

IBM Cúram Social Program Management
Versión 6.0.5

*Utilización del motor de correlación de
datos*

IBM

Nota

Antes de utilizar esta información y el producto al que hace referencia, lea la información que figura en el apartado "Avisos" en la página 33

Revisado: marzo de 2014

Esta edición se aplica a IBM Cúram Social Program Management v6.0.5 y a todos los releases posteriores mientras no se indique lo contrario en nuevas ediciones.

Materiales bajo licencia - Propiedad de IBM.

© Copyright IBM Corporation 2012, 2013.

© Cúram Software Limited. 2011. Reservados todos los derechos.

Contenido

Figuras v

Tablas vii

Desarrollo con el motor de correlación de datos 1

Introducción	1
Finalidad	1
Público al que va dirigida	1
Requisitos previos	1
Capítulos de esta guía	2
Conceptos básicos sobre la correlación de datos.	3
Introducción	3
Conceptos básicos sobre cómo se almacenan los datos en el CDS	3
Creación de correlaciones lógicas	4
Consideración de los problemas de validación en la correlación de datos	4
Cómo escribir especificaciones y configuraciones de correlación para pruebas estáticas y PDF	5
Introducción	5
Cómo escribir especificaciones de correlación	5
Especificación de correlación simple	5
Correlación de expresiones de condición	6
Correlación de valores de tablas de códigos	6
Correlación con varias entidades de destino	6
Correspondencia de una entidad padre con varias entidades hijas.	7
Correspondencia de patrones y seguimiento de asociaciones en el CDS	7
Correlación de miembros de una unidad familiar	9
Cómo escribir configuraciones de correlación	9
Para el generador de solicitudes de pruebas	10
Configuración de correlación simple	11
Gestión de campos caseParticipantDetails	11
Establecimiento de identificadores de entidades de destino	13
Para el generador de solicitudes en PDF.	14
Secciones y campos.	14
Cómo llenar campos de texto	15

Secciones repetidas y descripciones de tablas de códigos.	16
Recuadros de selección	16
Botones de selección	16
Cuadros combinados de selección	17
Cómo configurar el formulario de solicitud en PDF	17
Cómo escribir especificaciones y configuraciones de correlación para pruebas dinámicas	18
Introducción	18
Cómo escribir especificaciones y configuraciones de correlación para pruebas dinámicas	18
Metadatos de pruebas dinámicas simples	18
Especificación de correlación simple	19
Configuración de correlación simple	19
Correlación de pruebas dinámicas padre-hijo	20
Metadatos de pruebas dinámicas padre-hijo simples	20
Especificación de correlación padre-hijo simple	21
Configuración de correlación simple para relación padre-hijo	22
Correlación con terceros	22
Introducción	22
Cómo correlacionar terceros	23
Definición del creador de participantes	23
Crear participante	23
Esquema de correlación de ejemplo	23
Ejemplo de configuración de correlación.	25
Esquema para especificaciones de correlación	26
Esquema	26
Esquema para configuraciones de correlación	28
Esquema	28
Registro cronológico de errores y diagnóstico	31
Códigos de error	31

Avisos 33

Consideraciones sobre la política de privacidad	35
Información sobre interfaces de programación.	36
Marcas registradas	36

Figuras

- | | | | |
|---|---|--|----|
| 1. Ejemplo de estructura de datos en el almacén de datos de Cúram | 4 | 2. Ingresos de trabajo en CDS. | 8 |
| | | 3. Uso de datos en CDS | 10 |

Tablas

1.	Requisitos previos para el motor de correlación de datos de Cúram	1
2.	Ejemplo de formulario PDF	8
3.	Miembros de la unidad familiar	9
4.	Campos de un formulario en PDF para registrar miembros de la unidad familiar.	16

Desarrollo con el motor de correlación de datos

La correlación de datos es el proceso mediante el cual los datos del almacén de datos de Cúram se correlacionan con las entidades de prueba de Cúram. El motor de correlaciones de datos de Cúram correlaciona los datos utilizando correlaciones configuradas en el editor de correlaciones de datos de Cúram. Las correlaciones de un programa se crean en una solicitud de admisión.

Introducción

Finalidad

La finalidad de esta guía es describir cómo utilizar el motor de correlación de datos de Cúram (CDME) para correlacionar entre sí datos de ciudadanos capturados durante el proceso de admisión y examen:

- Entidades de pruebas y no de pruebas dentro de casos de Cúram, o
- Formularios de solicitud cumplimentados en formato PDF

Ambas opciones agilizan la solicitud del ciudadano para los programas de prestaciones. La correlación de datos con pruebas de casos permite que los datos se utilicen para determinar la elegibilidad como parte del proceso de casos de Cúram. La correlación de datos con formularios de solicitud en PDF acelera el proceso de cumplimentación de estos formularios para el envío manual.

Esta guía debe utilizarse junto con la publicación Guía del editor de correlaciones de datos de IBM Cúram. El editor de correlaciones de datos es la herramienta más sencilla para crear rápidamente correlaciones con las pruebas. El editor de correlaciones de datos guarda estas correlaciones en un lenguaje de correlación XML. Esta guía describe los detalles del lenguaje de correlación XML que resultan útiles para comprender cómo se ejecutan las correlaciones existentes, para entender correlaciones más avanzadas, para crear correlaciones con formularios PDF y para mantener correlaciones antiguas que posiblemente no sean compatibles con el editor de correlaciones de datos.

Público al que va dirigida

La correlación de datos es un proyecto colaborativo entre analistas empresariales y desarrolladores. El rol de los analistas empresariales consiste en correlacionar lógicamente datos de la ciudadanía con campos de un formulario de solicitud PDF o con entidades de pruebas. El rol de los desarrolladores consiste en traducir los esfuerzos de los analistas empresariales escribiendo especificaciones de correlación y configuraciones de correlación en XML.

Requisitos previos

En la tabla siguiente se enumeran los requisitos previos para el motor de correlación de datos y se proporcionan lecturas recomendadas para obtener conocimientos acerca de estos requisitos previos:

Tabla 1. Requisitos previos para el motor de correlación de datos de Cúram

Requisito previo	Lectura recomendada	Público objetivo
XML	http://www.w3.org/XML/	Desarrolladores

Tabla 1. Requisitos previos para el motor de correlación de datos de Cúram (continuación)

Requisito previo	Lectura recomendada	Público objetivo
Casos de Cúram	Descripciones de entidades de casos en el Modelo de referencia básico; Guía de gestión de casos integrados de Cúram	Analistas empresariales
	Inside Eligibility and Entitlement Using Cúram Express Rules (Comprensión de la elegibilidad y la titularidad con Cúram Express Rules)	Desarrolladores
Participantes de Cúram	Descripciones de entidades de participantes en el Modelo de referencia básico; Guía para participantes de Cúram	Analistas empresariales
Almacén de datos común (CDS)	Creación de esquemas de almacén de datos	Desarrolladores
IBM Cúram Universal Access	Guía de personalización de Universal Access	Desarrolladores
Pruebas de Cúram	Descripciones de entidades de pruebas en el Modelo de referencia básico; Guía de pruebas de Cúram	Analistas empresariales
	Diseño de soluciones de pruebas de Cúram	Desarrolladores
Formularios en PDF	http://www.adobe.com/products/acrobat/?promoid=BPDDU	Desarrolladores

Capítulos de esta guía

Esta guía incluye los siguientes capítulos:

Capítulo 2: Conceptos básicos sobre la correlación de datos

En este capítulo se describe cómo el motor de correlación de datos convierte datos de ciudadanos de Cúram (capturados en el portal de acceso universal y almacenados en el almacén de Cúram) en campos de formularios de solicitud en PDF o entidades de pruebas de casos.

Capítulo 3: Cómo escribir especificaciones y configuraciones de correlación para pruebas estáticas y PDF

En este capítulo se describe cómo escribir especificaciones de correlación y configuraciones de correlación en XML para pruebas estáticas y pdf. Las especificaciones de correlación correlacionan datos de un formato a otro, por ejemplo, de las entidades de la estructura de CDS a entidades de la base de datos. Las configuraciones de correlación describen cómo llenar solicitudes en PDF o convertir datos en entidades de pruebas.

Capítulo 4: Cómo escribir especificaciones y configuraciones de correlación para pruebas dinámicas

En este capítulo se describe cómo escribir especificaciones de correlación y configuraciones de correlación en XML para pruebas dinámicas.

Capítulo 5: Correlación con terceros

En este capítulo se describe cómo abordar la correlación de terceros como participantes de un caso con casos.

Apéndice A: Esquema para especificaciones de correlación

Este apéndice define la gramática que se debe seguir al escribir especificaciones de correlación en XML.

Apéndice B: Esquema para configuraciones de correlación

Este apéndice define la gramática que se debe seguir al escribir configuraciones de correlación en XML.

Apéndice C: Conformidad

En este apéndice se describe cómo desarrollar de forma compatible.

Conceptos básicos sobre la correlación de datos

Introducción

El motor de correlación de datos de Cúram (CDME) funciona con el portal de acceso universal de Cúram para ayudar a los ciudadanos a examinarse a sí mismos para saber si pueden obtener prestaciones. Durante el proceso de examen y admisión, un ciudadano presentará su información. CDME se encarga de convertir la información del ciudadano en un formulario de solicitud en PDF (cumplimentados con detalles del ciudadano) o en un caso de Cúram con entidades de pruebas nuevas.

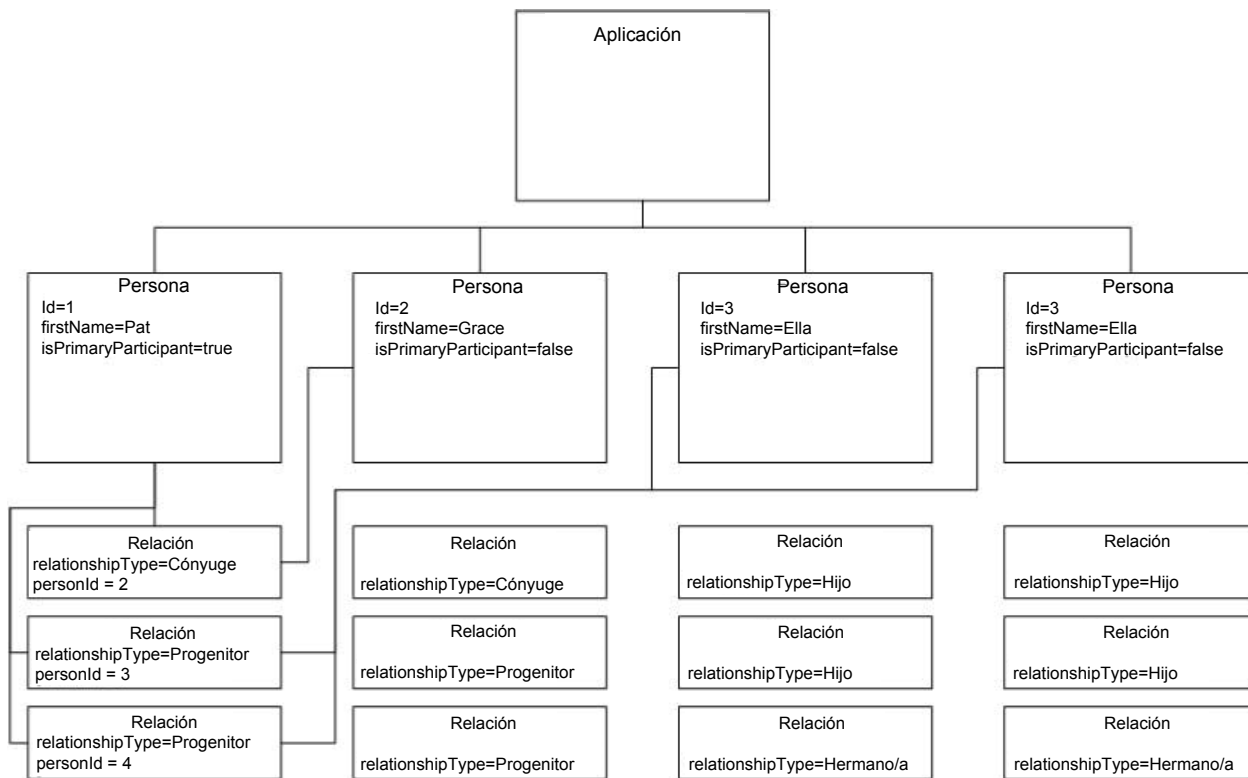
El CDME funciona de la siguiente manera. Cuando el ciudadano envía su información, dicha información se almacena en el almacén de datos de Cúram (CDS). El CDME lee los datos y utiliza reglas de una especificación de correlación para transformar los datos en información legible para un generador de solicitudes. Un generador de solicitudes en PDF utiliza la configuración de correlación para determinar cómo aparecerán los datos del ciudadano en un formulario de solicitud en PDF. Un generador de solicitudes en PDF utiliza una configuración de correlación para llamar a la API de pruebas para crear las entidades de pruebas nuevas para el nuevo caso.

Conceptos básicos sobre cómo se almacenan los datos en el CDS

Al correlacionar datos de ciudadanos con pruebas de casos o formularios de solicitud en PDF, la primera tarea consiste en examinar la salida deseada para los datos del ciudadano, es decir, como pruebas de un caso o como un formulario de solicitud cumplimentado. El paso siguiente consiste en examinar cómo se almacenan los datos de un ciudadano en el CDS como parte del proceso de admisión y examen. Cuando conozca la información necesaria en pruebas de un caso o formularios de solicitud en PDF y la información almacenada en el CDS, puede crear correlaciones lógicas entre los formatos de datos.

La finalidad del proceso de examen y admisión consiste en ayudar a los ciudadanos a solicitar prestaciones. Como mínimo, un formulario de solicitud o pruebas de un caso requerirán los nombres de los miembros de una familia y sus relaciones con la persona que solicita los beneficios. Los scripts IEG capturan esta información sobre un ciudadano y la almacenan en el CDS según un esquema predefinido. “Conceptos básicos sobre cómo se almacenan los datos en el CDS” muestra un ejemplo de una estructura de datos en el CDS para una sola solicitud

de prestaciones e incluye los miembros de una familia y sus relaciones.



No se muestran todas las relaciones

Figura 1. Ejemplo de estructura de datos en el almacén de datos de Cúram

Creación de correlaciones lógicas

Una correlación lógica desglosa información acerca de una persona almacenada en CDS en las entidades de pruebas del caso: miembro de la unidad familiar, acuerdo de convivencia y discapacidad.

Parte de la creación de una correlación lógica consiste en reconocer reglas empresariales que pueden afectar al modo en que CDME correlaciona los datos. Por ejemplo, si el ciudadano indica que es ciego y discapacitado, las reglas empresariales indican que deben crearse dos registros de pruebas de discapacidad para el ciudadano (uno para la ceguera y uno para la discapacidad).

Consideración de los problemas de validación en la correlación de datos

Como parte del proceso de admisión y examen, es necesario lograr un equilibrio entre las pruebas de validación para que se puedan insertar razonablemente a la vez que se garantiza que no se hace preguntas innecesarias al ciudadano. Un enfoque para considerar los problemas de validación en la correlación de datos es garantizar que durante la admisión, las pruebas se creen con validaciones mínimas, valores predeterminados o temporales que se puedan actualizar más adelante.

Cómo escribir especificaciones y configuraciones de correlación para pruebas estáticas y PDF

Introducción

Los desarrolladores pueden utilizar la correlación de datos lógica como especificación de requisitos para escribir especificaciones de correlación y configuraciones de correlación. Una especificación de correlación describe cómo correlacionar datos almacenados en una estructura determinada con una de diferente. Cada especificación de correlación hace referencia a un origen, de donde proceden los datos, y un destino, a donde se dirigen los datos.

Aunque la especificación de correlación contiene las reglas necesarias para transformar los datos de un formato a otro, se necesita más información para convertir los datos transformados en pruebas de caso o formularios de solicitud en PDF cumplimentados. Esta información adicional se proporciona en la configuración de correlación. Este capítulo proporciona ejemplos sobre cómo escribir especificaciones de correlación y configuraciones de correlación.

Cómo escribir especificaciones de correlación

El CDME utiliza especificaciones de correlación para transformar datos del CDS en otro formato. Todos los datos contenidos en la solicitud de admisión y examen de un ciudadano se pueden recuperar leyendo la entidad de solicitud, los datos contenidos en sus hijos, los hijos de sus hijos, y así sucesivamente. El motor de correlación de datos recorre los datos del CDS y aplica reglas expresadas en XML para completar la transformación de los datos.

Especificación de correlación simple

Esta especificación de correlación simple correlaciona una entidad de persona del CDS con una entidad de pruebas de miembro de la unidad familiar:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <map xmlns="http://www.curamsoftware.com/schemas/GUMBO/Map"
3   name="TestMapping">
4   <map-entity source="Person">
5     <target-entity name="HouseholdMember"
6       id="HouseholdMemberTarget">
7       <map-attribute from="isNativeAmerican"
8         to="natAlaskOrAmerInd"/>
9       <map-attribute from="comments" to="comments"/>
10    </target-entity>
11  </map-entity>
12 </map>
```

La línea 4 indica el origen de la correlación, mientras que la línea 5 indica el destino. Esta regla se puede parafrasear como "Para cada entidad Person encontrada en el CDS, crear una entidad HouseholdMember correspondiente". El elemento <target-entity> contiene dos elementos <map-attribute> en las líneas 6 y 7.

El elemento <map-attribute> de la línea 6 indica que el atributo isNativeAmerican de la entidad Person está correlacionado con el atributo natAlaskOrAmerInd de la entidad HouseholdMember. Los atributos no se correlacionan a menos que haya un elemento <map-attribute> específico. Esta es la razón por la que la línea 6 indica que el atributo de comentarios de Person está correlacionado con el atributo de comentarios de HouseholdMember.

En algunos casos, es necesario especificar que una correlación sólo se produce en circunstancias concretas. Por ejemplo, una entidad HeadOfHousehold sólo debe crearse en el sistema de destino cuando la correlación encuentra una entidad Person en el CDS que tiene un indicador isPrimaryParticipant establecido en true. El ejemplo anterior puede ampliarse para incluir esta regla tal como se indica a continuación:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <map xmlns="http://www.curamsoftware.com/schemas/GUMBO/Map"
3   name="TestMapping">
4   <map-entity source="Person">
5     <target-entity name="HouseholdMember"
6       id="HouseholdMemberTarget">
7       <map-attribute from="isNativeAmerican"
8         to="natAlaskOrAmerInd"/>
9       <map-attribute from="comments" to="comments"/>
10    </target-entity>
11  </map-entity>
12  <condition expression="Person.isPrimaryParticipant==true">
13    <target-entity name="HeadOfHousehold"/>
14  </condition>
15 </map>
```

Correlación de expresiones de condición

En este ejemplo, los valores Yes, No y Unanswered se representan como valores de la tabla de códigos y se utilizan para registrar si la persona es un ciudadano estadounidense o no. El valor ITYN4001 corresponde a que el cliente responda "Sí" a esta pregunta. Observe el uso de ". Se debe a que los símbolos de comillas "" no pueden utilizarse directamente en XML. La sintaxis para correlacionar atributos condicionalmente es exactamente igual.

```
1 <condition expression="Person.isBlind==&quot;ITYN4001&quot;">
2   <target-entity
3     name="Disability"
4     id="BlindDisabilityTarget"
5   >
6     <set-attribute
7       name="disabilityType"
8       value="DT1"
9     />
10  </target-entity>
11 </condition>
```

Correlación de valores de tablas de códigos

En algunas correlaciones, los valores de la tabla de códigos registrados en el CDS pueden convertirse directamente en valores de tabla de códigos utilizados en el modelo de destino. En estos casos, se puede utilizar una sección al principio del script de correlación para especificar las correlaciones de tabla de códigos. Por ejemplo:

```
1 <map-code-table source-codetable="CITIZENSTATUS"
2   target-codetable="AlienStatus">
3   <map-value source="US1" target="AS4"/>
4   <map-value source="US2" target="AS1"/>
5 </map-code-table>
```

En el Ejemplo 5, los valores de la tabla de códigos CITIZENSTATUS se correlacionan con los valores de la tabla de códigos AlienStatus.

Correlación con varias entidades de destino

A veces es necesario crear un grupo de entidades de destino. Esto se suele hacer al crear un grupo de entidades de pruebas, una de los cuales es padre y las demás

son hijas de esa entidad de pruebas padre. Vea el ejemplo siguiente para ver cómo crear grupos de entidades de destino relacionadas.

```
1      <target-entities>
2  <target-entity
3      name="BusinessAsset" id="BusinessAssetTarget"
4      type="parent"
5  >
6      <map-attribute
7          from="resourceAmount"
8          to="amount"
9      />
10     <map-attribute
11         from="amountOwed"
12         to="amountOwed"
13     />
14 </target-entity>
15 <target-entity
16     name="Ownership" id="OwnershipTarget"
17     type="child"
18 >
19     <set-attribute
20         name="percentageOwned"
21         value="100.0"
22     />
23 </target-entity>
24 </target-entities>
```

En este ejemplo, se crean dos entidades. La entidad BusinessAsset es una prueba padre, mientras que la entidad Ownership es una prueba hija. Al modelar la correlación de este modo se garantiza que se respeten los patrones de entidades de pruebas padre/hijo correctos cuando el generador de solicitudes de pruebas cree las pruebas.

Correspondencia de una entidad padre con varias entidades hijas

En “Correlación con varias entidades de destino” en la página 6, se creó un grupo de entidades de destino en el que las entidades padre e hija estaban relacionadas entre sí. En algunos casos, es necesario crear sólo una entidad padre para el caso en su totalidad. Todas las entidades hijas siguientes están relacionadas con la misma entidad padre. A continuación se muestra un ejemplo de cómo hacerlo.

```
1 <target-entities>
2 <target-entity name="HholdMealsGroup" type="parent"
3     attachment="case" id="MealGroup">
4     <set-attribute name="groupName" value="sample"/>
5 </target-entity>
6 <target-entity name="MealGroupMember" type="child"
7     id="MealGroupMember">
8 </target-entity>
9 </target-entities>
```

En este ejemplo, se crea un HholdMealsGroup y un MealGroupMember la primera vez que se ejecuta la regla. Cada vez posterior que se ejecuta la regla, sólo se crea un MealGroupMember y se asocia con la misma entidad HholdMealsGroup.

Correspondencia de patrones y seguimiento de asociaciones en el CDS

En el escenario siguiente, los clientes han solicitado que el motor de correlación se utilice para rellenar un formulario PDF parecido al siguiente:

Tabla 2. Ejemplo de formulario PDF

Nombre	Nombre del empleador	Fecha de inicio	Retribución anual antes de impuestos
Pat	The Gingerman Bakery	2/1/2004	30000
Grace	Jarmin Pharmaceutical	3/1/2002	50000

Cada campo de este formulario PDF tiene una identidad exclusiva. Por ejemplo, el campo que contiene el nombre, Pat, se identifica como Job0.Name. El campo que contiene 30000 se identifica como Job0.Salary.

Tenga en cuenta cómo puede almacenarse la información de la admisión en el CDS:

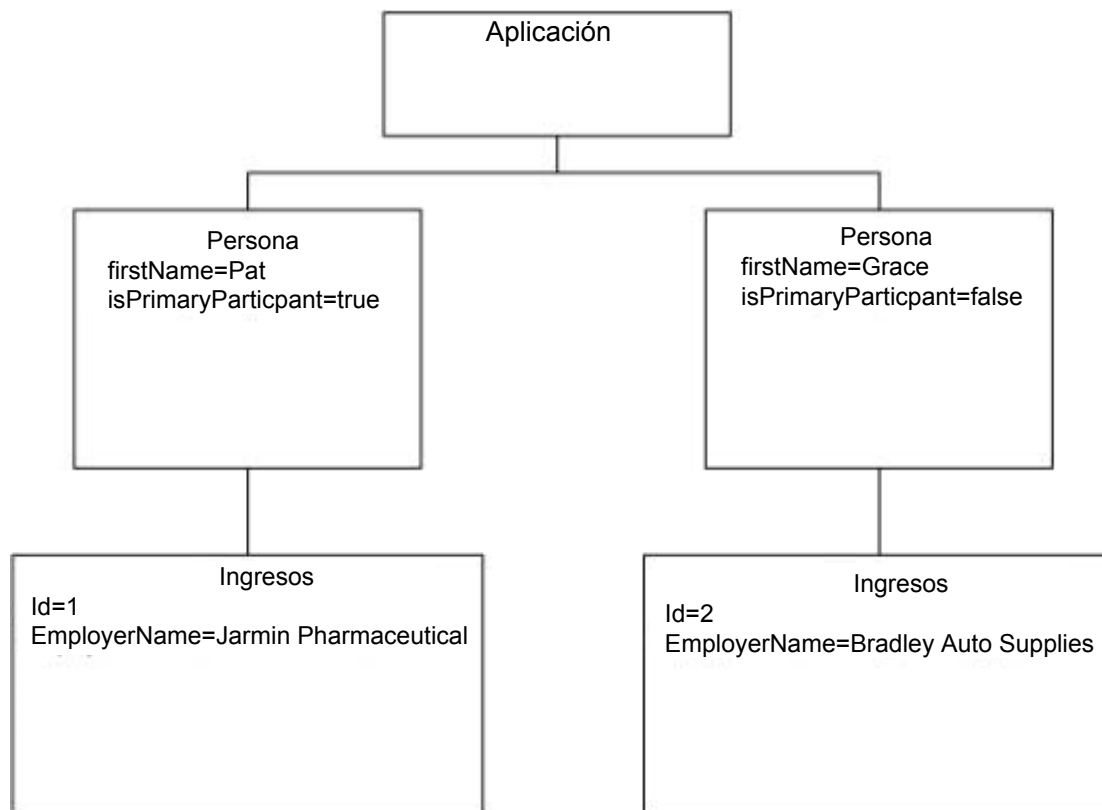


Figura 2. Ingresos de trabajo en CDS

Para rellenar el campo de nombre (Name) en el formulario PDF anterior, la especificación de correlación debe contener una regla que estipule que para cada ingreso (Income) perteneciente a una persona (Person), se debe proporcionar como salida el nombre (firstName) de la persona en el campo Name. En el lenguaje de correlación, esto puede expresarse tal como se indica a continuación:

```

1 <map-entity source="Person">
2 <map-entity source="Income">
3   <target-entity name="Job" id="JobTarget">
4     <map-attribute from="firstName" to="Name" entity="Person"/>
5     <map-attribute from="employerName" to="Employer"/>
  
```



```

6     ...
7   </target-entity>
8 </map-entity>
9 </map-entity>

```

Esta regla de correlación se puede parafrasear como "Para cada entidad Income contenida en una entidad Person, crear una entidad de destino de tipo Job". El atributo Name de la entidad Job se correlaciona desde el atributo firstName de la entidad Person que contiene la entidad Income que se está correlacionando."

Tenga en cuenta el uso de la sintaxis entity="Person" en la línea 4 para indicar que el atributo firstName procede de la entidad Person y no de la entidad Income. Un ejemplo más complejo de este tipo de especificación de correlación incluye las siguientes asociaciones o enlaces de una entidad a otra.

Correlación de miembros de una unidad familiar

La tabla siguiente muestra cómo se expresan generalmente las relaciones en un formulario de solicitud. El requisito es correlacionar las entidades de CDS con un formulario de solicitud previamente cumplimentado similar al que se muestra a continuación. La parte difícil en este caso consiste en rellenar el campo titulado "¿Qué relación tiene esta persona con usted?". Este campo se abrevia "RelType" en este ejemplo.

Tabla 3. Miembros de la unidad familiar

Nombre	¿Qué relación tiene esta persona con usted?	Fecha de nacimiento	Número de la seguridad social
Grace	Cónyuge	2/1/1981	209-57-9943
Ella	Hija	3/1/2002	987-23-1190

En este ejemplo, la correlación necesaria se escribe tal como se indica a continuación:

```

1 <condition expression="Person.isPrimaryParticipant == true">
2   <map-entity source="Person">
3     <map-entity source="Relationship">
4       <follow-association source="personID">
5         <target-entity name="Householder" id="Householder">
6           <map-attribute from="firstName" to="Name"/>
7           <map-attribute from="relationshipType" to="RelType"
8             entity="Relationship"/>
9         </target-entity>
10      </follow-association>
11    </map-entity>
12  </map-entity>
13 </condition>

```

Esto se puede parafrasear como "Para cada relación contenida en el participante primario, seguir la asociación con la persona a la que hace referencia dicha relación. Correlacionar el atributo firstName de esta entidad Person con el campo Name. Correlacionar el atributo relationshipType de la entidad Relationship con el campo RelType". La clave para entender el ejemplo está en la línea 7, donde el campo RelType se correlaciona desde un atributo de la entidad Relationship.

Cómo escribir configuraciones de correlación

Esta sección proporciona ejemplos de configuraciones de correlación para el generador de solicitudes de pruebas y el generador de solicitudes en PDF.

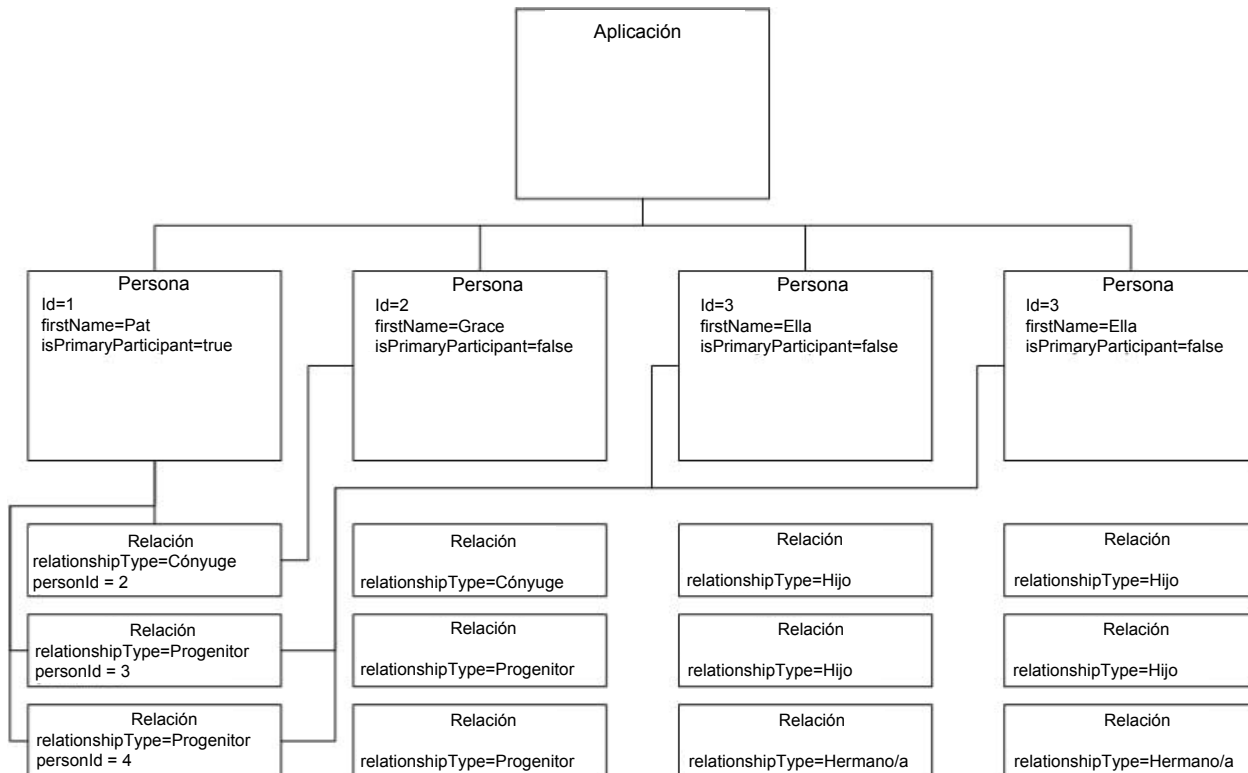
Para el generador de solicitudes de pruebas

Cuando se configura para utilizar el generador de solicitudes de pruebas, el trabajo del motor de correlación se divide en dos fases. En la primera fase, el motor de correlación crea un caso integrado que incluye miembros de caso y crea relaciones de rol de asunto entre los miembros de un caso. Los miembros del caso se llenan buscando entidades en el almacén de datos denominado "Person". El motor de correlación de datos está diseñado para tratar todas las entidades de almacén de datos común denominadas Person como referencia a una persona o candidato en un caso de Cúram.

Tomando "Para el generador de solicitudes de pruebas" como ejemplo, la fase 1 de la correlación del generador de solicitudes de pruebas creará un caso integrado con cuatro miembros de caso donde:

- Pat se denomina el participante primario.
- Se crea un par de registros de relación de rol de asunto para indicar que Grace y Pat están casados.
- El sistema crea también todas las demás relaciones de rol (para las relaciones parentales y de hermanos).
- También se crean registros de número de teléfono y dirección.

La fase 2 del proceso de correlación está relacionada con la creación de entidades de pruebas, de las que se incluyen muestras en el resto de esta sección.



No se muestran todas las relaciones

Figura 3. Uso de datos en CDS

Configuración de correlación simple

En el ejemplo siguiente se muestra una configuración para el generador de solicitudes de pruebas:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <application-builder-config
3   xmlns="http://www.curamsoftware.com/schemas/GUMBO/ApplicationBuilderConfig">
4   <evidence-config package="curam.evidence">
5     <entity name="HouseholdMember"/>
6     <entity name="HeadOfHousehold"/>
7   </evidence-config package="curam.evidence">
8 </application-builder-config>
```

En este ejemplo, el generador de solicitudes de pruebas se ha configurado para crear pruebas de HouseholdMember y HeadOfHousehold. En la línea 3, el nombre de paquete Java base se especifica como curam.evidence. El generador de solicitudes de pruebas utiliza esta información para deducir lo siguiente acerca de HouseholdMember:

1. El nombre de la clase de capa de servicios de pruebas es curam.evidence.service.HouseholdMember.
2. El nombre de la operación en esta clase utilizada para crear las pruebas es createHouseholdMemberEvidence().
3. El nombre de la clase que se pasa como argumento en esta llamada es curam.evidence.entity.struct.HouseholdMemberEvidenceDetails.

El generador de solicitudes de pruebas utiliza esta información para crear las pruebas de HouseholdMember para la persona actual que se está procesando.

Todo lo anterior se basa en la suposición de que las pruebas se codifican de acuerdo con determinados patrones. Se garantiza que esto es así si el generador de pruebas se utiliza para generar las pruebas. El generador de solicitudes de pruebas puede trabajar con pruebas codificadas manualmente siempre que siga los patrones utilizados por el generador de pruebas.

Gestión de campos caseParticipantDetails

Para poder ejecutar la operación createHouseholdMemberEvidence() en el HouseholdMember, el CDME tiene que rellenar el campo caseParticipantDetails de la estructura HouseholdMemberEvidenceDetails, de la que se muestra un extracto a continuación:

```
public final class HouseholdMemberEvidenceDetails
implements java.io.Serializable, curam.util.type.DeepCloneable {

    /** Atributo de la estructura. */
    public curam.core.sl.struct.CaseIDKey caseIDKey;

    /** Atributo de la estructura. */
    public curam.core.sl.struct.CaseParticipantDetails
        caseParticipantDetails;

    /** Atributo de la estructura. */
    public curam.core.sl.struct.EvidenceDescriptorDetails descriptor;

    /** Atributo de la estructura. */
    public curam.evidence.entity.struct.HouseholdMemberDtls dtls;
    ...
}
```

En líneas generales, los miembros de la estructura dtls se rellenan mediante los elementos <set-attribute> y <map-attribute> de la especificación de correlación. Por

ejemplo, la línea siguiente de la especificación de correlación hace que el campo natHawOrPaIsInd se rellene con un valor de la estructura dtls:

```
<map-attribute
    from="nativeAlaskanOrAmericanIndian"
    to="natHawOrPaIsInd"
/>
```

El campo caseParticipantDetails suele estar presente en una estructura EvidenceDetails. En este ejemplo, se crea un participante en un caso para Grace y caseParticipantDetails hace referencia a este participante en un caso. El motor de correlación de datos realiza esta acción automáticamente cuando se encuentra un campo denominado caseParticipantDetails en la estructura EvidenceDetails. A veces, sin embargo, existen variaciones necesarias en la gestión de los participantes en un caso, por ejemplo cuando la estructura de detalles de pruebas contiene participantes en un caso adicionales que hacen referencia a terceros. Considere lo siguiente:

```
public final class AnnuityEvidenceDetails
implements java.io.Serializable, curam.util.type.DeepCloneable {
    /** Atributo de la estructura. */
    public curam.core.sl.struct.CaseIDKey caseIDKey;

    /** Atributo de la estructura. */
    public curam.core.sl.struct.CaseParticipantDetails
        instCaseParticipantDetails;

    /** Atributo de la estructura. */
    public curam.core.sl.struct.EvidenceDescriptorDetails descriptor;

    /** Atributo de la estructura. */
    public curam.evidence.entity.struct.AnnuityDtls dtls;

    /** Atributo de la estructura. */
    public curam.evidence.entity.struct.AnnuityCaseParticipantDetails
        annuityCaseParticipantDetails;
}
```

En este ejemplo, se hace referencia al participante en un caso propietario de la anualidad en la estructura AnnuityCaseParticipantDetails agregada bajo el nombre de campo annuityCaseParticipantDetails. La institución titular de la anualidad se describe en la estructura CaseParticipantDetails y se agrega bajo el nombre de campo instCaseParticipantDetails. Esta variación se puede facilitar utilizando la siguiente configuración del generador de solicitudes de pruebas:

```
1 <entity
2   case-participant-class-name="curam.core.sl.struct.CaseParticipantDetails"
3   case-participant-relationship-name="annuityCaseParticipantDetails"
4   name="Annuity"
5 >
6   <ev-field
7     aggregation="instCaseParticipantDetails"
8     referenced-as="participantName"
9     target-name="participantName"
10  />
11  <ev-field
12    aggregation="instCaseParticipantDetails"
13    referenced-as="address"
14    target-name="address"
15  />
16 </entity>
```

Las líneas 2 y 3 indican al generador de solicitudes de pruebas que el nombre de campo annuityCaseParticipantDetails hace referencia a los detalles del participante

en el caso (caseParticipantDetails) de esta entidad de pruebas utilizando la estructura CaseParticipantDetails. Las líneas 5-9 indican al generador de solicitudes de pruebas que se puede hacer referencia al campo participantName de la estructura agregada instCaseParticipantDetails en la especificación de correlación como "participantName" (línea 7). Lo mismo sucede para la dirección institucional en las líneas 10-14. Utilizando el ejemplo siguiente, es posible correlacionar el nombre y la dirección de la institución titular de la anualidad:

```
1 <target-entity name="Annuity" id="AnnuityTarget">
2   <map-attribute
3     from="institutionName"
4     to="participantName"
5   />
6   <map-attribute
7     from="institutionAddress"
8     to="address"
9   />
10 </target-entity>
```

En algunos casos, se puede considerar demasiado pesado pedir al cliente que rellene todo este tipo de información de terceros como parte de una admisión en línea. En su lugar, puede utilizarse la especificación de correlación para adoptar los valores predeterminados para estos valores, que se pueden rellenar correctamente en la fase de la entrevista. A continuación se muestra un ejemplo de cómo obtener los valores predeterminados para un participante tercero como por ejemplo una institución financiera:

```
1 <target-entity name="Annuity" id="AnnuityTarget">
2   <map-attribute
3     from="resourceAmount"
4     to="annuityValue"
5   />
6   <set-attribute
7     name="participantName"
8     value="Unknown"
9   />
10  <set-attribute
11    name="address"
12    value="curam.blankaddress"
13  />
14 </target-entity>
```

El valor curam.blankaddress de la línea 12 hace que se inserte una dirección en blanco para el participante.

Establecimiento de identificadores de entidades de destino

En la línea 1 del ejemplo 13, al igual que en varios de los ejemplos anteriores, el elemento <target-entity> ha incluido un atributo de ID "AnnuityTarget". Aunque este atributo es opcional, es una buena práctica incluir un ID en todos los elementos <target-entity>. Esto permite que el motor de correlación de datos distinga entre varias correlaciones diferentes de la misma entidad con el mismo tipo de entidad de destino. Considere el ejemplo siguiente: la entidad Person del almacén de datos común tiene dos indicadores booleanos: isBlind y hasDisability. Ambos se correlacionan con el mismo tipo de entidad de destino, Disability, tal como se indica a continuación:

```
1 <map-entity source="Person">
2   <condition expression="Person.isBlind==true">
3     <target-entity
4       id="DisabilityBlind"
5       name="Disability"
6     >
7     <set-attribute
```

```

7         name="disabilityType"
8         value="DT1"
9     />
10 </target-entity>
11 </condition>
12 <!-- Crear un registro de discapacidad vacío. -->
13 <condition expression="Person.hasDisability==true">
14     <target-entity
15         id="DisabilityUnspecified"
16         name="Disability"
17     />
18 </condition>
19 </map-entity>

```

El primer destino de las líneas 1-11 garantiza que si un solicitante indica que es ciego se cree un registro de discapacidad del tipo ceguera. El segundo destino, líneas 13-18, comprueba el indicador `hasDisability` y, si está establecido en `true`, se crea un registro de discapacidad de tipo no especificado. Al asignar a las dos correlaciones un ID distinto, el motor de correlación puede diferenciar las dos correlaciones. Sin el ID, la segunda correlación no se procesará.

Para el generador de solicitudes en PDF

Un formulario PDF contiene diversos campos de distintos tipos. Cada campo tiene un nombre exclusivo. Los Servicios del espacio de trabajo de Cúram utilizan este nombre exclusivo para hacer referencia al campo para que pueda escribir datos en ese campo. Para que este proceso funcione, el autor del formulario PDF debe asignar un nombre a estos campos y establecer propiedades de estos campos de acuerdo con ciertos convenios. Si se siguen estos convenios, el generador de solicitudes en PDF podrá correlacionar datos con dichos campos.

Nota: En esta sección se describe el generador de solicitudes PDF personalizadas, que permite realizar correlaciones con un formulario PDF personalizado. Los clientes también pueden beneficiarse del generador de PDF genérico, consulte la sección "Cómo personalizar el PDF genérico para solicitudes procesadas" del capítulo Personalización del manejo de solicitudes enviadas de la Guía de personalización de Cúram Universal Access.

Secciones y campos

El convenio más básico es que los campos se agrupan en "Secciones". Estas secciones no corresponden necesariamente a las secciones del formulario, pero en muchos casos sí lo hacen. Por ejemplo:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <application-builder-config
3     xmlns="http://www.curamsoftware.com/schemas/GUMBO/ApplicationBuilderConfig">
4     <pdf-config>
5         <section name="Applicant">
6             <field name="Name" type="append" append-separator=" "/>
7             <field name="SSN"/>
8             <field name="DateofBirth"/>
9             <field name="Gender" type="button-radio"/>
10            <field name="USCitizen" type="button-radio"/>
11            <field name="blackOrAfricanAmerican"
12                type="button-checkbox"/>
13            <field name="nativeAlaskanOrAmericanIndian"
14                type="button-checkbox"/>
15            <field name="asian" type="button-checkbox"/>
16            <field name="nativeHawaiianOrPacificIslander"
17                type="button-checkbox"/>
18            <field name="whiteOrCaucasian"
19                type="button-checkbox"/>

```

```

15         <field name="EthnicOrigin" type="button-radio"/>
16     </section>
17 </pdf-config>
18</application-builder-config>

```

El ejemplo 14 muestra un extracto de una configuración del generador de solicitudes en PDF. Hace referencia a una sección denominada Applicant. De acuerdo con la línea 4 del ejemplo, el generador de solicitudes en PDF espera el formulario PDF de destino contenga un campo denominado "Applicant.Name", un campo de texto. La línea 8 hace referencia a un campo del formulario PDF denominado "Applicant.Gender". Este campo es un botón de selección, mientras que las líneas 10-14 hacen referencia a los campos que son botones de recuadros de selección.

Cómo llenar campos de texto

La línea 6 del ejemplo 14 hace referencia a un campo de texto de formato estándar. La correlación correspondiente podría ser similar a la siguiente:

```

<target-entity name="Applicant">
  <map-attribute from="ssn" to="SSN"/>
</target-entity>

```

La línea 5 es en cierto modo diferente. El tipo se marca como "append". Esto significa que se puede escribir varias veces en el mismo campo de texto y cada vez que el motor de correlación escribe en el campo de texto, el resultado se añade al valor actual del campo de texto en lugar de sobrescribirlo. Cada vez que se produce una adición, los datos nuevos se separan de los datos anteriores mediante el separador de adición, en este caso un solo carácter de espacio. Si se toma un archivo de correlación como el que se muestra en el ejemplo 16 y se combina con la configuración de correlación que se muestra en el ejemplo 14, el campo Applicant.Name se rellenará con el nombre, la inicial del segundo nombre y el apellido, por ejemplo "Pat A Kayek".

```

<target-entity name="Applicant">
  <map-attribute from="firstName" to="Name"/>
  <map-attribute from="middleInitial" to="Name"/>
  <map-attribute from="lastName" to="Name"/>
</target-entity>

```

Añadir campos de texto también resulta útil para crear una lista de elementos separada por comas. Considere un campo que solicita al cliente que proporcione una lista de personas de su unidad familiar que están embarazadas. Un extracto del XML de correlación podría tener normalmente un aspecto similar al siguiente:

```

<condition expression="Person.isPregnant == true">
  <target-entity name="Pregnancy">
    <map-attribute from="firstName" to="Pregnancy"/>
    <set-attribute name="HasPregnancies" value="Yes"/>
  </target-entity>
</condition>

```

La configuración de correlación correspondiente se muestra en la Figura 18. Cada vez que el motor de correlación de procesa a una persona de la unidad familiar para la que el indicador isPregnant esté establecido en true, el nombre de esa persona se añadirá al campo Pregnancy.Pregnancies.

```

<section name="Pregnancy">
  <field name="Pregnancies" type="button-checkbox"/>
  <field name="Pregnancy" type="append" append-separator=", "/>
</section>

```

Secciones repetidas y descripciones de tablas de códigos

Algunos formularios contienen secciones repetidas, por ejemplo "Listar los detalles de todas las personas de su unidad familiar" o "Listar todas las fuentes de ingresos del trabajo". El generador de solicitudes en PDF está diseñado para tratar con esto siempre que el autor del PDF asigne nombres a los campos de acuerdo con los convenios correctos. Por ejemplo, se pueden asignar los nombres siguientes a los campos utilizados para recopilar datos sobre los miembros de la unidad familiar:

Tabla 4. Campos de un formulario en PDF para registrar miembros de la unidad familiar

Nombre	¿Qué relación tiene esta persona con usted?	Fecha de nacimiento	Número de la seguridad social
OtherPerson0.Name	OtherPerson0 .RelType	OtherPerson0 .DateOfBirth	OtherPerson0 .SSN
OtherPerson1.Name	OtherPerson1 .RelType	OtherPerson1 .DateOfBirth	OtherPerson1 .SSN
OtherPerson2.Name	OtherPerson2 .RelType	OtherPerson2 .DateOfBirth	OtherPerson2 .SSN

La configuración de correlación correspondiente se escribiría de la siguiente manera:

```
1 <section name="Person" type="multiple">
2 <field name="Name" type="append" append-separator=" "/>
3 <field name="RelType" codetable-class="RelationshipTypeCode"/>
4 <field name="DateofBirth"/>
5 </section>
```

Observe que, en la línea 1, el atributo `type="multiple"` es lo que causa que la sección se repita. Observe el atributo `codetable-class` de la línea 3 de este ejemplo. Es un atributo muy útil que hace que los valores de la tabla de códigos se conviertan en descripciones localizadas. Al utilizarlo en el contexto anterior, el autor del script garantiza que la segunda columna se llene con valores localizados como "Progenitor" y "Hermano" en lugar de códigos de sentido como "RT1" o "RT3".

Recuadros de selección

Un recuadro de selección es un campo único que se puede marcar o dejar sin marcar. El generador de solicitudes en PDF presupone que establecer un campo de recuadro de selección con el valor "Yes" hará que se marque, mientras que establecerlo con el valor "No" hará que se deje sin marcar. El autor del formulario PDF debe asegurarse de que se siga este convenio. Siempre que el motor de correlación correlacione un valor booleano con un campo de recuadro de selección, se correlacionará automáticamente tal como se indica a continuación: True se correlaciona con "Yes" y false se correlaciona con "No".

Botones de selección

Una colección de botones de selección se trata como un solo campo en un formulario PDF. Los botones de selección sólo pueden seleccionar un elemento a la vez. Los elementos individuales del botón de selección se seleccionan escribiendo un valor determinado para el botón de selección. El autor del formulario PDF puede determinar qué elemento está seleccionado especificando un "valor de exportación" para cada elemento. Un uso típico para un botón de selección con Cúram consiste en utilizar valores de exportación para indicar un número de elementos de tabla de códigos.

Considere el ejemplo de un botón de selección que se utiliza para indicar es sexo del solicitante. Los valores de la tabla de códigos para hombre y mujer son "SX1" y "SX2", respectivamente. El autor del PDF crea un solo campo de botón de selección denominado "Applicant.Gender". El elemento "Male" se indica mediante el valor de exportación "SX1", mientras que el elemento "Female" se indica mediante el valor de exportación "SX2". La correlación es como la siguiente:

```
<target-entity name="Applicant">
  <map-attribute from="gender" to="Gender"/>
</target-entity>
```

La configuración de correlación correspondiente es similar a la siguiente:

```
<section name="Applicant">
  <field name="Gender" type="button-radio"/>
</section>
```

Cuadros combinados de selección

Un cuadro combinado de selección es una lista desplegable de elementos donde el usuario puede elegir un elemento. El nombre del elemento proporciona información suficiente para que el usuario del formulario pueda decidir qué elemento debe elegir. Como ejemplo, imagine que la persona que diseña el formulario desea proporcionar una lista desplegable para representar el estado de inmigrante de una persona. Cúram tiene una tabla de códigos AlienStatus que tiene códigos correspondientes a las descripciones siguientes:

- Inmigrante
- Ciudadano de los EE.UU.
- Inmigrante indocumentado
- Refugiado
- Nacional no ciudadano

El diseñador de formularios PDF crea un cuadro combinado de selección y establece el elemento de texto para cada elemento de la lista desplegable con los valores anteriores. Para garantizar que se envíe al formulario la descripción de la tabla de códigos en lugar del código de la tabla de códigos, se necesita la configuración siguiente:

```
<section name="AlienPerson" type="multiple">
  <field name="CitizenshipStatus" type="choice-combo"
  codetable-class="AlienStatus"/>
</section>
```

Cómo configurar el formulario de solicitud en PDF

En primer lugar, cargue el formulario PDF en el portal de acceso universal. Es importante utilizar un *formulario* PDF, no sólo un documento PDF. El PDF debe contener un formulario y el formulario debe contener campos. Si desea utilizar el generador de solicitudes en PDF, entonces cada campo del formulario debe tener un nombre exclusivo, por ejemplo PersonalDetails.surname o Child1Details.surname. Debe utilizar un programa como Adobe Acrobat Writer Professional o GlobalSCAPE CutePDF Pro para editar el formulario.

Tenga en cuenta que, si utiliza Adobe Acrobat Writer Professional, debe asegurarse de guardar el formulario como un AcroForm y no un XFA. Puede subir su PDF iniciando sesión en Cúram como administrador, seleccionando la sección de administración de acceso universal y seleccionando formularios en PDF. Desde aquí, puede subir el formulario y asignarle un nombre adecuado. Cada solicitud de admisión tiene un tipo de solicitud de admisión, por lo que lo siguiente que debe

hacer es asegurarse de que el tipo de solicitud de admisión esté asociado con el formulario PDF correcto. Puede hacerlo accediendo a Solicitudes de admisión, seleccionando el tipo de solicitud correspondiente y seleccionando Editar. Pulse el menú desplegable Formulario PDF y el formulario PDF que acaba de subir deberá aparecer en esta lista. Selecciónelo. A continuación debe especificar correlaciones de PDF para los programas específicos que le interesen.

Escriba un xml de correlación de PDF y un xml de configuración de PDF siguiendo las instrucciones de la sección anterior de esta guía. Vaya al elemento de menú Programas de la sección de administración de acceso universal. Aparecerá una lista de programas. Vea el programa que le interese. Seleccione la pestaña Correlación de la parte superior derecha. Cree una correlación nueva. Asegúrese de seleccionar "Creación de formulario PDF" en lugar de Creación de pruebas. Suba el archivo de configuración de correlación y el archivo de especificación de correlación. Pruebe la correlación realizando una admisión para el programa relevante. Al final de la admisión, seleccione el enlace para visualizar el archivo PDF.

Cómo escribir especificaciones y configuraciones de correlación para pruebas dinámicas

Introducción

La finalidad de este documento es correlacionar datos de la ciudadanía capturados durante el proceso de admisión en entidades de pruebas dinámicas.

Requisitos previos:

Se presupone que tiene conocimientos de los conceptos básicos de las pruebas dinámicas. En concreto, se presupone que comprende bien la naturaleza de las pruebas, la definición de tipos de pruebas dinámicas, la definición de versiones de pruebas y los metadatos XML de las pruebas dinámicas.

Cómo escribir especificaciones y configuraciones de correlación para pruebas dinámicas

Las pruebas dinámicas, por su naturaleza, no están modeladas. Hay una sola entidad (o un conjunto de entidades) que contiene datos de todos los tipos de pruebas dinámicas. Un tipo de pruebas dinámicas tendrá una o más versiones de tipo de pruebas a lo largo de su ciclo de vida. En cualquier momento concreto, sólo será vigente una versión del tipo de pruebas. Todos los elementos de metadatos (atributos, relaciones, etc.) se definirán al nivel de versión de tipo de pruebas. Por lo tanto, el desarrollador es responsable de proporcionar la especificación y las configuraciones de correlación correctas para la versión de tipo de pruebas vigente.

Metadatos de pruebas dinámicas simples

Los metadatos simples representan la estructura del tipo de pruebas dinámicas Miembro de la unidad familiar. A continuación se definen los distintos tipos de atributos, como Boolean, Codetable, Date y String:

```
<?xml version="1.0" encoding="UTF-8"?>
<EvidenceTypeVersion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://DynamicEvidenceMetadata.xsd">
  <Model>
    <Attributes>
      <Attribute>
        <DataAttribute name="blk0rAfrAmerInd">
```

```

        <DomainType dataType="Boolean" />
    </DataAttribute>
</Attribute>
<Attribute>
    <DataAttribute name="ssnStatus">
        <DomainType dataType="Codetable">
            <CodetableOptions codetableName="SSNApplicationStatus" />
        </DomainType>
    </DataAttribute>
</Attribute>
<Attribute>
    <DataAttribute name="startDate">
        <DomainType dataType="Date" />
    </DataAttribute>
</Attribute>
<Attribute>
    <DataAttribute name="endDate">
        <DomainType dataType="Date" />
    </DataAttribute>
</Attribute>
<Attribute>
    <DataAttribute name="comments">
        <DomainType dataType="String" />
    </DataAttribute>
</Attribute>
</Attributes>
</Model>
<Validations>
    <PatternValidations>
    </PatternValidations>
</Validations>
<UserInterface />
</EvidenceTypeVersion>

```

Especificación de correlación simple

Esta especificación de correlación simple correlaciona datos de la entidad HouseHoldMember de un almacén de datos (DataStore) con una entidad de pruebas dinámicas HouseHoldMember definida en la sección anterior.

Tenga en cuenta que el nombre de atributo mencionado en el campo 'to' debe coincidir con el campo 'name' del elemento 'DataAttribute' de los metadatos de pruebas dinámicas. En otras palabras, un desarrollador debe asegurarse de que el atributo con el que se correlacionarán los datos esté presente en los metadatos del tipo de pruebas dinámicas en cuestión.

```

<?xml version="1.0" encoding="UTF-8"?>
<map xmlns="http://www.curamsoftware.com/schemas/GUMBO/Map"
    name="EvidenceMapping">
    <map-entity source="HouseHoldMember">
        <target-entity name="HouseHoldMember">
            <map-attribute from="blkOrAfrAmerInd"
                to="blkOrAfrAmerInd" />
            <map-attribute from="ssnStatus" to="ssnStatus" />
            <map-attribute from="startDate" to="startDate" />
            <map-attribute from="endDate" to="endDate" />
            <map-attribute from="comments" to="comments" />
        </target-entity>
    </map-entity>
</map>

```

Configuración de correlación simple

La configuración de correlación simple siguiente se define para el tipo de pruebas dinámicas HouseHoldMember:

```

<?xml version="1.0" encoding="UTF-8"?>
<application-builder-config >
  <evidence-config package="curam.gumbo.evidence">
    <entity name="HouseholdMember" ev-type-code="DE_HMEMBER"/>
  </evidence-config>
</application-builder-config>

```

Un desarrollador debe especificar el código del tipo de pruebas dinámicas en el atributo 'ev-type-code'. Esto sirve para permitir que el sistema determine si las pruebas son estáticas o dinámicas. Si este atributo se deja en blanco o contiene una entrada no válida, el sistema presupone que el tipo de pruebas es estático y procederá a correlacionar los datos.

Correlación de pruebas dinámicas padre-hijo

En esta sección se describe cómo se realizaría una correlación para pruebas que tienen una relación padre-hijo.

Metadatos de pruebas dinámicas padre-hijo simples

Los metadatos siguientes tienen atributos para el tipo de pruebas dinámicas de adopción (Adoption). Estos metadatos de Adoption tienen dos atributos que describen que esta entidad tiene dos campos CaseParticipant relacionados. El campo "caseParticipantRoleID" es un CaseParticipantRole primario y "parCaseParticipantRoleID" es un CaseParticipantRole asociado.

```

<?xml version="1.0" encoding="UTF-8"?>
<EvidenceTypeVersion xmlns:xsi="http://www.w3.org/2001/
  XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../../../
  DynamicEvidence/source/curam/dynaminevidence/definition/impl/
  xmlresources/DynamicEvidenceMetadata.xsd">
  <Model>
    <Attributes>
      <Attribute>
        <DataAttribute name="adoptionFinalizedDate">
          <DomainType dataType="Date" />
        </DataAttribute>
      </Attribute>
      <Attribute>
        <RelatedCPAttribute name="caseParticipantRoleID"
          participantType="Person" volatile="true" />
      </Attribute>
      <Attribute>
        <RelatedCPAttribute name="parCaseParticipantRoleID"
          participantType="Person" />
      </Attribute>
    </Attributes>
  </Model>
  <UserInterface>
    <Cluster>
      <RelCPCluster fullCreateParticipant="true">
        <StandardField source="caseParticipantRoleID" />
      </RelCPCluster>
    </Cluster>
    <Cluster>
      <RelCPCluster fullCreateParticipant="true">
        <StandardField source="parCaseParticipantRoleID" />
      </RelCPCluster>
    </Cluster>
  </UserInterface>
</EvidenceTypeVersion>

```

Los metadatos AdoptionPayment tienen relación con sus pruebas dinámicas padre.

```

<?xml version="1.0" encoding="UTF-8"?>
<EvidenceTypeVersion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://DynamicEvidenceMetadata.xsd"
  javaHookClassNameForCalculatedAttributes="curam.dynamicEvidencetest.
  hook.impl.AdoptionPaymentCalculatedAttributeHook"
  useHookForCalculatedAttributes="true">
  <Model>
    <Attributes>
      <Attribute>
        <DataAttribute name="amount">
          <DomainType dataType="Money" />
        </DataAttribute>
      </Attribute>
      <Attribute>
        <CalculatedAttribute name="parentName">
          <DomainType dataType="String" />
        </CalculatedAttribute>
      </Attribute>
    </Attributes>
    <Relationships>
      <Relationship>
        <MandatoryParent name="adoptions"
          evidenceTypeCode="DET004" />
      </Relationship>
    </Relationships>
  </Model>
</EvidenceTypeVersion>

```

Especificación de correlación padre-hijo simple

La especificación de correlación tiene una relación padre-hijo y hay pocos atributos definidos para la entidad de adopción (Adoption) y todos se utilizan para crear un participante nuevo. Los valores de estos atributos se leen del almacén de datos de IBM Cúram.

```

<?xml version="1.0" encoding="UTF-8"?>
<map xmlns="http://www.curamsoftware.com/schemas/GUMBO/Map"
  name="ParentChildMapping">
  <map-entity source="Adoption">
    <target-entities>
      <target-entity name="Adoption" type="parent" id="parent">
        <map-attribute from="adoptionFinalizedDate"
          to="adoptionFinalizedDate" />
        <map-attribute from="adParentName"
          to="adParentName" />
        <map-attribute from="adParentStreet1"
          to="adParentStreet1" />
        <map-attribute from="adParentStreet2"
          to="adParentStreet2" />
        <map-attribute from="adParentCity"
          to="adParentCity" />
        <map-attribute from="adParentState"
          to="adParentState" />
        <map-attribute from="adParentZipCode"
          to="adParentZipCode" />
      </target-entity>
      <target-entity name="AdoptionPayment"
        type="child" id="child">
        <set-attribute name="amount" value="2200" />
      </target-entity>
    </target-entities>
  </map-entity>
</map>

```

Configuración de correlación simple para relación padre-hijo

Aquí se presenten los elementos <def-create-participant> y <create-participant>. Lo más importante que se debe tener en cuenta aquí es se ha añadido un atributo nuevo, 'dyn-evidence-primary-cpr-field-name' al elemento <entity>. La finalidad de este atributo es que el desarrollador debe especificar el nombre de atributo del rol de participante en un caso (CaseParticipantRole) primario definido en los metadatos. En este ejemplo, 'caseParticipantRoleID' es el CaseParticipantRole primario que se ha definido en la entidad de adopción (Adoption). De forma similar, el nombre del atributo CaseParticipantRole ('parCaseParticipantRoleID' en este ejemplo) se define en el campo 'name' del elemento <create-participant>. Nota: En el caso de las pruebas estáticas, este mismo campo 'name' del elemento <create-participant> se utiliza para mencionar el nombre de agregación correspondiente.

```
<?xml version="1.0" encoding="UTF-8"?>
<application-builder-config xmlns="http://www.curamsoftware.com/
schemas/GUMBO/ApplicationBuilderConfig">
  <evidence-config package="curam.evidence">
    <def-create-participant id="AdoptedParentDetails" type="RL13">
      <participant-name-field name="firstName"
        from="adParentName" order="1" />
      <participant-address type="AT3">
        <address-field name="addressLine1"
          from="adParentStreet1" />
        <address-field name="addressLine2"
          from="adParentStreet2" />
        <address-field name="city" from="adParentCity" />
        <address-field name="state" from="adParentState" />
        <address-field name="zip" from="adParentZipCode" />
      </participant-address>
    </def-create-participant>
    <entity name="Adoption" ev-type-code="DET004"
      dyn-evidence-primary-cpr-field-name="caseParticipantRoleID">
      <create-participant refid="AdoptedParentDetails"
        name="parCaseParticipantRoleID" role="" />
    </entity>
    <entity name="AdoptionPayment" ev-type-code="DET005"/>
  </evidence-config>
</application-builder-config>
```

Correlación con terceros

Introducción

La funcionalidad de admisión de acceso universal permite a los usuarios especificar detalles sobre sus circunstancias utilizando scripts IEG2. El script IEG2 inserta los detalles del cliente en un almacén de datos. Tras el envío, el contenido del almacén de datos se correlaciona con datos de pruebas para un caso de admisión. Muchos tipos de pruebas hacen referencia a terceros; estos terceros deben insertarse en el caso como participantes en el caso nuevos con su propio rol de participante en un caso exclusivo. Por ejemplo, un registro de embarazo puede tener un asociado "Padre". Si el padre está ausente, ese padre puede registrarse como un participante en un caso candidato. En otro ejemplo, las pruebas de un estudiante se deben asociar con una escuela. Se especifica una escuela como un participante en un caso "Representante". Estos participantes en un caso nuevos se deben crear sobre la marcha durante la correlación y deben contener tanta información como sea posible para facilitar el proceso de admisión para los asistentes sociales asignados que deben procesar el caso.

La correlación de direcciones resulta necesaria debido a la necesidad de asociar un participante nuevo o existente con una prueba nueva. El participante es normalmente un representante o una persona. Uno de los retos concretos de la creación de participantes nuevos es la correlación de direcciones. Los campos de dirección almacenados en el almacén de datos, como por ejemplo "ADD1" se deben agregar correctamente en una estructura de dirección de Cúram con formato correcto para garantizar que el participante se pueda crear correctamente.

Verá en este capítulo que la lógica de la creación de un participante nuevo y de la correlación de una dirección con dicho participante está aislada del generador de solicitudes de pruebas.

Cómo correlacionar terceros

El esquema de configuración del generador de solicitudes incluye elementos y atributos que se pueden utilizar para crear un participante nuevo y correlacionar una dirección con dicho participante.

Definición del creador de participantes

El elemento <def-create-participant> forma parte del elemento <evidence-config>. Este elemento se utiliza para definir el comportamiento para crear un participante. Este mismo comportamiento puede ser reutilizado por varias definiciones de <entity> mediante el ID. Lo más importante que se debe tener en cuenta aquí es que el tipo de datos de todos los atributos mencionados aquí se debe definir como 'String'.

```
<def-create-participant id="SchoolRepresentative" type="RL13">
  <participant-name-field name="firstName" from="participantName"
    order="1"/>
  <participant-address type="AT3">
    <address-field name="addressLine1" from="street1"/>
    <address-field name="addressLine2" from="street2"/>
    <address-field name="city" from="city"/>
    <address-field name="state" from="state"/>
    <address-field name="zip" from="zipCode"/>
  </participant-address>
</def-create-participant>
```

Crear participante

El elemento <create-participant> se ha añadido al elemento <entity>. Este elemento indica a ApplicationBuilder que debe crear un participante, tal como se ha definido en la definición del creador del participante.

```
<entity case-participant-class-name="curam.core.sl.struct.
  CaseParticipantDetails"case-participant-relationship-name=
  "curam.none" name="Student">
  <create-participant refid="SchoolRepresentative"
    name="schCaseParticipantDetails" role="SCH"/>
</entity>
```

Esquema de correlación de ejemplo

A continuación se proporciona el esquema que se debe seguir al escribir la especificación de correlación. Tenga en cuenta en la entidad Education los atributos correlacionados directamente con la entidad de pruebas 'Student'. Atributos como schoolName, schoolStreet1, schoolStreet2, etc. se utilizarán para crear un participante nuevo y una dirección.

```
<?xml version="1.0" encoding="UTF-8"?>
<map xmlns="http://www.curamsoftware.com/schemas/GUMBO/Map"
  from-schema="GumboDS" name="TestMapping"to-schema="CGISS"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="...\EJBServer\components\
WorkspaceServices\lib\Mapping.xsd"><map-entity source="Person">
  <target-entity id="householdMember" name="HouseholdMember">
    <map-attribute from="ssnStatus" to="ssnStatus"/>
    <map-attribute from="blackOrAfricanAmerican"
      to="blkOrAfrAmerInd"/>
    <map-attribute from="nativeAlaskanOrAmericanIndian"
      to="natHawOrPaIsInd"/>
    <map-attribute from="asian" to="asianInd"/>
    <map-attribute from="nativeHawaiianOrPacificIslander"
      to="natHawOrPaIsInd"/>
    <map-attribute from="whiteOrCaucasian"
      to="whiteOrCaucInd"/>
    <map-attribute from="isMigrantOrSeasonalFarmWorker"
      to="migrantFWorkerInd"/>
  </target-entity>
  <target-entity id="livingArrange" name="LivingArrange">
    <map-attribute from="accommodationType"
      to="livingArrangeType"/>
  </target-entity>
</map-entity>
<map-entity source="Education">
  <condition expression=
    "Education.highestGrade!=&quot;&quot;">
    <target-entity id="highestGrade" name="Student">
      <map-attribute from="highestGrade"
        to="highGradeCompleted"/>
      <map-attribute from="attendanceFrequency"
        to="studentStatus"/>
      <map-attribute from="schoolName"
        to="participantName"/>
      <map-attribute from="schoolStreet1" to="street1"/>
      <map-attribute from="schoolStreet2" to="street2"/>
      <map-attribute from="schoolCity" to="city"/>
      <map-attribute from="schoolState" to="state"/>
      <map-attribute from="schoolZipCode" to="zipCode"/>
    </target-entity>
  </condition>
</map-entity>
<map-entity source="HealthInsuranceExpense">
  <target-entity id="healthInsuranceExpense"
    name="MedicalInsurance">
    <map-attribute from="policyNumber" to="policyNumber"/>
    <map-attribute from="groupNumber" to="groupPolicyNumber"/>
    <map-attribute from="policyHolderParticipantName"
      to="policyHolderParticipantName"/>
    <map-attribute from="policyHolderStreet1"
      to="policyHolderStreet1"/>
    <map-attribute from="policyHolderStreet2"
      to="policyHolderStreet2"/>
    <map-attribute from="policyHolderCity"
      to="policyHolderCity"/>
    <map-attribute from="policyHolderState"
      to="policyHolderState"/>
    <map-attribute from="policyHolderZipCode"
      to="policyHolderZipCode"/>
    <map-attribute from="groupParticipantName"
      to="groupParticipantName"/>
    <map-attribute from="groupStreet1" to="groupStreet1"/>
    <map-attribute from="groupStreet2" to="groupStreet2"/>
    <map-attribute from="groupCity" to="groupCity"/>
    <map-attribute from="groupState" to="groupState"/>
    <map-attribute from="groupZipCode" to="groupZipCode"/>
    <map-attribute from="insuranceProvider"
      to="insuranceProvider"/>
    <map-attribute from="InsProviderStreet1"

```



```

                to="InsProviderStreet1"/>
<map-attribute from="InsProviderStreet2"
                to="InsProviderStreet2"/>
<map-attribute from="InsProviderCity"
                to="InsProviderCity"/>
<map-attribute from="InsProviderState"
                to="InsProviderState"/>
<map-attribute from="InsProviderZipCode"
                to="InsProviderZipCode"/>
<map-entity source="HealthInsuranceExpenseRelationship">
<target-entity id="healthInsuranceExpenseRelationship"
                name="Coverage">
    <map-attribute from="personID"
                to="caseParticipantRoleID"/>
</target-entity>
</map-entity>
</target-entity>
</map>

```

Ejemplo de configuración de correlación

A continuación se proporciona el XML de configuración que se debe seguir al escribir la especificación de correlación.

```

<?xml version="1.0" encoding="UTF-8"?><application-builder-config xmlns=
"http://www.curamsoftware.com/schemas/GUMBO/ApplicationBuilderConfig">
  <evidence-config package="curam.evidence">
    <def-create-participant id="SchoolRepresentative" type="RL13">
      <participant-name-field name="firstName" from=
        "participantName" order="1"/>
      <participant-address type="AT3">
        <address-field name="addressLine1" from="street1"/>
        <address-field name="addressLine2" from="street2"/>
        <address-field name="city" from="city"/>
        <address-field name="state" from="state"/>
        <address-field name="zip" from="zipCode"/>
      </participant-address>
    </def-create-participant>
    <def-create-participant id="MedicalInsurancePolicyHolder"
      type="RL7">
      <participant-name-field name="firstName"
        from="policyHolderParticipantName" order="1"/>
      <participant-address type="AT3">
        <address-field name="addressLine1"
          from="policyHolderStreet1"/>
        <address-field name="addressLine2"
          from="policyHolderStreet2"/>
        <address-field name="city" from="policyHolderCity"/>
        <address-field name="state" from="policyHolderState"/>
        <address-field name="zip" from="policyHolderZipCode"/>
      </participant-address>
    </def-create-participant>
    <def-create-participant id="MedicalInsuranceGroup"
      type="RL13">
      <participant-name-field name="firstName"
        from="groupParticipantName" order="1"/>
      <participant-address type="AT3">
        <address-field name="addressLine1"
          from="groupStreet1"/>
        <address-field name="addressLine2"
          from="groupStreet2"/>
        <address-field name="city" from="groupCity"/>
        <address-field name="state" from="groupState"/>
        <address-field name="zip" from="groupZipCode"/>
      </participant-address>
    </def-create-participant>
  </evidence-config>
</application-builder-config>

```

```

<def-create-participant id="MedicalInsuranceProvider"
                        type="RL13">
  <participant-name-field name="firstName"
                        from="insuranceProvider" order="1"/>
  <participant-address type="AT3">
    <address-field name="addressLine1"
                  from="InsProviderStreet1"/>
    <address-field name="addressLine2"
                  from="InsProviderStreet2"/>
    <address-field name="city" from="InsProviderCity"/>
    <address-field name="state" from="InsProviderState"/>
    <address-field name="zip" from="InsProviderZipCode"/>
  </participant-address>
</def-create-participant>

<entity name="HouseholdMember"/>
<entity name="HeadOfHousehold"/>
<entity case-participant-class-name="curam.core.sl.struct.
CaseParticipantDetails"case-participant-relationship-name=
"curam.none" name="Student">
  <create-participant refid="SchoolRepresentative"
                    name="schCaseParticipantDetails" role="SCH"/>
</entity>
<entity name="MedicalInsurance">
  <create-participant refid="MedicalInsurancePolicyHolder"
                    name="plchdrCaseParticipantDetails" role="MIPH"/>
  <create-participant refid="MedicalInsuranceGroup"
                    name="groupCaseParticipantDetails" role="GPP"/>
  <create-participant refid="MedicalInsuranceProvider"
                    name="insCaseParticipantDetails" role="MIP"/>
</entity>
</application-builder-config>

```

Esquema para especificaciones de correlación

Esquema

A continuación se proporciona el esquema que se debe seguir al escribir especificaciones de correlación:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.curamssoftware.com/schemas/GUMBO/Map"
xmlns:mp="http://www.curamssoftware.com/schemas/GUMBO/Map"
elementFormDefault="qualified">

  <xs:simpleType name="TargetEntityType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="parent"/>
      <xs:enumeration value="child"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="AttachmentType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="case"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="MapAttributeType">
    <xs:attribute name="from" type="xs:NCName" use="required"/>
    <xs:attribute name="to" type="xs:NCName" use="required"/>
    <xs:attribute name="mapping-function" type="xs:string"
use="optional"/>
    <xs:attribute name="mapping-rule" type="xs:string"
use="optional"/>
  </xs:complexType>

```

```

        <xs:attribute name="entity" type="xs:NCName" use="optional"/>
    </xs:complexType>

    <xs:complexType name="SetAttributeType">
        <xs:attribute name="name" type="xs:NCName"/>
        <xs:attribute name="value" type="xs:string"/>
    </xs:complexType>

    <xs:element name="set-attribute" type="mp:SetAttributeType"/>

    <xs:complexType name="TargetEntityType">
        <xs:sequence>
            <xs:element name="map-attribute" type="mp:MapAttributeType"
minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:set-attribute" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:condition" minOccurs="0"
                maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:NCName"/>
        <xs:attribute name="type" type="mp:TargetEntityRoleType"/>
        <xs:attribute name="attachment" type="mp:AttachmentType"/>
        <xs:attribute name="id" type="xs:ID" use="optional"/>
    </xs:complexType>

    <xs:element name="target-entity" type="mp:TargetEntityType"/>

    <xs:complexType name="ConditionType">
        <xs:choice>
            <xs:element ref="mp:target-entity" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:target-entities" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:set-attribute" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:map-entity" minOccurs="0"
                maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute name="expression" type="xs:string"/>
    </xs:complexType>

    <xs:element name="condition" type="mp:ConditionType"/>

    <xs:complexType name="MapEntityType">
        <xs:sequence>
            <xs:element ref="mp:target-entity" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:target-entities" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:condition" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:map-entity" minOccurs="0"
                maxOccurs="unbounded"/>
            <xs:element ref="mp:follow-association" minOccurs="0"
                maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="source" type="xs:NCName"/>
    </xs:complexType>

    <xs:element name="map-entity" type="mp:MapEntityType"/>

    <xs:element name="follow-association" type="mp:MapEntityType"/>

    <xs:complexType name="MapEntitiesType">
        <xs:sequence>
            <xs:element ref="mp:target-entity" minOccurs="0"

```

```

        maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="target-entities" type="mp:MapEntitiesType"/>

<xs:complexType name="MapCodeTableValueType">
    <xs:attribute name="source" type="xs:string"/>
    <xs:attribute name="target" type="xs:string"/>
</xs:complexType>

<xs:complexType name="MapCodeTableType">
    <xs:sequence>
        <xs:element name="map-value"
            type="mp:MapCodeTableValueType" minOccurs="1"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="context" type="xs:NCName" use="optional"/>
    <xs:attribute name="source-codetable" type="xs:NCName"/>
    <xs:attribute name="target-codetable" type="xs:NCName"/>
    <xs:attribute name="source-package" type="xs:NCName"
use="optional"/>
    <xs:attribute name="target-package" type="xs:NCName"
use="optional"/>
</xs:complexType>

<xs:complexType name="MapType">
    <xs:sequence>
        <xs:element name="map-code-table" type="mp:MapCodeTableType"
            minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="mp:map-entity" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:NCName"/>
    <xs:attribute name="from-schema" type="xs:NCName"/>
    <xs:attribute name="to-schema" type="xs:NCName"/>
</xs:complexType>

<xs:element name="map" type="mp:MapType"/>

</xs:schema>

```

Esquema para configuraciones de correlación

Esquema

A continuación se proporciona el esquema que se debe seguir al escribir configuraciones de correlación:

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    Materiales bajo licencia - Propiedad de IBM

    Copyright IBM Corporation 2012. Reservados todos los derechos.

    Derechos restringidos para los usuarios del gobierno de los EE.UU. - Uso, duplicación
    o divulgación restringida por el GSA ADP Schedule Contract con IBM Corporation.
-->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="

    http://www.curamsoftware.com/schemas/GUMBO/ApplicationBuilderConfig"
    xmlns:abc="

    http://www.curamsoftware.com/schemas/GUMBO/ApplicationBuilderConfig"

```

```

elementFormDefault="qualified">

  <xs:complexType name="EvFieldType">
    <xs:attribute name="referenced-as" type="xs:NCName" use="optional"/>
    <xs:attribute name="target-name" type="xs:NCName" use="optional"/>
    <xs:attribute name="aggregation" type="xs:NCName" use="optional"/>
    <xs:attribute name="is-reference-attribute" type="xs:boolean"
use="optional"/>
    <xs:attribute name="map-as-concernrole-id" type="xs:boolean"
use="optional"/>
  </xs:complexType>

  <xs:complexType name="ParticipantCreatorType">
    <xs:attribute name="refid" type="xs:string" use="required"/>
    <xs:attribute name="name" type="xs:NCName" use="required"/>
    <xs:attribute name="role" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="ParticipantNameFieldType">
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="from" type="xs:NCName" use="required"/>
    <xs:attribute name="order" type="xs:positiveInteger" use="optional"/>
  </xs:complexType>

  <xs:complexType name="AddressFieldType">
    <xs:attribute name="name" type="xs:NCName" use="required"/>
    <xs:attribute name="from" type="xs:NCName" use="required"/>
  </xs:complexType>

  <xs:complexType name="ParticipantAddressType">
    <xs:sequence>
      <xs:element ref="abc:address-field" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:element name="participant-name-field"
type="abc:ParticipantNameFieldType"/>
  <xs:element name="participant-address"
type="abc:ParticipantAddressType"/>
  <xs:element name="ev-field" type="abc:EvFieldType"/>
  <xs:element name="create-participant" type="abc:ParticipantCreatorType"/>
  <xs:element name="address-field" type="abc:AddressFieldType"/>

  <xs:complexType name="EntityType">
    <xs:sequence>
      <xs:element ref="abc:ev-field" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="abc:create-participant" minOccurs="0" maxOccurs=
"unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:NCName"/>
    <xs:attribute name="package" type="xs:string"/>
    <xs:attribute name="case-participant-relationship-name" type="xs:NCName"/>
    <xs:attribute name="case-participant-class-name" type="xs:NCName"/>
    <xs:attribute name="ev-type-code" type="xs:NCName" use="optional"/>
    <xs:attribute name="dyn-evidence-primary-cpr-field-name" type="xs:NCName"
use="optional"/>
    <xs:attribute name="target-entity-type" type="xs:NCName" use="optional"/>
  </xs:complexType>

  <xs:complexType name="ParticipantCreatorDefinitionType">

```

```

        <xs:sequence>
            <xs:element ref="abc:participant-name-field"
minOccurs="0" maxOccurs="unbounded" />
            <xs:element ref="abc:participant-address" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required" />
        <xs:attribute name="type" type="xs:string" use="required" />
    </xs:complexType>

    <xs:complexType name="EvidenceConfigType">
        <xs:sequence>
            <xs:element name="entity" type="abc:EntityType"

maxOccurs="unbounded" />
            <xs:element

name="def-create-participant" type=
"abc:ParticipantCreatorDefinitionType"

minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
            <xs:attribute name="package" type="xs:string" use="optional" />
        </xs:complexType>

    <xs:simpleType name="MutliplicityType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="multiple" />
            <xs:enumeration value="singleton" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="FieldType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="append" />
            <xs:enumeration value="button-radio" />
            <xs:enumeration value="button-checkbox" />
            <xs:enumeration value="choice-combo" />
            <xs:enumeration value="choice-multi-list" />
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="FieldConfig">
        <xs:attribute name="name" type="xs:NCName" />
        <xs:attribute name="type" type="abc:FieldType" use="optional" />
        <xs:attribute name="append-separator" type="xs:string" use="optional" />
        <xs:attribute name="codetable-class" type="xs:string" use="optional" />
    </xs:complexType>

    <xs:complexType name="SectionConfig">
        <xs:sequence>
            <xs:element name="field" type="abc:FieldConfig" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:attribute name="name" type="xs:NCName" use="required" />
        <xs:attribute name="type" type="abc:MutliplicityType" use="optional" />
    </xs:complexType>

    <xs:complexType name="PdfConfigType">
        <xs:sequence>
            <xs:element name="section" type="abc:SectionConfig"

maxOccurs="unbounded" />
            </xs:sequence>
            <xs:attribute name="pdf-form-name" type="xs:string" />
        </xs:complexType>

    <xs:complexType name="DatastoreConfigType">
        <xs:attribute name="targetSchema" type="xs:string" />

```

```

</xs:complexType>

<xs:element name="evidence-config" type="abc:EvidenceConfigType"/>

<xs:element name="pdf-config" type="abc:PdfConfigType"/>

<xs:element name="datastore-config" type="abc:DatastoreConfigType"/>

<xs:complexType name="ApplicationBuilderConfigType">
  <xs:choice>
    <xs:element ref="abc:evidence-config"/>
    <xs:element ref="abc:pdf-config"/>
    <xs:element ref="abc:datastore-config"/>
  </xs:choice>
</xs:complexType>
<xs:element name="application-builder-config"

type="abc:ApplicationBuilderConfigType"/>
</xs:schema>

```

Registro cronológico de errores y diagnóstico

Códigos de error

Durante el desarrollo de correlaciones, es habitual que el desarrollador se encuentre con errores inesperados. La aplicación puede configurarse para aumentar el nivel de registro cuando los desarrolladores se encuentran con dichos errores.

La propiedad de aplicación

```
curam.workspaceservices.application.processing.logging.on
```

puede establecerse a true para aumentar la granularidad de los mensajes.

La lista de códigos de error (y su descripción de tabla de códigos) devueltos por el motor de correlaciones es la siguiente:

APROCER001 Se ha producido un error al crear una persona.
 APROCER002 Se ha producido un error al crear un candidato.
 APROCER003 Se ha producido un error al crear un candidato.
 APROCER004 Se ha producido un error al crear un caso.
 APROCER005 Se ha producido un error al realizar una correlación "map-attribute".
 APROCER006 Se ha producido un error al realizar una correlación "set-attribute".
 APROCER007 Se ha producido un error al realizar una correlación "map-address".
 APROCER008 Error general de correlación.
 APROCER009 Error al crear la prueba.
 APROCER010 Hay más de un formulario PDF registrado contra el mismo tipo de programa.
 APROCER011 Error al establecer el tipo de id alternativo de un candidato.
 APROCER012 Valor de ID alternativo no válido.
 APROCER013 El generador de solicitudes de pruebas se ha configurado correctamente.
 APROCER014 El tipo de prueba no está listado en la configuración de correlación.
 APROCER015 No se ha encontrado ninguna entidad de prueba padre.
 APROCER016 Se ha producido un error al intentar deserializar el XML de aplicación.
 APROCER017 Se ha producido un error al intentar establecer el valor de un campo.
 APROCER018 Se ha producido un error al intentar crear el documento PDF.
 APROCER019 Se ha producido un error al intentar crear el documento PDF.
 No se ha podido correlacionar un código de formulario con una descripción de tabla de códigos.
 APROCER020 Se ha producido un error al probar un manejador de extensiones de correlación de WorkspaceServices.
 APROCER021 Falta el atributo de origen en la entidad de almacén de datos.
 APROCER022 Un atributo en la expresión no es válido.
 APROCER023 Error de configuración del generador de solicitudes.
 APROCER024 Error al crear DataStoreMappingConfig, no se ha especificado ningún nombre.
 APROCER025 Error al crear DataStoreMappingConfig, el nombre no es exclusivo.
 APROCER026 La correlación con el almacén de datos se ha tenido que anular porque el esquema

no está registrado.
APROCER027 Se ha producido un problema al analizar la especificación de la correlación.
APROCER028 Error general de correlación. Se incluye el XML de correlación.
APROCER029 No se pueden tener varios participantes primarios.
APROCER030 No se ha solicitado ningún programa.
APROCER031 Se ha producido un error al intentar correlacionar con datos de persona.
APROCER032 Se ha producido un error al intentar correlacionar con datos de relación.
APROCER033 Se ha producido un error al crear casos.
APROCER034 Se ha producido un error al crear pruebas.
APROCER035 No se ha solicitado ningún programa.
APROCER036 Se ha producido un error al leer datos del almacén de datos.
APROCER037 El tipo de caso especificado no existe.
APROCER038 El tipo de caso especificado no existe.
APROCER039 Se ha especificado un SSN duplicado.
APROCER040 Se ha especificado un SSN duplicado.
APROCER041 Se ha producido un problema con el proceso de flujo de trabajo.
APROCER042 No se ha identificado ningún participante primario como parte del proceso de admisión.

Este conjunto de códigos de error devuelto por la aplicación es el conjunto definido en el archivo de tabla de códigos CT_ApplicationProcessingError.ctx. El rango de códigos reservado para el procesamiento interno es APROCER001 – APROCER500. Esto significa que los clientes disponen del rango APROCER501 – APROCER999 en caso de que deseen registrar errores en un procesamiento personalizado, p.ej. una clase de extensión manejadora de correlaciones.

Los mensajes de error que aparecen en los registros se encuentran en los siguientes archivos de mensajes: EJBServer\components\WorkspaceServices\message\Mapping.xml, EJBServer\components\WorkspaceServices\message\WorkspaceServicesDataMapping.xml

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos. Es posible que IBM no ofrezca los productos, servicios o características que se describen en este documento en otros países. Póngase en contacto con el representante local de IBM para obtener información acerca de los productos y servicios que actualmente están disponibles en su zona. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM. IBM puede tener patentes o aplicaciones pendientes de patente que conciernan al tema descrito en este documento. La entrega de este documento no le otorga ninguna licencia sobre dichas patentes.. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

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

Intellectual Property Licensing

Legal and Intellectual Property Law.

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japón

El siguiente párrafo no se aplica al Reino Unido ni a ningún otro país en las que tales provisiones sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO, SEA EXPRESA O IMPLÍCITA, INCLUIDAS, AUNQUE SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO CONTRAVENCIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO DETERMINADO. Algunos estados no permiten la renuncia de garantías explícitas o implícitas en determinadas transacciones, por lo que es posible que este párrafo no se aplique en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectuarán cambios en la información aquí contenida; dichos

cambios se incorporarán en las nuevas ediciones de la publicación. BM puede realizar mejoras o cambios en los productos o programas descritos en esta publicación en cualquier momento y sin previo aviso.

Las referencias en esta información a sitios web que no son de IBM se proporcionan sólo para su comodidad y de ninguna manera constituyen una aprobación de estos sitios web. Los materiales de estos sitios Web no forman parte de los materiales de IBM para este producto y el uso que se haga de estos sitios Web es de la entera responsabilidad del usuario.

IBM puede utilizar o distribuir la información que se le suministre del modo que estime oportuno, sin incurrir por ello en ninguna obligación con el remitente. Los titulares de licencias de este programa que deseen tener información sobre el mismo con el fin de: (i) intercambiar información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) utilizar mutuamente la información que se ha intercambiado, deberán ponerse en contacto con:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

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

El programa bajo licencia que se describe en este documento y todo el material bajo licencia que se encuentra disponible para el programa se proporcionan de acuerdo con los términos del Acuerdo del Cliente de IBM, el Acuerdo Internacional de Licencia de Programas o cualquier acuerdo equivalente entre IBM y el Cliente.

Cualquier dato relacionado con el rendimiento que aquí se presente se ha obtenido en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Es posible que algunas medidas se hayan tomado en sistemas que se están desarrollando y no se puede garantizar que dichas medidas serán iguales en los sistemas disponibles en general. Además, es posible que algunas mediciones se haya estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles.

IBM no ha probado tales productos y no puede confirmar la precisión de su rendimiento, su compatibilidad ni ningún otro aspecto relacionado con productos que no son de IBM. Las preguntas relacionadas con las posibilidades de los productos que no son de IBM deben dirigirse a los proveedores de tales productos.

Todas las sentencias relativas a la dirección o intención futura de IBM están sujetas a modificación o retirada sin previo aviso, y sólo representan objetivos.

Todos los precios de IBM que se muestran son precios actuales de venta al por menor sugeridos por IBM y están sujetos a modificaciones sin previo aviso. Los precios del intermediario podrían variar.

Esta información se utiliza a efectos de planificación. Ver antes de que los productos descritos estén disponibles.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlas de la forma más completa posible, los ejemplos pueden incluir nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es totalmente fortuita.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir los programas de ejemplo de cualquier forma, sin tener que pagar a IBM, con intención de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con la interfaz de programación de aplicaciones (API) de la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado en profundidad bajo todas las condiciones. En consecuencia, IBM no puede garantizar ni afirmar la fiabilidad, utilidad o funcionalidad de estos programas. Los programas de ejemplo se proporcionan "TAL CUAL", sin ningún tipo de garantía. IBM no asumirá ninguna responsabilidad por daños ocasionados por el uso de los programas de ejemplo.

Cada copia o parte de estos programas de ejemplo o cualquier trabajo derivado de los mismos, debe incluir un aviso de copyright como el siguiente:

© (nombre de la empresa) (año). Algunas partes de este código se derivan de programas de ejemplo de IBM Corp.

© copyright IBM Corp. _especifique el año o años_. Reservados todos los derechos.

Si visualiza esta información en una copia software, es posible que no aparezcan las fotografías ni las ilustraciones en color.

Consideraciones sobre la política de privacidad

Los productos de IBM Software, incluidas las soluciones de software como servicio ("Ofertas de software") pueden utilizar cookies u otras tecnologías para recabar información de uso del producto, ayudar a mejorar la experiencia del usuario final, adaptar las interacciones con el usuario final u otros fines. En muchos casos, las Ofertas de software no recopilan información de identificación personal. Algunas de nuestras Ofertas de software pueden ayudar a recabar información de identificación personal. Si esta Oferta de software utiliza cookies para recabar información de identificación personal, a continuación se expone información específica sobre el uso de cookies de esta oferta.

Dependiendo de las configuraciones desplegadas, esta Oferta de software podrá utilizar cookies de sesión u otras tecnologías similares que recaben el nombre, la contraseña u otra información de identificación personal a efectos de gestión de la sesión, autenticación, usabilidad de usuario mejorada, configuración de un inicio

de sesión único u otros fines de seguimiento del uso y/o funcionales. Dichas cookies o tecnologías similares no se pueden inhabilitar.

Si las configuraciones desplegadas para esta Oferta de software le proporcionan a usted como cliente la capacidad de recabar información de identificación personal de usuarios finales por medio de cookies y otras tecnologías, deberá buscar su propio asesoramiento legal relativo a las leyes aplicables a dicha recopilación de datos, incluyendo cualquier requisito de aviso y consentimiento.

Para obtener información adicional relativa al uso de diversas tecnologías, incluidas las cookies, a tales fines, consulte la política de privacidad de IBM en <http://www.ibm.com/privacy> y la declaración de privacidad en línea de IBM en <http://www.ibm.com/privacy/details>, las secciones tituladas "Cookies, balizas web y otras tecnologías" y "Declaración de privacidad de los productos software y del software como servicio de IBM" en <http://www.ibm.com/software/info/product-privacy>.

Información sobre interfaces de programación

Esta publicación documenta las interfaces de programación que permiten al cliente escribir programas para obtener los servicios de IBM Cúram Social Program Management.

Marcas registradas

IBM, el logotipo de IBM e [ibm.com](http://www.ibm.com) son marcas registradas de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM u otras empresas. Encontrará una lista actual de marcas registradas de IBM en la web en "Copyright and trademark information" en <http://www.ibm.com/legal/us/en/copytrade.shtml>.

Adobe y Portable Document Format (PDF), son marcas registradas o marcas comerciales de Adobe Systems Incorporated en los Estados Unidos, otros países o ambos.

Java y todas las marcas y logotipos basados en Java son marcas registradas de Oracle y/o sus afiliados.

Otros nombres pueden ser marcas registradas de sus respectivos propietarios. Otros nombres de empresas, productos o servicios pueden ser marcas registradas o de servicio de terceros.



Impreso en España