

CODE GENERATORS USER MANUAL

In order to execute one of the four code generators, you must double click in the corresponding jar file whitin each code generator folder.

Then, the window asking you the selection of the XML document with the architectural description appears (Figure 1).

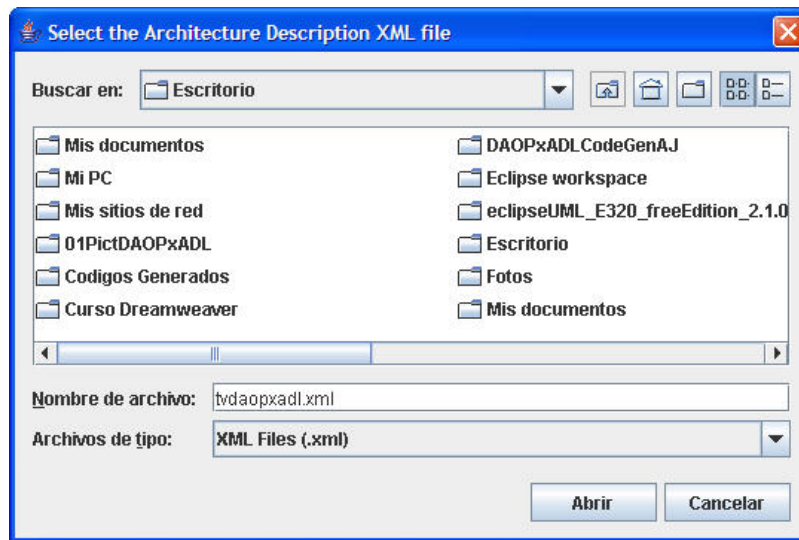


Figure 1. Window for the selection of the XML document with the architectural description.

After, the next window is shown (Figure 2) to choose the target folder of the generated application.

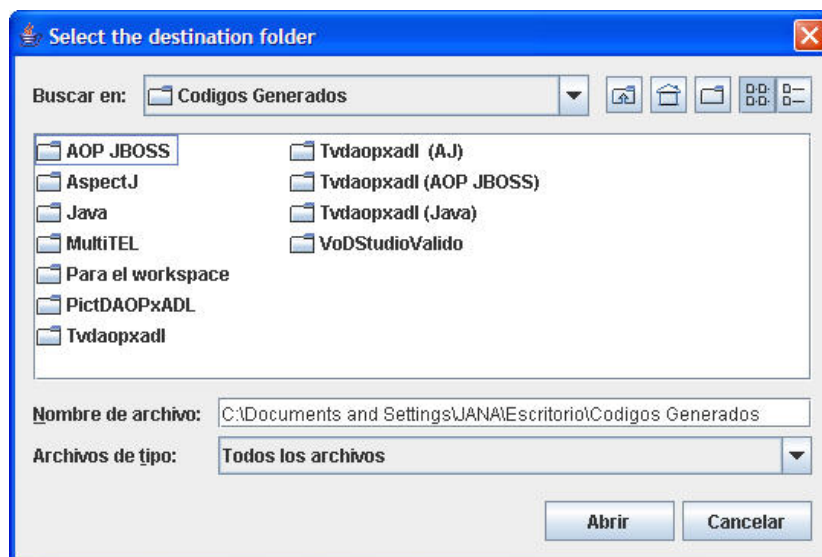


Figure 2. Window for the selection of target directory where the application will be created.

If the code was generated successfully, the window shown in Figure 3 will be appear. But if there are errors during the generation, windows with the errors information will be shown.



Figure 3. Window informing that the code was generated successfully.

If the application was terminated in a satisfying way, inside of the target directory a folder with the name of the XML document selected have been created. The Figure 4 shows this folder that contains the main class (*.java file) and for example, in the DAOPxADL case, contains two folders, one for the application components (*.java files) and the other for the aspects (*.java files or *.aj files in the AspectJ case).

Finally, when the code is generated, the programmer has to complete the code. Then, the programmer has to open every file generated and fill the methods and the rest of the things needed for the application.

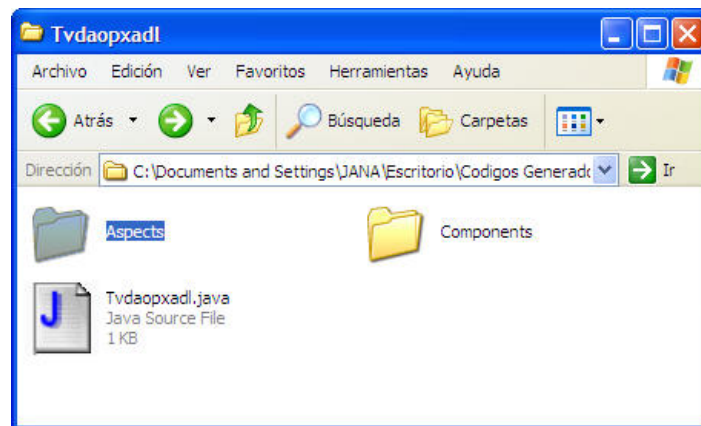


Figure 4. Code generated structure.

For example, for the next generated code, at least the programmer has to complete the methods `setChannel`, `getChannel` y `codifiedChannel`.

```
package Components;
import java.util.*;
public class Ctv tuner implements Ctv tuner Int {
    protected String role;
    protected String instancename;
    int channel;
    public Ctv tuner (String rol, String insname) {
        role = rol;
        instancename = insname;
    }
    public void setChannel (int newChannel) {
        //User code
    }
    public int getChannel () {
        //User code
    }
    public void codifiedChannel () {
        //User code
    }
    public void exit () throws Throwable {
        try {
            finalize ();
        } catch (Exception e) {
        }
    }
}
```