

Artículo

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

## Interacción y personalización Repositorio FHIR

Siempre que se trabaja con el repositorio FHIR, se tiene que interactuar con lógica de negocio propia de la institución donde se instala.

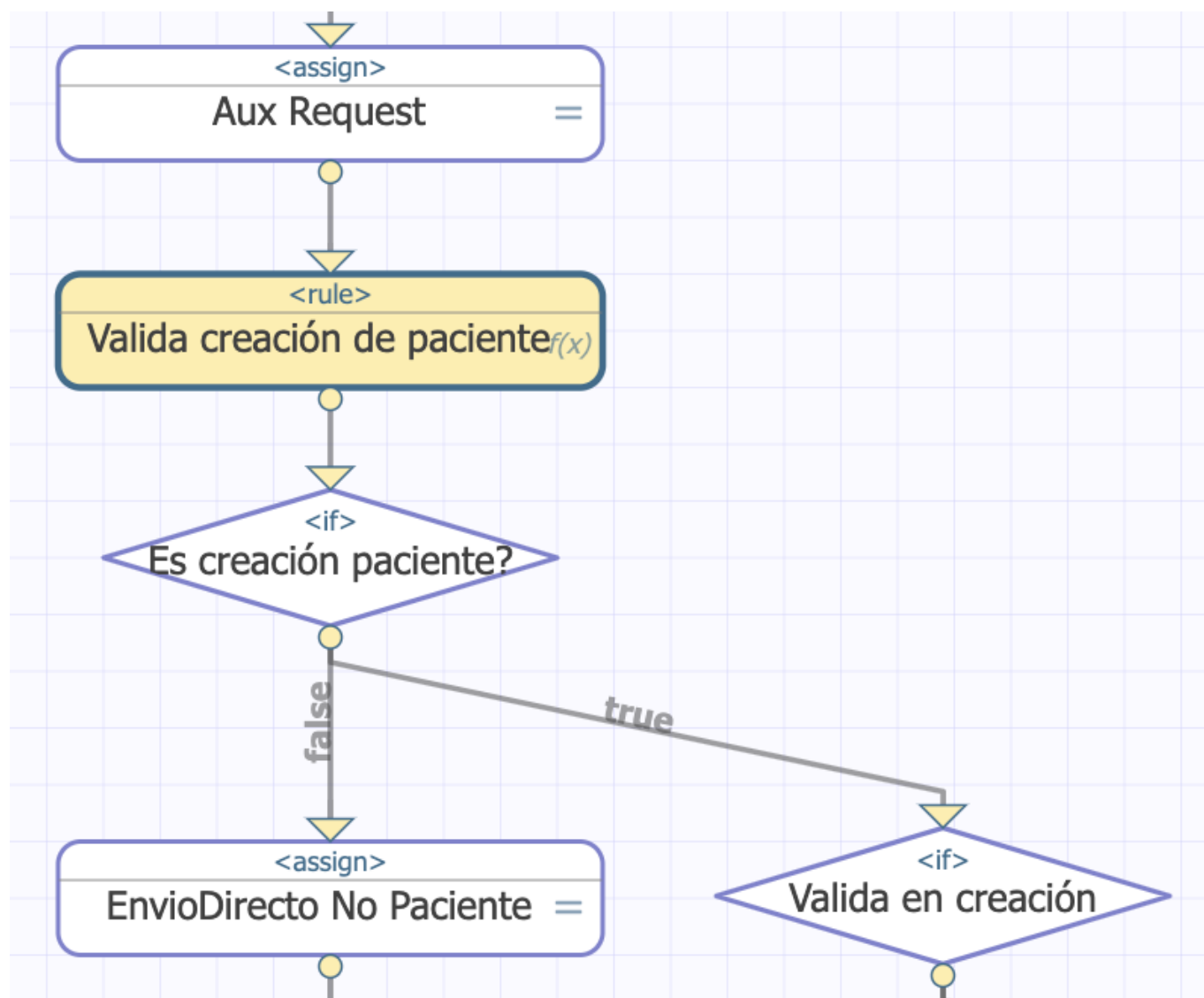
Surgió una regla bastante simple, para efectos de demostración, que consiste en validar la existencia de un paciente en el repositorio, aún si es una petición que contiene un Patient Resource enviado vía método POST, lo cuál por definición del estándar FHIR, corresponde a un Create Patient. Las actualizaciones de cualquier recurso FHIR se realizan a través del método HTTP UPDATE, sin embargo podría suceder que por error se nos envíe una petición POST y se creen numerosas instancias del mismo paciente.

La unificación de registros de pacientes, es algo que normalmente es resuelto por un Maestro de pacientes (MPI/EMPI), sin embargo, en casos de uso acotado, donde se pueda confiar en un único identificador nacional de paciente, pues se puede considerar que el repositorio realice la validación de la existencia del paciente a través de una consulta previa.

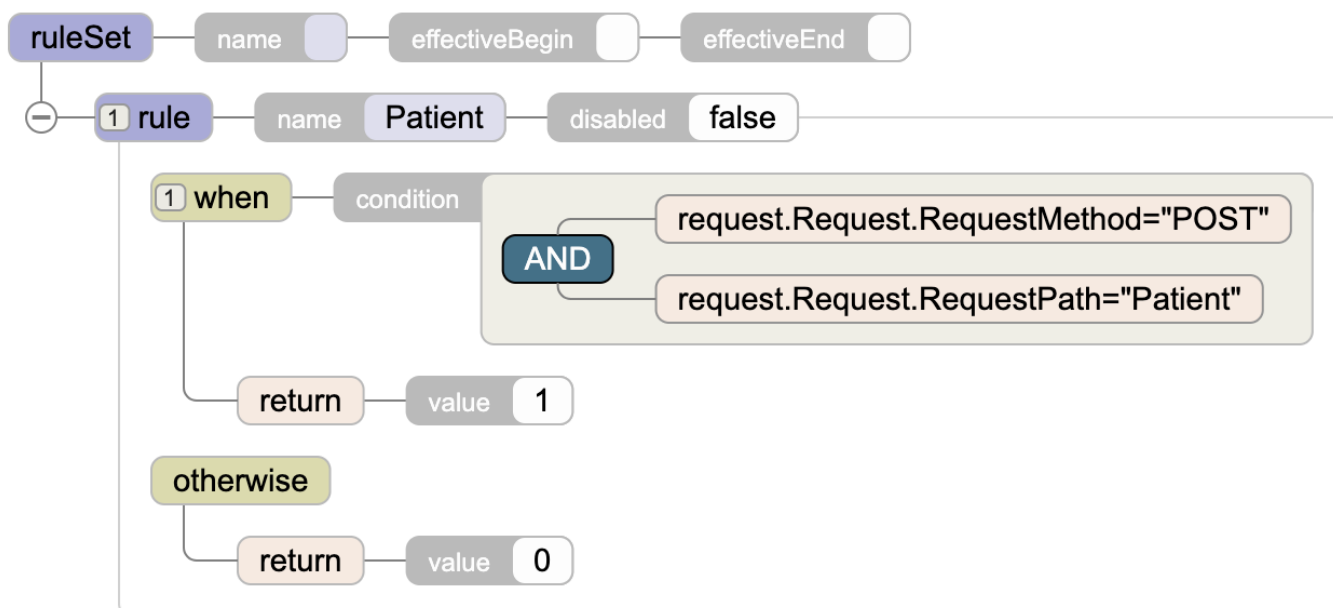
Entonces para nuestra demo usamos una producción que interactúa con un servidor FHIR, instalado con configuración default, con un Business Process (BP) orquestando la validación:

Producción en Ejecución			Categoría: FHIR	Leyenda	Configuración de producción
Servicios +	Procesos +	Operaciones +			
● HS.FHIRServer.Interop.Service	● MEDERI.BP.MPI.Process	● HS.FHIRServer.Interop.Operation			

Nuestro BP evalúa primeramente el tipo de mensaje con la finalidad de evaluar si debía aplicarle la lógica de validación o directamente enviarlo al repositorio FHIR:



La regla de negocio es bastante sencilla, dado que estaba centrada solo en identificar el mensaje del recurso FHIR del paciente. Pasamos el request de tipo HS.FHIRServer.Interop.Request al contexto del BP, para que fuera visible desde la regla:



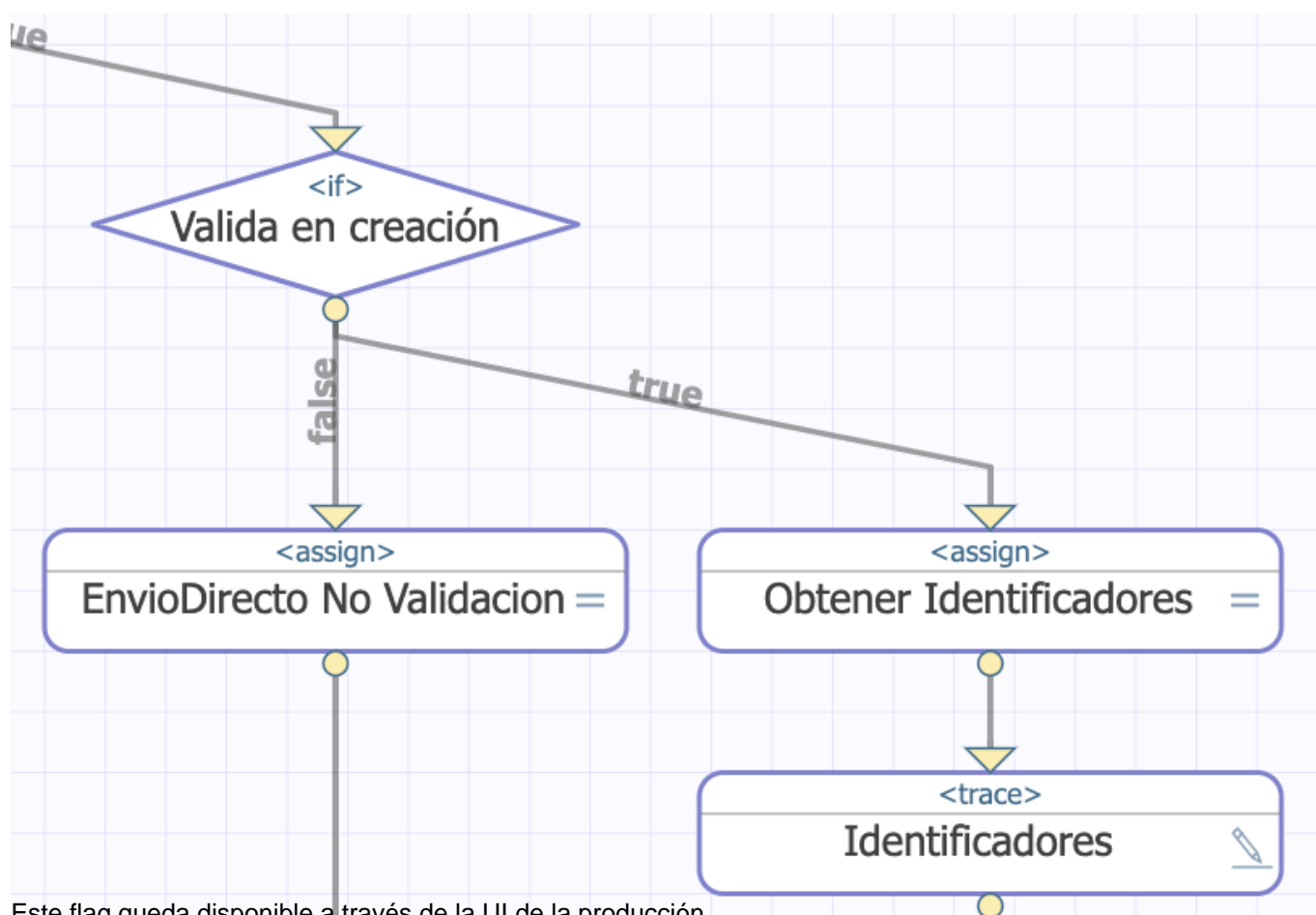
Incorporamos un flag en el BP que controla si se debe aplicar la validación, de lo contrario se envía el mensaje al servidor FHIR:

```
Class MEDERI.BP.MPI.Process Extends Ens.BusinessProcessBPL  
{
```

```
Property ValidarPacientesExistentesEnCreacion As %Boolean;
```


```
Parameter SETTINGS = "ValidarPacientesExistentesEnCreacion:Basic";
```

```
<if name='Valida en creación ' condition='process.ValidarPacientesExistentesEnCreacion > 0' xpos='470' ypos='600' xend='470' yend='2000' >  
<true>
```




# MEDERI.BP.MPI.Process

**Configuración** Cola Log Mensajes Procesos

**Aplicar** ▼  Buscar:

Descripción del Adaptador

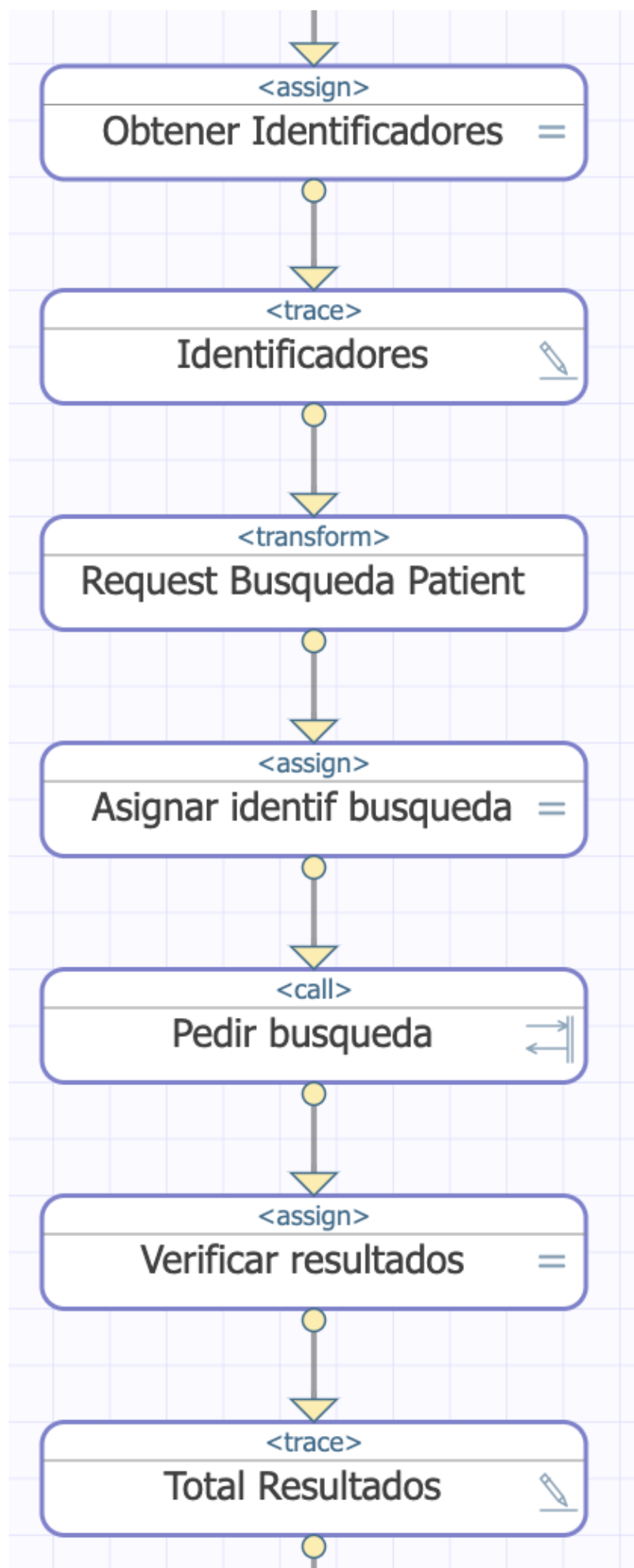
**Actor de Negocio**  
 ▼ 

▼ **Parámetros Básicos**

Habilitado  
☒

ValidarPacientesExistentesEnCreacion  
☒

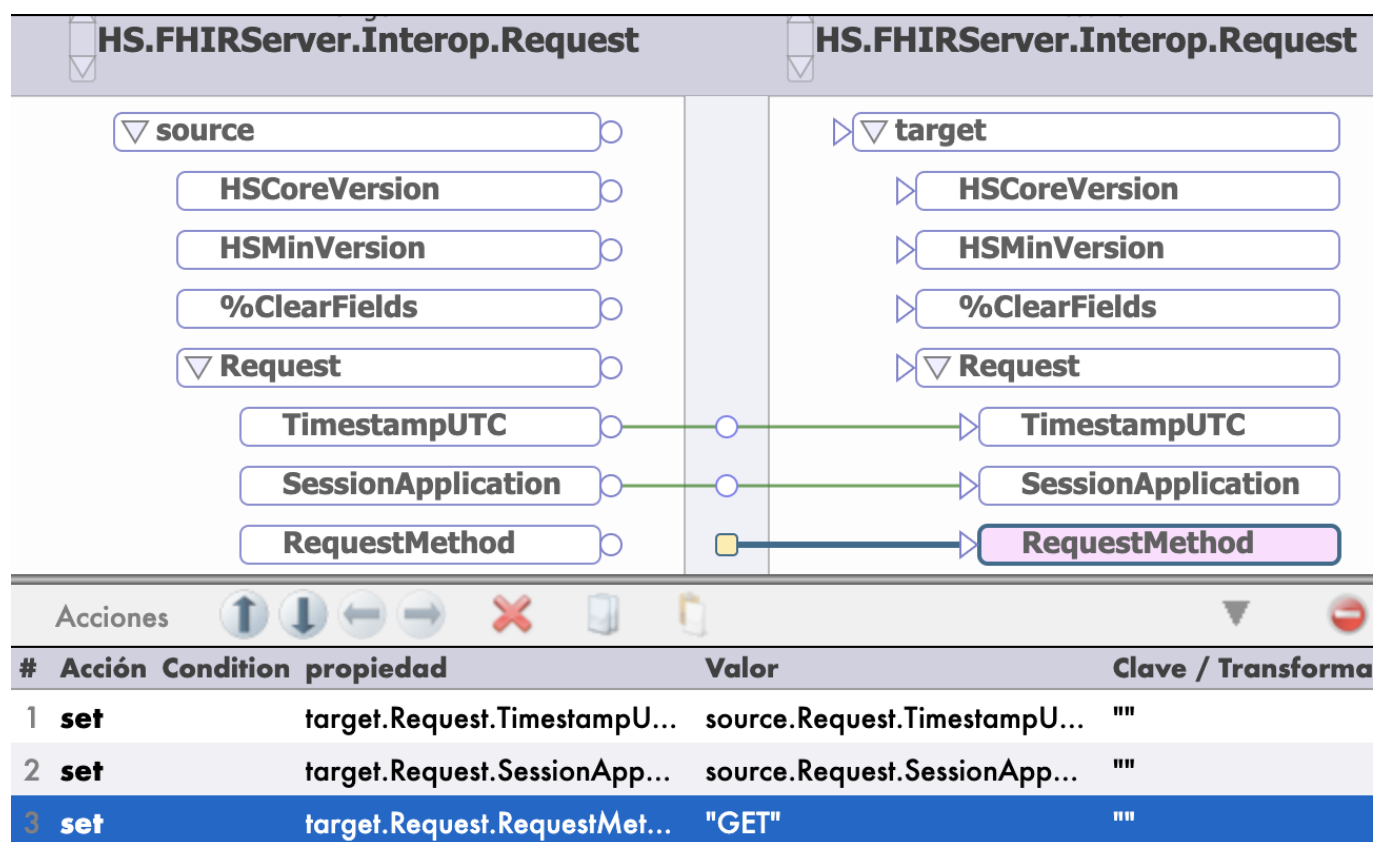
La serie de actividades de la validación, las vemos en la siguiente secuencia:



El BP llama a un método personalizado para obtener los identificadores desde el mensaje FHIR. Preferimos usar un método de clase, dado que como resultados necesitamos una cadena de texto concatenada para pasarla como parámetro de búsqueda de FHIR

```
<assign name="Obtener Identificadores" property="context.identificadores" value="##class(MEDERI.Facade.Patient).GetListaIdentificadores(request,.status)" action="set"/>
```

Cabe recordar que el Business Service (BS) FHIR core, genera un objeto del tipo HS.FHIRServer.Interop.Request, el cual nosotros tomamos como parámetro de nuestro BP. La gran parte de las propiedades del objeto Request de tipo HS.FHIRServer.Interop.Request para el BO FHIR, las obtenemos desde la Request original, simplemente asignando todas las propiedades que tienen valor, a través de una DTL. El cambio importante es el método HTTP, representado por la propiedad RequestMethod, al cuál le asignamos la cadena "GET"



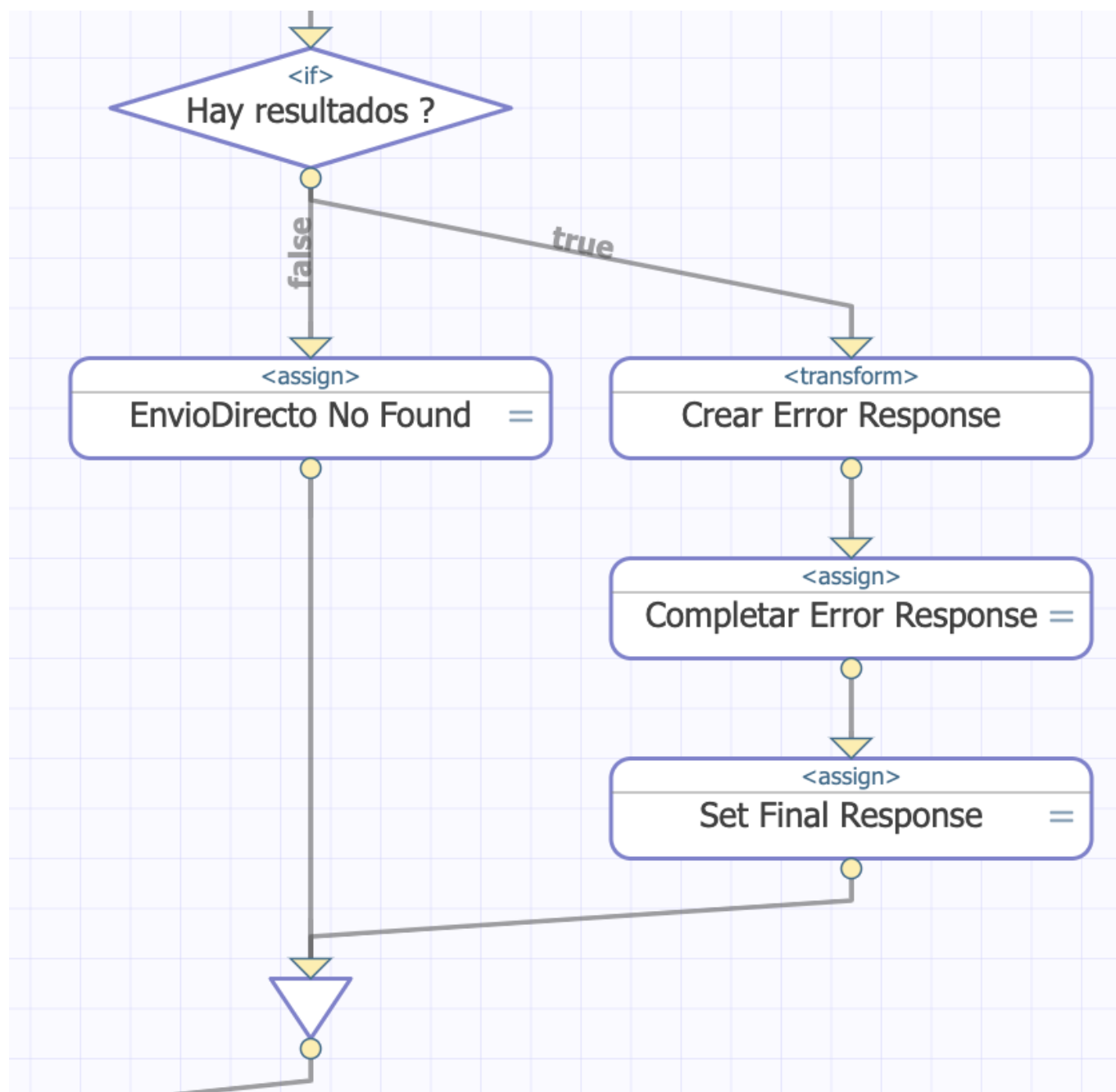
Y luego le asignamos la cadena de texto de los identificadores a la propiedad QueryString:

```
<assign name="Asignar identif busqueda" property="context.patientSearchRequest.Request.QueryString" value="context.identificadores" action="set"/>
```

Posteriormente enviamos este nuevo Request al repositorio FHIR. Este nos responderá con el resultado de la búsqueda, a través de un recurso del tipo Bundle, del cuál nos interesa la propiedad "total", el cuál rescatamos a través de un método de clase personalizado:

```
"resourceType": "Bundle",
"id": "7ebbc226-4bda-11ec-b91d-000c29e920c9",
"type": "searchset",
"timestamp": "2021-11-22T21:23:54Z",
"total": 2,
"link": [
  {
    "relation": "self",
    "url": "https://dm.intersystemschi."
  }
],
"entry": [
  {
    "fullUrl": "https://dm.intersystemschi."
  }
]
```

De encontrar pacientes que coincidan con los identificadores, entonces enviamos un mensaje de error personalizado. De no encontrarlo, le dejamos la responsabilidad al repositorio FHIR:



Este mensaje de error personalizado se modela como un objeto de tipo HS.FHIRServer.Interop.Response. Si llamamos vía Postman, lo veremos de la siguiente forma:



The screenshot shows a REST client interface with a POST request to `https://dm.intersystemschi.cl/fhir/r4/Patient...`. The request body is a JSON object representing a Patient resource with an extension for nationality. The response is a 400 Bad Request with a detailed error message in Spanish.

```
POST https://dm.intersystemschi.cl/fhir/r4/Patient...

{
  "resourceType": "Patient",
  "extension": [
    {
      "url": "nationality",
      "valueCodeableConcept": {
        "coding": [
          {
            "code": "242",

```

Body Cookies Headers (8) Test Results Status: 400 Bad Request Time: 1346 ms Size: 732 B Save Response

```

{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "invalid",
      "diagnostics": "<HSFHIR>NoCreateUpdate",
      "details": {
        "text": "Error personalizado: El paciente con estos identificadores ya existe (identifier=REGISTROCI|22222222-2, bajo id(s) ya existentes (,314540,314541,314542,314543). Se puede modificar sus datos usando el método PUT como lo establece FHIR, referenciando al id correspondiente."
      }
    }
  ]
}

```

Dado que el mensaje incluye información de los identificadores, entonces usamos un método de clase para crear la descripción del mensaje.

Pronto les compartiré el proyecto para que puedan visualizar el código fuente que generamos.

Espero esta demostración les pueda ayudar.

[#FHIR](#) [#HealthShare](#) [#InterSystems IRIS for Health](#)

URL de  
fuente: <https://es.community.intersystems.com/post/interacci%C3%B3n-y-personalizaci%C3%B3n-repositorio-fhir>