



APRENDERAPROGRAMAR.COM

## GUARDAR LOS PROYECTOS JAVA. COPIAS DE SEGURIDAD Y GESTIÓN DE VERSIONES (CU00631B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

**Resumen:** Entrega nº31 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

## GUARDAR LOS PROYECTOS JAVA. COPIAS DE SEGURIDAD Y GESTIÓN DE VERSIONES.

Muchos entornos de desarrollo, incluido BlueJ, trabajan guardando continuamente los cambios realizados en los proyectos. Si estamos trabajando y pulsamos directamente sobre el aspa de cierre del editor o sobre el aspa de cierre del programa, los cambios quedan guardados automáticamente.



Esto tiene la ventaja de que no tenemos que preocuparnos de estar pulsando un botón “guardar” periódicamente como puede ser habitual en editores de texto como Microsoft Word. Pero también tiene un inconveniente, que es el que modificaciones introducidas nos generen un perjuicio que no podamos evitar al no saber localizar exactamente el código “malo” que genera distorsiones en nuestro programa.

Cuando estamos trabajando en el desarrollo de un programa extenso, que pueda requerir días, semanas o meses de dedicación, conviene tomar precauciones para no perder código desarrollado por bloqueos del ordenador, fallos en soportes como discos duros o introducción de código “malo”.

En primer lugar y como precaución, básica, realiza una copia de seguridad de los archivos del proyecto a un soporte externo (como pendrive, disco duro o servidor externo) diariamente si el proyecto es importante. **Cada cierto tiempo, realiza una copia de seguridad** en un cd o dvd y mantenla en un lugar seguro. En segundo lugar, un proyecto evoluciona desde cero líneas de código hasta miles de líneas de código. Entre el principio y el final, hay muchos estados del proyecto. Es conveniente manejar estos estados como si se tratara de versiones o prototipos de programa de forma que en cualquier momento podamos recuperar un estado anterior si así lo deseamos. Existen herramientas de software libre para el manejo de versiones como *Subversion* o *Git*. Pueden ser de utilidad, pero no vamos a entrar aquí a explicar su uso. Vamos a proponer un manejo bastante más sencillo y rústico que consiste en lo siguiente:

1. Cuando comiences a trabajar en un proyecto importante crea primero una carpeta para almacenar las versiones o prototipos con el nombre general del proyecto (por ejemplo, “Cobra”). Luego comienza a trabajar con el entorno de desarrollo creando un proyecto dentro de la carpeta creada anteriormente y ponle al proyecto un nombre terminado en `_01`. Por ejemplo `Cobra_01`.
2. Cuando termines de trabajar en el proyecto en ese día cierra el entorno de desarrollo y busca la carpeta que contiene los archivos correspondientes al proyecto. Cópiala y renómbrala para que el nombre termine en `_011`, por ejemplo `Cobra_011`. Al día siguiente abre y usa como proyecto de trabajo el de numeración más grande, de modo que la carpeta anterior permanezca como copia de seguridad. BlueJ te permitirá seguir trabajando sin problema.
3. En días sucesivos vete repitiendo el proceso renombrando con terminaciones en `_012`, `_013`, etc.

4. Por otro lado, en un archivo de texto (que puede ser creado con el bloc de notas) al que puedes denominar changelog.txt, vete registrando las características de cada versión y los cambios relevantes introducidos respecto a la anterior. Por ejemplo una anotación podría ser la siguiente:

*En 0\_29 está implementado el constructor de la clase TruckSystem y probado que la generación del sistema de truckers es correcta. También está implementado el método getTruckBySize (String size) y probado que se localiza correctamente el truck solicitado.*

Nombre	Tamaño	Tipo
Cobra_01		Carpeta de archivos
Cobra_011		Carpeta de archivos
Cobra_012		Carpeta de archivos
Cobra_013		Carpeta de archivos
Cobra_014		Carpeta de archivos
Cobra_015		Carpeta de archivos
Cobra_016		Carpeta de archivos
changelog.txt	5 KB	Documento de texto

**Realiza copias de seguridad periódicas de todo el sistema de archivos.** De este modo, ante un fallo en el ordenador o soporte, no perderás el trabajo realizado. También te permitirá volver atrás si no estás satisfecho con la evolución del código y retomar el proyecto desde un punto dado. Por último, cuando tengas un programa que creas terminado, crea la copia de los archivos renombrando el proyecto con la terminación 1\_0 y para sucesivas versiones utiliza terminaciones como 1\_01, 1\_02, 1\_03, etc.

El sistema que hemos descrito es bastante casero frente a herramientas para mantenimiento profesional de versiones, pero nos puede servir perfectamente para dar nuestros primeros pasos como programadores Java evitando sustos indeseados.

Como último apunte, haremos referencia a la estabilidad de un entorno de desarrollo. Un IDE es, al fin y al cabo, software, es decir, un programa o aplicación. Al igual que cualquier programa se puede “bloquear”, “colgar” o dar algún tipo de problemas. BlueJ es un entorno bastante estable, pero ocasionalmente puede fallar por bloqueos, envío de mensajes de error de compilación sin motivo, visualizaciones extrañas o cierres inesperados. En primer lugar, indicarte que en el 99 % de los casos el problema será tuyo: no trates de achacárselo al entorno. En los casos excepcionales en que sospeches que se pueda tratar de un fallo del programa, procede en primer lugar a cerrarlo y abrirlo de nuevo para ver si el problema desaparece. En segundo lugar a apagar y reiniciar el ordenador. Si se te presenta un problema persistente, consulta en el sitio oficial del IDE o en foros de programación.

**Próxima entrega:** CU00632B

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=68&Itemid=188](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188)