

Artículo

[Yone Moreno](#) · 6 feb, 2022 Lectura de 2 min

" cuida tus ojos: usa el Cod "

¿Necesitas que tus compañeros y tú, tengan una forma más sencilla de leer el código?

¿Te preguntas cómo cuidar tu vista y hacer más agradable la lectura de código?

¿Buscas una forma de reducir la fatiga visual y mejorar la calidad de lo que escribes?

Por favor, sigue leyendo para averiguar las respuestas a las cuestiones anteriores.

1º Nos creamos una clase normal y corriente, que extiende de Ens.BP.Context

```
Class Procesos.PPCCx.GestionCitas.Code Extends Ens.BP.Context
{
```

2º Escribimos el método que necesitamos

```
Class Procesos.PPCCx.GestionCitas.Code Extends Ens.BP.Context
{
  ClassMethod XMLaER7(request As Ens.Response) As EnsLib.HL7.Message
  {
    set mensajeXML = ##class(%GlobalCharacterStream).%New()

    while (request.mensaje.AtEnd = 0){
      do mensajeXML.Write($zcvf(request.mensaje.Read()), "0", "UTF8")
    }

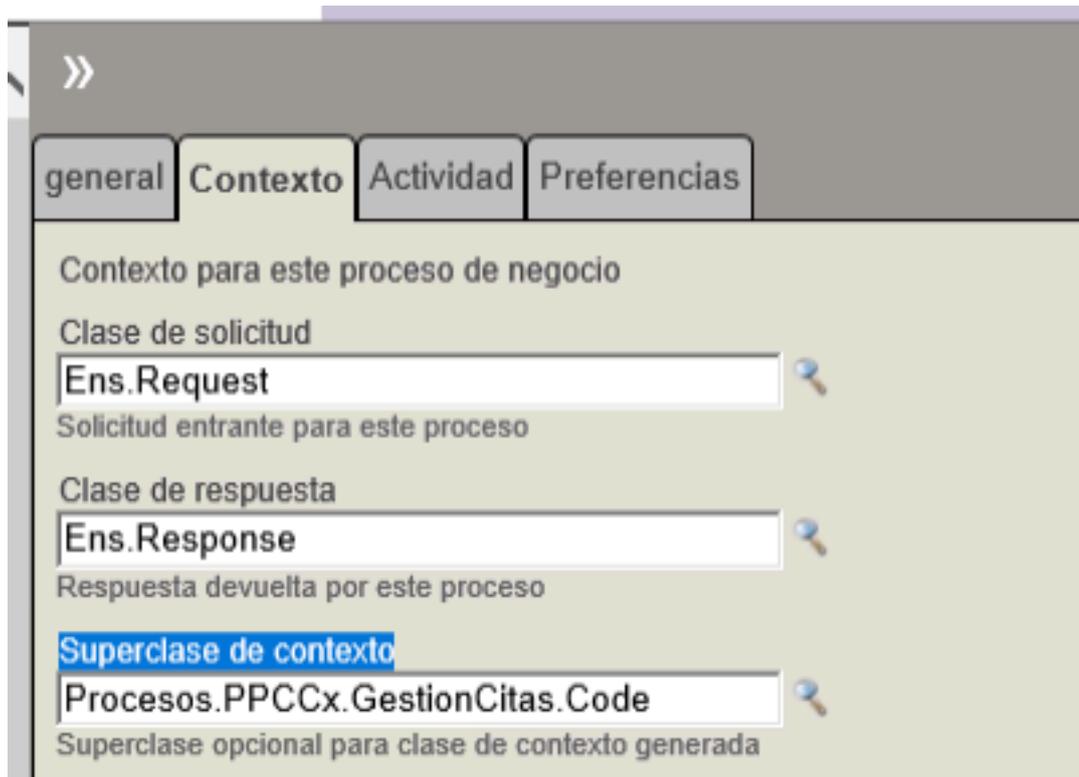
    $$$LOGINFO("En Procesos.PPCCx.GestionCitas.Code, mensajeXML.Read(): "_mensajeXML.Read())
    do mensajeXML.Rewind()
    set response = ##class(ITB.HL7.Util.Convert).XMLToER7(mensajeXML, .tSC, "2.5")
    $$$LOGINFO("tSC: "_tSC)

    quit response
  }
}
```

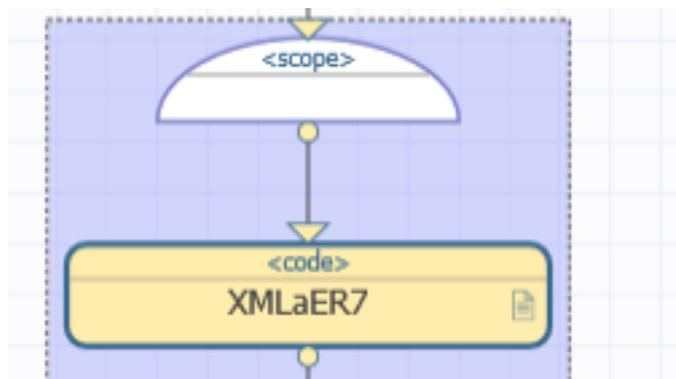
3º Enlazamos el Proceso de Negocio (BP) con nuestra clase Code:

Para ello en Contexto > Superclase de contexto

Anotamos el nombre de la clase Code, que hemos creado en el paso 1



4º Llamamos a nuestra super función desde el BP

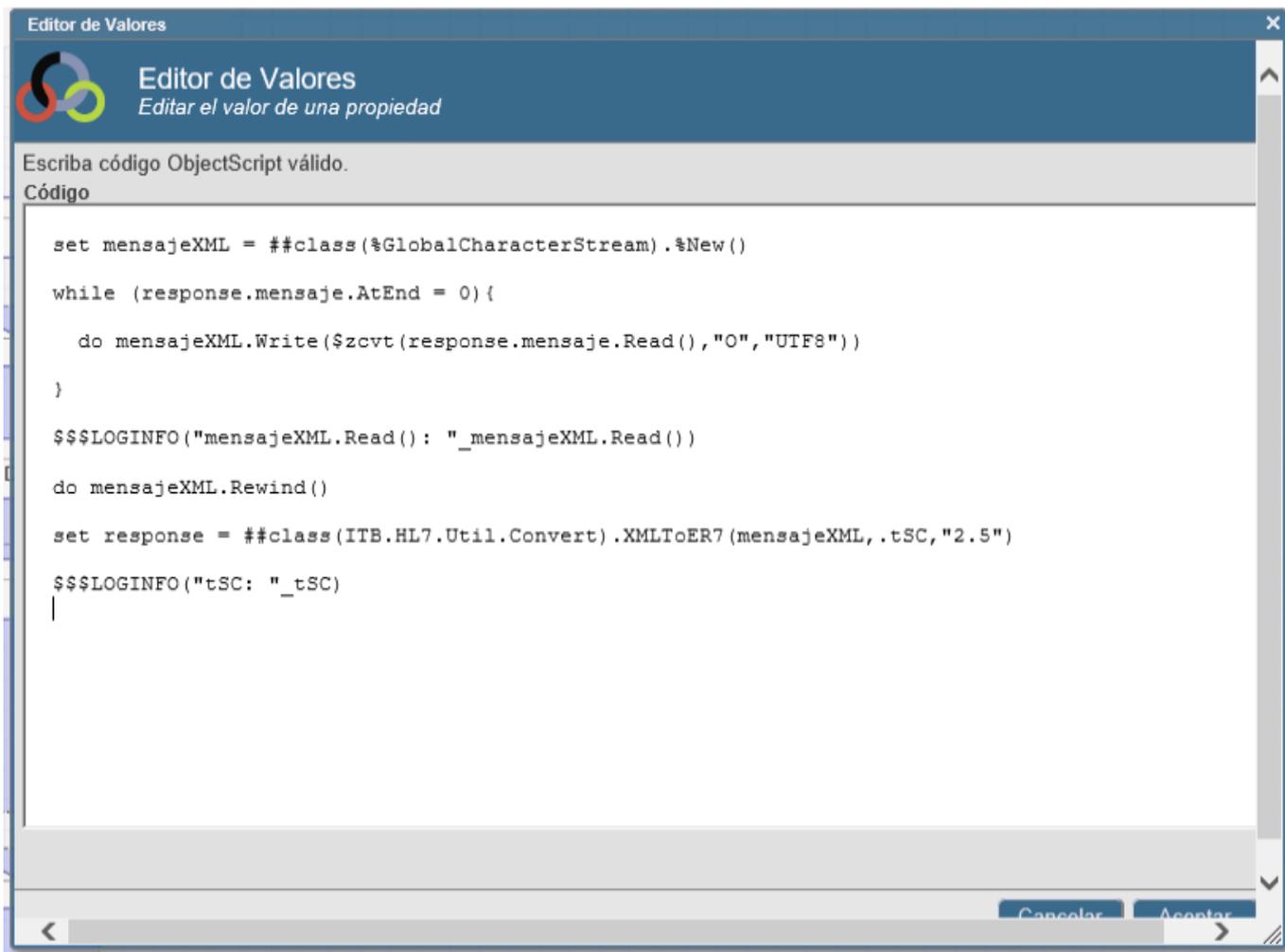




Para ello escribimos en un code:

```
set context.respuestaDragoER7 = context.XMLaER7(context.respuestaDragoXML)
```

De esta forma tendremos el alivio visual de en vez de poner toda la lógica en el Proceso, :



```
Editor de Valores
Editor de Valores
  Editar el valor de una propiedad

Escriba código ObjectScript válido.
Código

set mensajeXML = ##class(%GlobalCharacterStream).%New()

while (response.mensaje.AtEnd = 0){
  do mensajeXML.Write($zcvrt(response.mensaje.Read(),"0","UTF8"))
}

$$$LOGINFO("mensajeXML.Read(): "_mensajeXML.Read())

do mensajeXML.Rewind()

set response = ##class(ITB.HL7.Util.Convert).XMLtoER7(mensajeXML,.tSC,"2.5")

$$$LOGINFO("tSC: "_tSC)
|

Cancelar Aceptar
```

Escribirla de forma más sencilla en una clase a parte donde se ve la letra mejor :

```
Class Procesos.PPCCx.GestionCitas.Code Extends Ens.BP.Context
{
  ClassMethod XMLaER7(request As Ens.Response) As EnsLib.HL7.Message
  {
    set mensajeXML = ##class(%GlobalCharacterStream).%New()

    while (request.mensaje.AtEnd = 0){
      do mensajeXML.Write($zcvrt(request.mensaje.Read(),"0","UTF8"))
    }

    $$$LOGINFO("En Procesos.PPCCx.GestionCitas.Code, mensajeXML.Read(): "_mensajeXML.Read())
    do mensajeXML.Rewind()
    set response = ##class(ITB.HL7.Util.Convert).XMLtoER7(mensajeXML,.tSC,"2.5")
    $$$LOGINFO("tSC: "_tSC)

    quit response
  }
}
```

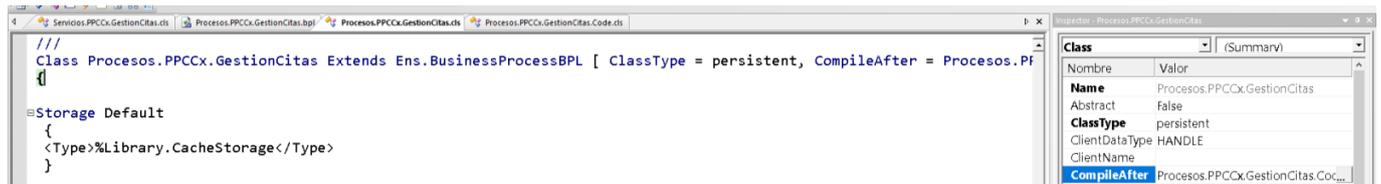
Atención: para que funcione al Importar y Compilar el proyecto en otra Instancia (INTEGRACION, PRE...)

Necesitamos expresar que el BP se compile DESPUÉS que el Code usando:

```
CompileAfter = Procesos.PPCCx.GestionCitas.Code
```

En el Proceso, añadimos:

```
Class Procesos.PPCCx.GestionCitas Extends Ens.BusinessProcessBPL [ ClassType = persistent, CompileAfter = Procesos.PPCCx.GestionCitas.Code, ProcedureBlock ]
```



De esta forma, hemos visto cómo hacer más agradable a la vista, el código, y cuidar la salud visual de nuestros compañeros, y la nuestra propia, al escribir de forma que se entienda y lea de forma más sencilla.

Gracias por leerlo

[#Concurso #Otro](#)

URL de fuente: <https://es.community.intersystems.com/post/%C6%88uida-tus-ojos-usa-el-cod%E1%BB%83-0>