

Versão 10 Release 0  
Junho de 2016

*Guia de API SOAP do IBM Campaign*

**IBM**

**Observação**

Antes de usar estas informações e o produto a que elas se referem, leia as informações em “Avisos” na página 63.

Esta edição se aplica à versão 10, liberação 0, modificação 0 do IBM Campaign e a todas as liberações e modificações subsequentes até que seja indicado de outra forma em novas edições.

© Copyright IBM Corporation 1998, 2017.

# Índice

## Capítulo 1. Visão geral da API SOAP do IBM Campaign . . . . . 1

Visão geral do design da API SOAP . . . . .	1
Mudanças de API SOAP por versão . . . . .	3
Implementações existentes usando o arquivo jar da API do cliente . . . . .	4
Implementações existentes usando o WSDL diretamente. . . . .	5
Referências da API SOAP . . . . .	7

## Capítulo 2. Usando a API SOAP do IBM Campaign . . . . . 9

Usando o arquivo .jar da API do cliente para chamar serviços do Campaign . . . . .	9
OfferAPI.java . . . . .	9
Usando o WSDL para chamar serviços do Campaign	11
Considerações de desempenho . . . . .	11

## Capítulo 3. Tipos de dados da API SOAP . . . . . 13

WSReference . . . . .	13
WSVersion. . . . .	13
WSServiceInfo . . . . .	13
WSAttributeTypeEnum . . . . .	14
WSAttributeStatusEnum . . . . .	14
WSAccessTypeEnum . . . . .	14
WSSelectTypeEnum. . . . .	14
WSRunStatusEnum . . . . .	14
WSRunTypeEnum . . . . .	15
WSAttribute . . . . .	15
WSAttributeMetadata . . . . .	16
WSCampaignInfo . . . . .	18
WSComponentOrFolderInfo . . . . .	18
WSTargetCellInfo . . . . .	18
WSMetricsInfo . . . . .	18
WSRunResults . . . . .	19
WSOfferInfo . . . . .	19
WSOfferCodeOrName. . . . .	19
WSOfferValidationInfo. . . . .	20
WSOfferTemplateInfo . . . . .	20
WSBulkOfferInfo . . . . .	20
WSOfferInfoStatus . . . . .	20

## Capítulo 4. Métodos de API SOAP . . . . . 21

Métodos de API SOAP: Serviço. . . . .	21
getServiceInfo . . . . .	21
Métodos de API SOAP: Atributos . . . . .	21
getAttributesByName . . . . .	21
updateAttributes . . . . .	22
getAttributeMetadataByName . . . . .	23
createAttributeMetadata . . . . .	24
updateAttributeMetadata . . . . .	25
deleteAttributeMetadata . . . . .	26
Métodos de API SOAP: Campanhas e fluxograma	27

generateCampaignCode . . . . .	28
deleteCampaigns . . . . .	28
createCampaign . . . . .	29
listCampaignsByPage . . . . .	30
stopFlowchart . . . . .	31
Métodos de API SOAP: Células de destino . . . . .	31
createTargetCell . . . . .	32
bulkCreateTargetCells . . . . .	33
listTargetCells . . . . .	34
bulkUpdateTargetCells. . . . .	35
getRunResultsByCell . . . . .	36
bulkDeleteTargetCells . . . . .	37
updateTemplateAttributes . . . . .	37
listBottomUpTargetCells . . . . .	38
Métodos de API SOAP: Analítica . . . . .	38
getCampaignMetrics . . . . .	39
Métodos de API SOAP: Ofertas, listas de ofertas, modelos de oferta . . . . .	39
listOffersAndFolders . . . . .	40
searchOffersBasic . . . . .	40
listOffersByPage . . . . .	41
createSmartOfferList . . . . .	42
createStaticOfferList . . . . .	43
getOffers . . . . .	44
validateOffers . . . . .	44
editOfferList . . . . .	45
createOffer. . . . .	46
retireOffers . . . . .	47
deleteOffers . . . . .	47
deleteOffersAndLists . . . . .	48
listOfferTemplates . . . . .	48
createTemplate . . . . .	49
getOfferTemplate . . . . .	50
retireOfferTemplates . . . . .	50
getOffersAndListsByPage. . . . .	51
bulkCreateOffers . . . . .	51
getOfferListDetails . . . . .	52
getOfferListMembers . . . . .	52
getOffersByQuery . . . . .	53
retireOfferLists . . . . .	53
createFolder . . . . .	54
editFolder . . . . .	55
getSubFoldersList . . . . .	55
moveFolders . . . . .	56
deleteFolders . . . . .	57

## Capítulo 5. Exceções comuns de API SOAP . . . . . 59

### Antes de contatar o suporte técnico do IBM . . . . . 61

### Avisos . . . . . 63

Marcas Comerciais . . . . . 65

Considerações sobre a Política de Privacidade e os  
Termos de Uso . . . . . 65

---

## Capítulo 1. Visão geral da API SOAP do IBM Campaign

A especificação da API SOAP do IBM® Campaign define a versão 3.0 da Interface de Programação de Aplicativos, também chamada de Serviços do Campaign. Essa especificação é entregue como parte do Kit de Ferramentas do Desenvolvedor do Software de Serviços Campaign (devkits) instalado com IBM Campaign.

O diretório <CAMPAIGN\_HOME>/devkits/CampaignServicesAPI que é fornecido pelo instalador inclui exemplos, construção e scripts de texto, Javadoc para classes públicas e interfaces, e notas sobre a liberação.

A API SOAP de Serviços do Campaign foi criada para:

- Fornece acesso de criação de baixa granularidade, descoberta, leitura e atualização para componentes IBM Campaign, enquanto isola os clientes dos detalhes de implementação subjacentes.
- Trabalhe em conjunto com a interface com o usuário IBM Campaign existente com mínimos efeitos.
- Garanta a validade dos dados.
- Atenda os serviços de segurança necessários do IBM Campaign.
- Dê suporte ao SOAP (Simple Object Access Protocol) padrão de mercado, incluindo autenticação segura.

---

### Visão geral do design da API SOAP

A API SOAP de Serviços do Campaign é uma fachada que proporciona uma visualização do cliente de uma instância do aplicativo do IBM Campaign em execução. Somente um subconjunto das capacidades de IBM Campaign são expostas, mas é suficiente conduzir aspectos chaves da funcionalidade Campaign.

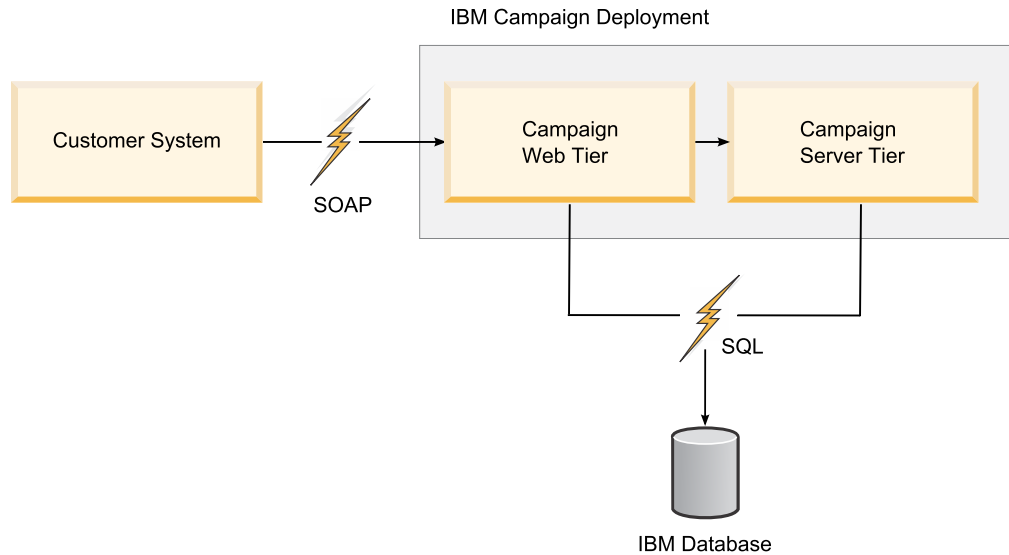
#### Recursos e diagrama

A API é usada simultaneamente com os usuários da web IBM Campaign e outros encadeamentos de API.

Geralmente, a API suporta os seguintes tipos de operações nas campanhas, ofertas e componentes de célula de destino.

- Criação do componente
- Diretório do componente
- Exclusão do componente
- Atributo do componente e criação, inspeção e modificação de metadados do atributo
- Busca de resultados de execução do fluxograma

O diagrama a seguir mostra uma implementação de amostra de CampaignServices 3.0.



## Autenticação do usuário

A autenticação trata do estabelecimento de uma identidade do usuário. A autenticação do usuário é a responsabilidade do aplicativo cliente.

## Autorização do usuário

A autorização trata das permissões que um usuário autenticado tem em relação aos componentes e operações que são expostos pela API.

É possível que um usuário seja autenticado com êxito, mas não tenha permissões suficientes para executar algumas operações, como editar as informações de resumo de uma campanha. Nesse caso, o método API emite `AuthorizationException`.

## Código de idioma

As solicitações de API fornecem um parâmetro **requestedLocale** opcional, que define o código de idioma a ser usado para realizar essa solicitação em específico. Se não definidos, a API será definida como padrão para o código de idioma preferencial do usuário IBM. O algoritmo usual de correspondência Java™ de melhor esforço é usado para retornar as mensagens e outro texto localizado no código de idioma solicitado.

Este parâmetro é da classe do tipo `java.util.Locale`.

**Nota:** Algum texto especificado pelo usuário, como descrições de campanha, fica no código de idioma do usuário que especificou o texto. O IBM Campaign não tenta localizar esses dados. Somente as mensagens informativas, de aviso e de erro são localizadas pela API.

## Gerenciamento de estado

A API CampaignServices não tem estado, o que significa que nenhuma informação por cliente é salva pela API nas chamadas.

Obviamente, as chamadas de API específicas podem alterar o estado de instâncias de componente subjacentes que são gerenciadas pelo Campaign e essas mudanças de estado podem ser persistidas para o banco de dados.

---

## Mudanças de API SOAP por versão

O propósito deste tópico é identificar mudanças na API SOAP do IBM Campaign para clientes que estão usando a API atualmente. Se você fez upgrade de uma versão anterior, revise as informações a seguir para determinar se precisa fazer mudanças em seu código de aplicativo.

### Versões e compatibilidade com anteriores

As futuras versões da API CampaignServices serão compatíveis com versões anteriores, com todas as liberações secundárias e de manutenção que compartilhem um número da versão principal. No entanto, IBM reserva para si o direito de interromper a compatibilidade com a versão anterior para liberações principais "dot zero" (x.0).

O número da versão principal da API será incrementada se alguma das modificações a seguir for efetuada:

- Interpretação de dados alterada
- Lógica de negócios alterada, ou seja, a funcionalidade do método de serviço
- Parâmetros de método e / ou tipos de retorno alterados

O número da versão secundária da API será incrementado, se alguma das modificações a seguir for feita.

- Novo método incluído
- O novo tipo de dados que é incluído e seu uso que é restrito para novos métodos
- Novo tipo incluído em um tipo enumerado
- uma nova versão de uma interface é definida

O IBM continua suportando o WSDL publicado, cliente SOAP e a versão de Apache Axis que é usada para implementar a oferta SOAP até, no mínimo, a próxima liberação principal do IBM . Falando de forma prática, isso é feito suportando diversos serviços da web específicos de versão simultaneamente. (O IBM já suporta diversas versões deste serviço internamente.)

### Mudanças na v9.1.2

Não houve mudanças de API na versão na v9.1.2.

**Nota:** Uma API REST foi introduzida na versão v9.1.2, além da API SOAP descrita neste guia. A API REST é descrita separadamente.

### Mudanças na v9.1.1

**stopFlowchart** inclui um novo parâmetro de entrada, **runId**, para suportar um ambiente de listener em cluster.

### Mudanças na v9.1

Não havia mudanças de API na versão 9.1.

## Mudanças na v9.0

Não havia mudanças de API na versão 9.0.

## Mudanças na v8.6

Mudanças na API implementadas na v8.6:

- O mecanismo SOAP foi atualizado do AXIS versão 1.4.1 para AXIS2 1.5.2.
- O WSDL foi reestruturado para lidar com problemas que tratam dos parâmetros obrigatórios e opcionais.
- O arquivo .jar de API do cliente alterado devido a mudanças WSDL. Como resultado, os stubs gerados e classes alteradas. Os parâmetros do método de API do cliente não foram alterados, mas os construtores dos objetos de valor de suporte foram modificados para o uso do conversor AXIS2 WSDL2Java.
- A URL de serviço da web aponta para:

`http://<host>:<port>/Campaign/services/CampaignServices30Service`

e o WSDL correspondente é recuperado em:

`http://<host>:<port>/Campaign/services/CampaignServices30Service?wsdl`

Se você estiver atualizando para IBM Campaign versão 8.6 ou superior e usar atualmente a API de Serviços Campaign, deve modificar seu nó de aplicativo.

Dependendo se você usa a API do cliente ou o WSDL, consulte as seções a seguir para obter detalhes:

- “Implementações existentes usando o arquivo jar da API do cliente”
- “Implementações existentes usando o WSDL diretamente” na página 5

## Implementações existentes usando o arquivo jar da API do cliente

Estas informações são pertinentes se você estiver fazendo upgrade para o IBM Campaign versão 8.6 ou superior e usar atualmente o arquivo .jar da API do cliente para interagir com o aplicativo da web do Campaign.

### Arquivo .jar de API do cliente

Seu aplicativo Java deve usar o arquivo .jar que está em:

`<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/  
CampaignServicesClient30.jar`

Para obter um exemplo Java que mostre a nova criação de oferta, consulte “OfferAPI.java” na página 9. O mesmo exemplo pode estar localizado na instalação da sua Campanha em:

`<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/samples/OfferAPI.java`

### Arquivos .jar dependentes

Como resultado do upgrade para AXIS2 versão 1.5.2, seu aplicativo Java também deve fazer upgrade usando os arquivos .jar de distribuição AXIS2 1.5.2, porque CampaignServicesClient30.jar depende desses arquivos .jar. Todos os arquivos



.jar dependentes devem ser incluídos no caminho de classe Java de seu aplicativo e podem estar localizados no arquivo Campaign.war que está em <CAMPAIGN\_HOME>/Campaign.war.

Extraia os arquivos .jar de Campaign.war, e inclua-os no caminho de classe Java.

## Construtor de API do cliente

Enquanto constrói o objeto de API do cliente, altere a URL de serviço da web e a assinatura de Exceção conforme mostrado neste exemplo.

```
try {
URL serviceURL = new URL(PROTOCOL, HOST, PORT,
"/Campaign/services/CampaignServices30Service");
CampaignServices30SoapClient client = new
CampaignServices30SoapClient(serviceURL, TIMEOUT);
} catch (RemoteException exception) {
exception.printStackTrace();
}
```

## Construtores parametrizados das classes de suporte

Com o mecanismo AXIS2, as classes geradas e os stubs não possuem construtores parametrizados. Em vez disso, essas classes possuem somente o construtor padrão sem argumento com setters e getters para os membros.

```
WSReference wsRef = new WSReference();
wsRef.setComponentTypeEnum(typeEnum);
wsRef.setId(id);
```

## Implementações existentes usando o WSDL diretamente

Estas informações são pertinentes se você estiver fazendo upgrade para o IBM Campaign versão 8.6 ou superior e você usar atualmente o WSDL para interagir com o aplicativo da web do Campaign. O WSDL do serviço da web do Campaign é usado para gerar status do lado do cliente e classes de suporte, usando qualquer ferramenta do conversor de terceiros. Os exemplos fornecidos aqui usam a ferramenta WSDL2Java a partir do Apache AXIS2 1.5.2.

### local do WSDL e URL de serviço

Um serviço da web Campaign para IBM Campaign é implementado em:

<http://host:port/Campaign/services/CampaignServices30Service>

O WSDL correspondente pode ser recuperado em:

<http://host:port/Campaign/services/CampaignServices30Service?wsdl>

### Gerando stubs e classes

A ferramenta WSDL2Java do Apache AXIS2 1.5.2 pode ser usada para gerar stubs e suportar classes Java do WSDL. Uma tarefa Ant de amostra é mostrada aqui.

A ferramenta também pode ser usada a partir da linha de comandos com conjunto de argumentos semelhantes. Os valores de argumento podem ser modificados para se adequarem ao seu ambiente.

**Nota:** A ligação ADB padrão é usada para o exemplo de conversor WSDL2Java a seguir:

```

<java classname="org.apache.axis2.wsd1.WSDL2Java" fork="true">
  <classpath refid="axis2.class.path"/> <!--Class path having
  AXIS2 libraries -->
  <arg value="-uri"/>
  <arg file="CampaignServices30.wsd1"/> <!--Actual location of
  WSDL -->
  <arg value="-s"/> <!-- Generate sync style code -->
  <arg value="-Euwc"/> <!-- Takes care of generating Wrapper
  java types for nillable = true elements. -->
  <arg value="-uw"/> <!-- Unwrap params -->
  <arg value="-u"/> <!-- Unpack classes -->
  <arg value="-ns2p"/> <!-- Namespace to package mapping. Customer
  can have their own package names. -->
  <arg value="http://webservices.unica.com/campaign/CampaignServices/
  3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
  <arg value="-o"/> <!-- Output directory -->
  <arg file="\${autogen.java.dir}"/>
</java>

```

## Usando stubs gerados e suportando classes

O stub pode ser usado da seguinte maneira:

```

CampaignServices30ServiceStub serviceStub = new
CampaignServices30ServiceStub(serviceURL);

serviceStub._getServiceClient().getOptions().setTimeOutInMilliseconds
(webServiceTimeout); //Timeout in milliseconds.

```

A oferta pode ser criada da seguinte maneira:

```

try{
  //Please change host and port to match your environment.
  String serviceURL = "http://host:port/Campaign/services/CampaignServices30Service";
  CampaignServices30ServiceStub serviceStub = new CampaignServices30ServiceStub(serviceURL);
  long webServiceTimeout = 2*60*1000; // 2 minutes
  serviceStub._getServiceClient().getOptions().setTimeOutInMilliseconds(webServiceTimeout); //Timeout in milliseconds.

  WSTextAttribute nameAttribute = new WSTextAttribute();
  nameAttribute.setMetadata(null);
  nameAttribute.setName("uacOfferDescription");
  nameAttribute.setValues(new String[]{"description " + System.currentTimeMillis()});

  WSTextAttribute[] wsAttributes = {nameAttribute};
  // convert to WSAttributeArrays
  WSAttributeArrays obj = new WSAttributeArrays();
  obj.setTextAttributes(wsAttributes);
  //Please change the values of following variables to match your environment.
  String authorizationLoginName = "asm_admin"; //login user name
  String partitionName = "partition1"; //Use your security policy of Campaign
  String securityPolicyName = "Global Policy"; //Use your security policy of Campaign

  String offerName = "1st Offer"; //Name of the offer to be created.
  String templateName = "Offer Template"; //Existing offer template name.
  long folderID = 100; //Actual ID of the folder where this offer will be created.
  //For folderID <=0, offer will be created at root level.

  CreateOffer createOfferObject = new CreateOffer();
  createOfferObject.setAuthorizationLoginName(authorizationLoginName);
  createOfferObject.setPartitionName(partitionName);
  createOfferObject.setRequestedLocale(Locale.US.toString());
  createOfferObject.setSecurityPolicyName(securityPolicyName);
  createOfferObject.setName(offerName);
  createOfferObject.setFolderID(folderID);
  createOfferObject.setTemplateName(templateName);
  createOfferObject.setAttributes(obj);
  // make campaign WebService call
  WSCreateOfferResponse wsResponse = serviceStub.createOffer(createOfferObject);
  // process status
  WSRequestStatus status = wsResponse.getStatus();
  // done
  WSOfferInfo offerInfo = wsResponse.getOfferInfo();
  System.out.println("status = "+status.getStatusType());
  System.out.println("offerInfo = "+offerInfo.getName());
}

```

```
} catch (Exception exception) {  
    //Handle the Exception here.  
    exception.printStackTrace();  
}
```

Neste exemplo, o `createOffer()` agora aceita somente um parâmetro do tipo `CreateOffer`.

Com o mecanismo `AXIS2`, as classes geradas e os stubs não possuem mais construtores parametrizados.

---

## Referências da API SOAP

As referências a seguir foram usadas para preparar a especificação da API SOAP do IBM Campaign.

- "Basic Profile Version 1.1", Organização de interoperabilidade de serviço da web (WS-I), 10 de abril de 2006. (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-0310.html>)
- "SOAP 1.2 (rascunho)", grupo de trabalho W3C Soap, 24 de junho de 2003 (<http://www.w3.org/TR/soap/>)
- "JAX-RPC 1.1", Sun Microsystems, 14 de outubro de 2003 (<http://java.sun.com/webservices/jaxrpc/index.jsp>)
- Grupo de trabalho de serviços da Web Apache (<http://ws.apache.org/axis2>)



---

## Capítulo 2. Usando a API SOAP do IBM Campaign

Para usar a API SOAP de Serviços da Web do IBM Campaign, é possível usar o arquivo .jar da API do cliente ou usar diretamente o WSDL. Um exemplo mostra como usar o método do arquivo .jar para criar uma oferta.

---

### Usando o arquivo .jar da API do cliente para chamar serviços do Campaign

O IBM Campaign fornece uma API de cliente que usa os serviços da web SOAP para interagir com o aplicativo da web Campaign. Esse wrapper é agrupado em um arquivo .jar que o aplicativo cliente pode usar para chamar a API Campaign.

O arquivo .jar pode ser localizado em:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/  
CampaignServicesClient30.jar
```

O exemplo a seguir mostra a criação de nova oferta no nível de pasta raiz da Oferta em Campaign. A mesma amostra pode ser localizada em:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/samples/OfferAPI.java
```

**Nota:** O exemplo usa alguns valores simulados para os parâmetros; seus valores reais podem ser diferentes.

Além disso, a URL para os serviços da web do Campaign é `http://host:port/Campaign/services/CampaignServices30Service`, em que o host e a porta se referem ao nome do host e número da porta da máquina em que o aplicativo da web Campaign é implementado.

Se você usa uma amostra fornecida, certifique-se de modificá-lo para ajustar seu ambiente de cliente.

#### OfferAPI.java

Para compilar e executar esta amostra Java, deve-se incluir todos os arquivos .jar dependentes no caminho de classe Java. O arquivo CampaignServicesClient30.jar depende dos arquivos .jar de mecanismo Apache AXIS2 SOAP e outros arquivos .jar Apache comuns, que podem ser localizados em <CAMPAIGN\_HOME>/Campaign.war. Extraia os arquivos .jar de Campaign.war e inclua-os no caminho de classe Java.

```
import java.net.URL;  
import java.util.Locale;  
import com.unica.publicapi.campaign.campaignservices.CampaignServicesException;  
import com.unica.publicapi.campaign.campaignservices.attribute.metadata.  
    IAttribute Metadata;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.  
    CampaignServices30SoapClient;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSAttribute;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSOfferInfo;  
import com.unica.publicapi.campaign.campaignservices.utils.WSAttributeUtils;
```

```
/**
```

```
 * Essa é a classe de cliente Java de amostra que mostra o uso da API
```

```

de serviços SOAP Campaign.
* Esta amostra usa a fachada CampaignServices30SoapClient para interagir com o
serviço da web Campaign.
* Aqui a criação da Oferta é mostrada. Consulte o guia da API para obter
mais detalhes.
*
* @author AGijare
*
*/
public class OfferAPI {

    /**
     * @param args
     */
    protected static CampaignServices30SoapClient CLIENT = null;

    private static void setup(){
        try {
            String protocol = "http"; //http or https
            String host = "localhost"; //Host name of deployed Campaign.
            Use os nomes próprios do host aqui.
            int port = 7001; //port number of deployed Campaign
            long timeOut = 2*60*1000; // 2 minutes
            String servicesURI = "/Campaign/services/CampaignServices30Service";
            CLIENT = new CampaignServices30SoapClient(
                new URL(protocol, host, port, servicesURI),
                timeOut);
        } catch (Exception exception) {
            exception.printStackTrace();
            System.exit(-1);
        }
    }

    public static void main(String[] args) {
        //Alterar os valores das seguintes variáveis para corresponder seu
        ambiente.
        String userName = "user_name"; //login user name
        String partitionName = "partition1"; //Usar nome próprio da partição de
        Campaign
        Locale loc = Locale.US;
        String securityPolicy = "Global"; //Usar sua política de segurança da
        Campaign

        String offerName = "Offer1";
        String offerTemplate = "Offer Template"; // Modelo do qual
        a Oferta será criada.
        long folderID = 1002; //ID real da pasta em que esta oferta
        será criada. Para folderID <=0, a oferta será criada no nível raiz.
        //Atributos de oferta
        WSAttribute[] wsAttributes = {
            WSAttributeUtils.getWSTextAttribute(IAAttributeMeta
            data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"description "
            + System.currentTimeMillis()})
        };

        setup();

        try {
            WSOfferInfo wsOfferInfo = CLIENT.createOffer(userName,
            partitionName, loc, securityPolicy,
            offerName, folderID, offerTemplate, wsAttributes);
            System.out.println("Created offer: " + wsOfferInfo.getName());
        } catch (CampaignServicesException e) {
            e.printStackTrace();
        }
    }
}

```

---

## Usando o WSDL para chamar serviços do Campaign

Os serviços do Campaign podem ser chamados usando o arquivo WSDL de serviços da Web Campaign: CampaignServices30.wsdl.

O arquivo CampaignServices30.wsdl pode estar localizado em:

`http://host:port/Campaign/services/CampaignServices30Service?wsdl`

ou na distribuição Campaign em:

`<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/`

O aplicativo Java do cliente deve usar as classes e os stubs que são gerados de WSDL usando qualquer ferramenta do conversor WSDL-to-Java de terceiros. O IBM recomenda o uso de Apache AXIS.

Os javadocs que são criados a partir de stubs e classes que são gerados a partir de WSDL usando o Apache AXIS2 podem ser localizados em:

`<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/javadocs/index.html`

**Nota:** Todos os arquivos .jar dependentes devem ser incluídos no caminho de classe Java. O arquivo CampaignServicesClient30.jar depende dos arquivos .jar do mecanismo Apache AXIS2 SOAP e outros arquivos .jar Apache comuns, que podem ser localizados no arquivo Campaign.war que está em <CAMPAIGN\_HOME>/Campaign.war. Extraia os arquivos .jar de Campaign.war e inclua-os no caminho de classe Java.

---

## Considerações de desempenho

O atual perfil de desempenho de implementação de API CampaignServices é semelhante ao do aplicativo, conforme experimentado através da GUI. Algumas APIs são projetadas explicitamente para desempenho. Em específico, a API `listCampaignsByPage()` permite a paginação relativamente eficiente.

A interface SOAP, por sua natureza, introduz a latência e sobrecarga porque todos os dados são convertidos no formulário XML, que em alguns casos é muito detalhado. Por exemplo, uma chamada SOAP de loopback simples pode tomar 100 ms em uma rede típica (Java 1.4.x era ainda mais lenta). A API é otimizada para o portal tópico e outros casos de uso de negócios do aplicativo cliente, como see `listOffersByPage()`, de modo que o desempenho SOAP deve ser adequado.

No entanto, o cliente deve tomar cuidado porque isso não coloca uma carga muito alta sobre CampaignServices normais que estejam atendendo as solicitações do usuário da Web. Em geral, espera-se que as necessidades de processamento do usuário de uma API não excedam aquelas de um usuário da web IBM Campaign típico.





---

## Capítulo 3. Tipos de dados da API SOAP

A API SOAP de Serviços do IBM Campaign usa os tipos de dados públicos a seguir.

---

### WSReference

Um wrapper simples ao redor de um ID do banco de dados:

- **componentTypeEnum**: um tipo enumerado que indica o tipo de componente para o qual o id se destina. Um dos seguintes:
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS\_CELL
  - OFFER
  - OFFER\_LIST
  - OFFER\_TEMPLATE
- *id*: um *Long*, definindo um identificador específico de banco de dados numérico exclusivo para a referência.

---

### WSVersion

Um tipo de wrapper que captura diversos componentes de uma versão, incluindo o seguinte:

- *major*: um número inteiro que defina o número da versão principal, como '8' da versão completa 8.1.2.3.
- *minor*: um Número inteiro que defina o número de versão secundário, como '1' da versão completa 8.1.2.3.
- *maintenance*: Número inteiro opcional que define o número de manutenção da versão, se aplicável, como '2' da versão completa 8.1.2.3. Nunca fornecido com uma versão de API.
- *patch*: Número inteiro opcional que define o número de liberação de correção, se aplicável, como '3' da versão completa 8.1.2.3. Nunca fornecido com uma versão de API.

---

### WSServiceInfo

Um tipo de wrapper simples em torno das informações sobre o serviço. Contém os campos a seguir:

- *apiVersion*: uma instância *WSVersion*, que define a versão mais atual da API suportada pelo serviço. (*apiVersion* inclui somente as informações de versão principais ou secundárias.)
- *campaignVersion*: uma instância *WSVersion*, que define a versão completa da instância IBM Campaign subjacente.
- *name*: o nome interno do serviço, como "CampaignServices30Service".

---

## WSAttributeTypeEnum

Um tipo enumerado que define todos os tipos de atributo possíveis, um entre:

- STANDARD: atributo padrão ou base que é definido por Campaign.
- CUSTOM: um atributo que é definido por outro aplicativo IBM , o cliente ou outro terceiro.
- INPUT\_PARAMETER: um parâmetro de entrada, como um atributo usado para executar um fluxograma IBM Campaign.
- OUTPUT\_PARAMETER: um parâmetro de saída, como um atributo cujo valor é completo como resultado de um fluxograma que é executado no IBM Campaign.

---

## WSAttributeStatusEnum

Uma enumeração de todos os códigos de status de atributo possíveis, um entre:

- ACTIVE: o atributo está ativo e pode ser usado à vontade.
- RETIRED: o atributo é removido do serviço e não deve ser usado.

---

## WSAccessTypeEnum

Um tipo enumerado que define todos os tipos de acesso de valor de atributo possíveis, um entre:

- READ\_ONLY: o valor de atributo pode ser lido e exibido, mas não modificado.
- READ\_WRITE: o valor de atributo pode ser lido, exibido e modificado.

O acesso ao atributo é acumulativo para permissões de segurança. Por exemplo, se a política de segurança para um usuário cliente negar acesso de leitura para um atributo específico, o acesso de atributo não poderá substituir essa configuração de segurança. Na realidade, a API nunca retornaria o atributo ao cliente.

---

## WSSelectTypeEnum

Define todos os tipos de seleção possíveis para um determinado valor de atributo, um entre:

- NONE: sem seleção (*hasOptions* é falso).
- SINGLE\_SELECT: somente uma opção de atributo da lista de possíveis opções pode ser escolhida por vez (válido somente se um atributo *hasOptions*).
- MULTIPLE\_SELECT: semelhante a SINGLE\_SELECT, exceto que uma ou mais opções possam se selecionadas por vez.

---

## WSRunStatusEnum

Um tipo enumerado de todo fluxograma, ramificação ou status de execução de célula possível, um entre:

- NOT\_STARTED: a execução é planejada, mas ainda não foi iniciada.
- RUNNING: execução em andamento.
- CANCELLED: a execução foi cancelada, por um usuário Campaign ou esta API.
- SUCCEEDED: a execução foi concluída com êxito.
- FAILED: a execução falhou; os detalhes de erro são relatados separadamente. (consulte “WSRunResults” na página 19.)

---

## WSRunTypeEnum

UM tipo enumerado de todos os tipos de execução possíveis, um de:

- NOT\_RUN
- TEST\_RUN
- PRODUCTION\_RUN
- RUN\_SKIPPED
- TEST\_FLOWCHART
- PRODUCTION\_FLOWCHART
- TEST\_BRANCH
- PRODUCTION\_BRANCH
- TEST\_PROCESS
- PRODUCTION\_PROCESS

---

## WSAttribute

Os atributos fornecem um mecanismo extensível simples para conectar os dados arbitrários às instâncias de componente acessíveis através da API, seja de dados padrão como um *nome* de campanha, parâmetros de entrada de execução de fluxograma como *gênero* ou dados customizados arbitrários que são especificados por outro aplicativo IBM ou cliente IBM .

**Nota:** Nesta API, os atributos são usados para modelar a maioria dos dados de componentes, não apenas os atributos customizados do Campaign.

Os componentes geralmente possuem vários atributos que são associados a eles, que são expostos pela API CampaignServices como um Mapa especialmente digitado que é chamado de *AttributeMap*. Os dados do atributo são representados como uma classe concreta digitada em toda a API, como *WSDecimalAttribute*, para atributos que contenham dados decimais (dupla precisão numérica).

Cada atributo inclui o seguinte:

- *Name*: o nome exclusivo do atributo. Este nome serve como a chave para acessar o atributo e seus metadados na instância do componente que ocorre. O formato do nome é indefinido; em alguns casos, é designado pelo serviço, pelo cliente ou por um usuário IBM Campaign.

Geralmente, este nome não é o nome de exibição que seria apresentado a um Campaign ou usuário cliente. Ele pode ser padronizado pela API, como *uacDescription*, pode ser designado pelo IBM Campaign ao publicar os fluxogramas ou pode ser designado pelo aplicativo IBM ou cliente ao definir os atributos customizados. No entanto, em todos os casos, o nome é exclusivo garantido.

- *Metadados*: (opcional) as informações sobre os dados do atributo, como tipo de dados de valor, nome de exibição, descrição, prompts, valor padrão, tipo de seleção, comprimento (texto), precisão (decimais), opções (se seleção única ou múltipla). Consulte o “WSAttributeMetadata” na página 16.
- *Values*: uma matriz de zero ou mais objetos de valor digitados. O campo de valores é fornecido pela classe de atributo concreto; o tipo de cada valor deve ser o mesmo e concordar com a definição de tipo no campo de metadados do atributo. No entanto, nem todos os atributos suportam múltiplos valores.

Os seguintes tipos de atributo concretos são suportados:

- **WSBooleanAttribute**: um atributo cujo valor é um booleano, ou seja, *true* ou *false*.
- **WSIntegerAttribute**: um valor de número inteiro (*java.lang.Long*).
- **WSDecimalAttribute**: um valor numérico decimal de dupla precisão (*java.lang.Double*).
- **WSCurrencyAttribute**: um valor de moeda composto, incluindo um código de moeda ISO 4217 opcional do valor de moeda, como "USD" para dólar dos EUA e valores de moeda capturados como *Double*. Se o código de moeda não for fornecido, o padrão usado por IBM Campaign será assumido.  
Consulte <http://www.xe.com/symbols.php> para obter uma lista de países, símbolos de moeda e códigos. O código de idioma que é usado para um valor de moeda pode ser diferente do código de idioma preferencial do usuário.
- **WSCalendarAttribute**: cujos valores são as datas do calendário ou data/hora, em algum fuso horário e código de idioma.
- **WSTextAttribute**: uma sequência de texto Unicode (possivelmente nulo ou vazio).

**Nota:** A lista de possíveis atributos geralmente é diferente para cada tipo de componente, mas as listas podem se sobrepor.

---

## WSAttributeMetadata

WSAttributeMetadata define informações sobre os dados de um determinado atributo digitado, como o tipo de dados do valor, texto localizado (nome de exibição, descrição, prompts), seu valor padrão, intervalo de valor permitido, tipo de seleção, opções (se seleção única ou múltipla). Assim como com os atributos, os metadados do atributo são digitados. Por exemplo, um WSDecimalAttribute *myNumber* deve ter um WS *DecimalAttributeMetadata* vinculado, e todos os valores, incluindo os valores de atributo, valor padrão de metadados e possíveis valores de opção, todos devem ser digitados *Dobrados*

Descrições, rótulos e outros textos de metadados de atributo são localizados; no entanto, o texto especificado pelo usuário pode estar disponível conforme foi digitado pelo usuário. Cada chamada de API inclui um código de idioma que o código de cliente pode usar para definir o código de idioma em que um determinado usuário gostaria que as mensagens localizadas fossem exibidas. As políticas normais de fallback de código de idioma Java são usadas para preencher a solicitação.

WSAttributeMetadata inclui os campos a seguir:

- *name*: O nome do atributo, padrão ou customizado; também o nome que é usado pelo atributo que se liga a esses metadados. Os atributos padrão são definidos pelo sistema e possuem nomes padrão em um espaço de nome reservado (ou seja, usam um prefixo "uac"), os nomes customizados podem usar qualquer outra convenção de nomenclatura.

**Nota:** O nome de atributo deve ser exclusivo, nunca é localizado e possui restrições de comprimento (que dependem do conteúdo do caractere e do banco de dados). O nome não faz distinção entre maiúsculas e minúsculas e pode ser composto por qualquer combinação de caracteres de dígito e letra Unicode, mais o caractere de sublinhado '\_', mas não pode iniciar com um dígito.

- *description*: descrição opcional do atributo. Adequado para uma dica de ferramenta ou outra apresentação da interface com o usuário.
- *Predicados*: predicados sortidos que descrevem o atributo:

- *isRequired*: verdadeiro, se o atributo for obrigatório.
- *isInternal*: verdadeiro, se o atributo for definido pelo sistema e for apenas para uso interno (não deve ser apresentado a um usuário).
- *isGenerated*: verdadeiro, se o valor ou os valores do atributo forem gerados automaticamente por IBM Campaign quando o componente for criado, como um código da célula de destino. Geralmente, o *accessTypeEnum* será *READ\_ONLY* para os valores gerados.
- *hasOptions*: verdadeiro, se o atributo tiver opções. Implica que as opções são definidas para esse metadados e que o *selectTypeEnum* é *SINGLE\_SELECT* ou *MULTIPLE\_SELECT*.
- *typeEnum*: um *WSAttributeTypeEnum* que define o tipo de atributo, como *STANDARD* ou *CUSTOM*.
- *statusEnum*: um *WSAttributeStatusEnum* que define o status do atributo, como *ACTIVE*.
- *accessTypeEnum*: um *WSAccessTypeEnum* que define o tipo de acesso para o valor de atributo, como *READ\_ONLY*.
- *selectTypeEnum*: um *WSAccessTypeEnum* que define o tipo de seleção que é usado para o atributo, como *SINGLE*. Deve ser *NONE* para os atributos de campanha e de célula, ou se nenhuma opção for fornecida.
- *componentTypeEnum*: um *WSComponentTypeEnum* de todos os componentes possíveis da Campanha que são expostos pela API, como *CAMPAIGN*, *FOLDER*.
- *defaultValue* (somente fluxograma): valor padrão de tipo opcional para o atributo. Este valor é fornecido pela classe de metadados do atributo concreto, como um valor padrão *WSTextAttributeMetadata's* do tipo *String*. (Consulte a descrição dos valores de Atributo). Para componentes que não sejam o fluxograma, o valor padrão é indefinido.
- *options*: lista opcional de opções para este atributo. Resumindo, as opções de um atributo definem o conjunto exato de valores permitidos para esse atributo; cada opção é digitada de modo que, por exemplo, um *WSTextAttributeMetadata* possa ter somente um *WSTextAttributeOption* vinculado.

**Nota:** Existe uma restrição nas opções; somente os atributos de texto são suportados.

Cada opção define o seguinte:

- *prompt*: prompt para a opção adequada para menus suspensos, como "Male" como uma opção de atributo de gênero. Diferente do prompt de metadados, os nomes de exibição de opção geralmente não incluem pontuação.
- *description*: descrição localizada da opção, como "Uma pessoa de persuasão masculina". Adequado para o texto de dica da ferramenta.
- *isDefault*: verdadeiro, se esta opção específica for o padrão. Para os tipos de seleção *MULTIPLE\_SELECT*, mas de uma opção pode ser marcada como um padrão.
- *value*: o valor de opção digitado. Como o *defaultValue* de metadados de atributo, esse valor é fornecido pela subclasse de opção concreta, como um valor *WSDecimalAttributeOption's* do tipo *Decimal*. (Consulte a descrição dos valores de Atributo). Continuando o exemplo *gender* acima, o valor poderia ser declarado como uma sequência "m" (*WSTextAttributeOption*) ou como um código numérico, 123 (*WSDecimalAttributeOption*).

---

## WSCampaignInfo

Um tipo de wrapper simples ao redor dos dados de atributo de campanha.

Contém os campos a seguir:

- *reference*: a referência de campanha.
- *name*: nome da campanha (*uacName*); exclusivo não garantido.
- *description*: descrição opcional da campanha (*uacDescription*).
- *campaignCode*: o código exclusivo da campanha (*uacCampaignCode*); designada pelo cliente ou Campaign.

---

## WSComponentOrFolderInfo

Contém uma combinação de campanha agrupada ou dados de atributo da pasta, como o nome de exibição, sua referência.

Contém os campos a seguir:

- *reference*: o componente ou a referência de pasta.
- *name*: nome do componente ou da pasta (*uacName*); exclusivo não garantido.
- *description*: descrição da pasta ou componente opcional (*uacDescription*).
- *componentCode*: código exclusivo para o componente ou nulo, se for uma pasta.

---

## WSTargetCellInfo

Um wrapper simples ao redor dos dados de atributo da linha da célula de destino.

Contém os campos a seguir:

- *reference*: a referência de célula.
- *name*: nome da célula (*uacName*); exclusivo não garantido.
- *description*: descrição da célula opcional (*uacDescription*).
- *cellCode*: o código de célula (*uacCellCode*); designado pelo cliente ou Campaign. Os códigos de célula podem ser forçados exclusivos, definindo o parâmetro de configuração IBM Campaign *DuplicateCellCodesAllowed* para falso.
- *flowchartName*: nome opcional do fluxograma ao qual a célula é ligada.

---

## WSMetricsInfo

Um tipo de wrapper simples em torno dos dados de analítica de campanha, incluído número de contatos. Contém os campos a seguir:

- *totalContacts*: uma lista longa que fornece o número total de contatos.
- *responses*: uma lista digitada de instâncias *WSMetricsResponse*, cada instância que define as informações de contato para uma resposta:
  - *typeCode*: uma sequência que define o código de tipo de resposta, como *PHC* para um contato de chamada telefônica.
  - *count*: uma lista longa que fornece o número de vezes em que esse contato aconteceu.

---

## WSRunResults

Um tipo de wrapper em torno dos resultados de um fluxograma, caixa de processo ou execução da célula, possivelmente ainda em progresso, incluindo o status de execução, início da execução do fluxograma e datas/horas de término e contagens.

Inclui os campos a seguir:

- *sourceReference*: referência opcional da origem do resultado de execução. De acordo com o contexto no qual os resultados de execução são buscados, isso pode se referir a um fluxograma, uma caixa de processo de fluxograma ou uma célula de destino. Em qualquer caso, os dados do resultado de execução restantes se referem a essa origem.
- *flowchartName*: o nome do fluxograma que foi executado.
- *flowchartId*: o ID do banco de dados para o fluxograma.
- *runId*: o ID do banco de dados da execução.
- *typeEnum*: um tipo enumerado que define qual execução gerou os resultados, como `PRODUCTION_PROCESS` (consulte *WSRunTypeEnum*).
- *statusEnum*: um tipo enumerado que define o status da execução, como `RUNNING` (consulte *WSRunStatusEnum*).
- *statusCode*: código opcional de status do número inteiro.
- *statusMessage*: mensagem de status opcional.
- *startDate*: data e hora de calendário opcional de quando a execução foi iniciada; é nulo se a execução não foi iniciada.
- *endDate*: semelhante a *startDate*, mas a data e hora em que a execução foi encerrada (êxito ou falha); é nulo se a execução não foi iniciada ou ainda não estiver finalizada.
- *count*: contagem total opcional de contatos que foram selecionados pela execução; pode ser zero ou nulo se a execução não foi concluída.

---

## WSOfferInfo

Um tipo de wrapper simples em torno da oferta ou dados de atributo da lista de ofertas.

Contém os campos a seguir:

- *reference*: a referência da oferta ou da lista de ofertas.
- *name*: nome da oferta ou da lista de ofertas (*uacName*); exclusivo não garantido
- *description*: descrição opcional (*uacDescription*).
- *offerCode*: o código de oferta (*uacOfferCode*), se uma oferta ou nulo, se uma lista de ofertas. (Não garantido para ser exclusivo.)

---

## WSOfferCodeOrName

Um tipo de wrapper simples em torno dos códigos de oferta ou dados de nomes da lista de ofertas.

Contém os campos a seguir:

- *isCode*: booleano que indica se o campo *codeOrName* é um código de oferta presumido (`true`) ou o nome de uma lista de ofertas (`false`).
- *codeOrName*: o código de oferta exclusivo (*uacOfferCode*) de uma oferta ou o nome da lista de ofertas.

---

## WSOfferValidationInfo

Um tipo de wrapper simples em torno das informações de validação da oferta.

Contém os campos a seguir:

- *errorCode*: se não nulo, define o código de erro de validação alfanumérica. Consulte a classe *IStandardDefinitions* para os códigos de erro.
- *errorMessage*: mensagem opcional localizada que descreve o erro (se tiver ocorrido um).
- *codeOrName*: o código de oferta validado ou nome da lista de ofertas.
- *reference*: a referência da oferta ou da lista de ofertas, se válida.

---

## WSOfferTemplateInfo

Um tipo de wrapper simples em torno dos dados de modelo de oferta.

Contém os campos a seguir:

- *reference*: a referência do modelo de oferta.
- *name*: nome do modelo de oferta; exclusivo garantido.
- *description*: descrição opcional (*uacDescription*).
- *offerTemplateID*: ID do banco de dados do modelo de oferta exclusivo.

---

## WSBulkOfferInfo

Usado para criar ofertas em massa.

Contém os campos a seguir:

- *offerName*: o nome da oferta sendo criada.
- *attributes*: uma matriz dos tipos *WSAttribute* que indica os atributos de oferta.

---

## WSOfferInfoStatus

Um tipo de retorno para o método de API *bulkCreateOffers()* que indica o status de criação de oferta em massa.

Contém os campos a seguir:

- *name*: o nome da oferta.
- *code*: o código de oferta. É nulo se a criação de oferta falhar.
- *description*: descrição da oferta.
- *reference*: *WSReference* da oferta criada. É nulo se a criação de oferta falhar.
- *status*: uma instância de *WSRequestStatus* que indica o status da criação de oferta.



---

## Capítulo 4. Métodos de API SOAP

A API SOAP de Serviços do IBM Campaign usa os métodos a seguir.

---

### Métodos de API SOAP: Serviço

A API SOAP do IBM Campaign oferece uma maneira de obter informações de identificação sobre o próprio serviço.

#### getServiceInfo

```
WSServiceInfo getServiceInfo()  
    throws CampaignServicesException;
```

Retorna informações sobre o serviço, como a versão de API mais atual que suporta, a versão completa da instância IBM Campaign subjacente.

**Nota:** Nenhuma informação de cliente é necessária para esta chamada e nenhuma permissão de segurança é aplicada.

#### Parâmetros

Nenhum.

#### Devoluções

Retorna uma instância *WSServiceInfo*.

#### Erros

Nenhum.

---

### Métodos de API SOAP: Atributos

A maioria dos dados de instância do componente pode ser exposta pela API SOAP do IBM Campaign como atributos ou metadados do atributo.

Em alguns casos, as definições de metadados de atributo são globais para IBM Campaign (como atributos customizados de campanha). Em outros casos, elas são restritas a um determinado componente, como variáveis de usuário do fluxograma. A menos que seja indicado de maneira diferente, todos os atributos podem ser lidos se o cliente tiver permissões de segurança suficientes para fazê-lo.

**Nota:** Somente componentes que estão ativos e aos quais o cliente tem acesso são expostos por essa API. O suporte público é limitado a um subconjunto das APIs disponíveis.

#### getAttributesByName

```
Map<String, WSAttribute>  
    getAttributesByName(String userCredential, String partitionName,  
        Locale requestedLocale,  
        WSReference reference,  
        String[] names)  
    throws CampaignServicesException;
```

Busca os atributos nomeados que estão associados à instância de componente especificada (pode estar vazio)

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: o código de idioma opcional a ser usado para esta solicitação; se não fornecido, as preferências do código de idioma do usuário IBM são usadas. O código de idioma normal que padroniza o algoritmo é aplicado, se necessário.

*partitionName*: nome opcional da partição do Campaign a ser usado. Se não definido, a partição padrão será usada.

*reference*: a referência para a instância do componente que contém os atributos desejados. *InvalidComponentException* é emitido se a referência for inválida ou o componente não existir.

*names*: a matriz opcional de nomes de atributos a serem buscados (não nomes de exibição); se não fornecidos, todos os atributos serão retornados. Lança *AttributeNotFoundException* se um dos atributos nomeados não existir.

## Devoluções

Um mapa digitado de zero ou mais atributos; o nome do atributo é a chave de entrada do mapa e a instância do atributo é o valor de entrada.

## Erros

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

**Nota:** Todas essas exceções são agrupadas no *CampaignServicesException*.

## updateAttributes

```
void updateAttributes(String userCredential, String partitionName,  
    Locale requestedLocale, WSReference reference,  
    boolean allowCreate,  
    WSAttribute[] attributes)  
    throws CampaignServicesException;
```

Atualize um ou mais atributos da instância do componente com os valores de atributo fornecidos.

## Lógica de atualização

A lógica de atualização é a seguinte.

Para cada atributo contido no mapa do atributo fornecido:

1. Se um nome do atributo corresponder a um atributo existente, tente substituir seu campo *valores* pelo campo de valores fornecidos.
2. Se o atributo ainda não existir, *allowCreate* será verdadeiro e seus metadados serão conhecidos, em seguida, crie o atributo. Isso se aplica aos metadados de atributo bem como atributos de instância (exceto fluxogramas).
3. Se o tipo de valor ou algum outro aspecto da definição de metadados do atributo não for atendido, ou um ou mais valores fornecidos for inválido ou estiver fora do intervalo, lance *InvalidAttributeException*.
4. Além disso, emita *AttributeNotFoundException*, se o atributo nomeado não existir.

**Nota:** No caso de uma exceção, nenhuma das atualizações é confirmada.

Esse método específico não suporta definir novos atributos customizados; use o método `createAttributeMetadata()` para isso.

Em todos os casos, a operação de atualização do atributo está sujeita a restrições usuais de segurança e validação. É responsabilidade do cliente determinar quais atributos são necessários para uma determinada instância de componente, os tipos corretos, etc.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*reference*: a referência para a instância do componente que contém os atributos a serem atualizados.

*allowCreate*: indica se um novo atributo deve ser criado, se ainda não existir para o componente. (Consulte “Lógica de atualização” na página 22.)

*attributes*: uma matriz de atributos a ser atualizada; o nome do atributo é usado para localizar o atributo a ser atualizado e os novos valores são usados para atualizar o valor de atributo existente como um único objeto do tipo apropriado ou uma matriz, se aplicável. (Consulte Capítulo 5, “Exceções comuns de API SOAP”, na página 59.)

### Devoluções

Nenhum.

### Erros

`InvalidComponentException`, `AttributeNotFoundException`,  
`InvalidAttributeException`

`AuthorizationException`, `DataException`

## getAttributeMetadataByName

```
Map<String, WSAttributeMetadata>  
getAttributeMetadataByName(String userCredential,  
    String partitionName, Locale requestedLocale,  
    WSReference reference, String[] names)  
throws CampaignServicesException;
```

Busca as definições de metadados de atributo nomeadas que são ligadas a um determinado componente, modelo ou globalmente definido.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*reference*: a referência opcional para o componente ou modelo que contém os metadados de atributo desejados. Se somente o `ComponentTypeEnum` for fornecido, a busca será restrita para os componentes desse tipo. Se a referência não

for fornecida, a busca retornará todas as definições de metadados globais, para todos os tipos de componentes. Lança *InvalidComponentException* se a referência fornecida for inválida.

*names*: matriz opcional de nomes de metadados de atributo a ser buscada. Se não for fornecida, serão retornados todos os metadados para o componente ou globalmente definidos caso nenhuma referência seja fornecida. Lança *AttributeNotFoundException*, se uma ou mais definições de metadados de atributo especificadas não existirem.

## Devoluções

Um mapa digitado de zero ou mais definições de metadados de atributo; o nome do atributo é a chave de entrada de mapa e os metadados do atributo é o valor de entrada.

## Erros

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

## createAttributeMetadata

```
void createAttributeMetadata(String userCredential,  
    String partitionName,  
    Locale requestedLocale, WSReference reference,  
    WSAttributeMetadata[] attributeMetadata)  
    throws CampaignServicesException;
```

Crie uma ou mais novas definições de metadados de atributo e opcionalmente ligue-as a um determinado componente ou modelo.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*reference*: referência opcional para o componente ou modelo aos quais os metadados devem estar ligados. Se não for fornecida, a definição de metadados criada será global. Se a referência for fornecida, mas não válida, emita *InvalidComponentException*.

*attributeMetadata*: uma matriz de definições de metadados do atributo a ser ligada. Se um ou mais metadados especificados estiverem ligados ao componente, ou seja, o nome não for exclusivo, então emita *AttributeExistsException*. Emita *InvalidAttributeException*, se ocorrer um problema com um ou mais metadados especificados, ou seja, estiver internamente inconsistente.

## Devoluções

Nenhum.

## Erros

*InvalidComponentException*, *AttributeExistsException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## updateAttributeMetadata

```
void updateAttributeMetadata(String userCredential,  
    String partitionName,  
    Locale requestedLocale, WSRreference reference,  
    boolean allowCreate,  
    WSAttributeMetadata[] attributeMetadata)  
    throws CampaignServicesException;
```

Atualize uma ou mais definições de metadados de atributo do componente ou modelo especificado, criando opcionalmente novas definições de metadados, se necessário.

### Lógica de atualização

A lógica de atualização é a seguinte.

Para cada definição de metadados de atributo contida na matriz fornecida:

1. Se o nome do atributo não corresponder a um metadado existente que seja ligado ao componente, execute o seguinte com base no valor do parâmetro *allowCreate*:
  - a. *True*: crie uma nova definição de metadados. Funcionalmente idêntico ao uso da solicitação `createAttributeMetadata()`.
  - b. *False*: lance *AttributeNotFoundException*.
2. Se o tipo de dados de metadados do atributo for diferente, lance *InvalidAttributeException*.
3. Tente sobrescrever a definição de metadados de atributo existente pelos valores do campo dos metadados fornecidos, e também lance *InvalidAttributeException*. Somente as atualizações a seguir são suportadas (também lance *InvalidAttributeException*):
  - a. *name*: não pode ser alterado (o nome é a chave!).
  - b. *displayName*: aceitar novo valor.
  - c. *description*: aceitar novo valor.
  - d. *isRequired*: somente permitir mudança de *true* para *false*.
  - e. *isInternal*: aceitar novo valor.
  - f. *isGenerated*: nenhuma mudança permitida.
  - g. *attributeTypeEnum*: nenhuma mudança permitida.
  - h. *accessTypeEnum*: aceitar novo valor.
  - i. *selectTypeEnum*: aceitar essas transições, se as opções forem fornecidas:
    - 1) NONE para SINGLE\_SELECT ou MULTIPLE\_SELECT
    - 2) SINGLE\_SELECT para MULTIPLE\_SELECT
  - j. *options*: as opções podem ser incluídas, mas não excluídas. Somente as mudanças de opção a seguir são suportadas (de acordo com a correspondência de valor):
    - 1) *displayName*: aceitar novo valor (sem cálculo automático).
    - 2) *description*: aceitar novo valor (sem cálculo automático).
    - 3) *isDefault*: aceitar novo valor; no entanto, deve corresponder a *SelectTypeEnum*.
    - 4) *value*: nenhuma mudança é permitida (o valor é a chave!).
  - k. *defaultValue*(somente fluxogramas): aceitar novo valor padrão.
  - l. *maxLength*(somente texto): aceitar novo comprimento, se maior.

4. Se a definição de metadados do atributo não for internamente consistente, lance *InvalidAttributeException*.
5. Se necessário, localize todas as instâncias do componente que fazem referência aos metadados do atributo atualizado e atualize conforme apropriado.

**Nota:** No caso de uma exceção, nenhuma das atualizações será confirmada.

Em todos os casos, a operação de atualização do atributo está sujeita a restrições usuais de segurança e validação.

Consulte `createAttributeMetadata()`, `deleteAttributeMetadata()`

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*reference*: referência opcional para a instância do componente que contém os atributos desejados. Se não fornecido, a atualização será restrita às definições globais de metadados. Lança *InvalidComponentException* se a referência fornecida for inválida.

*allowCreate*: se verdadeiro, as definições de metadados que atualmente não existem serão criadas (funcionalmente equivalente ao uso do método `createAttributeMetadata()`).

*attributeMetadata*: uma matriz de definições de metadados de atributo a serem atualizadas (e incluídas se o sinalizador *allowCreate* for verdadeiro). O nome do atributo é usado para localizar a definição de metadados a ser atualizada e os dados restantes são usados para atualizar a definição existente. (Consulte “Lógica de atualização” na página 25.)

## Devoluções

Nenhum.

## Erros

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## deleteAttributeMetadata

```
void deleteAttributeMetadata(String userCredential,  
    String partitionName,  
    Locale requestedLocale, WsReference reference,  
    String[] names)  
    throws CampaignServicesException;
```

Exclui uma ou mais definições de metadados de atributo nomeado do componente especificado, modelo (somente metadados de atributo customizado) ou definições de metadados de atributo global.

Como parte desta tarefa, o método localiza todos os componentes que fazem referência aos metadados excluídos e e atualiza conforme apropriado.

**Nota:** No entanto, no caso de uma exceção, nenhuma das exclusões é confirmada.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*reference*: referência opcional do componente ou modelo que contém os atributos a serem excluídos. Se não fornecido, a exclusão será restrita a definições de metadados globais. Lança *InvalidComponentException* se a referência fornecida for inválida.

**Nota:** Se a matriz de nomes opcional de metadados de atributo não for fornecida, esse método tentará excluir todos os metadados de atributo customizado que estão associados ao componente ou todas as definições globais, se a referência não foi fornecida.

*names*: matriz opcional de nomes dos metadados de atributo a serem excluídos. Lança *AttributeNotFoundException* se um ou mais metadados de atributo nomeado não existir. Lança *InvalidAttributeException*, se um atributo não pôde ser removido.

## Devoluções

Nenhum.

## Erros

*InvalidComponentException*, *AttributeNotFoundException*,  
*InvalidAttributeException*

*AuthorizationException*, *DataException*

---

## Métodos de API SOAP: Campanhas e fluxograma

A API SOAP do IBM Campaign dá suporte às operações a seguir no que diz respeito a campanhas e fluxogramas (sujeito a permissões de segurança).

- cria uma campanha
- descoberta (listar campanhas por diversos critérios)
- criação, leitura e atualização de atributo (via APIs de atributo)
- para execução do fluxograma

As campanhas possuem inúmeros atributos padrão que estão associados a elas que são expostos pela API. Essa lista pode ser estendida à vontade pelo cliente, incluindo os atributos customizados (consulte as APIs de Atributos).

Os atributos de campanha padrão são listados abaixo:

- *uacName*: nome da campanha (exclusivo não garantido).
- *uacDescription*: sequência opcional que descreve a campanha.
- *uacCampaignCode*: um código de sequência que identifica exclusivamente a campanha. Geralmente gerado pela campanha, mas pode ser fornecido pelo cliente.
- *uacCreateDate*: um Calendário que indica a data e a hora em que a campanha foi criada pelo servidor.

- *uacUpdateDate*: um Calendário que indica a data e a hora em que a campanha foi atualizada por último pelo servidor.
- *uacInitiative*: a sequência opcional que define a iniciativa da campanha.
- *uacObjectives*: sequência opcional que identifica os objetivos da campanha.
- *uacStartDate*: um Calendário opcional que fornece a data e a hora em que a campanha foi iniciada pelo servidor, ou que está planejada para iniciar.
- *uacEndDate*: assim como com o *uacStartDate*, mas define a data e a hora em que a campanha foi concluída ou está planejada para ser concluída. Deve ser após o *uacStartDate*.
- *uacLastRunDate*: um Calendário opcional que indica a data e a hora em que algum fluxograma ligado à campanha foi executado pela última vez (else null).
- *uacExternalLinkOwner*: uma sequência opcional que define o nome do proprietário de um link externo (consulte o atributo *uacExternalLinkReference*). Uso de IBM somente; deve ser um dos seguintes:
  - "Plan" (agora conhecido como IBM Marketing Operations)
  - "Collaborate" (agora conhecido como IBM Distributed Marketing)
- *uacExternalLinkId*: um ID do banco de dados opcional numérico que é designado por outro aplicativo IBM a um objeto vinculado a essa campanha. Uso de IBM somente: consulte também o atributo *uacExternalLinkOwner*.

## generateCampaignCode

```
String generateCampaignCode(String userCredential,
    String partitionName,
    Locale requestedLocale);
```

Gere um novo código de campanha.

Este código é garantido exclusivo e diferente do valor que é retornado por uma chamada anterior ou futura para este método ou o método `createCampaign()`, ou o valor que é gerado para uma campanha que é criada pela GUI IBM Campaign.

**Nota:** O uso deste método é opcional, porque a API `createCampaign()` gera um código de campanha para o cliente se uma não for fornecida.

Consulte `createCampaign()`.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado. Se houver somente uma partição na instalação do Campaign, este argumento pode ser nulo.

### Devoluções

O código de campanha gerado.

### Erros

AuthorizationException, DataException

## deleteCampaigns

```
public WSDelateCampaignsResponse deleteCampaigns(String userCredential,
    String partitionName, Locale requestedLocale, WSReference[] wsReferences)
    throws CampaignServicesException
```



Exclui as campanhas especificadas do sistema.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*wsReference*: faz referências de campanhas a serem excluídas.

## Devoluções

Retorna o objeto de tipo `WSDeleteCampaignsResponse`.

## Erros

Lança `CampaignServicesException`, se a campanha não existir ou a referência for inválida ou nenhuma referência for fornecida.

## createCampaign

```
CampaignInfo createCampaign(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    String securityPolicyName,  
    String name, Attribute[] attributes)  
    throws InvalidFolderException, AttributeNotFoundException,  
    InvalidAttributeException;
```

Crie uma nova campanha para o cliente, a partição e o `securityPolicyName`, aplicando os atributos especificados. Todas as campanhas que são criadas por esta API estão sob a pasta raiz.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*securityPolicyName*: nome opcional da política de segurança de campanha a ser usado para criar a campanha. Todas as operações subsequentes nesta campanha usam esta política. Se não definido, a política global será usada.

*name*: o nome para designar a nova instância de campanha (seu atributo "`uacName`").

*attributes*: uma matriz opcional de atributos de inicialização; quaisquer atributos fornecidos sobrescreverão os valores padrão da campanha; outros serão deixados inalterados. Por exemplo, se um atributo `uacCampaignCode` for fornecido, ele será usado em vez de um gerado automaticamente. Cabe ao cliente determinar os atributos que são obrigatórios para a campanha, seus tipos, etc.

Emite `AttributeNotFoundException` se um ou mais atributos nomeados não existir ou `InvalidAttributeException`, se um valor de atributo é inválido (como tipo de dados incorretos).

## Devoluções

Uma única instância de uma *CampaignInfo* para a campanha criada.

## Erros

*InvalidAttributeException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

## listCampaignsByPage

```
List<CampaignInfo>  
listCampaignsByPage(String userCredential, String partitionName,  
    Locale requestedLocale, Attribute[] attributes,  
    long pageOffset, int pageSize)  
throws AttributeNotFoundException, InvalidAttributeException,  
    RangeException;
```

Enumere uma "página" de campanhas que correspondem aos valores de atributo opcional, iniciando com o deslocamento de página especificado. As pastas são ignoradas.

Depois de recuperadas, cada *CampaignInfo* retornada pode ser usada no estado em que se encontra, tal como exibir uma lista de resumos ou os métodos de atributos podem ser usados para buscar ou atualizar os atributos da campanha.

Nenhum estado é mantido por esta API; portanto, é possível fazer chamadas para ela em qualquer ordem.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*attributes*: matriz opcional de atributos a serem correspondidos; o nome do atributo, tipo de dados e valores são usados para determinar a correspondência; se o atributo suportar as matrizes, todos os valores especificados devem ser correspondentes. O operador de correspondência implícito é um AND, portanto somente campanhas que correspondam a todos os valores de atributo fornecidos serão retornadas.

Lance *AttributeNotFoundException* se um nome de atributo não existir ou *InvalidAttributeException* se um ou mais atributos fornecidos for inválido.

*pageOffset*: a compensação inicial de todas as possíveis campanhas para iniciar a enumeração (de valor zero). Por exemplo, se a enumeração corresponder a 1000 campanhas e esse valor for configurado como 10, a página iniciaria no 11º componente. Uma *RangeException* é lançada se a compensação fornecida estiver fora do intervalo.

*pageSize*: o número máximo de campanhas correspondidas a serem retornadas para a página (não pode exceder 500).

## Devoluções

Uma Lista digitada de zero ou mais instâncias do wrapper de dados *CampaignInfo*, um para cada campanha correspondida na página.

## Erros

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

## stopFlowchart

```
stopFlowchart(int pid, int runid)
```

Esta API para um fluxograma em execução. Para uma única configuração de listener, uma execução de fluxograma pode ser exclusivamente identificada pelo PID que está associado à execução do fluxograma. O PID indica o ID do processo unica\_acsvr. Se múltiplos listeners IBM Campaign estiverem configurados, deve-se incluir o ID de execução que está associado à execução de fluxograma e ao PID.

## Parâmetros

*pid*: ID do processo unica\_acsvr associado a uma execução do fluxograma.

*runid*: ID de execução com uma execução do fluxograma. O parâmetro necessário para uma configuração de listener em cluster. Parâmetro opcional se um único listener for configurado.

## Devoluções

Nenhum

## Erros

Nenhum

---

## Métodos de API SOAP: Células de destino

As células de destino são uma abstração para alguns subconjuntos conhecidos de resultados de campanha que são gerenciados por IBM Campaign como uma Planilha de célula de destino (TCS). As células de destino podem ser globais para uma campanha ou podem estar associadas a um determinado fluxograma de campanha.

A API SOAP do IBM Campaign dá suporte às operações a seguir em células de destino:

- criar uma ou mais novas células de destino global
- atualizar em massa uma ou mais células de destino existentes
- descoberta (listagem de células de destino)
- criação, leitura e atualização de atributo (via APIs de atributo)
- excluir uma célula de destino existente
- buscar os resultados de execução que estão associados a uma ou mais células

As células de destino possuem inúmeros atributos padrão que estão associados a elas que são expostos pela API. Essa lista pode ser estendida à vontade pelo cliente, incluindo as definições de metadados de atributo customizado (consulte as APIs de Atributos). Cada metadados de atributo pode ser considerada uma coluna no TCS; o layout da planilha é opção do cliente.

Os atributos padrão da célula de destino são:

- *uacName*: nome da célula.
- *uacDescription*: sequência opcional que descreve o fluxograma.
- *uacCellCode*: uma sequência de códigos que identifica exclusivamente a célula. Geralmente, gerado automaticamente por Campaign, mas pode ser fornecido pelo cliente.
- *uacCreateDate*: uma instância de Calendário que fornece a data & hora em que a célula foi criada pelo servidor.
- *uacUpdateDate*: uma instância do Calendário que define quando foi a última vez que a célula foi atualizada pelo servidor.
- *uacIsControl*: um booleano que indica se essa é uma célula de controle (verdadeiro) ou não (falso). Outras células podem se referir a esta célula como uma célula de controle (consulte *uacControlCell*).
- *uacControlCell*: referência opcional da célula de controle (não permitido, se uma célula de controle). Consulte o atributo *uacIsControl*.
- *uacIsApproved*: um booleano que indica se a célula é aprovada (verdadeiro) ou não (falso).
- *uacIsReadOnly*: um booleano que indica se a célula é somente leitura (verdadeiro) ou não (falso).
- *uacDisplayOrder*: um número inteiro que fornece a ordem desta célula (linha) relativa a outras na planilha de células de destino.
- *uacIsTopDown*: um booleano que indica se a célula está de cima para baixo.
- *uacAssignedOffers*: uma matriz opcional de uma ou mais referências de ofertas ou listas de ofertas designada a essa célula (não permitida, se uma célula de controle).
- *uacFlowchartName*: nome opcional de fluxograma ao qual esta célula está vinculada (somente leitura deve ser configurada através da GUI IBM Campaign; não permitido se uma célula de controle).
- *uacFlowchartId*: ID do banco de dados opcional para o fluxograma ao qual esta célula está vinculada (somente leitura, como acima).

## createTargetCell

```

TargetCellInfo
    createTargetCell(String userCredential, String partitionName,
        Locale requestedLocale,
        Reference campaignReference,
        Attribute[] attributes)
        throws InvalidComponentException, CompositeException;

```

Crie uma nova linha de célula de destino específica de campanha, aplicando os atributos por célula especificados e informações do usuário.

Os atributos especificados podem ser padrão ou customizado; no entanto, se customizado, então devem existir definições de metadados de atributo global correspondente.

Depois que a célula de destino é criada, os valores de atributo podem ser alterados usando as APIs de atributos.

Consulte `listTargetCells()`, `bulkCreateTargetCells()`.

Consulte `createAttributeMetadata()`, `listAttributeMetadata()`, `getAttributesByName()`

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*campaignReference*: a referência da campanha que contém a planilha de células de destino a ser atualizada. Acumula um *InvalidComponentException*, se a campanha não existir ou a referência estiver inválida.

*attributes*: matriz opcional de atributos TCS para a nova célula. Cada elemento de atributo fornecido substitui os valores padrão do atributo de célula correspondente; outros são deixados inalterados. Cabe ao cliente determinar os atributos que são obrigatórios para a célula, seus tipos, etc. Acumula um *InvalidAttributeException*, se houver um problema com um atributo especificado.

Se alguma exceção for acumulada, esse método emitirá um *CompositeException* e todas criações serão desfeitas. A lista de causas da exceção inclui uma exceção para cada atributo que causou o erro e inclui um índice numérico em vez da referência, o nome do atributo e geralmente o valor ofensivo. A lista de causas é ordenada da mesma maneira que a *attributeList* de entrada.

## Devoluções

Um wrapper de dados *TargetCellInfo* para a célula TCS criada.

## Erros

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## bulkCreateTargetCells

```
List<TargetCellInfo>  
    bulkCreateTargetCells(String userCredential,  
        String partitionName,  
        Locale requestedLocale,  
        Reference campaignReference,  
        List<Attribute[]> attributesList)  
    throws InvalidComponentException, CompositeException;
```

Crie várias linhas de célula de destino específicas para nova campanha de uma vez, aplicando os atributos específicos por célula e as informações do usuário.

Os atributos especificados podem ser padrão ou customizado; no entanto, se customizado, então devem existir definições de metadados de atributo global correspondente.

Depois que a célula de destino é criada, os valores de atributo podem ser alterados usando as APIs de atributos.

Consulte `listTargetCells()`.

Consulte `createAttributeMetadata()`, `listAttributeMetadata()`, `getAttributesByName()`

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*campaignReference*: a referência da campanha que contém a planilha de células de destino a ser atualizada. Acumula um *InvalidComponentException*, se a campanha não existir ou a referência estiver inválida.

*attributeList*: lista opcional de matrizes do atributo por célula, uma para cada linha da célula de destino a ser criada. Qualquer atributo fornecido para um elemento de lista específico substitui os valores padrão do atributo de célula correspondente; outros são deixados inalterados. Cabe ao cliente determinar os atributos que são obrigatórios para a célula, seus tipos, etc. Acumula um *InvalidAttributeException*, se houver um problema com um atributo especificado.

Se alguma exceção for acumulada, esse método emitirá um *CompositeException* e todas criações serão desfeitas. A lista de causas da exceção inclui uma exceção para cada atributo que causou o erro e inclui um índice numérico em vez da referência e o nome do atributo, etc. A lista de causa é ordenada da mesma maneira que a entrada *attributeList*.

## Devoluções

Uma lista de wrappers de dados *TargetCellInfo*, uma para cada instância criada, ordenada de acordo com a ordem do elemento do parâmetro *attributesList* de entrada.

## Erros

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## listTargetCells

```
List<TargetCellInfo>  
listTargetCells(String userCredential,  
Reference campaignReference, Locale requestedLocale,  
Attribute[] attributes)  
throws InvalidComponentException, InvalidAttributeException;
```

Lista informações sobre todas as células de destino atualmente existentes que correspondem aos atributos especificados, seja para campanha especificada ou globalmente, se nenhuma campanha for especificada.

Consulte *getAttributeMetadata()*, *getAttributesByName()*.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*campaignReference*: referência da campanha pai. Lança *InvalidComponentException*, se a campanha não existir ou a referência estiver inválida.

*attributes*: matriz opcional de atributos a serem correspondidos. O operador de correspondência implícito é um AND; portanto, somente as células que correspondem a todos os valores de atributo fornecidos serão retornadas.

Lança *InvalidAttributeException*, se um ou mais atributos especificados forem inválidos.

### Devoluções

Retorna uma lista de zero ou mais instâncias *TargetCellInfo* para as células correspondidas.

### Erros

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## bulkUpdateTargetCells

```
void bulkUpdateTargetCells(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    Map<Reference, Attribute[]> attributesMap)  
    throws CompositeException;
```

Atualize os atributos de uma ou mais células de destino.

A lógica de atualização é a seguinte.

Para cada elemento no *attributesMap* fornecido, a chave de entrada é a referência da célula de destino para atualização e o valor de entrada é uma matriz dos atributos de atualização para essa célula. Se a célula de destino não existir, acumule um *InvalidComponentException*.

Depois que uma célula de destino é localizada, para cada atributo que é especificado, faça o seguinte:

1. Se um nome do atributo corresponder a um atributo existente, tente substituir seu campo de valores pelo campo de valores fornecidos.
2. Se o tipo de valor ou algum outro aspecto da definição de metadados do atributo não for atendido, ou um ou mais valores fornecidos for inválido ou estiver fora do intervalo, acumule um *InvalidAttributeException*.
3. Além disso, acumule *AttributeNotFoundException*, se o atributo nomeado não existir.

Se alguma exceção for acumulada, esse método emitirá um *CompositeException* e todas as atualizações serão desfeitas. A lista de causas da exceção inclui as exceções que são listadas acima. Para cada atributo que causou o erro, tanto a referência quanto o nome do atributo serão registrados.

Em todos os casos, a operação de atualização do atributo está sujeita a restrições usuais de segurança e validação. É responsabilidade do cliente determinar quais atributos são necessários para uma determinada instância de componente, os tipos corretos, etc.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*attributesMap*: um mapa de células de destino para atualização; a chave de entrada é a referência da célula para atualização e o valor de entrada é uma matriz de atributos de atualização. O nome do atributo é usado para localizar o atributo para atualização e os novos valores de atributo são usados para atualizar o valor de atributo existente como um único objeto do tipo apropriado ou uma matriz, se aplicável. Consulte as exceções acima.

## Devoluções

Nenhum.

## Erros

ComponentException

AuthorizationException, DataException

## getRunResultsByCell

```
List<RunResults>  
  getRunResultsByCell(String userCredential, String partitionName,  
    Locale requestedLocale,  
    Reference[] cellReferences)  
  throws InvalidComponentException;
```

Obtenha os resultados de execução de uma ou mais células de destino, possivelmente para um fluxograma que nunca foi iniciado ou ainda está em andamento.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*cellReferences*: uma matriz de referências das células de destino cujos resultados de execução são desejados. Lança *InvalidComponentException* se uma ou mais referências de célula forem inválidas ou fizer referência a uma célula não existente.

## Devoluções

Retorna uma lista digitada de resultados de execução para células nomeadas, ordenadas de acordo com a matriz de referência de entrada.

Cada status de execução é RUNNING se a caixa de processo do fluxograma subjacente ainda estiver em execução, FAILED se a execução tiver falhado por algum motivo ou NOT\_STARTED se a execução da caixa do projeto não foi iniciada. Os detalhes do status também são fornecidos.

## Erros

InvalidComponentException

AuthorizationException, DataException



## bulkDeleteTargetCells

```
void bulkDeleteTargetCells(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    Reference[] cellReferences)  
    throws CompositeException;
```

Exclui uma ou mais células de destino existentes e todos os seus componentes dependentes (ou seja, ligação do fluxograma, atributos).

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*cellReferences*: uma matriz ou uma ou mais referências de células a serem excluídas. *InvalidComponentException* será acumulado, se houver um problema com uma das referências especificadas ou uma célula não existir.

Se houver alguma exceção acumulada, esse método emitirá *CompositeException* e todas as exclusões serão desfeitas. A lista de causas da exceção inclui as exceções que são listadas acima. Para cada célula que causou o erro, a referência é registrada.

### Devoluções

Nenhum.

### Erros

CompositeException

AuthorizationException, DataException

## updateTemplateAttributes

```
updateTemplateAttributes  
(String userCredential, String partitionName, Locale requestedLocale,  
WSReference wsReference, boolean allowCreate,  
boolean clearExisting, WSAttribute[] wsStaticAttributes,  
WSAttribute[] wsHiddenAttributes, WSAttribute[] wsParametricAttributes)  
    throws CampaignServicesException
```

Atualiza os atributos dos modelos especificados

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*wsCampaignReference*: referência da campanha pai.

*allowCreate*: não usado agora.

*clearExisting*: sinaliza se configurado como verdadeiro, todos os valores anteriores no modelo que não são enviados na solicitação são limpos.

*wsStaticAttributes*: lista de atributos estáticos no modelo.

*wsHiddenAttributes*: lista de atributos ocultos no modelo.

*wsParametricAttributes*: lista de atributos parametrizados no modelo.

## Devoluções

Nenhum.

## Erros

Lança `CampaignServicesException` se o modelo de oferta não existir ou a referência for inválida ou nenhuma referência for fornecida.

## listBottomUpTargetCells

```
public List <WSTargetCellDetails>  
listBottomUpTargetCells(String userCredential, String partitionName,  
Locale requestedLocale, WSReference wsCampaignReference)  
throws CampaignServicesException
```

Lista de informações sobre todas as células de destino de baixo para cima que existem atualmente para a campanha especificada.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*wsCampaignReference*: referência da campanha pai.

## Devoluções

Retorna uma lista de zero ou mais instâncias `WSTargetCellDetails` para as células correspondidas

## Erros

Lança `CampaignServicesException`, se a campanha não existir ou a referência for inválida.

---

## Métodos de API SOAP: Analítica

A API SOAP do IBM Campaign dá suporte à recuperação de métricas simples do IBM Campaign.

## getCampaignMetrics

```
MetricsInfo getCampaignMetrics(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    Reference campaignReference)  
    throws InvalidComponentException;
```

Busca as métricas para a campanha especificada.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*campaignReference*: a referência da campanha pai. Lança *InvalidComponentException* se houver um problema com a referência de campanha ou se a campanha não existir.

### Devoluções

Retorna uma instância *MetricsInfo* para a campanha.

### Erros

*InvalidComponentException*

*AuthorizationException*, *DataException*

---

## Métodos de API SOAP: Ofertas, listas de ofertas, modelos de oferta

A API SOAP do IBM Campaign dá suporte às operações a seguir relacionadas a ofertas.

- *discovery*: listagem por pasta (ofertas, listas de ofertas e subpastas), atributo (ofertas e modelos de ofertas) ou o valor de procura (ofertas)
- validação
- recuperação de informações (atributos de recuperação para uma oferta específica ou modelo de ofertas)
- cria, editar, obsoleter e excluir ofertas

As ofertas possuem um número de atributos padrão que estão associados a elas. Esta lista pode ser estendida pelo cliente, incluindo definições de metadados de atributo customizado (consulte as APIs de Atributos).

Os atributos de oferta padrão são:

- *uacName*: nome da oferta.
- *uacDescription*: sequência opcional que descreve a oferta.
- *uacOfferCode*: uma sequência de códigos que identifica exclusivamente a oferta. Geralmente gerada pelo IBM Campaign, mas pode ser fornecido pelo cliente.
- *uacCreateDate*: uma instância do calendário que indica a data e a hora em que a oferta foi criada pelo servidor.
- *uacUpdateDate*: uma instância do calendário que indica a última vez em que a oferta foi atualizada pelo servidor.

Os modelos de oferta também possuem atributos padrão e customizados. Os atributos de modelo de oferta padrão são:

- *uacName*: nome do modelo de oferta.
- *uacDescription*: sequência opcional que descreve o modelo de oferta.
- *uacCreateDate*: uma instância do calendário que indica a data e a hora em que o modelo de oferta foi criado pelo servidor.
- *uacUpdateDate*: uma instância de Calendário que indica a última vez em que o modelo de oferta foi atualizado pelo servidor.

## listOffersAndFolders

```
List<WSComponentOrFolderInfo>  
listOffersAndFolders(String userCredential, String partitionName,  
    Locale requestedLocale,  
    WSReference parentReference)  
throws CampaignServicesException;
```

Liste todas as ofertas, listas de ofertas e pastas sob a pasta pai opcional.

Depois de recuperada, cada instância *WSComponentOrFolderInfo* que é retornada pode ser usada no estado em que se encontra, por exemplo, para exibir o próximo nível da hierarquia de pasta, as APIs de atributo podem ser usadas para buscar ou atualizar quaisquer ofertas contidas.

### Parâmetro

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*parentReference*: referência opcional da pasta-pai a ser listada. Somente as ofertas, listas de ofertas e pastas filhas imediatas de desta pasta pai serão enumeradas, de modo que as chamadas sucessivas a essa API sejam necessárias para navegar na hierarquia de pasta inteira (no entanto, geralmente é muito superficial). Se nenhum pai for fornecido, todos os componentes e pastas na raiz serão retornados.

Lança *InvalidFolderException* se ocorrer um problema com a referência de pasta pai especificada.

Uma *Lista* digitada de zero ou mais instâncias wrapper de dados *WSComponentOrFolderInfo*, uma para cada componente ou pasta correspondida.

### Erros

*InvalidFolderException*

*InvalidExecutionContextException*, *AuthorizationException*

## searchOffersBasic

```
List<WSOfferInfo>  
searchOffersBasic(String userCredential, Locale requestedLocale,  
    String partitionName, long folderID,  
    String searchCriteria, boolean includeRetired,  
    int pageOffset, int pageSize)  
throws CampaignServicesException;
```

Enumere uma "página" de ofertas que contenham os critérios de pesquisa fornecidos nos campos de nome, descrição, createdBy ou código de oferta, iniciando com a compensação da página especificada. A procura é baseada na entrada da Pasta opcional. (Se um folderID de 0 for fornecido, a pasta de oferta raiz será usada por padrão). As correspondências são retornadas com base em uma correspondência "contains" para a sequência de procura.

Depois de recuperadas, cada *WSOfferInfo* retornada pode ser usada no estado em que se encontra, por exemplo, para exibir uma lista de resumo ou os métodos de atributo podem ser usados para buscar ou atualizar os atributos de oferta.

Nenhum estado é mantido por esta API; portanto, é possível fazer chamadas para ela em qualquer ordem.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*folderID*: o ID da pasta de Oferta a ser procurado; se o ID da pasta de 0 for especificado, a pasta raiz é procurada.

*searchCriteria*: a frase de procura.

*includeRetired*: o valor booleano especifica se os resultados de procura incluem ofertas obsoletas. Os valores válidos são TRUE e FALSE, com TRUE indicando que as ofertas obsoletas são incluídas e FALSE indicando que as ofertas obsoletas não são incluídas.

*pageOffset*: a compensação inicial de todos os componentes possíveis para iniciar a enumeração (valor zero). Por exemplo, se a enumeração corresponder a 1000 ofertas e esse valor for configurado como 10, a página iniciaria no 11º componente. Uma *RangeException* é lançada se a compensação fornecida estiver fora do intervalo.

*pageSize*: o número máximo de componentes correspondidos para retornar para a página (não pode exceder 500).

## Devoluções

Retorna uma Lista digitada de zero ou mais instâncias de wrapper de dados de *Oferta*, uma para cada oferta retornada na página.

## Erros

*RangeException*

## listOffersByPage

```
List<OfferInfo>  
listOffersByPage(String userCredential, String partitionName,  
                 Locale requestedLocale, Attribute[] attributes,  
                 long pageOffset, int pageSize)  
throws AttributeNotFoundException, InvalidAttributeException,  
                 RangeException;
```

Enumere uma "página" de ofertas que corresponda aos valores de atributo opcional, iniciando com o deslocamento de página especificado. As pastas são

ignoradas. As correspondências são retornadas com base em uma correspondência "like" para as sequências (em que a correspondência é considerada suficiente, se uma sequência contiver o valor consultado) e a correspondência exata para datas e números.

Depois de recuperada, cada *OfferInfo* retornada pode ser usada no estado em que se encontra, tal como exibir uma lista de resumos ou os métodos de atributos podem ser usados para buscar ou atualizar os atributos de oferta.

Nenhum estado é mantido por esta API; portanto, é possível fazer chamadas para ela em qualquer ordem.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*attributes*: matriz opcional de atributos a serem correspondidos; o nome do atributo, tipo de dados e valores são usados para determinar a correspondência; se o atributo suportar as matrizes, todos os valores especificados devem ser correspondentes. O operador de correspondência implícito é um OR; portanto, os componentes que correspondem a qualquer valor de atributo fornecido serão retornados.

Lança *AttributeNotFoundException* se um nome de atributo não existir ou *InvalidAttributeException* se um ou mais atributos fornecidos forem inválidos.

*pageOffset*: a compensação inicial de todos os componentes possíveis para iniciar a enumeração (zerovalued). Por exemplo, se a enumeração corresponder a 1000 ofertas e esse valor for configurado como 10, a página iniciaria no 11º componente. Uma *RangeException* é lançada se a compensação fornecida estiver fora do intervalo.

*pageSize*: o número máximo de componentes correspondidos para retornar para a página (não pode exceder 500).

## Devoluções

Uma Lista digitada de zero ou mais instâncias do wrapper de dados *OfferInfo*, um para cada componente correspondido na página.

## Erros

*AttributeNotFoundException*, *InvalidAttributeException*, *RangeException*

*InvalidExecutionContextException*, *AuthorizationException*

## createSmartOfferList

```
public WSCreateSmartOfferListResponse createSmartOfferList
(String userCredential, String partitionName, Locale requestedLocale,
String name, String description, String policyName,
WSReference parentFolder, WSSmartListInfo offerListInfo,
WSApplicationTypeEnum createdBy, long creatorObjectId)
throws CampaignServicesException
```

Cria uma nova lista de ofertas inteligentes.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*name*: nome da pasta a ser criada.

*description*: descrição para a nova pasta.

*securityPolicyName*: nome da política de segurança a ser usada.

*parentFolder*: ID da pasta na qual a lista de ofertas deve ser criada.

*offerListInfo*: objeto do tipo WSSmartListInfo.

*createdBy*: (opcional) objeto do tipo WSApplicationTypeEnum indica qual aplicativo criou a pasta. Possíveis valores - Campaign/Plan/Collaborate/eMessage. Se não fornecidos, a campanha será usada.

*creatorObjectId*: (opcional) usado pelo Plano (IBM Marketing Operations) para vincular uma pasta no IBM Campaign com uma pasta no Marketing Operations.

## Devoluções

Retorna objetos do tipo WSCreateSmartOfferListResponse.

## Erros

Emite CampaignServicesException se o ID parentFolder for inválido ou offerListInfo não for fornecido.

Emite CampaignServicesException, se o nome da lista estiver duplicado.

## createStaticOfferList

```
public WSCreateStaticOfferListResponse createStaticOfferList
(String userCredential, String partitionName, Locale requestedLocale, String name,
String description, String policyName, WSReference parentFolder, WSReference[]
listMembers, WSApplicationTypeEnum createdBy, long creatorObjectId)
throws CampaignServicesException
```

Cria nova lista de ofertas estática.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*name*: nome da pasta a ser criada.

*description*: descrição para a nova pasta.

*securityPolicyName*: nome da política de segurança a ser usada.

*parentFolder*: ID da pasta na qual a lista de ofertas deve ser criada.

*listMembers*: faz referência a ofertas a serem incluídas na lista de ofertas.

*createdBy*: (opcional) objeto do tipo `WSApplicationTypeEnum` indica qual aplicativo criou a pasta. Possíveis valores - Campaign/Plan/Collaborate/eMessage. Se não for indicado, o Campaign será usado.

*creatorObjectId*: (opcional) usado pelo Plano (IBM Marketing Operations) para vincular uma pasta no IBM Campaign com uma pasta no Marketing Operations.

## Devoluções

Retorna objetos do tipo `WSCreateStaticOfferListResponse`

## Erros

Emite `CampaignServicesException`, se o ID *parentFolder* for inválido ou os *listMembers* for inválidos.

Emite `CampaignServicesException`, se o nome da lista estiver duplicado.

## getOffers

```
public WSGetOffersResponse getOffers  
(String userCredential, String partitionName,  
Locale requestedLocale, WSReference[] wsReferences)  
throws CampaignServicesException
```

Lista os detalhes de ofertas conforme especificado na solicitação.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*wsCampaignReference*: referência da campanha pai.

## Devoluções

Retorna o objeto do tipo `WSGetOffersResponse`.

## Erros

Lança `CampaignServicesException`, se a oferta não existir ou a referência for inválida ou nenhuma referência for fornecida.

## validateOffers

```
List<OfferValidationInfo>  
validateOffers(String userCredential, String partitionName,  
Locale requestedLocale,  
OfferCodeOrName[] codeOrNames);
```



Valide os códigos de oferta fornecidos ou os nomes da lista de oferta e retorne as informações de validação para cada um. "Validação" consiste em verificar se somente uma oferta de correspondência ou lista de ofertas existe no banco de dados.

O objeto `OfferValidationInfo` contém uma mensagem de erro em vez de Informações de ofertas, se zero ofertas ou listas de ofertas forem encontradas correspondendo ao código ou nome fornecido. Um erro também é retornado em vez de uma correspondência, se o código ou nome fornecido corresponder a várias ofertas ou listas de ofertas. A lista é retornada na mesma ordem fornecida. Os códigos de oferta e os nomes da lista de oferta são validados com base na exata correspondência com ofertas.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*codeOrNames*: uma matriz de todos os códigos de oferta ou nomes de lista de oferta a serem validados.

**Nota:** Nenhuma exceção é lançada por este método; em vez disso, as informações de validação são retornadas para todos os códigos ou nomes fornecidos.

### Devoluções

Uma Lista digitada de zero ou mais instâncias do wrapper de dados `OfferValidationInfo`.

### Erros

Nenhum.

## editOfferList

```
public WSEditOfferListResponse editOfferList(String userCredential,
String partitionName, Locale requestedLocale, WSReference listReference,
boolean isSmartList,String name, String description,
WSReference[] listMembers,WSSmartListInfo offerListInfo,
Long creatorObjectId, boolean clearExisting)
throws CampaignServicesException
```

Atualiza os detalhes da lista de ofertas estática e inteligente.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*listReference*: referência à lista de ofertas.

*isSmartList*: sinalizador indica se a lista é inteligente ou estática.

*name*: nome da pasta a ser criada.

*description*: descrição para a nova pasta.

*listMembers*: faz referência a ofertas a serem incluídas na lista de ofertas.

*offerListInfo*: objeto do tipo WSSmartListInfo.

*creatorObjectId*: (opcional) usado pelo Plano (IBM Marketing Operations) para vincular uma pasta no IBM Campaign com uma pasta no Marketing Operations.

*clearExisting*: sinalizador indica se as informações existentes devem ser limpas. Se verdadeiro, os membros da lista existentes são limpos antes de incluir novos membros. Se falso, os novos membros são anexados aos membros existentes.

## Devoluções

Retorna objetos do tipo WSEditOfferListResponse.

## Erros

Lança CampaignServicesException, se o ID parentFolder for inválido ou offerListInfo não for fornecido ou os listMembers forem inválidos.

Lança CampaignServicesException, se o nome da lista estiver duplicado.

## createOffer

```
OfferInfo createOffer(String userCredential, String partitionName,
    Locale requestedLocale,
    String securityPolicyName,
    String name, String templateName,
    Attribute[] attributes)
    throws InvalidFolderException, AttributeNotFoundException,
    InvalidAttributeException;

public WSOfferInfo createOffer(String authorizationLoginName, String
    partitionName, Locale requestedLocale, String
    securityPolicyName, String name, long folderID,
    String templateName, WSAttribute[] wsAttributes)
    throws CampaignServicesException;
```

Crie uma nova oferta para o cliente, aplicando os atributos especificados.

## Parâmetros

*authorizationLoginName*: Nome do usuário que cria a oferta. Os usuários devem receber permissão para Incluir ofertas para usar esse método.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*securityPolicyName*: nome opcional da política de segurança de campanha a ser usado para criar a oferta. Todas as operações subsequentes nesta oferta usarão essa política. Se não estiver definida, a política *Global* será usada.

*name*: o nome para designar a nova instância da oferta (seu atributo *uacName*).

*folderID*: o ID da pasta de Oferta em que a oferta é criada. Este ID é validado para exatidão e uma exceção será emitida se o ID for inválido.

*templateName*: nome obrigatório (exclusivo) de um Modelo de oferta existente que deve ser usado para a nova Oferta.

*wsAttributes*: uma matriz de atributos de inicialização; quaisquer atributos fornecidos sobrescrevem os valores padrão da oferta; outros são deixados inalterados. Por exemplo, se um atributo *uacOfferCode* for fornecido, ele será usado em vez de um gerado automaticamente. Cabe ao cliente determinar os atributos que são obrigatórios para a oferta, seus tipos, etc.

Lança *CampaignServicesException* se ocorrer uma das condições a seguir:

- O parâmetro *folderID* é inválido (não existente ou não do tipo de oferta).
- O usuário não está autorizado a executar essa operação.
- Os atributos inválidos são fornecidos em *wsAttributes*.
- Ocorrem outras exceções de tempo de execução.

## Devoluções

Uma instância única de *OfferInfo* para a oferta criada.

## Erros

*CampaignServicesException*

## retireOffers

```
void retireOffers(String userCredential, String partitionName,  
                 Locale requestedLocale, WSReference[] references)  
    throws CampaignServicesException;
```

Retira uma ou mais ofertas existentes.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*references*: uma matriz de referências das ofertas a serem obsoletadas.

*InvalidComponentException* é lançada se houver um problema com uma determinada referência ou uma oferta não existir.

## Devoluções

Nenhum.

## Erros

*InvalidComponentException*

*AuthorizationException*, *DataException*

## deleteOffers

```
void deleteOffers(String userCredential, String partitionName,  
                 Locale requestedLocale, WSReference[] references)  
    throws CampaignServicesException;
```

Exclui uma ou mais ofertas existentes.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*reference*: uma matriz de referências das ofertas a serem excluídas.

*InvalidComponentException* é emitido, se ocorrer um problema em uma referência especificada ou uma oferta não existir.

## Devoluções

Nenhum.

## Erros

*InvalidComponentException*

*AuthorizationException*, *DataException*

## deleteOffersAndLists

```
public WSDeleteOffersAndListsResponse deleteOffersAndLists  
(String userCredential, String partitionName, Locale requestedLocale,  
WSReference[] offers)  
throws CampaignServicesException
```

Exclui as ofertas e as listas especificadas.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*offers*: matriz de referência da oferta ou da lista de ofertas.

## Devoluções

Retorna os objetos do tipo *WSGetOfferListMembersResponse*.

## Erros

Lança *CampaignServicesException*, se o ID de oferta ou o ID da lista de ofertas for inválido.

## listOfferTemplates

```
List<WSOfferTemplateInfo>  
listOfferTemplates(String userCredential, String partitionName,  
Locale requestedLocale)  
throws CampaignServicesException;
```

Liste todos os modelos de oferta que o usuário tem permissão para visualizar.

Depois de recuperada, cada instância *WSOfferTemplateInfo* que é retornada pode ser usada no estado em que se encontra, ou uma ou mais APIs do atributo podem ser usadas para buscar ou atualizar qualquer modelo listado.

### Parâmetro

*userCredential*: a credencial de usuário do cliente.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*partitionName*: nome opcional da partição do Campaign a ser usado.

### Devoluções

Uma Lista digitada de zero ou mais instâncias do wrapper de dados *WSOfferTemplateInfo*, um para cada modelo retornado.

### Erros

*InvalidExecutionContextException*, *AuthorizationException*

*DataException*

## createTemplate

```
createTemplate(String userCredential, String partitionName, Locale requestedLocale,  
String name, String securityPolicyName, WSAttribute[]  
wsStaticAttributes, WSAttribute[] wsHiddenAttributes,  
WSAttribute[] wsParametricAttributes)  
throws CampaignServicesException
```

Cria um novo modelo de oferta.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*name*: nome do modelo de oferta recém-criado.

*securityPolicyName*: nome da política de segurança a ser usada.

*wsStaticAttributes*: lista de atributos no modelo.

*wsHiddenAttributes*: lista de atributos ocultos no modelo.

*wsParametricAttributes*: lista de atributos parametrizados no modelo.

### Devoluções

Retorna o objeto do tipo *WSCreateTemplateResponse*.

### Erros

Lança *CampaignServicesException* se o modelo de oferta não existir ou a referência for inválida ou nenhuma referência for fornecida.

## getOfferTemplate

```
public WSGetOfferTemplateResponse getOfferTemplate(String userCredential,  
String partitionName, Locale requestedLocale, WSReference[] wsReferences)  
throws CampaignServicesException
```

Lista os detalhes dos modelos de oferta conforme especificado nas referências.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*wsCampaignReference*: referência da campanha pai.

### Devoluções

Retorna o objeto do tipo `WSGetOfferTemplateResponse`.

### Erros

Lança `CampaignServicesException` se o modelo de oferta não existir ou a referência for inválida ou nenhuma referência for fornecida.

## retireOfferTemplates

```
public WSGenerateOfferCodeResponse generateOfferCodes  
(String userCredential, String partitionName, Locale requestedLocale,  
String offerName, WSReference template)  
throws CampaignServicesException
```

Obsoleta um ou mais modelos de oferta especificados.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*wsCampaignReference*: referência da campanha pai.

### Devoluções

Retorna objetos do tipo `WSRetireOfferTemplatesResponse`.

### Erros

Lança `CampaignServicesException` se o modelo de oferta não existir ou a referência for inválida ou nenhuma referência for fornecida.

## getOffersAndListsByPage

```
public WSGetOffersAndListsByPageResponse getOffersAndListsByPage
(String userCredential, String partitionName, Locale requestedLocale,
WSComponentTypeEnum type,
int pageSize, int pageOffset)
throws CampaignServicesException
```

List offers or offerlists by page.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*type*: tipo que indica se as ofertas são solicitadas ou listas de ofertas.

*pageSize*: o número máximo de componentes correspondidos a serem retornados para a página.

*pageOffset*: a compensação inicial de todos os componentes possíveis para iniciar a enumeração (valor zero). Por exemplo, se a enumeração corresponder a 1000 ofertas e esse valor for configurado como 10, a página iniciaria no 11º componente. Uma RangeException é lançada se a compensação fornecida estiver fora do intervalo.

### Devoluções

Retorna objeto do tipo WSGetOffersAndListsByPageResponse.

### Erros

Nenhum.

## bulkCreateOffers

```
WSOfferInfoStatus[] bulkCreateOffers(String authorizationLoginName,
String partitionName, Locale requestedLocale,
String securityPolicyName, String templateName, long folderID,
WSBulkOfferInfo[] offers)
throws CampaignServicesException;
```

Cria ofertas em massa com os atributos para cada oferta que está especificada no parâmetro *offers*. Todas as ofertas são criadas sob o *folderID* especificado, usando o *templateName* especificado.

### Parâmetro

*authorizationLoginName*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*securityPolicyName*: nome opcional da política de segurança de campanha a ser usado para criar a oferta. Se não definido, a política global será usada.

*templateName*: Nome do modelo de oferta existente no sistema. Todas as ofertas são criadas, usando este modelo.

*folderID*: o ID da pasta Oferta em que as ofertas foram criadas. Este ID é validado e uma exceção será emitida, se o ID estiver inválido.

*offers*: uma matriz de objetos *WSBulkOfferInfo* que define o nome da oferta e os atributos. Consulte o tipo de dados *WSBulkOfferInfo* para obter mais detalhes.

### **Devoluções**

Uma matriz de instâncias *WSOfferInfoStatus* para cada oferta. Contém o status e informações sobre a oferta. O status indica se a criação da oferta foi bem-sucedida ou não.

### **Erros**

*CampaignServicesException*

## **getOfferListDetails**

```
public WSGetOfferListDetailsResponse getOfferListDetails(String userCredential,
String partitionName, Locale requestedLocale, WSReference listReference)
    throws CampaignServicesException {
```

Listar detalhes da lista de ofertas especificada.

### **Parâmetros**

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*listReference*: referência à lista de ofertas.

### **Devoluções**

Retorna o objeto do tipo *WSGetOfferListDetailsResponse*.

### **Erros**

Lança a *CampaignServicesException*, se as referências da lista forem inválidas.

## **getOfferListMembers**

```
public WSGetOfferListMembersResponse getOfferListMembers
(String userCredential, String partitionName, Locale requestedLocale,
WSReference listReference)
    throws CampaignServicesException {
```

Lista as informações de ofertas em uma lista de ofertas especificada.

### **Parâmetros**

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.



*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*listReference*: referência à lista de ofertas.

## Devoluções

Retorna o objeto do tipo `WSDeleteOffersAndListsResponse`.

## Erros

Lança `CampaignServicesException`, se o id offerlist estiver inválido.

## getOffersByQuery

```
public WSGetOffersByQueryResponse getOffersByQuery(String user_credential,
String partition_name, Locale locale, String query, Integer maxSize,
Boolean includeSubFolder, WSReference[] scopeFolders)
throws CampaignServicesException
```

Lista ofertas que correspondem às ofertas fornecidas.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*query*: consulta para localizar as ofertas. O formato de consulta é o mesmo que o usado na lista de ofertas inteligente.

*maxSize*: registros máximos a serem listados.

*includeSubFolder*: sinalizador que indica se uma subpasta deve ser incluída na procura.

*scopeFolders*: lista de referências de pasta a ser procurada para ofertas.

## Devoluções

Retorna objeto do tipo `WSGetOffersByQueryResponse`.

## Erros

Lança `CampaignServicesException`, se as referências da pasta forem inválidas.

## retireOfferLists

```
public void retireOfferLists(String user_credential, String partition_name,
Locale locale, WSReference[] wsReferences)
throws CampaignServicesException
```

Obsoleta uma ou mais listas de ofertas especificadas.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*wsReferences*: referência a listas de ofertas.

## Devoluções

Nenhum.

## Erros

Lança a CampaignServicesException, se as referências da lista forem inválidas.

## createFolder

```
public WSCreateFolderResponse createFolder(String userCredential,  
String partitionName, Locale requestedLocale,String name,  
String description,String securityPolicyName,  
long parentFolderId,WSFolderTypeEnum folderType,  
WSApplicationTypeEnum createdBy,long creatorObjectId)  
throws CampaignServicesException
```

Cria uma nova pasta do tipo campaign/offer/sessions/segments.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*name*: nome da pasta a ser criada.

*description*: descrição para a nova pasta.

*securityPolicyName*: nome da política de segurança a ser usada.

*parentFolderId*: (opcional) ID da pasta-pai. Se não fornecido, a pasta será criada na raiz.

*folderType*: tipo de pasta-Offer/session/campaign/segment.

*createdBy*: (opcional) objeto do tipo WSApplicationTypeEnum indica qual aplicativo criou a pasta. Os valores possíveis são Campaign, Plan, Collaborate e eMessage. Se não for indicado, o Campaign será usado.

**Nota:** Plan=IBM Marketing Operations. Collaborate=IBM Distributed Marketing.

*creatorObjectId*: (opcional) usado pelo Plano (IBM Marketing Operations) para vincular uma pasta no IBM Campaign com uma pasta no Marketing Operations.

## Devoluções

Retorna o objeto do tipo `WSCreateFolderResponse`.

## Erros

Emite `CampaignServicesException` se o tipo de pasta for inválido ou duplicado.

## editFolder

```
public WSEditFolderResponse editFolder(String userCredential,  
String partitionName, Locale requestedLocale, long folderId,  
String name,String description, WSFolderTypeEnum folderType,  
Long creatorObjectId, boolean clearExisting)  
    throws CampaignServicesException
```

Atualiza a pasta especificada.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*id*: id da pasta a ser atualizada.

*name*: nome da pasta a ser criada.

*description*: descrição para a nova pasta.

*folderType*: tipo de pasta-Offer/session/campaign/segment.

*creatorObjectId*: (opcional) usado pelo Plano (IBM Marketing Operations) para vincular uma pasta no IBM Campaign com uma pasta no Marketing Operations.

*clearExisting*: não usado agora.

## Devoluções

Retorna o objeto do tipo `WSEditFolderResponse`.

## Erros

Lança `CampaignServicesException`, se o tipo de pasta for inválido ou se o ID da pasta estiver inválido.

## getSubFoldersList

```
public WSGetSubFolderListResponse getSubFoldersList(String user_credential,  
String partition_name, Locale locale, WSReference parentFolder,  
WSFolderTypeEnum folderType)  
    throws CampaignServicesException
```

Lista todas as subpastas na pasta especificada.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*parentFolder*: referência de pasta à qual todas as subpastas são solicitadas.

*folderType*: tipo de pasta.

## Devoluções

Retorna o objeto do tipo `WSGetSubFolderListResponse`.

## Erros

Lança `CampaignServicesException` se a referência de pasta estiver inválida.

## moveFolders

```
public WSMoveFolderResponse moveFolders(String userCredential,  
String partitionName, Locale requestedLocale, Long[] folderId,  
long parentFolder, long destinationFolder,  
WSFolderTypeEnum folderType)  
throws CampaignServicesException
```

Move a pasta especificada para outra pasta-pai.

## Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*folderId*: ID da pasta a ser excluída.

*parentFolder*: ID da pasta pai.

*destinationFolder*: id da pasta de destino em que a pasta especificada é movida.

*folderType*: tipo de pasta-Offer/session/campaign/segment.

## Devoluções

Retorna os objetos do tipo `WSMoveFolderResponse`.

## Erros

Lança `CampaignServicesException`, se o tipo de pasta ou ID da pasta-pai for inválido ou se o ID da pasta for inválido.

## deleteFolders

```
public WSDeleteFolderResponse deleteFolders(String userCredential,  
    String partitionName, Locale requestedLocale, Long[] folderId,  
    long parentFolder, boolean deleteChilds,  
    WSFolderTypeEnum folderType)  
    throws CampaignServicesException
```

Exclui as pastas especificadas no sistema juntamente com todos os itens na pasta.

### Parâmetros

*userCredential*: a credencial de usuário do cliente.

*partitionName*: nome opcional da partição do Campaign a ser usado.

*requestedLocale*: código de idioma opcional a ser usado para essa solicitação.

*folderId*: ID da pasta a ser excluída.

*parentFolder*: ID da pasta pai.

*deleteChilds*: sinalizador indica se todas as dependências da pasta devem ser excluídas. Se falso, nenhuma dependência será excluída.

*folderType*: tipo de pasta-Offer/session/campaign/segment.

### Devoluções

Retorna objeto do tipo WSDeleteFolderResponse.

### Erros

Lança CampaignServicesException, se o tipo de pasta for inválido ou se o ID da pasta for inválido.



---

## Capítulo 5. Exceções comuns de API SOAP

A API SOAP do IBM Campaign poderá lançar as exceções comuns a seguir. Todas as mensagens localizadas de exceção ficarão no código de idioma solicitado se disponíveis para IBM Campaign. As políticas de fallback de código de idioma Java usual se aplicam.

### **RemoteException**

Este item se aplica somente à interface SOAP.

Todas as chamadas SOAP para a API podem emitir uma RemoteException, se for encontrado um erro de nível de sistema, como um problema na camada de processamento do envelope SOAP (Axis), uma restrição que é definida no WSDL de serviço da web foi violada por algum motivo.

As exceções de PI rotineiras marcadas e desmarcadas, como DataException, são retornadas como um status de erro, não como uma RemoteException.

Consulte a seção de interface SOAP para obter detalhes.

### **AuthenticationException**

O usuário não pôde ser autenticado para a partição Campaign especificada. Verifique a função de usuário que está definida em IBM Marketing Platform.

### **Exceção de autorização**

O usuário não está autorizado a executar a operação solicitada. Esta exceção pode ser emitida por qualquer método de API, portanto é não declarada (desmarcada). Verifique as permissões que estão designadas à função do usuário no IBM Marketing Platform.

### **Exceção de dados**

Ocorreu uma exceção fatal na camada de banco de dados subjacente em Campaign (desmarcado).

Consulte o fluxograma de IBM Campaign e os logs do listener para obter detalhes.

### **Exceção de bloqueio**

Uma exceção provisória que é emitida quando o cliente tenta atualizar um componente, como um fluxograma, enquanto estiver sendo editado por outro usuário. Geralmente, essa exceção pode ser recuperada esperando e, em seguida, recuperando a operação. A lógica de nova tentativa é a responsabilidade do cliente.

### **InvalidComponentException**

Foi feita uma tentativa de referenciar um componente inválido ou desconhecido (campanha, fluxograma, célula de destino). O método `getComponentReference()` de exceções retorna a referência do componente ofensivo.

## **InvalidAttributeException**

Uma exceção que é emitida quando o cliente fornece ou referencia um atributo inválido, como usasse o tipo de dados incorreto ou usasse uma matriz de valores em que nenhum é permitido. O método `getAttributeName()` da exceção retorna o nome do atributo do problema, `getAttributeValue()` retorna o valor e `getComponentReference()` identifica o componente (ou índice em massa).

## **AttributeExistsException**

Emitido quando o cliente tenta definir um metadados de atributo duplicado para um componente. O método `getAttributeName()` da exceção retorna o nome do atributo duplicado; `getComponentReference()` identifica o componente (ou índice em massa).

## **AttributeNotFoundException**

Emitido sempre que o cliente tenta referenciar um atributo desconhecido (campanha, fluxograma, célula de destino, etc.). O método `getAttributeName()` da exceção retorna o nome do atributo não correspondido; `getComponentReference()` identifica o componente (ou índice em massa).

## **CompositeException**

Um `CompositeException` é usado por algumas APIs para relatar múltiplos erros de volta para o responsável pela chamada. Geralmente possui mais de uma causa ligada a ela; todas as causas são capturadas como uma lista na ordem em que ocorreram. O método `getCauseList()` da exceção retorna esta lista, que pode ser inspecionada melhor para obter detalhes de cada erro.

**Nota:** Geralmente, a API é concluída com êxito ou retrocede seu trabalho antes de emitir uma exceção composta. Consulte, por exemplo, as APIs de planilha de células de destino em massa conforme descrito em “Métodos de API SOAP: Células de destino” na página 31.



---

## Antes de contatar o suporte técnico do IBM

Se você encontrar um problema que não pode resolver consultando a documentação, o contato responsável por suporte de sua empresa poderá registrar uma chamada no suporte técnico da IBM. Siga estas diretrizes para assegurar que seu problema seja resolvido de modo eficiente e bem-sucedido.

Se você não possuir um contato responsável por suporte em sua empresa, entre em contato com o administrador do IBM para obter mais informações.

**Nota:** O Suporte Técnico não grava ou cria scripts de API. Para obter assistência para implementar nossas ofertas de API, entre em contato com os Serviços profissionais da IBM.

### Informações para reunir

Antes de contatar o suporte técnico IBM, reúna as informações a seguir:

- Uma breve descrição da natureza de seu problema.
- Mensagens de erro detalhadas que você vê quando ocorre o problema.
- Etapas detalhadas para reproduzir o problema.
- Arquivos de log relacionados, arquivos de sessão, arquivos de configuração e arquivos de dados.
- Informações sobre seu produto e ambiente do sistema , que você pode obter conforme descrito em "Informações do sistema".

### Informações sobre o sistema

Ao ligar para o suporte técnico da IBM, pode ser solicitado que forneça informações sobre o seu ambiente.

Se seu problema não impedir que você efetue login, muitas destas informações estão disponíveis na página Sobre, que fornece informações sobre seus aplicativos do IBM instalados.

É possível acessar a página Sobre selecionando **Ajuda > Sobre**. Se a página Sobre não estiver acessível, verifique o arquivo `version.txt` que se localiza no diretório de instalação do aplicativo.

### Informações de contato para o suporte técnico IBM

Para obter maneiras de contatar o suporte técnico IBM, consulte o website de Suporte Técnico do Produto IBM: ([http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)).

**Nota:** Para inserir uma solicitação de suporte, deve-se efetuar login em uma conta do IBM. Essa conta deve estar vinculada ao seu número de cliente IBM. Para saber mais sobre como associar sua conta ao seu número de cliente do IBM, consulte **Recursos de suporte > Suporte de software autorizado** no Portal de suporte.



---

## Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante IBM local para obter informações sobre os produtos e serviços atualmente disponíveis em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM ou quaisquer outros direitos da IBM poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, é responsabilidade do usuário avaliar e verificar a operação de qualquer produto, programa ou serviço não IBM.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos descritos nesta publicação. O fornecimento desta publicação não lhe garante direito algum sobre tais patentes. Pedidos de licenças devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138-146  
Botafogo  
Rio de Janeiro  
CEP: 22290-240

Para pedidos de licenças com relação a informações sobre DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos, por escrito, para:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições da publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a Web sites não-IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a estes Web sites. Os materiais contidos nesses Web sites não fazem partes dos materiais deste produto IBM e a utilização desses Web sites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138-146  
Botafogo  
Rio de Janeiro  
CEP: 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriados, incluindo em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que tais medidas sejam iguais em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Os resultados reais podem ser diferentes. Os usuários desta publicação devem verificar os dados aplicáveis para seu ambiente específico.

As informações referentes a produtos não IBM foram obtidas junto aos fornecedores desses produtos, anúncios publicados ou outras fontes de publicidade disponíveis. A IBM não testou aqueles produtos e não pode confirmar a precisão do desempenho, compatibilidade ou qualquer outra reivindicação relacionada a produtos que não sejam da IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a mudanças ou cancelamento sem aviso prévio e representam apenas metas e objetivos.

Todos os preços da IBM mostrados, são preços de varejo sugeridos pela IBM, são atuais e estão sujeitos à mudança sem aviso. Os preços para o revendedor podem variar.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos

incluem nomes de pessoas, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

#### LICENÇA DE COPYRIGHT:

Estas informações contêm programas aplicativos de amostra no idioma de origem, que ilustram técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra de qualquer forma, sem a necessidade de pagar a IBM, para os propósitos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativos para a plataforma operacional para a qual os programas de amostra são escritos. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar confiabilidade, manutenção ou função destes programas. Os programas de amostra são fornecidos "NO ESTADO EM QUE SE ENCONTRAM", sem garantia de nenhum tipo. A IBM não se responsabilizará por nenhum dano derivado do uso dos programas de exemplo.

Se estas informações estiverem sendo exibidas em cópia eletrônica, as fotografias e ilustrações coloridas podem não aparecer.

---

## Marcas Comerciais

IBM, o logotipo IBM e [ibm.com](http://ibm.com) são marcas ou marcas comerciais da International Business Machines Corp., registradas em muitos países no mundo todo. Outros nomes de serviços e produtos podem ser marcas comerciais da IBM ou de outras empresas. Uma lista atual de marcas comerciais da IBM está disponível na Web em "Copyright and trademark information" em [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Considerações sobre a Política de Privacidade e os Termos de Uso

Produtos de Software IBM, incluindo soluções de software como serviço, ("Ofertas de Software") podem usar cookies ou outras tecnologias para coletar informações de uso do produto, para ajudar a melhorar a experiência do usuário final, para customizar interações com o usuário final ou para outros propósitos. Um cookie é uma parte dos dados que o website pode enviar para o seu navegador, e que pode ser armazenada em seu computador como uma tag que identifica o seu computador. Em muitos casos, nenhuma informação pessoal é coletada por esses cookies. Se uma Oferta de Software que você está usando permitir a você coletar informações pessoais por meio de cookies e tecnologias semelhantes, informamos a você sobre as especificações abaixo.

Dependendo das configurações implementadas, esta Oferta de Software pode usar cookies de sessão e persistentes que coletam o nome de usuário de cada usuário e outras informações pessoais com propósitos de gerenciamento de sessão, usabilidade de usuário aprimorada ou outros propósitos de rastreamento de uso ou funcionais. Esses cookies podem ser desativados, mas a desativação também eliminará a funcionalidade ativada por eles.

Várias jurisdições regulamentam a coleta de informações pessoais por meio de cookies e tecnologias semelhantes. Se as configurações implementadas para esta Oferta de Software fornecer ao Cliente a capacidade de coletar informações pessoais dos usuários finais usando cookies e outras tecnologias, você deverá procurar o aviso legal sobre qualquer lei aplicável a essa coleta de dados, incluindo todos os requisitos para fornecer avisos e consentimento quando adequado.

A IBM requer que os Clientes (1) forneçam um link claro e evidente para os termos de uso do website do Cliente (por exemplo, política de privacidade) que inclui um link para a coleção de dados da IBM e do Cliente e práticas de uso, (2) notifiquem que cookies e gifs/web beacons claros estão sendo colocados no computador do visitante pela IBM em nome do Cliente juntamente com uma explicação do propósito de tal tecnologia, e (3) até a extensão requerida por lei, obtenham o consentimento dos visitantes do website antes de se colocar cookies e gifs/web beacons claros pelo Client ou IBM em nome do Cliente nos dispositivos do visitante do website

Para obter mais informações sobre o uso de várias tecnologias, incluindo cookies, para estes propósitos, consulte Declaração de Privacidade Online da IBM em: <http://www.ibm.com/privacy/details/us/en> na seção intitulada "Cookies, Web Beacons e Outras Tecnologias".





Impresso no Brasil