

IBM Unica Campaign  
Versión 8 Release 6  
30 de abril 30 de 2012

*Guía PDK de validación*

**IBM**

**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado "Avisos" en la página 15.

---

## Contenido

### IBM Unica PDK (Plug-in Developer's Kit) de validación . . . . . 1

|   |   |
|---|---|
| Ayuda adicional de PDK de validación . . . . .    | 1 |
| Contenido del PDK de validación . . . . .         | 1 |
| Componentes del PDK de validación . . . . .       | 2 |
| Buscar material de referencia . . . . .           | 2 |
| Prestaciones del PDK de validación. . . . .       | 2 |
| Cargar el plug-in en IBM Unica Campaign . . . . . | 3 |
| Validación de oferta versus campaña . . . . .     | 4 |
| Configuración del PDK de validación . . . . .     | 4 |
| validationClass . . . . .                         | 4 |
| validationClasspath . . . . .                     | 5 |
| validatorConfigString . . . . .                   | 5 |
| Desarrollar un plug-in . . . . .                  | 6 |
| Visión general . . . . .                          | 6 |
| Pasos principales para crear plug-ins . . . . .   | 7 |
| Configuración . . . . .                           | 7 |

|  |    |
|--|----|
| Crear los validadores . . . . .  | 8  |
| Configurar IBM Unica Campaign . . . . .                                | 8  |
| Probar la configuración del validador . . . . .                        | 8  |
| Crear un validador . . . . .   | 9  |
| Ejecutable de ejemplo . . . . .  | 9  |
| Configurar Campaign para el plug-in ejecutable<br>de ejemplo . . . . . | 10 |
| Interfaz de uso de ejecutable esperado . . . . .                       | 10 |
| Ejemplo . . . . .  | 11 |
| Impedir ediciones de campaña . . . . .                                 | 11 |

### Contactar con el soporte técnico de IBM Unica . . . . . 13

### Avisos . . . . . 15

|                              |    |
|------------------------------|----|
| Marcas registradas . . . . . | 17 |
|------------------------------|----|



---

## IBM Unica PDK (Plug-in Developer's Kit) de validación

El PDK (Plug-in Developer's Kit) de validación de IBM® Unica le permite desarrollar lógica de validación personalizada en IBM Unica Campaign. El PDK de validación es una subclase de una estructura de plug-in más genérica que se proporciona con IBM Unica Campaign.

Las secciones clave de este capítulo incluyen:

- “Contenido del PDK de validación”
- “Prestaciones del PDK de validación” en la página 2
- “Configuración del PDK de validación” en la página 4
- “Desarrollar un plug-in” en la página 6
- “Pasos principales para crear plug-ins” en la página 7
- “Ejecutable de ejemplo” en la página 9
- “Ejemplo” en la página 11

---

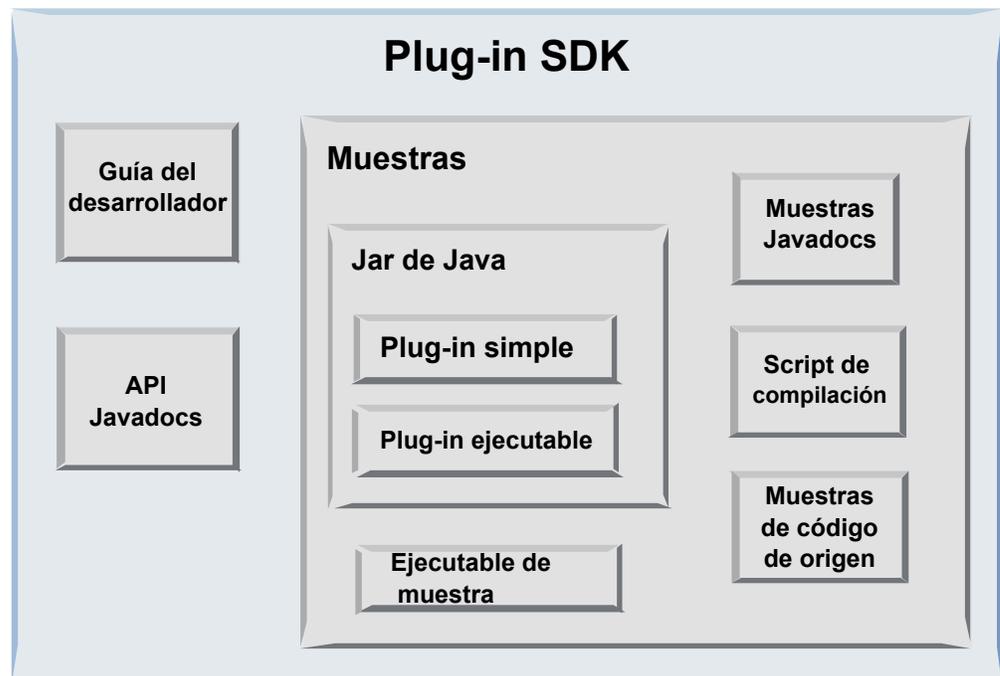
### Ayuda adicional de PDK de validación

Si necesita ayuda al utilizar el PDK (Plug-in Developer's Kit) de validación, póngase en contacto con el soporte técnico de IBM Unica para efectuar sus preguntas. Para obtener detalles, consulte “Contactar con el soporte técnico de IBM Unica ” en la página 13.

---

### Contenido del PDK de validación

El PDK de validación contiene todo lo que necesita para desarrollar ejecutables de la línea de comandos Java para añadir validación a IBM Unica Campaign.



## Componentes del PDK de validación

El PDK (Plug-in Developer's Kit) de validación tiene los siguientes componentes.

| Componente                | Descripción  |
|---------------------------|--|
| Guía PDK de validación    | Este documento.  |
| Ejemplos                  | Los ejemplos son ejemplos construibles y documentados de cómo utilizar el PDK.   |
| Archivo .jar Java         | Un archivo JAR de ejemplo que contiene los plug-ins de ejemplo. El jar contiene: <ul style="list-style-type: none"><li>• <b>Plug-in simple:</b> un ejemplo de una clase de validador autónomo.</li><li>• <b>Plug-in ejecutable:</b> un validador de ejemplo que ejecuta un ejecutable de la línea de comandos definido por el usuario para realizar la validación.</li></ul> |
| Ejecutable de ejemplo     | Un ejecutable de la línea de comandos que se puede utilizar con el plug-in ejecutable en UNIX.   |
| Javadocs de ejemplo       | Javadocs que describen las clases de validador de ejemplo.   |
| Script de creación        | Un script Ant que crea el código fuente incluido en plug-ins de validador utilizables.   |
| Código fuente de ejemplos | El código fuente Java para el validador simple y el validador ejecutable.  |

## Buscar material de referencia

El PDK de validación contiene información de referencia para la API de plug-in y el código de ejemplo. La información de referencia tiene formato de Javadocs. Para ver la documentación, abra el archivo siguiente:

```
C:\Inicio_Unica\Inicio_Campaign\devkits\validation\javadoc\index.html
```

Donde *Inicio\_Unica* es la ruta al directorio de instalación raíz de IBM Unica y *Inicio\_Campaign* es la ruta al directorio de instalación de Campaign.

Por ejemplo, la ruta a la documentación puede ser similar al ejemplo siguiente:

```
C:\IBM\Unica\Campaign\devkits\validation\javadoc\index.html
```

Puede ver los Javadocs en cualquier navegador web.

---

## Prestaciones del PDK de validación

Un plug-in creado con el PDK de validación puede ejecutar lógica de validación personalizada para campañas, ofertas o ambas cosas. Algunas de las posibles utilizaciones de la lógica de validación son:

- Comprobar los atributos ampliados (por ejemplo, válido versus opcional o campos dependientes relacionados)
- Proporcionar servicios que están fuera del ámbito de IBM Unica Plataforma de Marketing (por ejemplo, validar cuáles son los usuarios con permiso para editar y cuáles son los atributos ampliados).

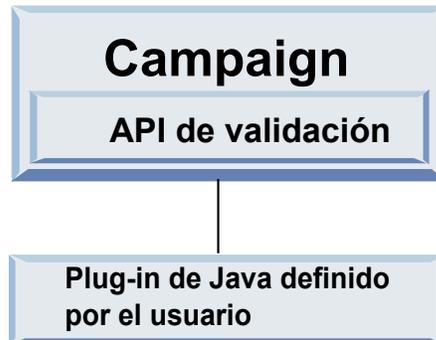
## Cargar el plug-in en IBM Unica Campaign

Existen dos formas de utilizar la API:

- Utilizarla para crear un plug-in de clase Java que se carga en la aplicación.
- Utilizar uno de los plug-ins incluidos para llamar a una aplicación ejecutable para manejar la validación.

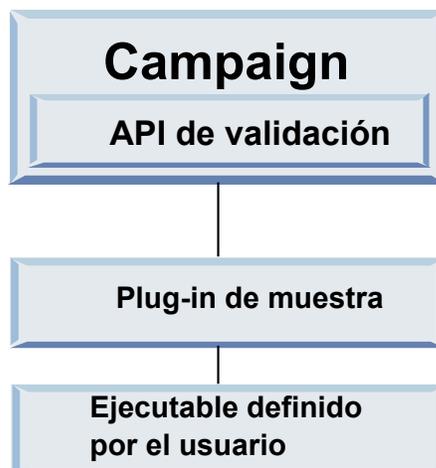
### Crear un plug-in de clase Java que se carga en la aplicación

El PDK de validación proporciona las interfaces, las clases de asistente y herramientas del desarrollador para desarrollar estas clases.



### Llamar a un aplicación para manejar la validación

El segundo modo de utilizar el PDK de validación consiste en utilizar uno de los plug-ins incluidos para llamar a una aplicación ejecutable para manejar la validación:



El ejecutable puede estar escrito en cualquier lenguaje, pero debe residir en el servidor de IBM Unica Campaign y ejecutarse en el servidor. El plug-in que llama al ejecutable envía un archivo XML que contiene la información que debe validarse; por ejemplo, el usuario que edita el objeto y los valores before/after para todos los atributos estándar y ampliados de dicho objeto). IBM Unica Campaign espera que la información de los resultados se devuelva con el formato de un archivo XML. Para obtener más información, consulte el apartado "Desarrollar un plug-in" en la página 6.

## Validación de oferta versus campaña

El PDK de validación puede validar ofertas y campañas. Si hay definido un plug-in de validación, lo llama automáticamente IBM Unica Campaign cada vez que se guarda un objeto de oferta o campaña. IBM Unica Campaign establece un indicador cuando llama al método de validación del plug-in. IBM Unica Campaign pasa los siguientes indicadores:

- `ValidationInputData.CAMPAIGN_VALIDATION`, cuando se añade o cambia una campaña
- o
- `ValidationInputData.OFFER_VALIDATION`, cuando se añade o edita una oferta.

A continuación, puede utilizar estos indicadores para construir reglas de validación aplicables a ofertas y campañas.

---

## Configuración del PDK de validación

El PDK de validación utiliza parámetros de configuración que indican a Campaign cómo encontrar la clase de plug-in que debe utilizarse y ofrece un modo de pasar información de configuración a estos plug-ins.

Todos los valores se encuentran en la página de configuración:

Campaign > particiones > partición[n] > validación

**Nota:** La validación funciona con varias particiones; `partición[n]` se puede cambiar por cualquier nombre de partición para proporcionar rutinas de validación también para estas particiones.

Esta sección contiene los siguientes valores:

- “`validationClass`”
- “`validationClasspath`” en la página 5
- “`validatorConfigString`” en la página 5

### validationClass

| Propiedad            | Descripción  |
|----------------------|--|
| Descripción          | Es el nombre de la clase que se utilizará para la validación. El valor de la propiedad <code>validationClasspath</code> indica dónde reside esta clase.  |
| Detalles             | La clase debe estar completamente calificada con su nombre de paquete. Si no se ha establecido correctamente, Campaign no realiza ninguna validación personalizada.  |
| Ejemplo              | <code>com.unica.campaign.core.validation.samples.SimpleCampaignValidator</code><br><br>Esto establece <code>validationClass</code> en la clase <code>SimpleCampaignValidator</code> del código de ejemplo. |
| Valor predeterminado | De forma predeterminada, no se establece ninguna ruta:<br><br><code>&lt;property name="validationClass" /&gt;</code>   |

## validationClasspath

| Propiedad            | Descripción   |
|----------------------|---|
| Descripción          | Es la ruta a la clase que se utiliza para la validación personalizada. Puede ser una ruta completa o una ruta relativa.   |
| Detalles             | <p>Si la ruta finaliza con una barra inclinada (una barra inclinada / para UNIX o una barra inclinada invertida \ para Windows), Campaign supone que es una ruta a un directorio que contiene la clase de plug-in Java que debe utilizarse.</p> <p>Si la ruta no finaliza con una barra inclinada, Campaign supone que es el nombre de un archivo .jar que contiene la clase Java. Por ejemplo, el valor</p> <pre>/opt/Unica/Campaign/devkits/validation/lib/validator.jar?</pre> <p>es la ruta en una plataforma UNIX que apuntaría al archivo JAR que está listo para utilizar con el kit del desarrollador de plug-in.</p> <p>Si la ruta es relativa, el comportamiento depende del servidor de aplicaciones que ejecuta Campaign. WebLogic utiliza la ruta al directorio de trabajo del dominio que, de forma predeterminada, es</p> <pre>c:\bea\user_projects\domains\mydomain</pre> <p>Si el valor no contiene una cadena, Campaign no intenta cargar un plug-in.</p> |
| Ejemplo              | <pre>/opt/Unica/Campaign/devkits/validation/lib/validator.jar?</pre> <p>Es la ruta en una plataforma UNIX que apunta al archivo JAR que viene empaquetado con el kit del desarrollador de plug-in.</p>  |
| Valor predeterminado | De forma predeterminada, no se establece ninguna ruta:<br><pre>&lt;property name="validationClasspath" /&gt;</pre>  |
| Consulte también     | Consulte "validationClass" en la página 4 para obtener información sobre cómo designar la clase que se utilizará.   |

## validatorConfigString

| Propiedad   | Descripción   |
|-------------|---|
| Descripción | Es una cadena que se pasa al plug-in de validador cuando Campaign lo carga. |

| Propiedad            | Descripción  |
|----------------------|--|
| Detalles             | <p>La forma en que el plug-in utiliza esta cadena depende del diseñador. Puede utilizarlo para enviar una cadena de configuración al plug-in cuando el sistema lo carga.</p> <p>Por ejemplo, ExecutableCampaignValidator (del plug-in ejecutable de ejemplo incluido con el PDK) utiliza esta propiedad para indicar el ejecutable que se ejecutará.</p> |
| Ejemplo              | <p>Para ejecutar el script de shell Bourne como el script de validación, establezca</p> <pre>validatorConfigString en  /opt/unica/campaign/devkits/validation/ src/com/unica/campaign /core/validation/samples/validate.sh</pre>   |
| Valor predeterminado | <p>De forma predeterminada, no se establece ninguna ruta:</p> <pre>&lt;property name="validatorConfigString" /&gt;</pre>   |

## Desarrollar un plug-in

### Visión general

Un plug-in es una clase Java que se carga durante el tiempo de inicio y se llama cada vez que se valida una campaña o una oferta. Esta validación se produce cada vez que un usuario guarda una campaña. Puede crear sus propios plug-ins Java utilizando las herramientas que proporciona el PDK de validación. Contiene código fuente para los plug-ins de ejemplo y un archivo Ant (Apache Ant es una herramienta de creación basada en Java) que se utiliza para compilar plug-ins. Las siguientes secciones le muestran cómo configurar el entorno para desarrollar un plug-in i, a continuación, le guía por la creación de su propio plug-in.

### Validadores de ejemplo

En la instalación estándar de Campaign se incluyen dos validadores de ejemplo.

- SimpleCampaignValidator es un plug-in autónomo que muestra cómo realizar acciones como la autorización personalizada y la validación de nombres de campaña permitidos. Puede encontrarlo en la ruta siguiente:

```
devkits\validation\src\com\unica\campaign\core\validation\
samples\SimpleCampaignValidator.Java
```

Es recomendable realizar una copia de la clase mientras se trabaja con ella, en lugar de editarla directamente.

- ExecutableCampaignValidator es un plug-in Java que llama a una aplicación ejecutable para realizar la validación. El código fuente para ExecutableCampaignValidator se incluye en el mismo directorio que SimpleCampaignValidator.

Sin embargo, la finalidad real de este ejemplo su utilización como un ejecutable de línea de comandos para la validación. Este archivo se encuentra en la ruta siguiente:

```
devkits/validation/src/com/unica/campaign/  
core/validation/samples/validate.sh
```

Este archivo es un ejecutable de bucle de retorno de ejemplo, que ilustra los tipos comunes de trabajo de validación.

## Arnés de prueba

La programación extrema y otras metodologías ágiles son útiles y populares. Un aspecto importante de estas metodologías es que son de prueba central. Los clientes que utilizan estas metodologías utilizan ampliamente la prueba de unidad. El PDK de validación da soporte a estas metodologías ofreciendo un arnés de prueba para ejecutar un plug-in fuera de Campaign. El hecho de poder probar el código sin colocarlo en IBM Unica Campaign acelera el proceso del desarrollador de plug-in.

Para utilizar el arnés de prueba:

1. Modifique el caso de prueba de unidad para reflejar la lógica de validación en el plug-in.
2. Ejecute el script de creación:
  - Para crear el plug-in sin realizar pruebas de unidad, ejecute los scripts de creación utilizando el comando "ant jar".
  - Para crear el plug-in y además realizar la prueba de unidad, ejecute los scripts de creación utilizando el comando "ant run-test".

## Scripts de creación

Los scripts de creación del PDK compilan todas las clases en un directorio y las colocan en un jar adecuado para utilizarlo en IBM Unica Campaign. El directorio que utiliza el script de creación proporcionado es:

```
devkits/validation/src/com/unica/campaign/core/validation/samples/
```

---

## Pasos principales para crear plug-ins

Estos son los pasos principales para crear plug-ins:

- "Configuración"
- "Crear los validadores" en la página 8
- "Configurar IBM Unica Campaign" en la página 8
- "Probar la configuración del validador" en la página 8
- "Crear un validador" en la página 9

Las siguientes secciones muestran cómo crear el origen para crear el archivo .jar.

## Configuración

El PDK de validación se puede instalar en cualquier máquina, pero los plug-ins que crea con él deben estar en la máquina en que se ejecuta IBM Unica Campaign. Se recomienda instalar el PDK en la máquina en la que prueba los plug-ins.

El PDK requiere que disponga de un kit de desarrollador Java de Sun y Jakarta Ant en la máquina plug-ins Java plug-ins. Se recomienda utilizar los paquetes de Ant y JDK que se proporcionan con el servidor de aplicaciones para garantizar la compatibilidad.

**Nota:** Puede utilizar otro JDK. Sin embargo, si utiliza el JDK que se proporciona con el servidor de aplicaciones, los plug-ins que crea pasan comprobaciones de

compatibilidad de tiempo de compilación. Por ejemplo, si utiliza WebLogic 8.1, que utiliza 1.4 JDK para la ejecución, cualquier clase específica de 1.5 JDK causaría errores de "clase no encontrada" cuando Campaign intente utilizar el plug-in. La utilización de 1.4 JDK para compilar impide la utilización de clases específicas de 1.5 JDK y garantiza la compatibilidad del plug-in con Campaign.

Para configurar el entorno para utilizar el PDK de validación:

1. Añada la carpeta que contiene el ejecutable Ant a la ruta. Por ejemplo, si WebLogic 8.1 está instalado en el directorio predeterminado en Windows, añada lo siguiente a la ruta:  
c:\bea\weblogic81\server\bin
2. Establezca la variable de entorno JAVA\_HOME en el directorio que contiene los directorios bin y lib del JDK. Por ejemplo, para WebLogic 8.1 en Windows, establezca JAVA\_HOME en:  
c:\bea\jdk141\_03

## Crear los validadores

El PDK proporciona un script Ant que puede crear todo el código en los archivos de ejemplo. El comportamiento predeterminado del script es crear un jar que contenga las clases de validación. Opcionalmente, también puede crear Javadocs y ejecutar pruebas en los validadores para asegurarse de que funcionan en Campaign antes de intentar utilizar el plug-in en producción.

Para crear el validador:

1. Cambie al directorio de PDK,  
<Inicio\_Unica\Inicio\_Campaign>\devkits\validation\build  
Verá el script Ant, build.xml, en este directorio.
2. Ejecute el jar Ant en la línea de comandos.  
Ant ejecuta el script y produce un archivo JAR denominado validator.jar en el directorio:  
<Inicio\_Unica\Inicio\_Campaign>\devkits\validation\build\lib

Ahora dispone de un validador personalizado que se puede utilizar en IBM Unica Campaign. En el apartado siguiente se explica cómo configurar Campaign para que utilice este validador.

## Configurar IBM Unica Campaign

Después de crear un plug-in de validador, debe indicar su ubicación a IBM Unica Campaign para poder utilizarlo.

Para utilizar SimpleCampaignValidator, establezca las propiedades que se describen en "Configuración del PDK de validación" en la página 4 del modo siguiente:

- validationClasspath: Unica\campaign\devkits\validation\lib\validator.jar
- validationClass:  
com.unica.campaign.core.validation.samples.SimpleCampaignValidator
- No es necesario establecer validatorConfigString para utilizar SimpleCampaignValidator porque no utiliza una cadena de configuración.

## Probar la configuración del validador

Después de crear el archivo validator.jar que contiene la clase SimpleCampaignValidator y realizar los cambios necesarios en la configuración, está

listo para probar y utilizar el plug-in. Este plug-in impide que los usuarios guarden una campaña denominada "badCampaign."

Para probar la configuración:

1. Vuelva a desplegar el servidor de aplicaciones para que los cambios entren en vigor. Para obtener detalles sobre cómo volver a desplegar el servidor de aplicaciones, consulte la documentación del servidor.
2. Inicie sesión en IBM Unica Campaign y vaya a la página de creación de campaña.
3. Cree una campaña con el nombre **badCampaign** e intente guardarla.

Si se ha configurado correctamente todo, no podrá guardar la campaña nueva. Debería recibir un mensaje de error del validador.

## Crear un validador

En esta sección, se crea un plug-in de validación que se parece bastante a SimpleCampaignValidator, pero se impide la creación de campañas denominadas "badCampaign2."

Para crear un validador:

1. Realice una copia del validador de ejemplo SimpleCampaignValidator.java, que se encuentra en  

```
<Inicio_Unica\Inicio_Campaign>\devkits\validation\src\com  
\unica\campaign\core\validation\samples
```
2. Denomine la copia MyCampaignValidator.java y déjela en el mismo directorio que el origen.
3. Abra MyCampaignValidator.java en un editor. Busque la palabra "badCampaign" en el documento y sustitúyala por la palabra "badCampaign2."
4. Guarde el archivo y cierre el editor.
5. Vuelva a crear los validadores. Para obtener más detalles, consulte "Crear los validadores" en la página 8.

**Nota:** Si el servidor de aplicaciones bloquea el archivo validate.jar mientras se utiliza, debe detener el servidor antes de crear los validadores.

6. Vuelva a configurar campaign\_config.xml para utilizar la nueva clase:  

```
<property name="validationClass"  
value="com.unica.campaign.core.validation.samples.MyCampaignValidator">
```
7. Pruebe el validador. Para obtener más detalles, consulte "Probar la configuración del validador" en la página 8.

No debería poder guardar las campañas denominadas "badCampaign2."

---

## Ejecutable de ejemplo

Los validadores de ejemplo también incluyen un validador, EjecutableCampaignValidator, que puede ejecutar un ejecutable desde la línea de comandos.

Esta sección:

- Muestra cómo configurar IBM Unica Campaign para ejecutar el plug-in ejecutable de ejemplo y

- Describe cómo crear los propios plug-ins ejecutables que son compatibles con el uso de la interfaz de uso de ejecutable.

## Configurar Campaign para el plug-in ejecutable de ejemplo

Para utilizar ExecutableCampaignValidator, establezca la propiedades que se describen en “Configuración del PDK de validación” en la página 4 del modo siguiente:

- validationClasspath:  
`<Campaigninicio_>\devkits\validation\lib\validator.jar`
- validationClass:  
`com.unica.campaign.core.validation.samples.ExecutableCampaignValidator`
- validatorConfigString:  
`<Campaigninicio_>\pdk\bin\validate.sh`

El script de ejemplo que se proporciona con el PDK es un script de shell Bourne para UNIX. Niega la creación de la campaña a cualquiera que tenga el nombre de usuario "badUser." Puede ver el código para el ejecutable en el directorio siguiente:

```
devkits\validation\src\com\unica\campaign\core\
validation\samples\validate.sh
```

Debe desarrollar su propio script que realice la validación pertinente para su implementación. Los lenguajes de script como PERL y Python son buenos candidatos para los scripts de proceso de texto como este; sin embargo, cualquier lenguaje que se pueda ejecutar desde la línea de comandos es aceptable.

## Interfaz de uso de ejecutable esperado

ExecutableCampaignValidator llama a un ejecutable con un línea de comandos que contiene los argumentos siguientes:

- nombre\_ejecutable: es la cadena establecida en validatorConfigString en la plataforma de IBM Unica Marketing.
- nombre\_archivo\_datos: es el nombre del archivo que el ejecutable lee como entrada. Los datos de entrada deben tener formato XML.
- nombre\_archivo\_resultados\_esperados: es el nombre del archivo que el ejecutable debe enviar como salida. Los resultados esperados deben tener el formato datos XXX.xml donde XXX es un número.
  - A continuación se proporciona un ejemplo de cómo se envían datos satisfactorios:  
`<ValidationResult result="0" generalFailureMessage="" />`
  - A continuación se proporciona un ejemplo de cómo se envían datos anómalos:  
`<ValidationResult result="1" generalFailureMessage="">`  
`<AttributeError attributeName="someAttribute" errorMessage="something" />`  
`<AttributeError attributeName="someAttribute2" errorMessage="something2" />`  
`</ValidationResult>`
  - El texto del archivo XML debe codificarse con caracteres ASCII normales o UTF-8.

**Nota:** Es muy recomendable proporcionar mensajes de error fáciles de comprender para que puedan corregir el problema antes de volver a intentar otra operación de guardar.

---

## Ejemplo

Esta sección contiene un ejemplo de un escenario de validación.

### Impedir ediciones de campaña

Si intenta impedir que alguien que edita una campaña pueda cambiar el código de campaña, puede utilizar una rutina de validación de campaña personalizada. La rutina garantizará que cuando se guarde la campaña se realice la siguiente comprobación:

```
código_campaña_nuevo == código_campaña_anterior
```

Para manejar el caso en que la campaña se crea por primera vez, pase al indicador de rutina indicando si la campaña que se valida es nueva (creación) o existente (edición). Si este indicador indica **edición**, realice la comparación de los códigos de campaña.

La aplicación Campaign establece este indicador en el objeto `InputValidationData` que a continuación pasa al plug-in. El plug-in lee el indicador cuando determina si la validación es para una campaña nueva o modificada.



---

## Contactar con el soporte técnico de IBM Unica

Si tiene algún problema que no puede resolver consultando la documentación, el contacto responsable del soporte técnico de la empresa puede registrar una llamada con el soporte técnico de IBM Unica . Utilice la información de esta sección para asegurarse de que el problema se ha resuelto de forma eficaz y satisfactoria.

Si no es un contacto responsable del soporte técnico en su empresa, póngase en contacto con el administrador de IBM Unica para obtener información.

### Información que debe recopilar

Antes de ponerse en contacto con el soporte técnico de IBM Unica , debe recopilar la información siguiente:

- Una breve descripción de la naturaleza del problema.
- Los mensajes de error detallados que aparecen cuando se produce el problema.
- Los pasos detallados para reproducir el problema.
- Los archivos de registro, archivos de sesión, archivos de configuración y archivos de datos relacionados.
- Información sobre el producto y el entorno del sistema, que se puede obtener tal como se describe en "Información del sistema."

### Información del sistema

Cuando llama al soporte técnico de IBM Unica , es posible que le soliciten que proporcione información sobre el entorno.

Si el problema no le impide iniciar la sesión, gran parte de esta información está disponible en la página Acerca de, que proporciona información sobre las aplicaciones de IBM Unica que hay instaladas.

Puede acceder a la página Acerca de seleccionando **Ayuda > Acerca de**. Si no se puede acceder a la página Acerca de, puede obtener el número de versión de cualquier aplicación de IBM Unica visualizando el archivo `version.txt` que se encuentra en el directorio de instalación de cada aplicación.

### Información para contactar con el soporte técnico de IBM Unica

Para obtener información sobre cómo contactar con el soporte técnico de IBM Unica , consulte el sitio web de soporte técnico del producto IBM Unica : (<http://www.unica.com/about/product-technical-support.htm>).



---

## Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en EE.UU.

Es posible que IBM no ofrezca los productos, servicios o funciones que se tratan en este documento en otros países. Consulte al representante local de IBM para obtener información sobre los productos y servicios disponibles actualmente en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende indicar ni implica que sólo se pueda utilizar este producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. No obstante, queda bajo la responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que aborden temas descritos en este documento. El suministro de este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar consultas acerca de licencias, por escrito, a la dirección siguiente:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
EE.UU.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe sus consultas, por escrito, a:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japón

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones sean incompatibles con la legislación vigente: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" sin garantías de ningún tipo NI EXPLÍCITAS NI IMPLÍCITAS, INCLUYENDO PERO NO LIMITÁNDOSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO. Algunas legislaciones no contemplan la declaración de limitación de responsabilidad, ni implícitas ni explícitas, en determinadas transacciones, por lo que cabe la posibilidad de que esta declaración no se aplique en su caso.

Esta información puede incluir imprecisiones técnicas o errores tipográficos. La información aquí contenida está sometida a cambios periódicos; tales cambios se irán incorporando en nuevas ediciones de la publicación. IBM puede realizar en cualquier momento mejoras o cambios en los productos o programas descritos en esta publicación sin previo aviso.

Todas las referencias hechas en este documento a sitios web que no son de IBM se proporcionan únicamente para su información y no representan en modo alguno una recomendación de dichos sitios web. Los materiales de estos sitios web no forman parte de los materiales destinados a este producto de IBM, y el usuario será responsable del uso que se haga de estos sitios web.

IBM puede utilizar o distribuir cualquier información que se le proporcione en la forma que considere adecuada, sin incurrir por ello en ninguna obligación para con el remitente.

Los propietarios de licencia de este programa que deseen tener información sobre el mismo con el fin de poder: (i) intercambiar información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) utilizar de forma mutua la información que se ha intercambiado, deberán ponerse en contacto con:

IBM Corporation  
170 Tracer Lane  
Waltham, MA 02451  
EE.UU.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluyendo en algunos casos el pago de una tarifa.

El programa bajo licencia que se describe en este documento y todo el material bajo licencia disponible los proporciona IBM bajo los términos de las Condiciones Generales de IBM, Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento contenidos en esta publicación se han determinado en un entorno controlado. Por tanto, los resultados obtenidos en otros entornos operativos pueden variar de forma significativa. Es posible que algunas mediciones se hayan realizado en sistemas en nivel de desarrollo y no existe ninguna garantía de que estas mediciones sean las mismas en sistemas disponibles generalmente. Además, algunas mediciones podrían haberse estimado mediante extrapolación. Los resultados reales podrían variar. Los usuarios de este documento debería verificar los datos aplicables para su entorno específico.

La información relativa a productos no IBM se ha obtenido de los distribuidores de dichos productos, de anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha comprobado estos productos y no puede confirmar la precisión de su rendimiento, compatibilidad ni contemplar ninguna otra reclamación relacionada con los productos que no son de IBM. Las preguntas relacionadas con las funciones de los productos que no son de IBM deberán dirigirse a los proveedores de estos productos.

Todas las declaraciones relativas a la dirección o intenciones futuras de IBM pueden cambiar o ser retiradas sin aviso, y representan sólo propósitos y objetivos.

Todos los precios IBM que se muestran son precios de venta al público sugeridos por IBM, son actuales y están sujetos a cambios sin previo aviso. Los precios de los distribuidores pueden variar.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen los nombres de personas, compañías, marcas y productos. Todos estos

nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por empresas comerciales reales es pura coincidencia.

#### LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente, que ilustra las técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier modo sin realizar ningún pago a IBM, con el fin de desarrollar, utilizar, comercializar o distribuir programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Dichos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas. Los programas de ejemplo se proporcionan "TAL CUAL", sin garantía de ningún tipo. IBM no será responsable de los daños que surjan por el uso de los programas de ejemplo.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

---

## Marcas registradas

IBM, el logotipo de IBM e [ibm.com](http://ibm.com) son marcas registradas o marcas comerciales registradas de International Business Machines Corp., registradas en muchas jurisdicciones de todo el mundo. Otros nombres de servicios y productos podrían ser marcas registradas de IBM o de otras empresas. Hay disponible una lista actual de marcas registradas de IBM en la web en "Información de marca registrada y copyright en "[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).







Impreso en España