



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.C. Jorge Rommel Santiago Arce

Asignatura: Programación Orientada a Objetos

Grupo: 2

No de Práctica(s): 11

Integrante(s): Cruz Rangel Leonardo Said
Ibañez Guzman Osvaldo
Maya Ortega Maria Fernanda
van der Werff Vargas Pieter Alexander

*No. de Equipo de
cómputo empleado:* 45-48

No. de Lista o Brigada: 7

Semestre: 2020-1

Fecha de entrega: 4/11/2019

Observaciones:

CALIFICACIÓN: _____

Práctica 11: “Manejo de archivos”

Objetivos:

Implementar el intercambio de datos (lectura y escritura) entre fuentes externas (archivos y/o entrada y salida estándar) y un programa (en un lenguaje orientado a objetos).

Desarrollo:

Actividad 1. El repositorio.

<https://github.com/OsvaldoIG/Equipo7/blob/Practica11>

Actividad 2. Archivos, directorios, rutas...



```
Prueba de la clase File
/
existe
no es un archivo
es un directorio
es una ruta absoluta
Ultima modificacion: 1550797077000
Longitud: 1024
Ruta: /
Ruta absoluta: /
Padre: null

Contenido del directorio:
home
usr
.Spotlight-V100
net
.DS_Store
.PKInstallSandboxManager-SystemSoftware
bin
installer.failurerequests
Network
sbin
.file
etc
var
Library
LC_AB
System
.fseventsd
private
.DocumentRevisions-V100
.vol
Users
Applications
opt
dev
Volumes

anaconda3
tmp
Insight Files
cores
```



```
Prueba de la clase File
/Users/poo02alu37/
poo02alu37 existe
no es un archivo
es un directorio
es una ruta absoluta
Ultima modificacion: 1572363110000
Longitud: 512
Ruta: /Users/poo02alu37
Ruta absoluta: /Users/poo02alu37
Padre: /Users

Contenido del directorio:
Music
.DS_Store
.CFUserTextEncoding
Pictures
Desktop
Library
.oracle_jre_usage
.bash_sessions
Public
Movies
.Trash
Documents
Downloads
.viminfo
```

El programa *Archivo* recibe en un campo de texto el nombre de un archivo o una ruta, para desplegar toda la información relacionada a esa ruta. Para saber esto se ocupan distintos métodos de la clase *File*, así como un operador ternario para cada método; ya que devuelven un booleano; de manera que se imprima si es o no un archivo, directorio y ruta absoluta.

Posteriormente se imprimen sus atributos: *lastModified* para la última modificación, *length* para la longitud del archivo, *getPath* para la ruta, *getAbsolutePath* para la ruta absoluta y *getParent* para el padre.

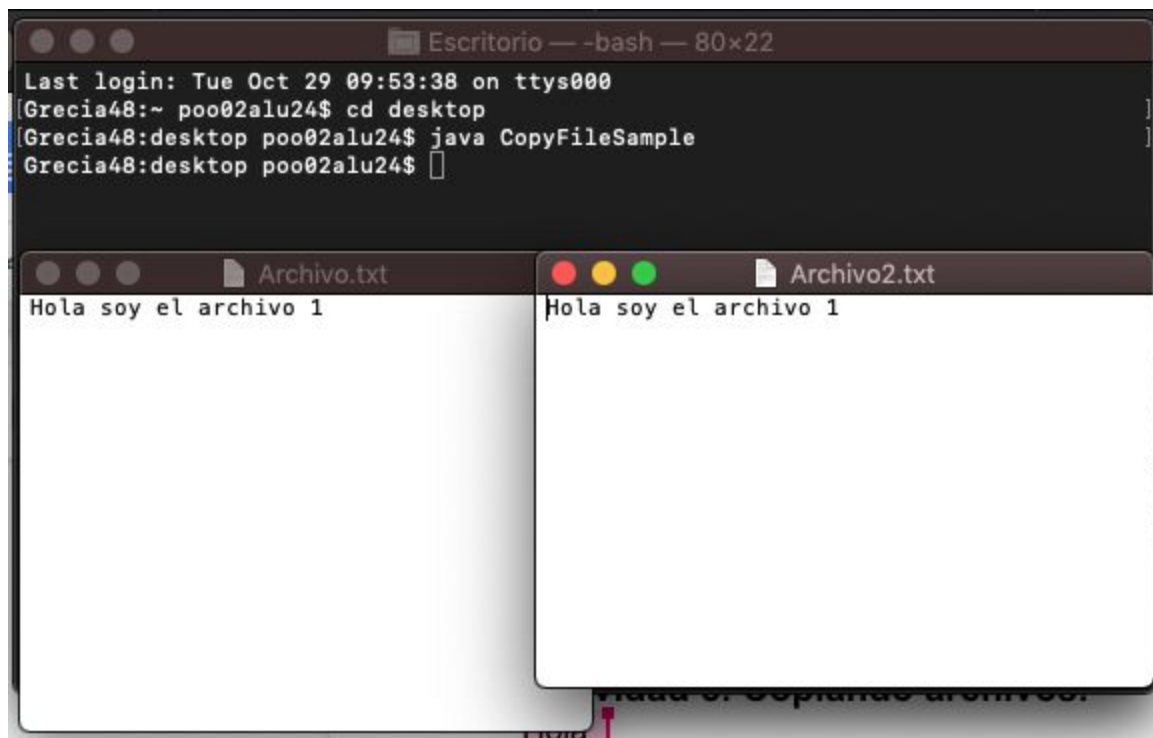
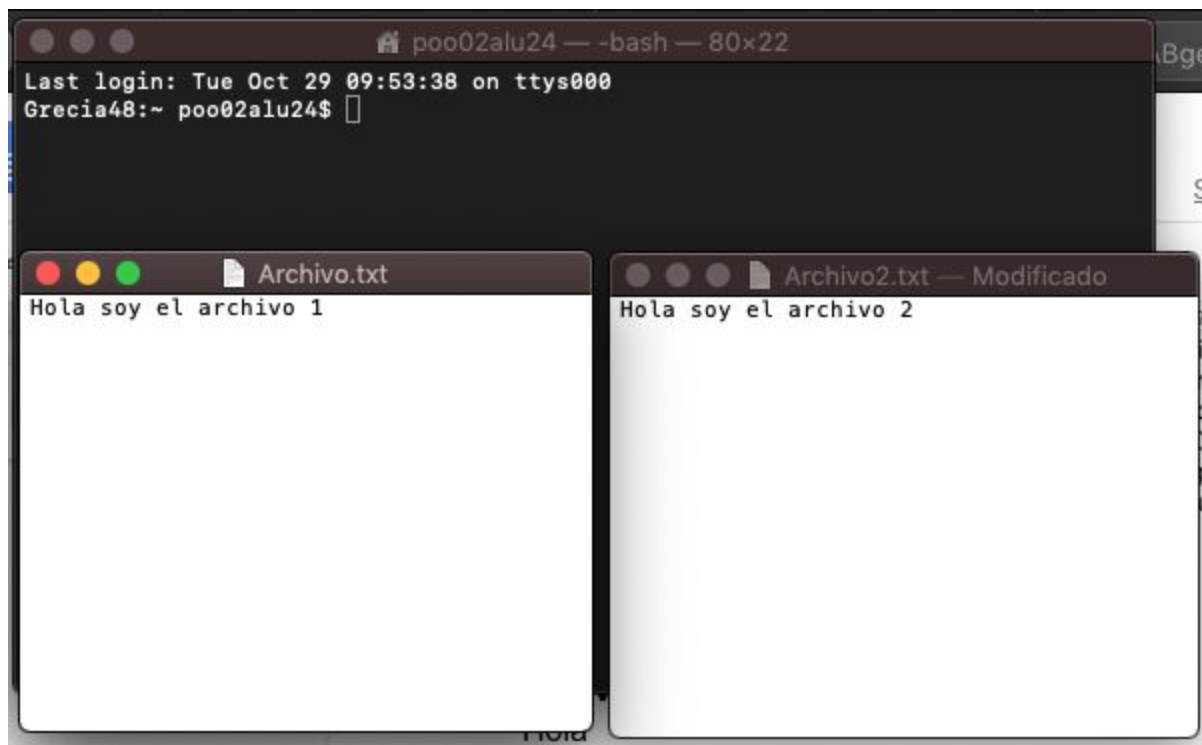
Por último se imprime el texto contenido en el archivo, utilizando la clase *BufferedReader*.



```
Granada47:Desktop poo02alu37$ javac Archivo.java Animal.java ObjectStreamSample.java
Granada47:Desktop poo02alu37$ java Archivo
Granada47:Desktop poo02alu37$ touch ejemplo.txt
Granada47:Desktop poo02alu37$ java Archivo
ejemplo.txt existe
es un archivo
no es un directorio
no es una ruta absoluta
Ultima modificacion: 1572363053000
Longitud: 0
Ruta: ejemplo.txt
Ruta absoluta: /Users/poo02alu37/Desktop/ejemplo.txt
Padre: null
```

Actividad 3. Copiando archivos.

Para copiar el texto de un archivo a otro, se tiene el método dado, en el que se utiliza la clase *InputStream* y también *OutputStream* asociados a los distintos archivos pasados como parámetro el archivo origen y el destino. Con ayuda de un ciclo while va leyendo los caracteres del archivo origen y los escribe en el destino.



Actividad 4. Serialización.

```
Gambia45:Desktop poo02alu11$ javac ObjectStreamSample.java
Gambia45:Desktop poo02alu11$ java ObjectStreamSample
[Animal [name=Tommy Tiger, age=5, type=T], Animal [name=Peter Penguin, age=8, type=P]]
Gambia45:Desktop poo02alu11$
```

Se pretende ejemplificar el uso de la clase Serializable, para esto el profesor nos proporcionó las clases ObjectStreamSample y Animal.

Animal, implementa la clase Serializable, ésta nos permite convertir el estado de un objeto de Animal en un arreglo de bytes, lo que nos va a permitir escribirlo en un archivo y en consecuencia tener la información en memoria secundaria.

ObjectStreamSample cuenta con los métodos:

- `public static List<Animal> getAnimals(File dataFile) throws IOException, ClassNotFoundException:` Obtiene el estado del objeto guardado en un archivo e imprime los datos en pantalla.
Regresa una lista de objetos de Animal.
- `public static void createAnimalsFile(List<Animal> animals, File dataFile) throws IOException:` A partir de un objeto serializable, escribe su estado en un archivo de tipo data.
- `public static void main(String[] args) throws IOException, ClassNotFoundException:` Crea una lista de objetos de tipo Animal, agrega dos objetos con sus datos, crea un objeto de tipo File, manda llamar a la clase `createAnimalsFile` e imprime el contenido del archivo a partir del llamado del método `getAnimals`.

Actividad 5. Console.

La clase Console implementa varios métodos entre ellos están :

`readPassword()`, que como su nombre lo dice sirve para leer una contraseña, esto haciendo que lo que se escriba no se muestre en pantalla , además entre los paréntesis se puede escribir un mensaje para mandar a pantalla.

`flush()` , ayuda a eliminar lo de la contraseña anterior para poder escribir una nueva, que en este caso será la segunda contraseña para comparar.

Posteriormente se comparan las 2 cadenas devolviendo un booleano esto con ayuda de un método de la clase Array, finalmente ambas cada carácter de las cadenas es sustituido por una X , para que de esta forma no sea posible recuperar la contraseña.

Para terminar se manda pantalla diciendo si las contraseñas coinciden o no, además de mostrar las cadenas pero solo como X's.

```
[Georgia46:desktop poo02alu19$ javac PasswordCompareSample.java
[Georgia46:desktop poo02alu19$ java PasswordCompareSample
[Enter your password:
[Enter your password again:
Your password was incorrect

Password : xxx
Verify : xx
```

Conclusiones:

Cruz Rangel Leonardo Said:

En los archivos podemos guardar información en memoria secundaria, tenemos la posibilidad de recuperarla después de cierto tiempo o cuando sea necesario, situación que nos permite, de cierta forma, aumentar la eficiencia de un programa, pues éste no necesita tener toda la información disponible en ciertos momentos.

Ibañez Guzman Osvaldo:

El manejo de archivos a mi parecer se vuelve un poco más sencillo con el uso de Java además de que el proyecto que se realizó, nos ayudó a comprender mejor las bases y el funcionamiento de todos estos métodos, los objetivos de la práctica se cumplieron en su totalidad ya que se llegó a conocer las diferentes operaciones que se pueden realizar con archivos, además de su funcionamiento, a mi parecer esto nos abre un poco el panorama de lo que en un futuro serán las bases de datos ya que aprendemos a organizar información ya tratar estos datos a nuestra conveniencia

Maya Ortega María Fernanda (づゝづ)

Al programar muchas veces se requiere guardar la información generada o trabajar con información guardada previamente, es por eso que se necesita tener un buen manejo de archivos. Java tiene ya predeterminadas clases para su manejo y poder generar mejores implementaciones.

van der Werff Vargas Pieter Alexander:

Si se quiere guardar información de una manera más permanente, en memoria secundaria, se requiere el uso de archivos externos. Para su manejo existen diferentes clases, como lo son *File*, *FileReader*, *FileWriter*, etc. que nos ayudan a crear, eliminar y modificar y leer archivos que se hayan guardado en memoria secundaria.