

Artículo

[Eduardo Anglada](#) · 22 nov, 2021 Lectura de 3 min

## Cómo importar archivos CSV a las tablas existentes en InterSystems IRIS

¡Hola amigos!

A veces necesitamos importar datos a InterSystems IRIS desde archivos de tipo CSV. Esto puede hacerse, por ejemplo, mediante la herramienta [csvgen](#), que genera una clase e importa todos los datos a ella.

Pero, ¿qué pasa si tienes tu propia clase y quieres importar datos desde un archivo CSV a una tabla que ya existe previamente?

Hay varias formas de hacerlo, ¡pero puedes utilizar csvgen (o [csvgen-ui](#)) otra vez! Preparé un ejemplo y estoy encantado de compartirlo con todos. ¡Vamos allá!

El concepto es el siguiente: Tengo la clase A y quiero los datos del archivo.csv, ya que este archivo contiene una columna que necesito para mi clase.

Los pasos son los siguientes:

1. Crear la clase B utilizando csvgen
2. Efectuar una actualización en SQL para añadir los datos de la clase B en la clase A
3. Eliminar la clase B

Para explicar el concepto, creé una [demostración sencilla del proyecto](#). El proyecto importa el [conjunto de datos Countries](#), que contiene la clase `dc.data.Country` con información variada sobre los países, incluido el GNP (PIB en español).

```
ClassMethod ImportCSV() As %Status
{
    set sc = ##class(communit.csvgen).GenerateFromURL("https://raw.githubusercontent.com/evshvarov/test-ipad/master/gnp.csv", ",", "dc.data.GNP")

    Return sc
}
```

Pero los datos del GNP están desactualizados y los más recientes se encuentran en [este CSV](#). Este es el método que muestra el GNP, por ejemplo, para Angola:

```
ClassMethod ShowGNP() As %Status
{
```

```
Set sc = $$$OK

&sql(

SELECT TOP 1 name,gnp into :name,:gnp from dc_data.Country

)

if SQLCODE < 0 throw ##class(%Exception.SQL).CreateFromSQLCODE(SQLCODE,"Show Country GNP")

write "Country ",name," gnp=",gnp,!

Return sc

}
```

Entonces importo el CSV en una clase generada con una línea:

```
ClassMethod ImportCSV() As %Status

{

set sc = ##class(community.csvgen).GenerateFromURL("https://raw.githubusercontent.com/evshvarov/test-ipad/master/gnp.csv","", "dc.data.GNP")

Return sc

}
```

y con la segunda línea realizo una consulta en SQL que importa los datos actualizados del GNP a mi clase `dcdata.Country`.

```
ClassMethod UpdateGNP() As %Status

{

Set sc = $$$OK

&sql(

UPDATE dc_data.Country

SET Country.gnp=GNP."2020"

FROM

dc_data.Country Country

INNER JOIN dc_data.GNP GNP

On Country.name=GNP.CountryName

)

}
```

```
if SQLCODE < 0 throw ##class(%Exception.SQL).CreateFromSQLCODE(SQLCODE,"Importing data")
```

```
w "Changes to GNP are made from dc.data.GNP",!
```

```
Return sc
```

```
}
```

Y luego, elimino la clase generada (con todos sus datos) porque ya no la necesito.

```
ClassMethod DropGNP() As %Status
```

```
{
```

```
Set sc = $$$OK
```

```
&sql(
```

```
DROP TABLE dc_data.GNP
```

```
)
```

```
if SQLCODE < 0 throw ##class(%Exception.SQL).CreateFromSQLCODE(SQLCODE,"Drop csv table")
```

```
write "dc.data.DNP class is deleted.",!
```

```
Return sc
```

```
}
```

Y este es el método que hace todo al mismo tiempo:

```
ClassMethod RunAll() As %Status
```

```
{
```

```
Set sc = $$$OK
```

```
zw ..ImportDataset()
```

```
zw ..ShowGNP()
```

```
zw ..ImportCSV()
```

```
zw ..UpdateGNP()  
  
zw ..ShowGNP()  
  
zw ..DropGNP()  
  
Return sc  
  
}
```

Por supuesto, esta solo es una de las formas de solucionar el problema, pero espero que sea útil. ¡Me encantaría recibir vuestros comentarios!

[#CSV](#) [#Importación y Exportación de Datos](#) [#SQL](#) [#InterSystems IRIS](#)

---

URL de  
fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-importar-archivos-csv-las-tablas-existentes-en-intersystems-iris>