
Artículo

[Alberto Fuentes](#) · 20 mayo, 2022 · Lectura de 3 min

[Open Exchange](#)

Git para entornos de desarrollo compartidos

Si estás desarrollando soluciones en IRIS y quieres utilizar Git, ¡es genial! Sólo tienes que utilizar VSCode con un repositorio git local y enviar los cambios al servidor, así de fácil.

Pero qué sucede si:

- Colaboras con otros desarrolladores en un entorno de desarrollo remoto compartido y quieres evitar la edición simultánea del mismo archivo
- Utilizas editores basados en el portal de administración para BPL, DTL, tablas dinámicas, cuadros de mando, etc. y quieres incluir esos elementos en el control de código de forma sencilla.
- Aún utilizas Studio para realizar algunas cosas y/o ocasionalmente vuelves a usarlo desde VSCode. O tu equipo todavía no ha adoptado completamente VSCode y algunos miembros del equipo todavía quieren utilizar Studio
- Trabajas en varios proyectos diferentes al mismo tiempo en el mismo namespace - digamos, en varios paquetes definidos mediante InterSystems Package Manager - y quieres trabajar con todos ellos desde una vista de edición de isfs (en vez de en varios proyectos distintos) con los cambios rastreados automáticamente en el repositorio de git apropiado

En ese caso no era tan sencillo... hasta finales del año pasado, cuando lanzamos Git para Entornos de Desarrollo Compartidos ([Open Exchange](#) / [GitHub](#)).

Puedes descargar esta extensión mediante InterSystems Package Manager:

```
zpm "install git-source-control"
```

Antes de esto, las opciones para controlar las fuentes con Git eran una [antigua extensión de Git, principalmente para Windows, y sólo para entornos de desarrollo local](#) y un [proyecto más reciente de Open Exchange basado en él pero más sencillo](#). También está [Port](#), que únicamente maneja archivos y es independiente del sistema de control de versiones.

¿Qué tiene git-source-control que no tienen estos paquetes?

- Una integración sencilla con git, basada en un menú, que funciona en cualquier sistema operativo
- Una interfaz de usuario de git para cubrir un conjunto de operaciones comunes de git, sin tener que conectarse con SSH al entorno remoto.
- Control de concurrencia para múltiples usuarios que trabajan en el mismo entorno al mismo tiempo. Una vez que realizas cambios en una clase/rutina/etc., se vuelve tuya hasta que se descarten o se confirmen los cambios. (Sin embargo, tenemos formas de evitarlo cuando es necesario)
- Integración con el administrador de paquetes: simplemente usa zpm mediante "load -dev /path/to/package" y si "/path/to/package/.git" existe, los cambios en los recursos de tu paquete se reflejarán automáticamente en el sitio correcto del sistema de archivos del servidor. La interfaz de usuario también funciona con esto, basada en la clase/etc. desde la que se ejecuta.

Todo esto funciona desde VSCode:

Spoiler

Y Studio:

Spoiler

Para que tengas el control y la visión de tu repositorio de git:

Spoiler

Esperamos que esto permita desarrollar con éxito las soluciones basadas en IRIS.

¡Agradecemos vuestros comentarios!

NOTA: Para ver la presentación del lanzamiento en el Global Summit de 2021, consultad este artículo:

<https://community.intersystems.com/post/video-git-gitlab-shared-developm...>

[#Código abierto](#) [#Git](#) [#GitHub](#) [#Interoperabilidad](#) [#InterSystems Package Manager \(IPM\)](#) [#Mejores prácticas](#)
[#ObjectScript](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#) [#VSCode](#)
[Ir a la aplicación en InterSystems Open Exchange](#)

URL de fuente: <https://es.community.intersystems.com/post/git-para-entornos-de-desarrollo-compartidos>