

Artículo

[Bernardo Linarez](#) · 22 nov, 2021 · Lectura de 3 min

CALL asíncronicas dinámicas y sentencia SYNC

En algún momento, puede surgir la necesidad de llamar (sentencia CALL en BPL) de manera dinámica, porque hay mucha lógica de negocio compartida por distintos orígenes de datos, que da pie a que esa lógica se comparta y no se individualice cada llamada (CALL).

Esto tiene ventajas a nivel de mantenimiento de la lógica definida, catch de errores y administrar los hilos de ejecución de manera centralizada.

Entonces lo normal es que se defina una sentencia FOREACH que ejecute la sentencia CALL según cada ítem que se necesite.

```
<foreach name='Origenes activos' property='context.FeedOrigenes' key='context.countFeedOrigenes' >
```

```
<call name='CallHost' target='Host' async='1'>
</foreach>
```

Entonces, con la finalidad de utilizar mejor el tiempo, estas distintas CALL se solicitan de manera asíncrona (async='1'), pero es necesario esperar que todas ellas sean respondidas desde los orígenes de datos. Entonces tenemos que usar la sentencia SYNC. Normalmente lo usaríamos de esta manera:

```
<sync name='Sincroniza Calls' calls='CallHost' />
```

Sin embargo, aquí empieza la dificultad, dado que al indicarle esto al BP, este sólo espera hasta que la primera llamada termine su ciclo, de todas las llamadas dinámicas realizadas, y continua la ejecución de la lógica posterior. Como es de esperarse, esto nos generará errores, porque nuestra lógica necesita de la información de todos los orígenes de datos.

Para suerte nuestra, IRIS permite que tanto la sentencia CALL como la sentencia SYNC, se puedan definir con variables invocadas por indirección (indirection). En otras palabras, podemos definir variables de contexto, que podemos luego usar para definir propiedades de sentencias como CALL y SYNC.

Entonces en nuestra lógica del ejemplo, generaremos no sólo la sentencia CALL de manera dinámica, sino también el NAME de la sentencia CALL, de la siguiente manera:

```
<context>
<property name='strCrearCallName' type='%String' instantiate='0' >
```

```
<property name='strSyncCallName' type='%String' instantiate='0' >
```

```
...
```

```
</context>
```

```
...
```

```
<foreach name='Origenes activos' property='context.FeedOrigenes' key='context.countFeedOrigenes' >

//Definimos el valor de la propiedad Name de nuestra sentencia CALL, usando el contador del foreach
<assign name="Crear CallName" property="context.strCrearCallName" value="&quot;Call&quot;_context.countFeedOrigenes" action="set"/>

//Acumulamos los valores de los sucesivos Name para poderlos usar en la sentencia SYNC. Notar que la concatenación se realiza a través de una ","
<assign name="Añadir Cadena Sync" property="context.strSyncCallName" value="context.strSyncCallName_&quot;;&quot;_context.strCrearHCCCAllName" action="set">

//Llamamos al Host respectivo con un Name en una variables de contexto, a través de indirección
<call name='@context.strCrearCallName' target='Host' async='1'>
</foreach>

...

//Finalmente llamamos al SYNC con la variable de contexto que acumulaba todos los Name, a través de indirección.
<sync name='Sincroniza Calls' calls='@context.strSyncCallName' timeout='72000' type='all'/>
```

Espero que les sirva.

[#Ensemble #InterSystems IRIS](#)

URL de
fuente: <https://es.community.intersystems.com/post/call-as%C3%ADncronicas-din%C3%A1micas-y-sentencia-sync>