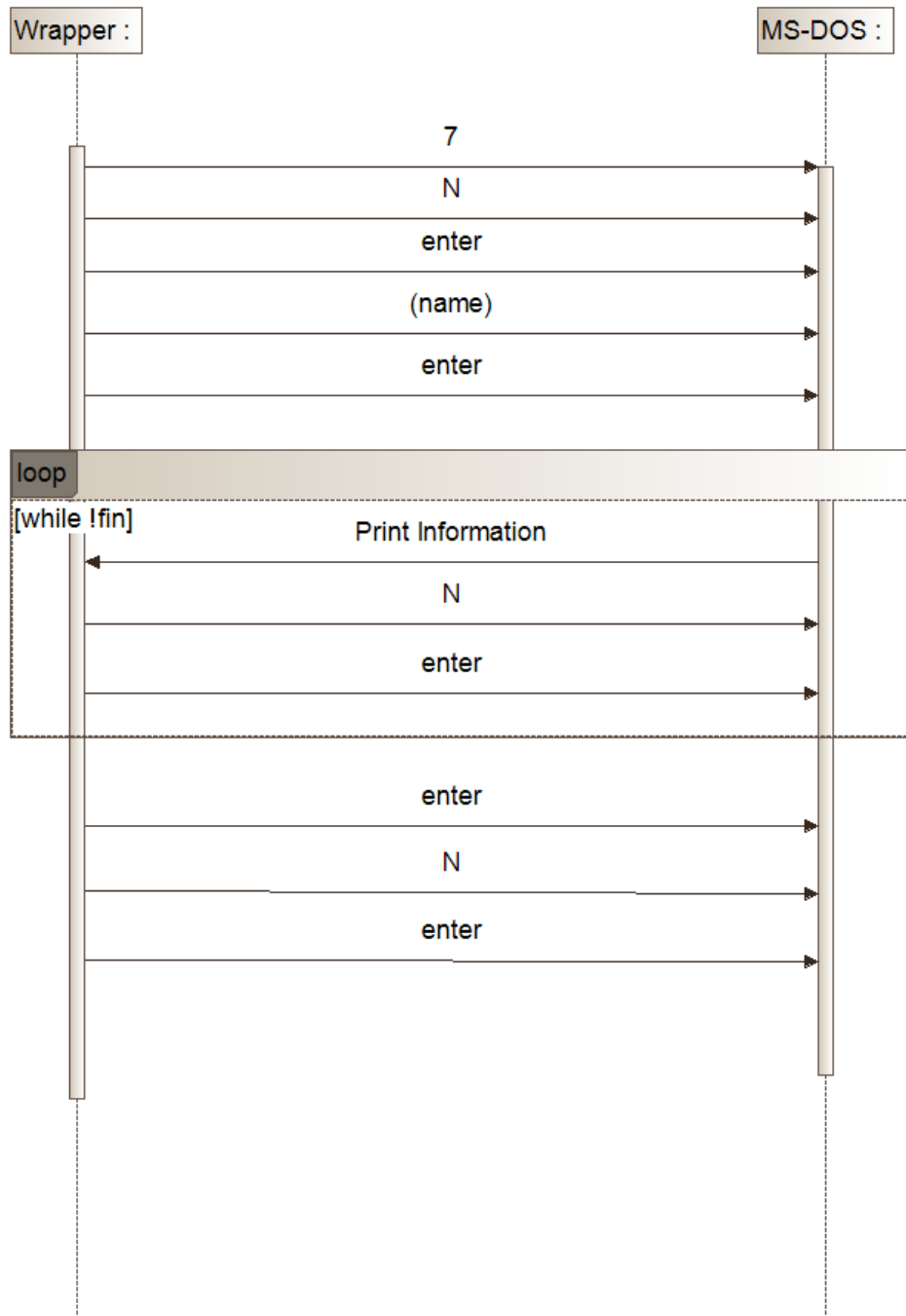


Figura 4. Diagrama “Búsqueda de programa por nombre”

```

2 - PAINTBOX  UTILIDAD  CINTA:A
ES ESTE ? (S/N) ? _
  
```

```

117 - PAINTER  ARCADE  CINTA:I
ES ESTE ? (S/N) ? _
  
```

Figura 5. Información de los programas(Opción 7)

La última funcionalidad consiste en listar todos los programas que se encuentran en una cinta determinada, ordenados por la antigüedad. La dificultad es que la aplicación legada no dispone de esta opción a priori y además, un programa puede estar en dos cintas.

La idea intuitiva es emplear la opción 6 para obtener el listado entero y posteriormente extraer los que son de la cinta, aunque también se dispone de la opción 3 para ordenar el listado antes de capturarlo. Tras analizar las dos opciones y la información de la cinta, tenemos que la ordenación por cintas; por un lado, para los números sigue el criterio de cadenas de texto, es decir, la cinta 11 va antes que la 2 pero por otro lado, las letras van detrás de los números; entonces, el orden por iniciales es [0...9,A..Z,a..z]. Cuando el programa se encuentra en dos cintas, se expresa en el formato C_1-C_2, siendo C_1 la cinta con el identificador de menor valor en la tabla ASCII(letras) o un valor numérico menor, por ejemplo, Q-U, E-29 o 5-28.

Del análisis anterior y teniendo en cuenta que la opción 3, tarda unos 30 segundos en completarse, podemos diferenciar la búsqueda de las cintas con identificador numérico (Figura 6) de las de identificador alfabético(Figura 7). Las numéricas necesitan recorrer prácticamente toda la lista, si tenemos que buscar la cinta 2, podemos tener las de 20 y tantos en medio y además puede estar también en la parte de identificadores alfabéticos, entonces no tiene sentido dedicar 30 segundos para ordenarlo y luego recorrerlo completamente, que tarda minuto y medio más. Para las alfabéticas, podemos ordenarlos anteriormente con la opción 3, y en la opción 6, saltarse toda la parte de cintas numéricas y también puede pararse antes porque los programas en dos cintas aparecen con la letra que va antes en el listado y no es necesario llegar hasta el final.

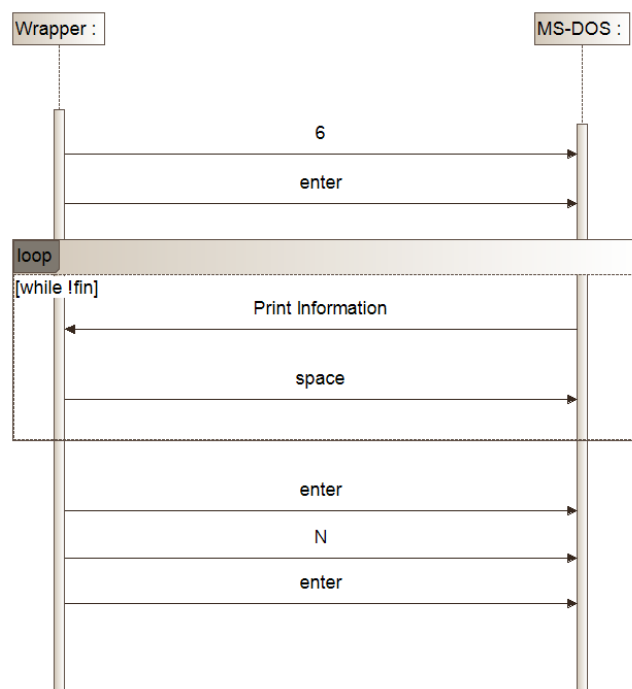


Figura 6. Diagrama “Búsqueda de programas en cintas numéricas”

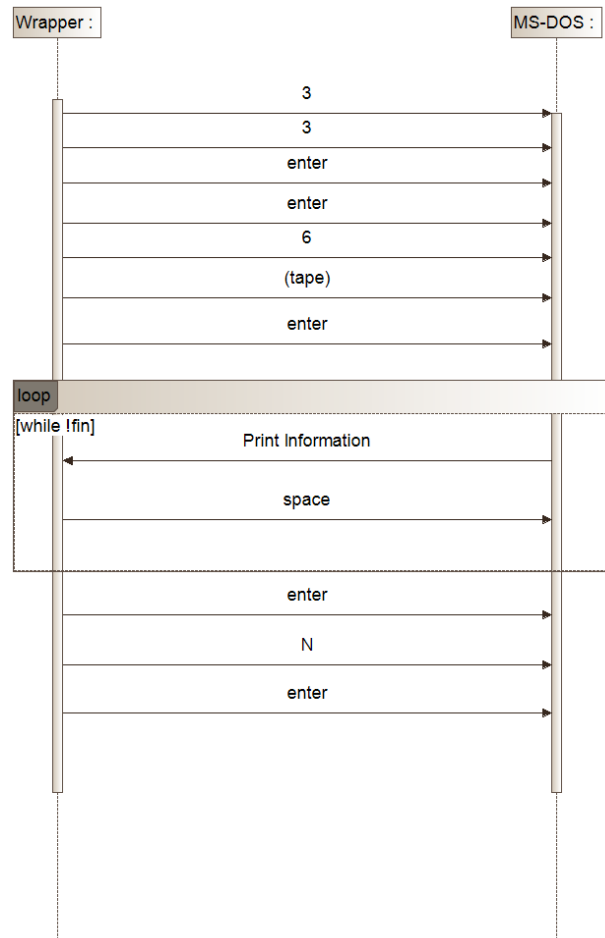


Figura 7. Diagrama “Búsqueda de programas en cintas alfabéticas”

2. Frontend

Se ha desarrollado, en ReactJS, una aplicación de una sola página. Al acceder se muestra el número de registros con el formulario para buscar programas(ver Figura 8), tras introducir la palabra clave y darle al botón de Buscar, se actualizará la página con la información de los programas en tabla(ver Figura 9).

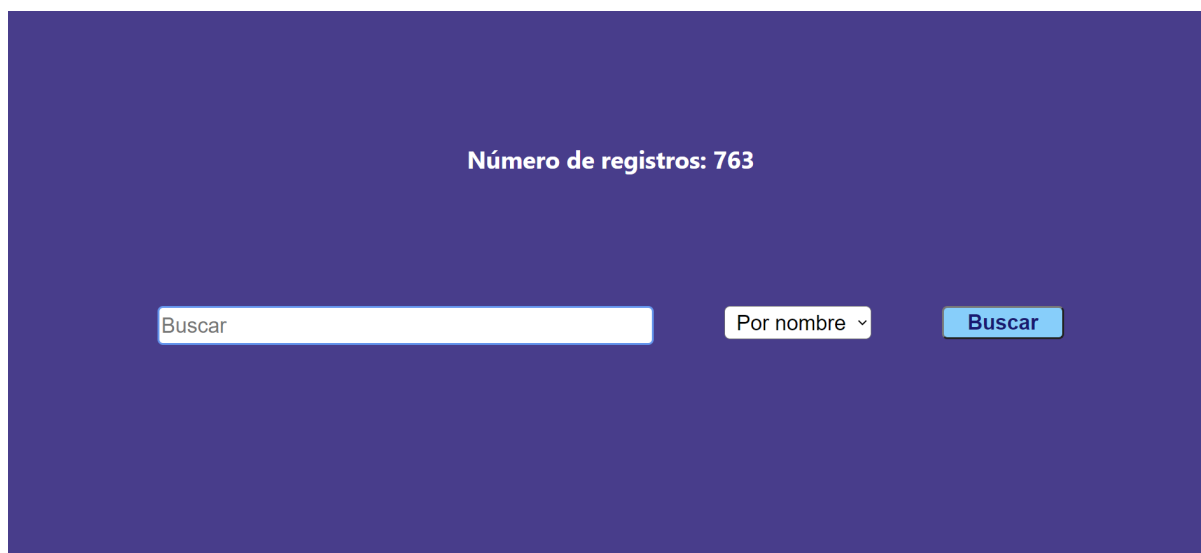


Figura 8. Interfaz inicial



Figura 9. Interfaz actualizada

3. Backend

Primero, hemos iniciado un proyecto de Spring Web con [Spring Initializr](#), en Java, para facilitar la configuración y el despliegue. Se expone la API REST con tres endpoints “/numRegister”, “/searchByName/{name}” y “searchByTape/{tape}” que corresponden a las tres funcionalidades requeridas.

Cada vez que llega una petición al controlador, se inicia un proceso que invoca al emulador DOSBox, la entrada a la aplicación legado, solamente por teclado, se simula con la clase [Robot](#) de Java que genera eventos de bajo nivel.

La salida de la aplicación sólo se puede conseguir haciendo capturas de las pantallas y un reconocimiento óptico de caracteres(OCR) de las mismas, esto provocó bastantes problemas:

Antes de hacer las capturas, hay que localizar las coordenadas de la ventana correspondiente, esto se ha resuelto utilizando [JNA](#); luego, las capturas se realizan también con la clase Robot, al principio, se ha probado con el método createScreenCapture y para realizar el OCR, se ha usado la librería [Tess4J](#) con el fichero de datos de entrenamiento proporcionado al inicio de la práctica spa.traineddata; pero apenas reconocía, solo algún carácter. Se ha probado también la librería [Asprise OCR](#) pero no hubo ninguna mejora. Más tarde, pasamos a utilizar la función createMultiResolutionScreenCapture, de Robot también, pero que consigue unas imágenes de mejor resolución, en este caso, el reconocimiento ha sido bastante bueno, sólo se confunde con algunos caracteres, aunque en general, se equivoca más en los caracteres sueltos, sobretodo en la columna de la cita, que en las palabras completas. Se ha probado a entrenarlo con el módulo de [tesstrain](#) de Tesseract-OCR, generando para ello una inmensa cantidad de capturas de las pantallas en formato TIFF y los ficheros TXT equivalentes con el resultado esperado(se encuentran en el back/tesstrain.zip). También, se ha encontrado una fuente muy similar, [Perfect DOS VGA 437](#). Desgraciadamente, el resultado fue peor; finalmente se ha intentado entrenar desde cero partiendo de las capturas de pantallas, empleando la herramienta JTessBoxEditor, para corregir los ficheros BOX manualmente y generando automáticamente los datos de entrenamiento (ficheros de entrenamiento y de configuración en back/jtessbox.zip). Esta vez, dio un resultado similar al conjunto inicial, pero con mejor precisión en los caracteres sueltos, aunque en las palabras confunde bastante las letras V, W,X,Y,Z; pero, excepto los tres programas duplicados en la cinta V, todos los demás registros e identificadores de cinta los distingue perfectamente.

4. Creación del ejecutable

No se ha podido desplegar la aplicación Web, ni se ha creado un ejecutable EXE para Windows. Sólo se proporciona un run.bat en el directorio que lanza el front y el back en local, suponiendo que se tiene el npm instalado. Se abrirá el emulador y el navegador en la URL "<https://localhost:3000>" juntos, entonces el posible que no haga bien la captura al principio, puede reiniciar la página.

Anexo. Esfuerzos dedicados

Binhui Chen Zhou

Descripción de la actividad	Total
Análisis de la aplicación legada	3
Wrapper de la MSDOS	16
REST API	0.5
Implementación de la GUI	1
Integración	1.5
Documentación	5
Esfuerzos totales	27

Rubén Subías Rodríguez

Salomé Rea Avila