

Artículo

[Alberto Fuentes](#) · 31 mayo, 2022 · Lectura de 2 min

[Open Exchange](#)

Actualización de la plantilla Dockerfile estándar para desarrollo con InterSystems IRIS

¡Hola desarrolladores!

Gracias a todos los que habéis empezado el desarrollo con InterSystems IRIS desde la [plantilla de desarrollo básico](#)!

Recientemente, gracias a las contribuciones de [@Dmitry Maslennikov](#), he actualizado el [Dockerfile](#) para hacer el desarrollo más sencillo, las imágenes más ligeras y el proceso de construcción más rápido. Y también parece más bonito ;)

Esto es lo que ha cambiado:

```
Dockerfile > ARG
9 FROM $IMAGE
10
11- USER root
12
13- WORKDIR /opt/irisbuild
14- RUN chown ${ISC_PACKAGE_MGRUSER}:${ISC_PACKAGE_IRIS} /opt/irisbuild
15- USER ${ISC_PACKAGE_MGRUSER}
16
17- #COPY Installer.cls .
18- COPY src src
19- COPY Installer.cls Installer.cls
20- COPY module.xml module.xml
21- COPY iris.script iris.script
22-
23- RUN iris start IRIS \
24-     && iris session IRIS < iris.script \
25-     && iris stop IRIS quietly
26

5 FROM $IMAGE
6
7+ WORKDIR /home/irisowner/irisbuild
8
9+ ARG TESTS=0
10
11+ RUN --mount=type=bind,src=.,dst=. \
12+     iris start IRIS && \
13+     iris session IRIS < iris.script && \
14+     ([ $TESTS -eq 0 ] || iris session iris "#cL
15+     iris stop IRIS quietly
16
```

Vamos a repasar los cambios:

1. Las líneas 11-15 han sido sustituidas por una [única línea 7](#), que define el WORKDIR.

```
WORKDIR /home/irisowner/irisbuild
```

De hecho, si WORKDIR señala a una carpeta de inicio del usuario que ejecuta iris, no hace falta realizar ajustes de seguridad.

Por supuesto, debería cambiar [la línea en iris.script](#) para cargar el módulo zpm desde otra carpeta workdir

2. Las líneas 17-21 eran comandos COPY que yo solía usar para copiar diferentes archivos desde la carpeta origen del repo a una imagen para usarla para iris en el entorno.

Ahora todas esas líneas han sido sustituidas por una línea

```
RUN --mount=type=bind,src=.,dst=. \
```

Esta sintaxis usa la funcionalidad buildkit de Docker que permite montar los archivos que necesitas en la imagen, sin copiarlos realmente. Esto ahorra espacio y acelera el proceso de construcción de la imagen.

Así que con todo esto puedes olvidarte de los comandos COPY: todo lo de la carpeta repo se montará y estará disponible en un WORKDIR durante la fase de construcción. Bonito, ¿eh? Asegúrate de tener esta opción habilitada en tu Docker:



3. Lo que se ha añadido es la línea con parámetro TEST:

```
ARG TESTS=0
```

Si TESTS=1 hace que Dockerfile llame test unitarios para el módulo ZPM que desarrollas en el proyecto. Esto se controla con [la siguiente línea](#):

```
([ $TESTS -eq 0 ] || iris session iris "##class(%ZPM.PackageManager).Shell(\"test $MO  
DULE -v -only\",1,1)") && \
```

Este parámetro TESTS también ayuda a realizar test unitarios automáticamente durante el CI. Esto merece un tema de discusión adicional y continuaré con ello en el próximo artículo.

O puedes mirarlo en el [código de Dmitry](#) si no puedes esperar :D

Espero que os haya gustado. Y cualquier comentario, duda o pull request es bienvenido.

Happy coding!

[#Docker #Principiante #InterSystems IRIS](#)
[Ir a la aplicación en InterSystems Open Exchange](#)

URL de
fuente: <https://es.community.intersystems.com/post/actualizaci%C3%B3n-de-la-plantilla-dockerfile-est%C3%A1ndar-para-desarrollo-con-intersystems-iris>
