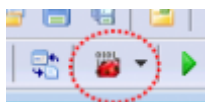


El modo Debug

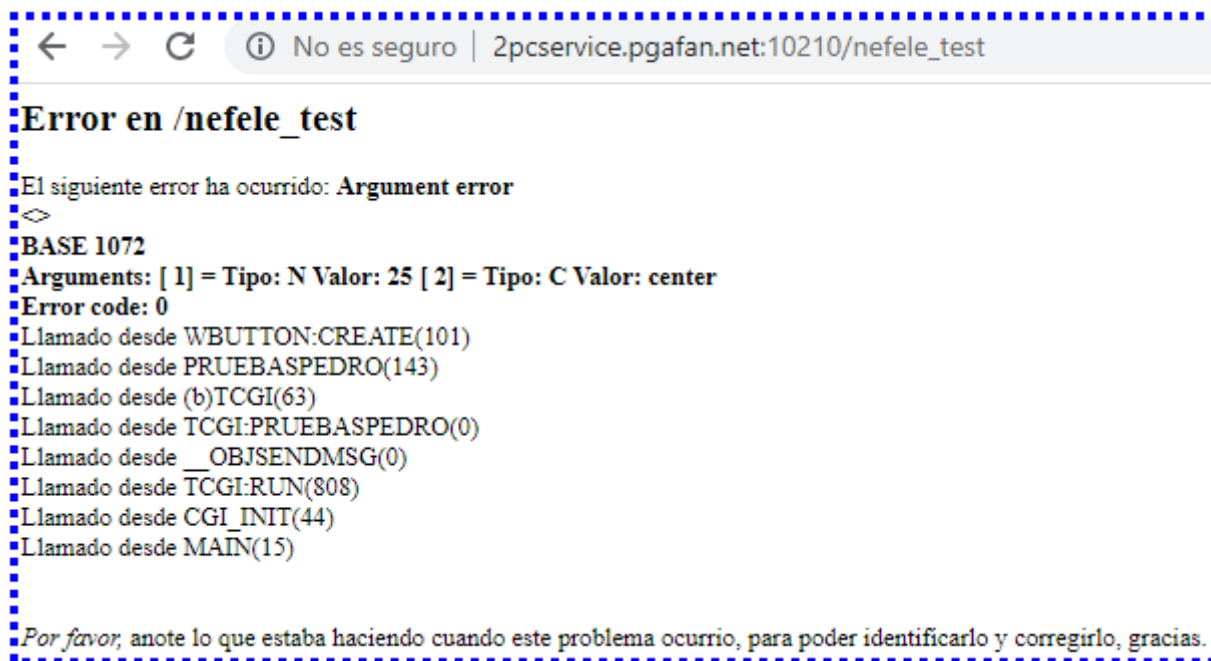
Como Néfele lo hemos creados entre desarrolladores de software de gestión que utilizamos habitualmente [Xailer](#), estamos acostumbrados a utilizar el modo Debug de Xailer, en Néfele no tiene sentido ya que no puedes ejecutar tu aplicación dentro del IDE, pero teníamos claro que queríamos algo similar o mejor.

Utilizamos el propio IDE para habilitar y deshabilitar el modo Debug en nuestras aplicaciones.



Esto nos permite que nuestra aplicación sepa si esta en modo Debug solo consultado a la función de Xailer **IsDebug()**.

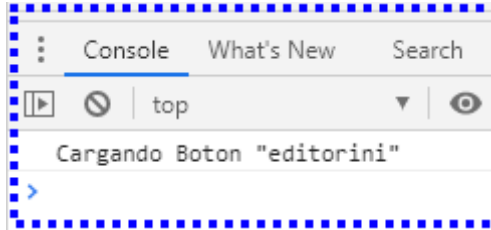
Una vez habilitado, si se produce un error en nuestra aplicación nos mostrara el error en el navegador que realizo la consulta.



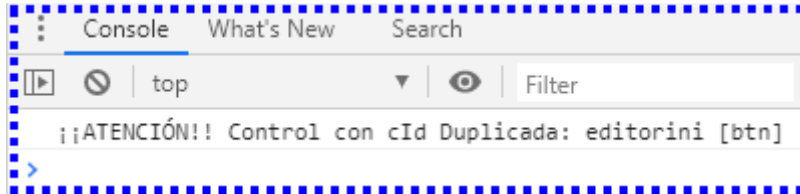
También podremos enviar mensajes a la consola del navegador con [LogDebug\(\)](#).

```
WITH OBJECT WBevel():New(:WO)
  :LogDebug("Cargando Boton 'editorini'")
WITH OBJECT WButton():New(:WO)
  :cId      := "editorini"
  .
  .
```

Al llegar a ese punto de la ejecución en el Navegador nos aparecerá...



De esta manera también se nos comunicara Néfele con algunos errores que aunque no producen la rotura de la pagina si es conveniente resolverlos, por ejemplo cuando por error le pongamos el mismo **cId** a dos controles.

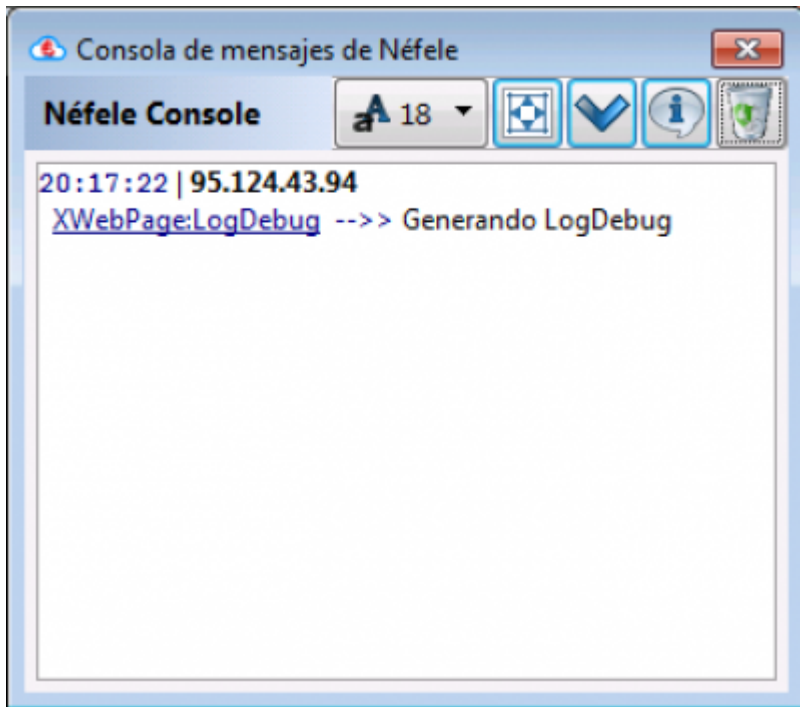


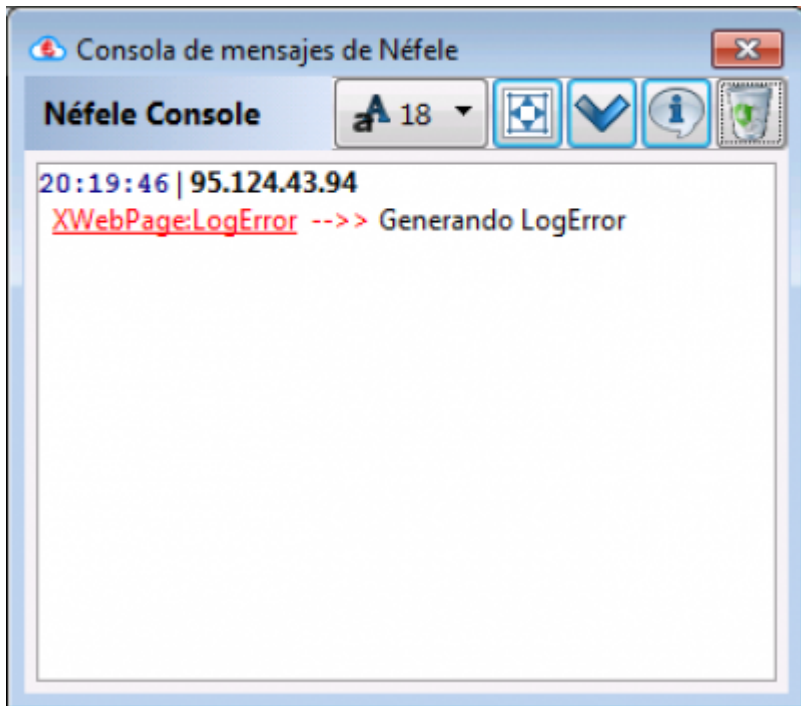
La tercera funcionalidad del modo Debug nos la da [Néfele Console](#) que es parte de las herramientas que acompañan a Néfele.

Desde nuestros programas podemos enviar mensajes a [Néfele Console](#) con la instrucción `oCGI:Console()` de esta manera.

```
oCGI:Console("Mensaje que aparecerá en Néfele Console")
```

También nos aparecerán en NefeleConsole los mensajes generados con los métodos [LogDebug](#) y [LogError](#) de cualquier control de Néfele.





El modo Debug de Néfele también nos añade comentarios en el código HTML generado para que nos sea más fácil depurar los posibles errores, y nos presenta los parámetros de una forma legible.

```
1.0.0/js/materialize.min.js"></script>
<!-- Scripts Propios -->
<script>
  function editorini_onclick(e = 'Onclick') {return
  autoform({'FUNCTION':'EditorINI','editorini': e,
  'INIFILE':'appgestion'}, "", "");}
  <!-- -->
  function formulario_onclick(e = 'Onclick') {return
  autoform({'FUNCTION':'formulario','formulario': e}, "", "");}
  <!-- -->
```

Cuando no estamos en modo Debug los parámetros son algo menos legibles.

```
1.0.0/js/materialize.min.js"></script>
<script>
  function editorini_onclick(e = 'Onclick') {return
  autoform({'np_47575046574d53534f': 'np_46666b7772764d534e60', 'np_66
  666b7772766d736e6e': e,
  'np_4a504b494c50494b': 'np_6272726a6877786e74746f'}, "", "");}
  function formulario_onclick(e = 'Onclick') {return
  autoform({'np_47575046574d53534f': 'np_67717470787065776e7570', 'np_
  67717470787065776e7570': e}, "", "");}
  </script>
```

Modo Debug en Producción

Cuando nuestro CGI esta en producción y los hemos compilado sin Debug, podemos forzar a que pase a modo Debug añadiendo en el directorio donde este nuestro CGI un fichero con el mismo nombre del CGI con la extensión **.debug**.

Al forzar el modo Debug en nuestro CGI en producción se habilita también la generación de ficheros error.log con información de los errores producidos e información de nuestro CGI en el momento de producirse el error.

Este fichero se comporta como un *.ini* por lo que podemos utilizarlo para deshabilitar estos parámetros de manera individual.

```
[Debug]
Force = .T.
LogError = .T.
```

From:

<https://www.nefele.dev/wiki/> - **Nefele Project**

Permanent link:

<https://www.nefele.dev/wiki/tecnicas/debugger?rev=1605913744>

Last update: **20/11/2020 23:09**

