

# El modo Debug

Como Néfele lo hemos creado 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 está en modo Debug solo consultando a la función de Néfele [nfl\\_IsDebug](#).

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

The screenshot shows a browser window with a blue dashed border. At the top, there are navigation buttons (back, forward, refresh) and a status bar that says "No es seguro | 2pcservice.pgafan.net:10210/nefele\_test". Below the status bar, the main content area displays an error message:

**Error en /nefele\_test**

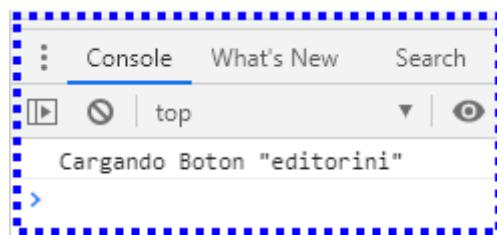
El siguiente error ha ocurrido: **Argument error**  
=>  
**BASE 1072**  
**Arguments: [ 1 ] = Tipo: N Valor: 25 [ 2 ] = Tipo: C Valor: center**  
**Error code: 0**  
Llamado desde WBUTTON:CREATE(101)  
Llamado desde PRUEBASPEDRO(143)  
Llamado desde (b)TCGI(63)  
Llamado desde TCGI:PRUEBASPEDRO(0)  
Llamado desde \_\_OBJSENDMSG(0)  
Llamado desde TCGI:RUN(808)  
Llamado desde CGI\_INIT(44)  
Llamado desde MAIN(15)

*Por favor, anote lo que estaba haciendo cuando este problema ocurrió, para poder identificarlo y corregirlo, gracias.*

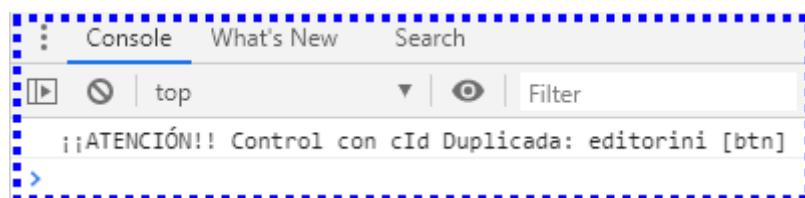
También podremos enviar mensajes a la consola del navegador con [LogDebug\(\)](#).

```
WITH OBJECT WBevel():New(:W0)
:LogDebug("Cargando Botón 'editorini'")
WITH OBJECT WButton():New(:W0)
: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.

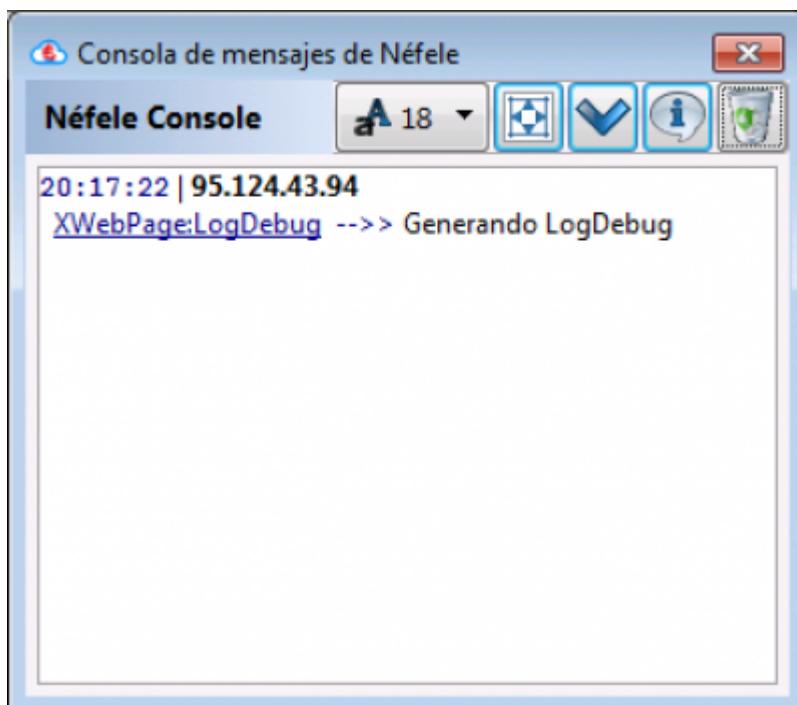


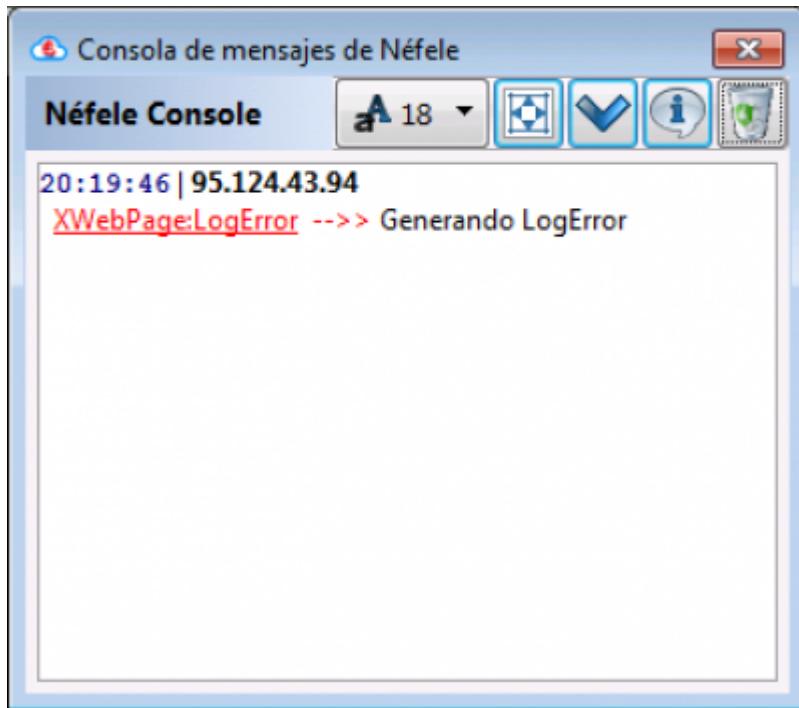
La tercera funcionalidad del modo Debug nos la da [Néfele Console](#) que es parte de las herramientas que acompañan a Néfele.

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 NéfeleConsole los mensajes generados con los métodos [LogDebug](#) y [LogError](#) de cualquier control de Néfele.





El modo Debug de Nefele 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,'',''});}
<!-- -->
```

The code block shows a snippet of JavaScript from a file named "materialize.min.js". It contains two functions: "editorini\_onclick" and "formulario\_onclick". Both functions call the "autoform" function with specific parameters. The code is presented in a readable, commented-out style typical of browser developer tools in debug mode.

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_666b7772766d736e6e': e,
        'np_4a504b494c50494b':'np_6272726a6877786e74746f','',''});
    function formulario_onclick(e = 'Onclick') {return
        autoform({'np_47575046574d53534f':'np_67717470787065776e7570','np_67717470787065776e7570': e,'',''});}
```

This code block shows the same JavaScript code as above, but it is heavily obfuscated. All variable names and parameter values are replaced by long, random strings of lowercase letters and numbers, making the code difficult to read and understand without context.

## Modo Debug en Producción

Cuando nuestro CGI esta en producción y lo 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 **.debug** se comporta como un **.ini** por lo que podemos utilizarlo para deshabilitar estos parámetros de manera individual.

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

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

Permanent link:  
<https://nefele.dev/wiki/tecnicas/debugger>

Last update: **20/03/2022 18:29**

