

Artículo

[Ricardo Paiva](#) · 11 mar, 2022 · Lectura de 8 min

[InterSystems IRIS por primera vez] Interoperabilidad: Crear componentes (Business Process)

Este artículo es una continuación de [esta publicación](#).

En ese artículo, analizamos la creación de business operations a partir de los componentes necesarios para la integración del sistema.

En este artículo, mostraré cómo crear un business process que llama a las dos business operations definidas en el orden de la secuencia.

- [Producción](#)
- [Mensaje](#)
- **Componentes**
 - Business Services
 - **Business Processes**
 - [Business Operations \(publicación anterior\)](#)

El business process actúa como coordinador (centro de mando) del proceso.

Los ajustes en el proceso que se pueden implementar en la muestra son los siguientes:

Paso 1: Proporcionar el nombre de la ciudad a una API web externa y solicitar **información meteorológica**.

Paso 2: Registrar el resultado de la consulta (**información meteorológica**) del paso 1 y el nombre del producto comprado que se recibió al inicio de la producción.

En el business process de muestra, esperaremos la respuesta al paso 1 y ajustaremos el paso 2 para que funcione.

En el proceso de espera de una respuesta (es decir, de sincronización), por ejemplo, ¿qué ocurre si el paso 1) no responde durante unos días?

Si se entregan nuevos mensajes al business process mientras se espera una respuesta durante unos días, los mensajes no serán descartados ya que se almacenan en una **cola**. Sin embargo, el business process no procesará nuevos mensajes y habrá un retraso en la operación.

Nota: Los business processes y las business operations tienen colas.

Por lo tanto, en producción, cuando hay una llamada sincronizada, hay dos maneras de que el business process pueda moverse: **A) Hacer una sincronización perfecta**, y **B) Guardar el estado del propio business process en la base de datos y transferir el entorno de ejecución para que otros procesos puedan ejecutarse mientras esperan una respuesta.**

A) Cómo hacer una sincronización perfecta:

Mientras se realiza una llamada sincronizada, el procesamiento del business process está en marcha, y esperando a que se procese el siguiente mensaje hasta que se complete todo el procesamiento.

Esta función se utiliza cuando se necesita garantizar el orden del procesamiento en el método "primero en entrar, primero en salir".

B) El método para guardar el estado del propio business process en la base de datos y transferir el entorno de ejecución para que otros procesos puedan ejecutarse mientras esperan una respuesta es

Cuando se realiza una llamada sincronizada, el proceso guarda su estado en la base de datos. Cuando se recibe un mensaje de respuesta, y es el momento de procesar el mensaje, se abre la base de datos y se ejecuta el siguiente proceso.

(IRIS gestionará el almacenamiento y la reapertura de los business processes en la base de datos).

Se utiliza cuando es aceptable cambiar el orden para procesar los mensajes (es decir, cuando se permiten procesar más y más mensajes diferentes recibidos mientras se espera una respuesta).

En la muestra, se utiliza B).

Hay dos tipos de editores para crear business processes: un Editor de Business Processes que permite colocar cuadros de procesamiento (actividades) e implementarlos mientras se define su ejecución, y un método para crearlos mediante ObjectScript en Studio o VSCode.

Si utilizas el Editor de Business Processes, utilizarás la actividad de llamada para invocar al componente, pero esta actividad está implementada de la forma B). Por supuesto, también puedes implementar el método A) en el Editor de Business Processes, excepto que en ese caso no utilizará la actividad de llamada (utilizará la actividad del código).

En esta sección, explicaré cómo crearla.

Si se utiliza el Editor de Business Processes, deben escribirse en el Portal de Administración.

También puedes abrir el business process desde la página de configuración de la producción. La siguiente imagen muestra el procedimiento.

Interoperability > Production Configuration - (Start.Production)

Production Configuration

Start Stop

Production Running Category: All Legend Production Settings

Services Processes Operations

- EnsLib.JavaGateway.Service
- Start.FileBS
- Start.NonAdapterBS
- Start.WS.WebServiceBS

Start.WeatherCheckProcess

- Start.GetKionOperation
- Start.InsertOperation
- Start.SQLInsertOperation

- Click business process name
- Click Settings tab
- Expand [Informational Settings]
- Click icon on the right side of class name

Start.WeatherCheckProcess

Settings Queue Log Messages Jobs

Apply Search:

Informational Settings

Comment

Category

Class Name Start.WeatherCheckProcess

Interoperability > Business Process Designer - (Start.WeatherCheckProcess)

New Open Save Save As Compile 75% -Add Activity- -Group Items-

Business Process Start.WeatherCheckProcess
Last modified: Tuesday, January 26, 2021, 01:14:04PM

```
graph TD; Start((<start>)) --> Call1[<call> 気象情報取得]; Call1 --> Call2[<call> 気象情報DB登録]; Call2 --> End((<end>));
```

General Context Activity Preferences

General settings for this Business Process

Language ObjectScript Basic

Layout Automatic Manual Width 2000

Annotation

Includes

Optional list of include files

Version

Version number

Is component

Iconos como en este editor se llaman actividades, y las que están marcadas como son actividades que pueden invocar otros componentes.

Este símbolo indica que se devolverá un mensaje de respuesta (es decir, se realizará una llamada sincronizada). La actividad se ajusta de forma predeterminada a la configuración de la llamada no sincronizada, que puede cambiarse según sea necesario.

Ahora observemos los business processes, que son componentes que se invocan al recibir un mensaje de solicitud, así como las business operations.

En la muestra, el mensaje de solicitud: Se pone en marcha cuando recibe un [Start.Request](#) y no devuelve un mensaje de respuesta.

En el business process, los mensajes aparecen en varias situaciones.

Mensajes de solicitud que se envían a los business processes.

Mensaje de solicitud (más un mensaje de respuesta) que se envía cuando se llama a otro componente usando la actividad.

En el Editor de Business Processes, los nombres de los objetos que almacenan mensajes están claramente separados para poder ver qué mensaje se envió desde qué destino.

- solicitud (requisitos básicos)

El mensaje que activó el inicio del business process, en nuestro ejemplo, es [Start.Request](#) (el mensaje que se debe especificar en la configuración de la Solicitud en la pestaña Contexto dentro del Editor de Business Processes)

- respuesta (respuesta básica)

Mensaje de respuesta para devolver a la persona que llama al business process (no se utiliza en el ejemplo) (mensaje que se debe especificar en la configuración de la respuesta que aparece en la pestaña Contexto en el Editor de Business Processes)

- callrequest (mensaje de solicitud)

Mensaje de solicitud que se envía al llamar al componente determinado por la actividad.

- callresponse (mensaje de respuesta)

Mensaje de respuesta devuelto desde el componente especificado por la actividad.

callrequest y callresponse son objetos que se eliminarán cuando se complete el procesamiento de llamada de la actividad. Todos los demás objetos no desaparecerán hasta que finalice el business process.

Ahora se presenta el problema cuando desaparece callresponse.

Esto es porque, como se puede ver en este ejemplo, Cuando se llama a un componente, si se quiere utilizar el resultado de la respuesta de un componente llamado previamente, se perderá el mensaje de respuesta, y se borrará la información que se iba a utilizar en el siguiente componente.

Es un problema

¿Qué deberíamos hacer?

En este caso, se puede utilizar el objeto de contexto.

El objeto de contexto, al igual que la solicitud/respuesta, es un objeto que sobrevive hasta el final del business process.

Además, como el contexto es un objeto genérico, se puede definir en el editor de proceso.

Además del contexto, también se puede utilizar el objeto de respuesta si tiene una propiedad que coincida con lo que guarda la información heredada.

Ahora, vamos a repasar los pasos de nuevo.

El mensaje de respuesta en el [globo azul claro](#): [Start.Response](#) es un objeto que se eliminará cuando termine el proceso.

Como queremos utilizar el mensaje de respuesta ([Start.Response](#)) que contiene la información meteorológica como el mensaje que se enviará a la siguiente [Business Operation para la actualización de la base de datos], tenemos que implementar el objeto de contexto de tal forma que todos los valores de la propiedad del mensaje de respuesta ([Start.Response](#)) se puedan asignar a él.

Entonces, ¿cuál es la configuración para la propiedad de contexto?

Las propiedades se definen en "Context Properties" en la pestaña Context del Editor de business processes.

En este caso, nos gustaría guardar todas las propiedades del mensaje de respuesta ([Start.Response](#)) en el objeto

de contexto. Por lo tanto, la especificación del tipo de propiedad se establece en [Start.Response](#).

A continuación, consulta la configuración en la actividad.

Los mensajes de solicitud y de respuesta tienen un botón llamado `Builder`.

Al hacer clic en este botón se iniciará un editor de líneas que permite especificar lo que se quiere registrar en las propiedades de cada mensaje.

Después de esto, la business operation para solicitar una actualización de la base de datos ([Start.SQLInsertOperation](#) o [Start.InsertOperation](#)) se llama de la misma manera con la actividad, y todo estará listo.

(Para más información, consulta [Configuring para Business Processes](#)).

Cuando se haya completado la verificación, se podrá probar. El método de prueba es el mismo que se utiliza para probar las business operations (consulta [este artículo](#)).

El seguimiento después de la prueba es el siguiente:

Como el business process es el coordinador, pudimos ver que invocaba de forma secuencial los componentes definidos, lo que mantiene la ejecución sincronizada.

Nota 1: Este ejemplo solo se refiere a la actividad sobre las llamadas, pero hay otras actividades, como la transformación de datos.

Nota 2: Los business processes creados únicamente por ObjectScript, distintos al Editor de Business Processes, se heredan de la clase `Ens.BusinessProcess`. Si se crea en el Editor de Business Processes, se hereda de la clase `Ens.BusinessProcessBPL`.

El business process es el coordinador del proceso de integración del sistema.

El Editor de Business Processes ofrece los siguientes tipos de variables para los mensajes (`request/response/callrequest/callreponse/context`).

Un business process creado con el Editor de Business Processes puede funcionar de forma que no retrase otros mensajes, incluso si hay sincronización en la llamada del componente.

En el próximo artículo, mostraré finalmente cómo desarrollar el último componente: los business services.

[#Interoperabilidad](#) [#Principiante](#) [#Proceso empresarial](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de
fuente: <https://es.community.intersystems.com/post/intersystems-iris-por-primera-vez-interoperabilidad-crear-componentes-business-process>