

Objetivo

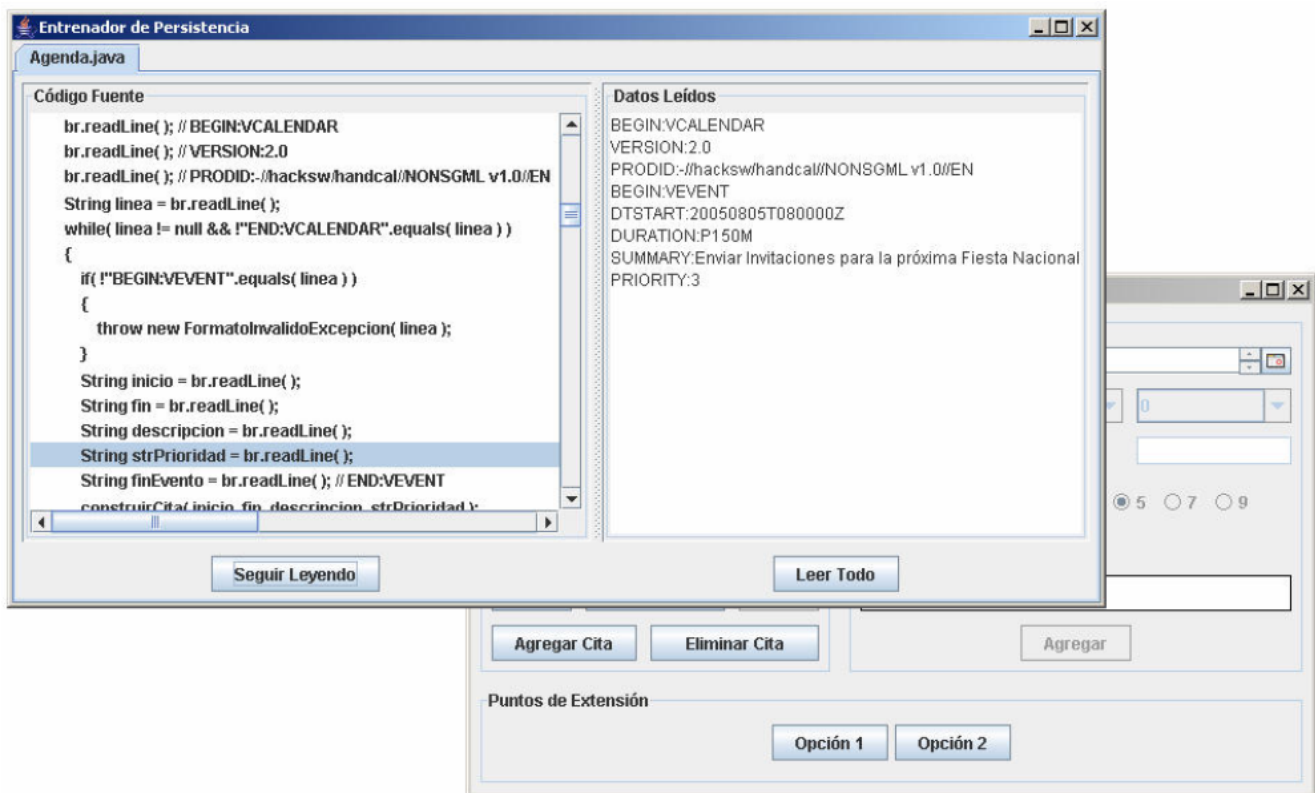
El objetivo de este entrenador es ayudar a:

- Practicar los algoritmos para cargar datos desde un archivo de texto
- Practicar los algoritmos para salvar datos en un archivo de texto

Como cada contexto tiene particularidades para los procesos de cargar y salvar, este entrenador puede utilizarse para muchos casos diferentes, en los cuales será posible ver tanto lo que se está leyendo / escribiendo, como el código que está siendo utilizado para leer / escribir. De esta forma se espera que los usuarios del entrenador ejerciten las habilidades necesarias para diseñar y verificar algoritmos para cargar o guardar información.

Descripción

Este entrenador se utiliza desde adentro de una aplicación que está siendo implementada. Cuando esta se ejecute una ventana mostrará cómo las instrucciones que permiten cargar o salvar se van ejecutando una por una, al tiempo que se va mostrando el contenido de los archivos leídos o salvados.



En la imagen anterior puede verse cómo, en el ejemplo n9_agenda, se puede seguir el algoritmo encargado de cargar las citas.

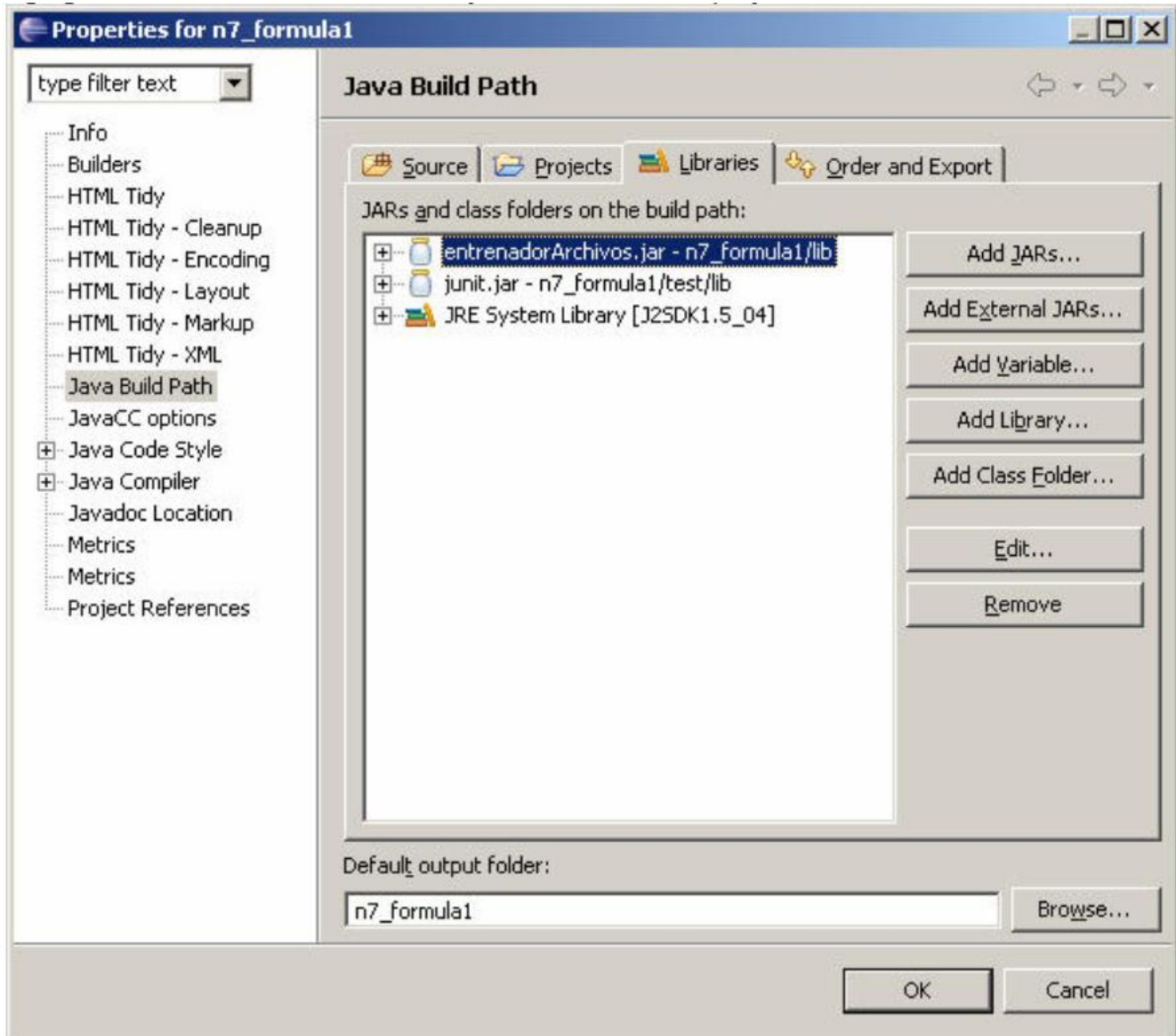
Instrucciones

1. Descargue el archivo `entrenadorArchivos.jar` y cópielo a la carpeta `lib` de su proyecto.
2. Modifique las propiedades de su proyecto para que incluya el archivo `entrenadorArchivos.jar` dentro de las librerías del proyecto. Para esto haga clic derecho sobre su proyecto y seleccione la opción *Propiedades* o *Properties*. En la opción llamada *Java Build Path* seleccione la pestaña que dice *Libraries*. Haga clic en el botón *Add Jar* y seleccione al archivo `entrenadorArchivos.jar` que está dentro de su proyecto. Haga clic en *Ok* y cierre la ventana con las propiedades del proyecto.
3. Ubique en qué clases de su aplicación está construyendo objetos de las clases `PrintWriter` o `BufferedReader`: modifique las líneas que se encuentran en la parte superior de esas clases y que se encargan de importar las clases usadas en ese archivo.
 - a. En lugar de `import java.io.BufferedReader;` escriba `import uniandes.cupi2.entrenadorArchivos.mundo.BufferedReader;`
 - b. En lugar de `import java.io.PrintWriter;` escriba `import uniandes.cupi2.entrenadorArchivos.mundo.PrintWriter;`
4. Salve los archivos modificados y ejecute su aplicación. Cuando ahora cargue o salve un archivo debería aparecer una ventana donde podrá ver el comportamiento de sus algoritmos.
5. Cuando ya no necesite la ventana de ayuda del entrenador, deberá modificar nuevamente los *imports* y devolverlos a su forma original.

Ejemplo: Fórmula 1

A continuación se describe el proceso completo para agregar el entrenador de persistencia al ejemplo `n8_formula1`.

- En primer lugar descargue el ejemplo `n8_formula1` y el archivo `entrenadorArchivos.jar` de la página de `cupi2` y copie este último archivo a la carpeta `lib` del ejemplo.
- Cargue el proyecto en Eclipse.
- Agregue el archivo `entrenadorArchivos.jar` a las librerías del proyecto.



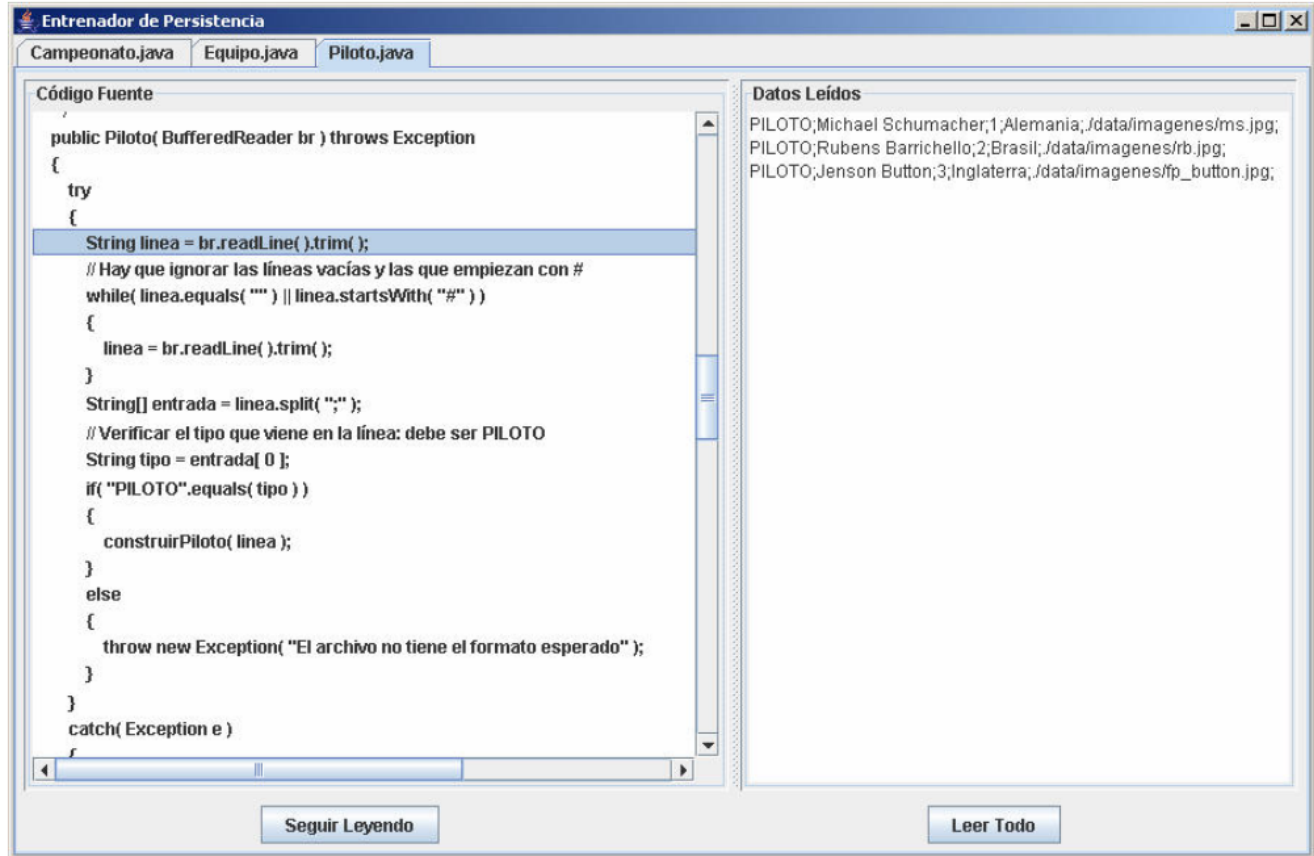
- Busque la clase `uniandes.cupi2.formula1.mundo.Campeonato`
- Modifique las líneas que importan las clases `PrintWriter` y `BufferedReader`.
En lugar de:

```
import java.io.BufferedReader;  
import java.io.PrintWriter;
```

Deben decir:

```
import uniandes.cupi2.entrenadorArchivos.mundo.BufferedReader;  
import uniandes.cupi2.entrenadorArchivos.mundo.PrintWriter;
```

- Ejecute la aplicación y vea cómo ahora puede seguir sus algoritmos de cargar y salvar paso a paso.



The screenshot shows a Java IDE window titled "Entrenador de Persistencia". It has three tabs: "Campeonato.java", "Equipo.java", and "Piloto.java". The "Piloto.java" tab is active, showing the following source code:

```
public Piloto( BufferedReader br ) throws Exception
{
    try
    {
        String linea = br.readLine( ).trim( );
        // Hay que ignorar las líneas vacías y las que empiezan con #
        while( linea.equals( "" ) || linea.startsWith( "#" ) )
        {
            linea = br.readLine( ).trim( );
        }
        String[] entrada = linea.split( ";" );
        // Verificar el tipo que viene en la línea: debe ser PILOTO
        String tipo = entrada[ 0 ];
        if( "PILOTO".equals( tipo ) )
        {
            construirPiloto( linea );
        }
        else
        {
            throw new Exception( "El archivo no tiene el formato esperado" );
        }
    }
}
catch( Exception e )
```

On the right side, the "Datos Leídos" panel displays the following data:

```
PILOTO;Michael Schumacher;1;Alemania;./data/imagenes/ms.jpg;
PILOTO;Rubens Barrichello;2;Brasil;./data/imagenes/rb.jpg;
PILOTO;Jenson Button;3;Inglaterra;./data/imagenes/fp_button.jpg;
```

At the bottom of the IDE, there are two buttons: "Seguir Leyendo" and "Leer Todo".