

# 36019475. CSIFC03. MP0485. Programación

**XUNTA DE GALICIA**CONSELLERÍA DE CULTURA, EDUCACIÓN  
E ORDENACIÓN UNIVERSITARIA

Páxina principal ► Os meus cursos ► Formación Profesional a Distancia ► Curso 2016-2017 ►  
36019475 IES de Rodeira ► CSIFC03 Desenvolvemento de aplicacións web ►  
125\_36019475\_ZSIFC03\_MP0485\_A ► Unidade didáctica 6 ► 6 - Traballando con Ficheiros de Acceso Aleatorio ...

## NAVEGACIÓN

Páxina principal

• Amiña área persoal

Páxinas do sitio

O meu perfil

Curso actual

125\_36019475\_ZSIFC03\_MP0485\_A

Participantes

Distincións

Xeral

Unidade didáctica 1

Unidade didáctica 2

Unidade didáctica 3

Unidade didáctica 4

Unidade didáctica 5

Unidade didáctica 6

Orientaciones para el alumnado. PROG06.

Solución a la tarea para PROG06.

PROG06 Guiada.- Almacenando datos.

Actividades presenciales de la UD6 en la tutoría c...

Tarefa 6 - Soluciona da Titoria Presencial

Foro para PROG06.

Mapa conceptual para PROG06.

PROG06 Completa.- Almacenando datos.

Recursos complementarios UD06.

1.- Introducción a Entrada/Saída en Java

Tarefa a Entregar 1

Solución Tarefa 1

2.- Fluxos Binarios e Fluxos de Caracteres

3.- Acceso a Ficheiros

Tarefa a Entregar 2

Solución

4.- Traballando con Streams Binarios

5 - Traballando con Fluxos de Caracteres

Tarefa a Entregar 3

Solución

**6 - Traballando con Ficheiros de Acceso Aleatorio ...**




Introducción á comunicación con Sockets

Leeweb.java

Chat

Tarea para PROG06.

Solución Alternativa

 Tarefa a Entregar 4 Solución (Versión sen clase para acceso a ficheiros) Solución (Con clase independente para acceder aos ... Tarefa a entregar 5 Solución Tarefa a Entregar 6 Modificacións Propostas Examen de la UD06. Aplicación Hotel Completa

Unidade didáctica 7

Unidade didáctica 8

Unidade didáctica 9

Unidade didáctica 10

Unidade didáctica 11

Os meus cursos

## ADMINISTRACIÓN

Administración do curso

Configuración do meu perfil

## 6 - Traballando con Ficheiros de Acceso Aleatorio (RandomAccessFile)

Os fluxos de datos (streams) son útiles para o envío e lectura de información de xeito secuencial, pero non permiten o posicionamento en datos concretos o que nos limita moito. Por exemplo, mediante streams, o único modo de eliminar información dun ficheiro é leelo e xenerar ao mesmo tempo un novo ficheiro saltándose a parte a borrar, e renomeando por último o novo ficheiro. Tampouco é posible modificar información dun xeito eficiente, xa que o proceso ten que ser igual que o borrado.

Todo esto nos leva a que si traballamos con Streams, teñamos que almacenar na memoria a práctica totalidade do ficheiro (nun ArrayList, nun List.. etc)

Para este tipo de circunstancias (borrados, actualizacións, consultas...) é moito mellor utilizar a clase RandomAccessFile. Esta clase nos permite ler/escrobir información nun ficheiro, pero tamén nos permite posicionarnos na parte do ficheiro que desexemos. O ficheiro se trata como unha secuencia de bytes, e mantén internamente a posición na que se realizará a seguinte lectura/escritura. Cada vez que se lee/escribe, se incrementa esa posición de lectura/escritura no número de bytes leídos/escritos de xeito automático, posición que se pode modificar mediante o método **seek**, ou averiguar mediante o método **getFilePointer**.

Estas operacións anteriores nos permiten traballar coa información dentro do propio ficheiro (si a organizamos convenientemente). Podemos facer modificacións posicionándonos (seek) xusto na posición da información a modificar e escribindo, podemos "borrar" a información escribindo encima un dato que nos indique que non é válida, podemos ler directamente a información que necesitemos si coñecemos a súa posición.... etc.

O máis simple para poder facer todas esas cousas é almacenar a información utilizando rexistros de lonxitude fixa, xa que de ese xeito coñeceremos a posición de comezo de cada rexistro de datos do ficheiro, e podemos tratalo casi como un Array, pero se poden artellar métodos para utilizar rexistros de lonxitude variable (como almacenar ao comezo de cada rexistro a súa lonxitude...).

Esta clase nos proporciona ademáis métodos para ler e escribir os tipos de datos básicos, de xeito similar a DataInputStream.

O principal problema para utilizar ficheiros de lonxitude fixa é que as cadeas de texto (String) poden utilizar caracteres que ocupan un número variable de bytes por carácter. A solución é escribir métodos que se encarguen de ler e escribir Strings restrinxindo o seu tamaño. Por exemplo:

```
{
    StringBuffer buffer = new StringBuffer();
    if (str!=null)
        buffer.append(str);
    buffer.setLength(dim);
    file.writeChars(buffer.toString());
}

private String readString (RandomAccessFile file, int dim) throws IOException
{
    char campo[] = new char[dim];
    for (int i=0; i<dim; i++)
        campo[i] = file.readChar();
    return new String(campo).replace("\0", ' ');
}
```

Última modificación: Luns, 30 de Xaneiro do 2017, 11:27





