

Diferencia entre Check-in y Shelve

En sistemas de control de versiones como Unity DevOps, **Check-in** y **Shelve** son dos acciones distintas que permiten gestionar los cambios realizados en el código. Aquí te explicamos las diferencias con un ejemplo práctico.

1. Check-in

El **check-in** es el proceso mediante el cual **confirmas y subes los cambios realizados en tu espacio de trabajo al repositorio central**. Esto implica que tus cambios se convierten en parte del historial del proyecto y son accesibles para otros miembros del equipo.

Ejemplo de Check-in:

Imagina que estás trabajando en un proyecto de Unity llamado "Bad Blood". Tienes un archivo llamado `EnemyAI.cs` que has modificado para mejorar la inteligencia artificial de los enemigos.

- Después de hacer varias modificaciones y probar el código, decides que todo está listo para ser compartido con el equipo.
- Realizas un **check-in** de tu archivo, lo que significa que el código que has escrito ahora está en el repositorio compartido y tus compañeros pueden verlo y acceder a él.
- Además, puedes agregar un comentario como: **"Mejorada la IA de los enemigos, agregando nuevas reglas de comportamiento"**.

El **check-in** hace que tus cambios sean permanentes y visibles para todos, siendo parte del historial del proyecto.

2. Shelve

El **shelve** es un proceso mediante el cual guardas temporalmente los cambios que has hecho, **sin comprometerlos en el repositorio**. Los cambios quedan en tu espacio de trabajo local, pero no son visibles para otros miembros del equipo hasta que decidas realizar un **check-in** más tarde.

Es útil cuando estás trabajando en algo, pero necesitas **interrumpir temporalmente** tu tarea y no quieres que esos cambios sean visibles para el resto del equipo todavía. También es útil cuando no estás seguro si los cambios son correctos y quieres tener la posibilidad de retomarlos más tarde.

Ejemplo de Shelve:

Siguiendo con el ejemplo anterior, supongamos que mientras trabajabas en la **IA de los enemigos**, recibes una tarea urgente de tu jefe para realizar un arreglo rápido en otro archivo. Tienes la opción de **guardar temporalmente** tu trabajo sin perderlo o comprometerlo al repositorio, para que puedas retomarlo más tarde cuando tengas tiempo.

- Usas la opción **Shelve** para guardar los cambios que hiciste en el archivo `EnemyAI.cs`, y esos cambios se guardan en un "cajón" temporal.
- Ahora puedes empezar a trabajar en la tarea urgente sin que los demás vean ni tengan acceso a los cambios de la IA hasta que decidas hacer el **check-in** más tarde.

Diferencias clave entre Check-in y Shelve:

1. Check-in:

- Los cambios son permanentes y se suben al repositorio.
- Son accesibles para otros miembros del equipo.
- Se registran como parte del historial del proyecto.
- Se utiliza cuando los cambios están listos para ser compartidos y finalmente se integran en el proyecto.

2. Shelve:

- Los cambios se guardan temporalmente sin comprometerlos al repositorio.
- No son accesibles para otros miembros del equipo hasta que decidas hacer un check-in.
- No quedan registrados como parte del historial del proyecto.
- Se utiliza cuando necesitas interrumpir temporalmente tu trabajo y retomarlo más tarde sin perderlo.

Resumen visual:

- **Check-in** = "Confirmar los cambios y subirlos al repositorio".
- **Shelve** = "Guardar los cambios de forma temporal, sin subirlos al repositorio".

En resumen, si los cambios que estás haciendo están listos para ser compartidos con el equipo, realiza un **check-in**. Si no están listos y necesitas hacer una pausa en tu trabajo, utiliza **shelve** para guardarlos temporalmente.